9th 3DGeoInfo Conference 2014

Proceedings

Editors: Martin Breunig, Mulhim Al-Doori, Edgar Butwilowski, Paul Vincent Kuper, Joachim Benner, Karl-Heinz Haefele

Karlsruhe Institute of Technology

AUD
AMERICAN
UNIVERSITY
IN DUBAI

Published by the Conference Chairs of 3DGeoInfo 2014 in Karlsruhe, Germany.
Part of 3DGeoInfo conference proceedings series.

SEMANTIC VALIDATION OF GML-BASED GEOSPATIAL DATA

Detlev Wagner[1] Volker Coors[1] and Joachim Benner[2]

Abstract

General awareness for data quality of virtual 3d city models is increasing with more complex application scenarios. Investigations and algorithms for validation of geometric properties show the importance of clearly specified data quality of geometry features for downstream applications such as simulation or building energy demand estimation. Beyond geometry, semantic information is a relevant part of information models (such as CityGML or XPlanGML) and has a wide influence on the usability of the data. In this paper, approaches to specify and validate rules and restrictions for semantic information according to the conformance requirements of the CityGML 2.0 standard and XPlanGML are discussed. In addition, a sample use case referring to the product specification for 3D models of AdV (German working group of surveying authorities) is presented.

1 Introduction

A large percentage of the electronic data available in public authorities as well as in private industry is spatially related. For the interoperable exchange of data between different application systems, open standards become more and more important. Many of these standards, modelling two-dimensional (2D) as well as three-dimensional (3D) geospatial data, are extensions (*Application Schemata*) of

[1] Detlev Wagner(✉), Volker Coors
HFT Stuttgart-University of Applied Sciences, Faculty C, Schellingstraße 24,
70174 Stuttgart, Germany
e-mail: detlev.wagner@hft-stuttgart.de, volker.coors@hft-stuttgart.de

[2] Joachim Benner
Karlsruhe Institute of Technology, Institute for Applied Computer Science
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen, Germany
email: joachim.benner@kit.edu

the base standard Geography Markup Language (GML) of the Open Geospatial Consortium (OGC).

The specification of a GML Application Schema defines rules, which have to be observed by any *valid* instance document. The set of rules normally consists of two parts: One or more *XML-Schemata*, defining formal syntax rules for instance documents, and a set of additional *Conformance Rules*, which may either be specified informally, or by using a formal language like Object Constraint Language (OCL 2014) or *Schematron* (Schematron 2014).

An application schema always extends the basic standard GML. The specification of GML also refers to XML-Schemata and additional conformance rules (OGC 2007), which also have to be regarded by any valid instance document. The most important GML conformance rules concern geometry. In particular, every geometry object occurring in the document must be mathematically correct, and the corresponding coordinate reference system must (directly or indirectly) be known.

For special application areas or user groups, the basic requirements of a GML application schema may be sharpened, intending to facilitate the usage of the standard by eliminating ambiguities, or to ensure the availability of needed information. The definition of an *application profile* can be formalized as restriction of the basic XML-Schemata, and/or as a set of additional conformance rules.

This paper generally discusses validation and testing of GML-based geospatial data. As already indicated, the conformance rules of the base standard GML are mainly concerned with correct geometry. Especially for 3D geospatial data, this problem has already been studies by several authors (Wager et. al. 2013), (Zhao et al. 2012). In these papers, mathematical definitions for correct 3D geometry objects (esp. for surfaces and solids) are specified, as well as algorithms and procedures for checking geometrical correctness are described. In our contribution, we therefore concentrate on conformance and profile rules for specific GML application schemata, based on geometrical, and/or attributive properties of feature types, and on plausibility checks for the correct usage of semantic concepts or the consistency of geometry properties with corresponding attributive information. In the following, this is called *semantic validation,* and the corresponding rules are generally denoted as *semantic rules*.

Section 2 addresses a number of techniques for formally specifying semantic rules and describe the state of the art for validating XML-based data. In section 3, semantic validation is generally defined and basic requirements for a successful semantic validation are described. The rest of the paper illustrates the concept with two examples: CityGML (OGC 2012) as example for a 3D data model, and XPlanGML (Benner and Krause 2007) as example for a 2D data model. The corresponding semantic rules are summarized in section 4, and section 5 depicts prototypic use cases for semantic validation.

2 State of the art

Validation of XML instance documents can be done step by step, on different levels:

- a validation against the corresponding XML schema,
- compliance with the specified conformance requirements
- geometry checks, and
- as far as possible, the semantic validation, often referring to conformance rules and geometry.

During schema validation the elements and the structure of an instance document are compared with the definitions in the schema. Reliable tools are available (Oxygen, XMLSpy etc.) to perform this task. We don't discuss the details of this step here, as this is common for all types of XML-documents.

In many cases, the schema does not contain all relevant information. Additional restrictions for certain features are often defined elsewhere due to the lack of integrating assertions about the presence or absence of patterns in XML trees. One way to validate these kinds of constraints is Schematron (Schematron 2014), a rule-based validation language for markup languages. It can perform similar tasks and has a similar relationship with XML as OCL (OMG 2014) is suitable to formulate constraints for UML.

GML defines special geometric elements. It is not sufficient to check for their formal structure within the instance document, what could be done by the above mentioned techniques. The underlying geometric model needs special attention in two ways. First, parse the instance document in order to extract relevant geometric objects and their properties, while deriving implicit information (e.g. topology). Second, once a suitable geometry representation is retrieved after this unmarshalling process, the correctness of the geometric objects needs to be validated against a rule set based on mathematical definitions. Rules for 3D objects have been defined by Gröger & Plümer (2011) and Ledoux (2013).

Geometry validation of XML-based 3D city models is evolving around the building as the object of major interest. Subsequently, geometric constraints are implemented and validation rules are developed to satisfy the special needs for geometric validation of building objects. XML-based validation seems to be dealt with mainly by conducting schema validation in a preprocessing step. However, certain relevant aspects are not covered by this approach. Mäs (2009) developed abstract semantic integrity constraints for spatial data, mainly tailored towards topology and 2D maps. The concept might be useful in certain aspects of 3D data as well, as it extends the work of Egenhofer & Franzosa (1991).

The relationship of semantics and geometry can become rather complex, and difficult to model. Research towards spatio-semantic coherence shows that semantic 3D city models contain a high chance of being not coherent (Stadler & Kolbe 2007). Examinations of validating semantics of CityGML models are discussed in

Wagner et al. 2012. An approach for IFC, based on the usage of the Query Language for Building Information Models, is presented by Daum & Borrmann, 2013.

This paper extends the proposal of a formal specification of model properties and a validation plan in Wagner et al. (2014) and describes a related approach using GML-Toolbox (Benner 2012).

3 Necessity, requirements and approaches for semantic validation

As already addressed in section 1, in GML-based data models the XML schema rules very often are complemented by (formally or informally specified) conformance or profile rules. The additional rules should be formulated that, for instance documents validating against the XML schemata, an automatic processing is at least principally possible. In the following, this process is called *semantic validation*.

A main reason for the necessity of semantic validation lies in the structure of typical GML-based data models. They normally contain a large number of different *feature types*, each of them representing a specific real-world object. Furthermore, a feature type very often contains a large number of *optional feature properties*, which represent specific aspects of the corresponding real-world object. Here, the term "*feature property*" is defined quite generally and comprises:

- *Simple non-geometric attributes*, defined by XML schema built-in types (e.g. `string` or `anyUri`) or restrictions resp. extensions of such types (e.g. `gml:MeasureType`);
- *Simple geometric attributes*, defined by GML geometry types (e.g. `gml:SolidPropertyType`);
- *Relations* to other feature types;
- *Complex attributes*, defined as (eventually hierarchically structured) aggregation simple attributes and relations.

Very often, different feature types of an application schema address the same real-world object, or different properties of a feature type are related with the same real-world aspect of the corresponding object. As an example, the CityGML standard (see section 4.1) enables to represent the exterior shell of a building with different feature types, either representing a building as a whole (`Building` or `BuildingPart`), or representing specific parts of a building (e.g. walls or roofs) with dedicated feature types (`WallSurface`, `RoofSurface`), or both. Furthermore, `Building` supports a number of attributes for geometrically representing the building's exterior shell, which differ in semantic meaning and the geometry type in use. In addition, there are a number of simple non-geometric attributes (number of storeys, storey heights, building height), which also address the geometrically represented real-world object.

This shows that *semantic dependencies* may exist between different feature types or between different feature properties of one feature type. In consequence, instances of these feature types (subsequently called *feature objects*), or instances of feature properties (*property values*) must be consistent and cannot be generated independently. There are two main strategies to ensure semantic consistency or at least to enable an automatic consistency checking: The omission of consistency problems by restricting the general data model, and the explicit modelling of consistency requirements of semantically dependent feature types/properties by conformance rules.

An explicit restriction of a GML-based data model normally is called a *profile*. Besides the omission of consistency problems, a frequent motivation for defining profiles is to facilitate the interpretation of instance documents for dedicated application domains, and to ensure that all needed information really is contained in the data set. Typical techniques used for the definition of profiles are:

- The exclusion of selected feature types;
- The exclusion of selected *optional* feature properties of a feature type;
- The transformation of optional feature properties into required properties;
- The restriction of the admissible range of values for certain properties.

However, not in every case it is possible or reasonable to avoid semantic dependencies by restricting the base standard. In many cases, multiple, semantically dependent feature types or feature properties are needed to digitally represent a real-world situation completely and correctly. In other cases, different alternative ways to represent the same situation must be supported for application reasons. In such cases, the consistency requirements should be represented as conformance rules.

In many cases, conformance rules are specific for a feature type and specify restrictions in the usage of its optional properties. Thus, it frequently occurs that dedicated properties cannot be used simultaneously, or that the actual value of one property implies restrictions on the admissible values of other properties.

Semantic dependencies between different feature types also exist frequently. In many cases, involved feature objects are linked by a relation, which expresses a specific semantic meaning. In very special cases, semantic dependencies between groups of unrelated feature objects may also exist; section 4.2 gives an example for the standard XPlanGML.

4 Semantic validation for existing data models

The last section very generally introduced the concept of semantic validation. For actually performing this process, a set of rules is needed, restricting or sharpening the XML schemata of the basic data model, supporting semantic consistency on feature type or property level, or enabling plausibility checks for the correct usage of semantic concepts. In this section, semantic rules for two existing standards and possible ways to verify them shall be presented and discussed.

Validation of CityGML documents

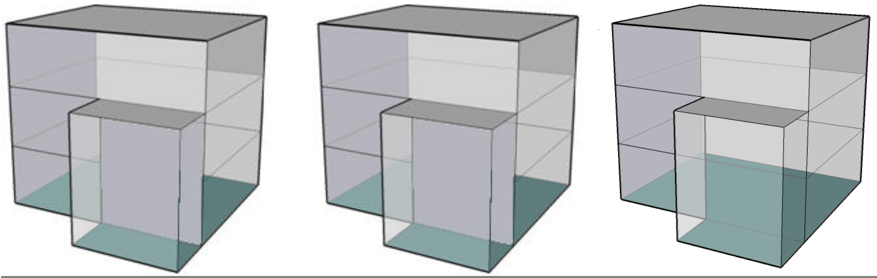
CityGML is a standard of the OGC that provides a comprehensive open data model for modeling, storage and exchange of virtual semantically enriched 3D city models in different levels of detail. The standard allows different varieties for modeling the same real world features, which is conductive to the wide acceptance. It is widely used as an exchange format for 3D city models.

A consequence of the open nature of the standard is that in some cases data sets are structured in unfavorable or inconsistent ways, and features are modeled with inefficient or even wrong geometry, let alone semantics. An example for misleading geometry is to model a building with `MultiSurface` geometry although it obviously has a generic *solid* geometry.

Semantic properties of virtual city models are defined less strict than geometry. Semantic validation of GML-based geospatial data is not restricted to semantic consistency between different feature objects or feature property values. For the correct usage of a data model it is also essential to correctly use the semantic concepts of feature types and properties. In CityGML, the feature type `Building` is designated for the representation of buildings, its usage for bridges or tunnels would be wrong. In the same way, it would be wrong to use the class `WallSurface` (exterior part of a wall) for modelling the roof or the ground surface of a building.

As an example, the surface normal of a building wall will mostly be horizontal (within a tolerance), while a roof normal has a finite positive angle against the (x,y) plane. This characterization enables at least a rough check for `WallSurface` and `RoofSurface` feature objects.

Another issue is the division of a complex structure into `Building` and `BuildingPart` features. According to the standard, a `BuildingPart` “is used to model a structural part of a building” (p.70). Further details are found in the conformance requirements, namely the base requirement of the Building module: “if a building is composed of individual structural segments, it shall be modelled as a `Building` element having one or more additional `BuildingPart` elements. Only the geometry and non-spatial properties of the main part of the building should be represented within the aggregating `Building` element” (p.84). Several alternatives exist, which are in use regardless of this requirement, most notably to model the building itself as a container without own geometry that contains several `BuildingParts`, which in turn contain their geometry. Some alternative ways to model `BuildingParts` are discussed in Wagner et al. (2013) and depicted in figure 1. The semantic attribution `Building` refers only to the main structure in the left part, but to the full structure in the other two examples, where also two `BuildingParts` exist. Relation between geometric and semantic elements is hence different.



1 Building (Solid) 1 BuildingPart (Solid) combined geometry: CompositeSolid	1 Building (no geometry) 2 BuildingParts (Solid) combined geometry: Solid	1 Building (no geometry) 2 BuildingParts (MultiSurface) combined geometry: MultiSurface

Fig. 1: Different alternatives of modeling a Building and a BuildingPart.

Since there might be a reason to decide for a certain alternative, the validation result cannot be expressed as true or false without additional information. It is rather an indicator of plausibility to reveal major errors in the usage of semantic concepts.

Conformance Requirements

A conformance requirement defines restrictions to the model which can be described in a formal way and be validated automatically. Conformance requirements in the standard are often defining additional spatio-semantic restrictions, while others resemble modeling guidelines. They can be grouped into different types, according to the possibility to derive a crisp validation rule. In the following, we give examples of conformance requirements which refer to semantic properties for the Building module of CityGML. Requirements dealing with the interior structure of a building are not considered here.

- *Starting from LOD2, the exterior shell of an _AbstractBuilding may be semantically decomposed into _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of _AbstractBuilding. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and ClosureSurface as subclasses of _BoundarySurface are allowed. The boundedBy property (not to be confused with the gml:boundedBy property) shall not be used if the building is only represented in LOD1.*

The requirement is two-fold, both parts can be validated unambiguously. The validation result is a non-conformance error. A similar conformance requirement is defined for `BuildingInstallation` elements.
Suggestion for a formal specification:

if (_AbstractBuilding.boundedBy.BoundarySurface != NULL) **then** (LOD(_AbstractBuilding) ≥ 2) **and** (typeOf (BoundarySurface) != CeilingSurfaceType) **and** (typeOf (BoundarySurface) != FloorSurfaceType) **and** (typeOf (BoundarySurface) != InteriorWallSurfaceType)

As a consequence, interior surfaces in LOD 4 would be recognized as erroneous. As they are belonging to the Room element, this result would be expected.

- *Starting from LOD2, the outerBuildingInstallation property (type: BuildingInstallationPropertyType) of _AbstractBuilding may be used to model BuildingInstallation elements. BuildingInstallation elements shall only be used to represent outer characteristics of a building which do not have the significance of building parts. The outerBuildingInstallation property shall not be used if the building is only represented in LOD1.*

A similar problem with the definition of `BuildingInstallation` elements as discussed above for `BuildingParts` occurs. “Significance” of a `BuidlingInstallation` is always relative and can refer to size, shape or material of the `BuildingInstallation`. One could think about user-provided definitions for size or shape parameters.
Formal specification is only possible for the remaining part:

if (_AbstractBuilding.BuildingInstallation != NULL) **then**
LOD(_AbstractBuilding) > 1

- *Starting from LOD3, openings of _BoundarySurface elements may be modelled using the opening property (type: OpeningPropertyType) of _BoundarySurface. This property shall not be used for _BoundarySurface elements only represented in LOD2. Accordingly, the surface geometry representing a _BoundarySurface in LOD2 must be simply connected.*

Validation of _Opening elements occurring only in LOD 3 and 4 is simple and unambiguous. Formal specification:

if (_AbstractBuilding.boundedBy.BoundarySurface._Opening != NULL) **then**
(LOD(_AbstractBuilding) ≥ 3) and (BoundarySurface.InteriorRing) != NULL)

Since complex spatio-semantic relations are described and CityGML has no explicit topological model, considering the type of the embrasure surfaces is only possible after neighborhood of surfaces has been extracted from the data. However, upper and lower parts of window embrasures can be modeled as `OuterCeilingSurfaces` or `OuterFloorSurfaces`, respectively, which would be a violation of the requirement.

Other restrictions concerning the coherency of geometric attributes and geometry are not stated as explicit conformance requirements, although an attribute such as `measuredHeight` or `storeysAboveGround` can be easily compared with the actual geometry.

Validation of XPlanGML documents

XPlanGML (XPlanung 2014) is a two-dimensional, GML-based data model for spatial planning documents in Germany. It is designed to represent the spatial content of German national planning instruments like “Bebauungsplan”, “Flächennutzungsplan” or “Regionalplan”, regulating the possible utilization of land in the future. For this, the feature types and feature properties of XPlanGML directly implement national planning legislation (Baugesetzbuch, Bau-Nutzungsverordnung). In many aspects (Benner et al. 2013) the data model is similar to the European INSPIRE PLU standard for planned land use (INSPIRE 2013). XPlanGML as well as INSPIRE PLU support the *Zoning Concept*, enforcing that the primary land use is uniquely defined for the whole planning area. Furthermore, the primary regulations can be complemented or restricted by additional conditions and constraints.

The XML schemata of the actual version XPlanGML 4.1 are supplemented by more than 230 conformance rules (Benner 2014). They are defined informally, but the specification in a formal language and an automatic checking is principally possible. A short overview on this set of rules is depicted subsequently.

A very important global rule is concerned with the Zoning Concept. For XPlanGML instance documents, the implementation of this concept requires that the surface geometry of feature objects, which are identified to represent primary land use regulations, must not overlap and must altogether seamlessly cover the whole planning area.

The remaining conformity rules regulate the consistency of attributive information. Frequently, a specific numeric restriction of land use (e.g. the admissible number of building storeys or the admissible size of the constructible surface) can be specified in different manner: As maximal value, as range of values or as exactly prescribed value, and (for physical measures) it is possible to specify absolute values or relative parameters. In these cases, additional rules must ensure that the specifications are complete and unambiguous.

As INSPIRE PLU, XPlanGML partially supports a hierarchical classification of land use. In these cases, XPlanGML normally uses two or more independent attributes, whose admissible values are defined by specific enumerations or code lists. The specified conformity rules ensure the consistency of the attributive information and guarantee that always a rough classification is available.

5 Use cases

In this section, two use cases for semantic validation of GML-based geospatial data will be presented. The first one is the checking of conformance requirements for CityGML and XPlanGML, which can actually be performed with the software GML-Toolbox (Benner 2012). The second example is concerned with a specific CityGML profile, being used by the German Cadastral and Surveying Authorities.

5.1 Validation of conformance rules with GML-Toolbox

The GML-Toolbox was originally developed as prototypic implementation of the XPlanGML standard. Later on, the software was enhanced for supporting various two- and three dimensional, GML-based data models including CityGML and CityGML Application Domain Extensions (ADE).

All errors detected in the checking process are interactively displayed and documented in an XML-based error protocol. The error documentation covers a detailed error description, including information which conformity rule has been hurt, and information needed to locate and correct the error (line numbers, identifiers, names and values of objects and properties).

In the GML-Toolbox, some semantic validation functions are hard-coded in the software. This comprises all checks based on geometric algorithms, like, e.g., the validation of the Zoning Concept. Furthermore, the software enables for each supported data model to specify a list of notations for coordinate reference systems and units of measurements, which must be used by instance documents. It also can be checked automatically whether the value of code list attribute is really contained in the dedicated code list. The user interface enables to activate or deactivate all these checks.

For both standards CityGML and XPlanGML, the majority of the existing conformance rules can be specified in a formal language, which enables a general rule-based checking strategy. The GML-Toolbox uses a specially designed, XML-based format for the specification of conformance rules, which shall be illustrated by some CityGML examples.

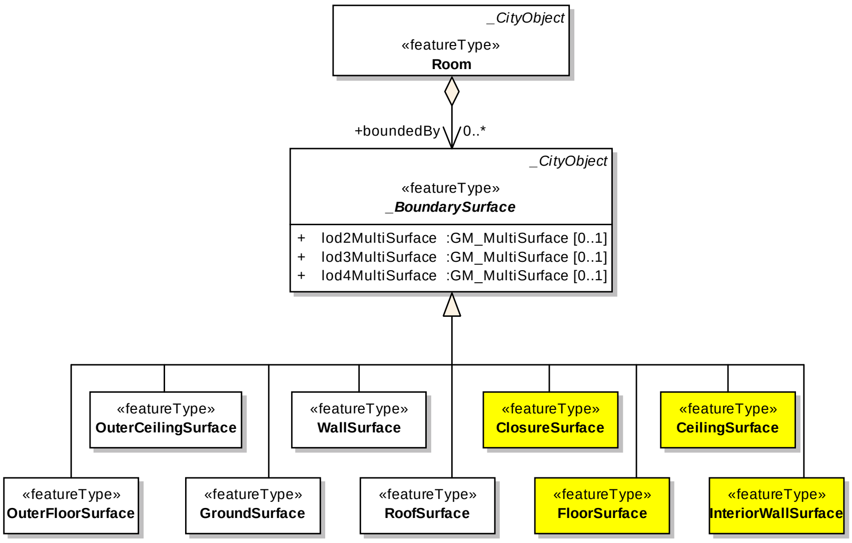


Fig.2: CityGML feature type `Room`

Many CityGML conformance rules restrict relations between CityGML feature types. For example, due to the XML-schema (see Fig. 2) the relation `boundedBy` of feature type `Room` may point to an arbitrary boundary surface (abstract feature type `_BoundarySurface`). Another conformance rule restricts the feasible feature types to `FloorSurface`, `CeilingSurface`, `InteriorWallSurface` and `ClosureSurface`.

The formalisation of this restriction is as follows:

```
<Rule class="blgd:Room">
    <Required>
      <RelationFilter relationName="bldg:boundedBy" minOccurs="0"
maxOccurs="unbound">
        <GmlType>bldg:InteriorWallSurface</GmlType>
        <GmlType>bldg:FloorSurface</GmlType>
        <GmlType>bldg:CeilingSurface</GmlType>
        <GmlType>bldg:ClosureSurface</GmlType>
      </RelationFilter>
    </Required>
  </Rule>
```

A number of CityGML conformance rules regulate, which feature types or feature properties can be used for a CityGML model of specific Level of Detail (LoD). In particular, modelling of the building interior (feature type `Room`) is only allowed in LoD4. This implies that all `_BoundarySurface` objects referenced by `Room` must only use the geometry property `lod4MultiSurface`, which can be formalized as follows:

```
<Rule class="blgd:Room">
    <If>
      <RelationFilter relationName="bldg:boundedBy" minOccurs="1"
maxOccurs="unbound"/>
    </If>
    <Required combination="AND">
      <RelationFilter relationName="bldg:boundedBy"
geometryPropertyNameInRelatedObject="lod2MultiSurface" minOccurs="0"
maxOccurs="0"/>
      <RelationFilter relationName="bldg:boundedBy"
geometryPropertyNameInRelatedObject="lod3MultiSurface" minOccurs="0"
maxOccurs="0"/>
    </Required>
  </Rule>
```

The next example formalizes a geometric restriction of a feature property. In the CityGML feature types `Building` and `BuildingPart` (both derived from `_AbstractBuilding`), the geometry properties `lod0FootPrint` and `lod0RoofEdge` are defined as arbitrary multi-surfaces. The conformance rule 10.3.9.2 implies a further restriction: In both cases the multi-surfaces must be *horizontal*, which means that the z-values of all coordinates must be identical. The corresponding rule in the GML-Toolbox formalization is:

```
<Rule superclassType="bldg:_AbstractBuilding">
    <Required combination="AND">
      <GeometryFilter geometryPropertyName="lod0FootPrint "
geometryRestriction="Horizontal"/>
      <GeometryFilter geometryPropertyName="lod0RoofEdge "
geometryRestriction="Horizontal"/>
    </Required>
  </Rule>
```

5.2 *AdV product specification CityGML*

The working group of surveying agencies (AdV) of Germany has defined a product specification for unified CityGML 1.0 building models in LOD1 and LOD2, which shall be provided nation-wide. The profile definition is available as document 1071R2 from adv-online.de and contain the following restrictions:

- Buildings must have Solid geometry.
- A building data set must contain the following mandatory attributes: object identifier, building function, quality information, official municipality key. Optional attributes are number of storeys, location definition, name. Code lists are given for the exact specification of the building height including the location of the ground and roof point, respectively, plus the origin of these values.

A concept for the validation of the AdV profile is described in (Wagner et.al. 2014), where also the formal specification of a validation plan is given. The validation plan is focused on geometry checks, where each check has a unique identifier plus parameters such as tolerance values and dependencies on other checks in some cases. This validation plan will be extended here by a more flexible approach which can express semantic and attribute specifications such as defined in the AdV profile (Listing 1).

```
 <cd:check>
  <cd:checkId>SEM_ATTRIBUTE</cd:checkId>
  <cd:parameter>
    <cd:parameterName mandatory=true>
        bldg:function</cd:parameterName>
    <cd:parameterValue>[URI of code list]</cd:parameterValue>
  </cd:parameter>
  <cd:parameter>
    <cd:parameterName mandatory=true>
        bldg:roofType</cd:parameterName>
    <cd:parameterValue>[URI of code list]</cd:parameterValue>
  </cd:parameter>
  <cd:parameter>
    <cd:parameterName mandatory=true>
        BezugspunktDach</cd:parameterName>
    <cd:parameterValue>[URI of code list]</cd:parameterValue>
  </cd:parameter>
  <cd:parameter>
    <cd:parameterName mandatory=true>
        DatenquelleDachhoehe</cd:parameterName>
    <cd:parameterValue>[URI of code list]</cd:parameterValue>
  </cd:parameter>
  <cd:parameter>
    <cd:parameterName mandatory=true>
        DatenquelleBodenhoehe </cd:parameterName>
    <cd:parameterValue>[URI of code list]</cd:parameterValue>
  </cd:parameter>
  <cd:parameter>
    <cd:parameterName mandatory=true>
        DatenquelleLage</cd:parameterName>
    <cd:parameterValue>[URI of code list]</cd:parameterValue>
  </cd:parameter>
  <cd:parameter>
    <cd:parameterName mandatory=false>
        bldg:storeysAboveGround </cd:parameterName>
    <cd:parameterValue>isInt</cd:parameterValue>
  </cd:parameter>
</cd:check>
<check>
  <cd:checkId>SEM_GML_ID_EXISTS</cd:checkId>
</check>
```

Listing 1: Snippet with validation plan for quality metadata of AdV profile

The check with ID SEM_ATTRIBUTE handles all attributes of the building. Attributes are referred by their name as parameters of the check, regardless whether it is a generic or a predefined attribute in the standard. The attribute value is specified as a parameter value, which can refer to an external code list specified in the instance document. In this case, the actual attribute values are compared with all values occurring in the code list. Other attributes can be checked for correct data types. The validation plan is parsed by an application which executes the actual checks. An error is thrown in case no corresponding attribute value is found in the code list. This is a flexible way to specify attributes and allowed values. The existence of a GML-ID is validated by the check SEM_GML_ID_EXISTS. Uniqueness of this ID is ensured within the validated data set.

The approach is straight forward and provides a simple way to deliver data sets of different origin with the same characteristics to the user. Once validated, these data are guaranteed to be conformant with the published specification, which enables stable downstream processing.

6 Outlook

For geometry validation, a validation plan based on predefined checks proves quite efficient, due to a relatively small variety of different geometric modeling alternatives. Semantic information, however, can be of very distinct and domain-specific nature, which makes it difficult to provide an adequate predefined set of checks. A first step is to enable free definition of attribute names and the allowed range of values.

These rules could be extended to other semantic features such as `BoundarySurface` types or `Building/BuildingPart` relationships.

The formal specification of rules and restrictions is possible in many different ways. Two examples are presented in the paper. Further discussion on improvement and/or extension of these formats would enable the introduction of a commonly accepted specification of validation rules and model properties. This could help to avoid misunderstandings when exchanging data and provide a higher level of confidence for the relationship of model producers and their clients.

7 References

Benner, J., Krause, K.-U. (2007) XPlanung – Ein GIS-Standard zum Austausch digitaler Bauleitpläne. Flächenmanagement und Bodenordnung, 6/2007, pp. 274 – 280.

Benner, J. (2012) XPlanGML-Toolbox – ein flexibler Werkzeugkasten zur Unterstützung des Standards XPlanung. In: M.-O. Löwner, F. Hillen, R. Wohlfahrt (edrs.), Geoinformatik 2012 „Mobilität und Umwelt“, Braunschweig, 28. – 30. März 2012, pp.251 – 255.

Benner, J, Häfele, K.-H., Geiger, A., (2013) Transnational Planning Support by the European Geodata Infrastructure INSPIRE. In: Proceedings REAL CORP 2013, Rome, 20 – 23 May 2013, pp. 2009 – 1017.

Benner, J. (2014) Konformitätsbedingungen XPlanGML 4.1.

Egenhofer, M. J., & Franzosa, R. D. (1991) Point-set topological spatial relations. International Journal of Geographical Information System, 5(2), 161-174.

Gröger, G., & Plümer, L. (2011) How to achieve consistency for 3D city models. Geoinformatica 15, Vol. 1, 137-165.

INSPIRE (2013) D2.8.III.4 Data Specification on Land Use – Technical Guidelines.

Ledoux, H. (2013) On the validation of solids represented with the international standards for geographic information. Computer Aided Civil and Infrastructure Engineering, 28(9), 693-706.

Mäs, S. (2009) On the consistency of spatial semantic integrity constraints (Doctoral dissertation, Universität der Bundeswehr München).

OCL (2014) Object Management Group, Available at: http://www.omg.org/ spec/OCL/. Accessed 16 June 2014.

Open Geospatial Consortium (OGC) (2007) OGC 07-036, OpenGIS® Geography Markup Language (GML) Encoding Standard, Version 3.2.1.

Open Geospatial Consortium (OGC) (2012) OGC 08-007r2, OpenGIS® City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0,

Schematron (2014) Available at: http://www.schematron.com/. Accessed 16 June 2014.

Stadler, A., & Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3D city models. In: Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede.

OMG (2014) Object Constraint Language (OCL). Available at: http://www.omg.org/spec/OCL/. Accessed 28 June 2014.

Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., & Coors, V. (2013) Geometric-semantical consistency validation of CityGML models. In *Progress and New Trends in 3D Geoinformation Sciences* (pp. 171-192). Springer Berlin Heidelberg.

Wagner, D., Kolbe, T., Coors, V. (2014) Spezifikation von Prüfplänen und Prüfergebnissen zur Validierung von 3D-Stadtmodellen. Gemeinsame Tagung 2014 der DGfK, der DGPF, der GfGI und des GiN (DGPF Tagungsband 23 / 2014). CD-ROM.

XPlanung (2014) XPlanung: Standardised data model for land-use plans, Available at: http://www.xplanung.de. Accessed 20 June 2014.

Zhao, J., Ledoux, H., & Stoter, J. (2013) Automatic repair of CityGML LOD2 buildings using shrink-wrapping. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-2 W, 1, 309-317.

Height references of CityGML LOD1 buildings and their influence on applications

Filip Biljecki
Delft University of Technology, Section GIS technology, The Netherlands
f.biljecki@tudelft.nl

Hugo Ledoux
Delft University of Technology, Section GIS technology, The Netherlands
h.ledoux@tudelft.nl

Jantien Stoter
Delft University of Technology, Section GIS technology, The Netherlands
Kadaster, Product and Process Innovation, Apeldoorn, The Netherlands
Geonovum, Amersfoort, The Netherlands
j.e.stoter@tudelft.nl

Abstract

The geometry of the LOD1 block model of CityGML may be represented in a multitude of valid variants, e.g. it may represent individual buildings or aggregated blocks. The geometric representations within each of these, such as the reference of the footprint of a building, are not standardised in CityGML and are not informed in the metadata. The lack of knowledge of the used representation may possibly cause errors when the models are used for spatial analysis. Further, the effect of using different variants is not investigated. In this paper we (1) discuss this known, but frequently overlooked topic; (2) overview the possible geometric references and show how employing them may cause drastic differences for a GIS operation and/or use-case; (3) focus on the vertical (height) references for the top surface in LOD1, and show how to determine the optimal variant within a use-case with experiments using the Monte Carlo method; and (4) discuss the adoption and extension of metadata in INSPIRE to CityGML to reflect the employed geometric reference.

Key words: Level of detail, LOD1, Geometric references, INSPIRE, CityGML

1. Introduction

The level of detail (LOD) concept of the Open Geospatial Consortium (OGC) standard CityGML provides guidelines about the acquisition of buildings and other real-world phenomena [1]. The five defined LODs describe a model's complexity and the spatio-semantic coherence that should be achieved in each [2].

The coarsest volumetric representation that the standard contains is the LOD1 model, which is described as ``the well-known blocks model comprising prismatic buildings with flat roof structures'' [1]. A further clarification is presented by Gröger and Plümer [3]: ``In LOD1 volume objects (buildings, vegetation objects, etc.) are modelled in a generalised way as prismatic block models with vertical walls and horizontal 'roofs'.''.

The block models are usually acquired with extrusion from 2D footprints in combination with points acquired with airborne laser scanning [4], and generalisation from finer LODs [5], for instance, as a bounding box of an LOD2 [6], [7], or of an LOD3 including features such as antennas on roofs [8]. CityGML defines the LOD1 representation for thematic classes such as tunnels and bridges, but in practice it is used only for buildings, hence, this paper is focused on them.

Since LOD1 models are relatively inexpensive to acquire and are not complex, they are ubiquitous, representing a large share of 3D models worldwide [9]. While they are the coarsest volumetric model defined by CityGML, LOD1 models may be very accurate and they find their use in a number of applications [10]. For instance, they have been extensively used in the analysis of the traffic noise for assessing the effect on the quality of life and for the placement of noise barriers [11] [12], in shadow analysis in areas with predominantly flat roofs [13], heat energy demand modelling [14], real estate mass valuation in the urban areas [15], and estimation of the population in a given area [16]. Further, LOD1 models are useful in enhancing the visual representation of other data, such as in the 3D visualisation of air quality data [17]. For other visualisation applications, such as the visual enhancement of navigation, LOD1 models containing individual buildings are deemed too complex, so they have to be further generalised [18].

The LOD1 model contains no semantics on constituting geometries, and its geometric representation is realised by a `gml:solid` and/or a `gml:MultiSurface` [19]. This means that LOD1 is the only volumetric model in CityGML where the semantics is out of focus. Taking into account the above, the LOD1 model represents a favourable ratio between the costs and possible uses, being the main factor contributing to their popularity.

However, since the LOD concept of CityGML, besides a narrative description, does not provide specifications and instructions how to geometrically model the features [20], this results in multiple different models being considered legal variants of the same LOD (see [21] for examples).

This also affects the LOD1 model. While being a rough and fairly simple model without semantics, it encompasses a myriad of possibilities how to model its geometry.

The ambiguities caused by the standard may lead to misunderstandings between stakeholders and to the misutilisation of 3D models. Despite being of the same LOD, the geometry of two models acquired with different practices may considerably deviate from each other, potentially leading to errors in the use of the data if such metadata is not known and/or it is not regarded.

It is our experience that most of the ambiguities are caused by the following: (1) the varying reference point for the elevation of the horizontal top surface of the LOD1 block model (see Fig. 1); and (2) are the vertical surfaces (i.e. walls) captured at the footprint of the building or as the projection of the roof edges. The latter is also an ambiguity for LOD2 models.

These different practices, which we name geometric references, have not been much researched. They may also be available under different terms, such as modelling choices [22].

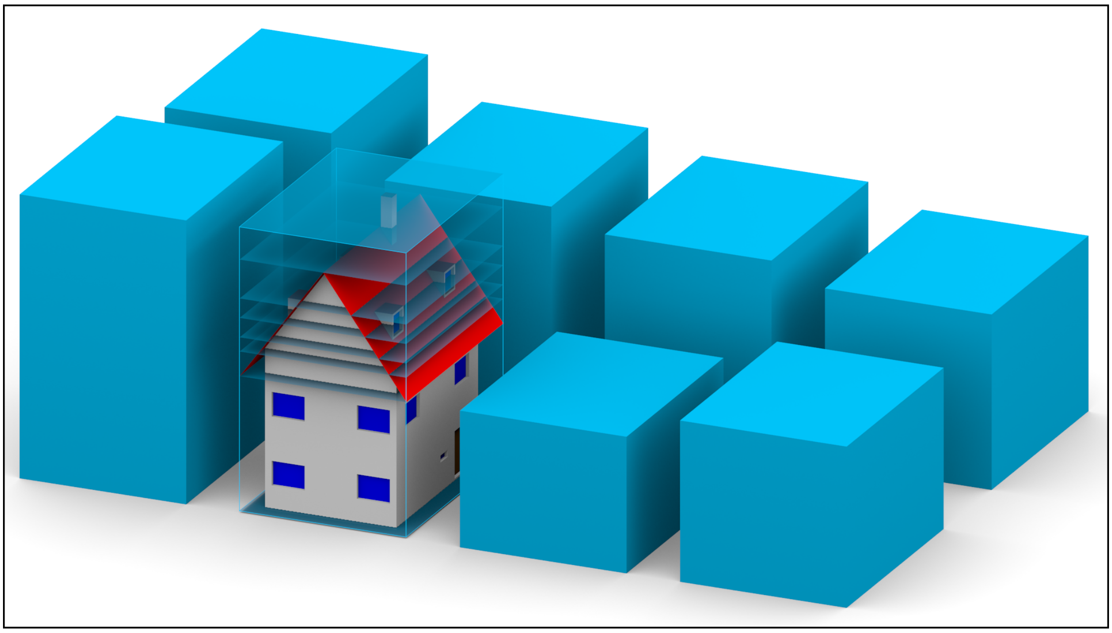


Figure 1: A building represented using different geometric representations in LOD1 (orthographic rendering to preserve the dimensions). All of them are valid according to the CityGML standard. The difference between the resulting geometric representations is significant.

This paper focuses on the geometric reference of representing the height of the block model in LOD1. This topic is important because LOD1 is used widely and it finds its purpose in a number of applications, as shown above. In Sec. 2 we list the frequently used references for the top surface of an LOD1 block model, and relate them to the INSPIRE Data Specification on Buildings [16], [23], which provides a list of references that we further refine to reflect the current practices of the data producers.

We show the importance of this topic by determining the effect of employing different variants to the result of a GIS operation, and how to determine what is the optimal geometric representation with respect a GIS operation and/or use-case. In this process we demonstrate with examples that the differences between the variants may have a drastic influence on the result of an operation (Sec. 3). This is achieved with a Monte Carlo simulation by generating multiple CityGML models of a large number of buildings, and by comparing the results of the computations of volumes of buildings. This is a prominent 3D GIS operation being used in a number of use-cases, such as energy demand estimation [14].

We argue that the LOD concept should continue following the current practices and allow flexibility with the geometric references in each LOD, however, it is something that should be noted in the metadata, which explicitly explain how the geometries have to be interpreted. Because CityGML does not provide such, we work towards implementing the metadata for the geometric references in it (Sec. 4).

2. Background and context

2.1 Geometric LOD1 variants

Notwithstanding its relative simplicity, an LOD1 model has a multitude of valid geometric representations. For instance, LOD1 may represent individual buildings, but also multiple buildings that are aggregated in blocks [24]. Within each of these, there may be a number of possible variants. On one hand, the bases of the blocks may not necessarily be only rectangular, but they may represent finer footprints. On the other hand, an LOD1 model may contain differentiated flat roof tops, rather than a single top surface [25], [26]. Further, the top surface of the block model may represent a number of different references, depending on the height that was determined as the reference value, e.g. height at the roof edges or height at the half point of the roof. Multiplying the different categories, it is easy to count over a few dozen variants of LOD1, many of them occurring in practice.

The variants can be grouped into two categories: LOD-based, and reference-based. The first group covers the fineness of the representation, i.e. a model of an individual building with differentiated roof tops is finer than a model with a single top surface. Because CityGML is not precise enough with defining the LODs, these variants are considered of the same LOD by the standard, which may be disputed from the computer graphics perspective. The latter group refers to the reference of the features, i.e. what does the position of the top surface represent. This paper focuses on the latter.

Fig. 1 shows seven LOD1 block models of an individual building generated according to different references for the height of the top surface, which we will discuss later in the paper. The centre of the Figure shows the model in higher detail (LOD3) as a reference, with the overlapped top surfaces for comparison. It is obvious that the difference between the geometry of the models may be significant. However, CityGML does not provide metadata for expressing such variants, and in practice for many datasets the used geometric reference is not known.

2.2 INSPIRE Building Model references

The INSPIRE initiative's document Data Specification on Buildings-Technical Guidelines presents spatial data specification for European data related to the theme “Buildings” [16]. The document covers both 2D and 3D representations, and gives a focus on the different concepts of footprint and elevation for buildings, representing a good foundation for this paper. It provides a (code) list and definition of several elevation references of buildings that may serve both as attributes and descriptors of the geometric representation. These references are represented by a value type `ElevationReferenceValue`, i.e. a list of self-explanatory elements considered to capture a vertical geometry. The list contains not only elements that represent higher points of a building, but also lower points such as the elevation at the entrance point to the building, and the bottom of the construction.

These values are primarily intended as attributes of objects regardless of their geometry. However, for the 3D representation in LOD1, the standard mandates that the level of a building that was chosen to represent its top has to be documented. This is realised through the attribute `verticalGeometryReference3DTop`, and preferably using the following values of the subset of all values from the above introduced code list `ElevationReferenceValue`, namely: `generalRoofEdge`, `lowestRoofEdge`, `highestRoofEdge`, `lowestEave`, `generalEave`, `highestEave`, `generalRoof`, and `topOfConstruction`. The list is extensive by also taking into account less common cases, for instance, the case where the height of the eaves is not equal (e.g. the eaves on one side of the building are higher than the eaves on the other side). On the other hand, the value `generalRoof` is somewhat ambiguous because it may refer to any point on the roof surface.

For most of the buildings, many values are equal and may be considered as identical. For instance, in reality `lowestEave` is usually equal to `highestEave`, and all-together they correspond to `generalEave`.

While the standard recommends the above listed values as the references for the top surface in LOD1, we do not believe that this list is complete. For instance, another relevant value may be `highestPoint`, the height level that includes non-roof elements such as chimney and antennas, and that is frequent in generalisation. This level is a possible value in the code list

ElevationReferenceValue, but for some reason it is not listed as a recommended value for verticalGeometryReference3DTop.

The second relevant concept is the footprint, which is also covered by the INSPIRE Building model. The reference for the geometry of the footprint is expressed through the HorizontalGeometryReferenceValue, with possible values such as footPrint and roofEdge. The reference may be used for both LOD1 and LOD2 models.

2.3 Overview of frequently used references for the top of the block model

In this Section we focus on the reference of the height of the top surface of a building, partly based on the INSPIRE Building model presented in the previous Section. It is our experience that this is the variable that causes most of the ambiguity in CityGML, and may cause significant errors in the utilisation of the models if not regarded.

While INSPIRE provides an extensive list of geometric references for the top surface of the building in an LOD1 block model, we have thoroughly researched are there other values occuring in practice. We did this by examining papers that deal with the production of LOD1 [27]-[30], and by contacting a few producers of 3D models. We have found additional values that require extending the references in the INSPIRE Building model, and this research also gave us insights for understanding the source of the different variants.

In airborne laser scanning and photogrammetry, the building's height is usually taken from the median height of the points positioned within the footprint of a building (in most cases it roughly corresponds to the half of the height of the roof), but we have also encountered other values such as one third or two thirds of the height of the roof.

In the extrusion from footprints according to attribute values, some practitioners extrude the footprint to the height in an attribute from OpenStreetMap [31], number of floors [32], or the height available in the cadastral records [33]. For many of these the lineage of the data is not known, propagating to the uncertainty of the height of the generated block model. However, in some cases this height may represent the height at the roof eaves, a value which cannot be acquired from airborne platforms because it is usually obscured.

In the generalisation from finer LODs, the block model is usually derived as a 3D bounding box of the fine LOD [7], [8]. Depending on the starting LOD, this means that in most of the cases the top surface represents either the top of the roof or the highest point of the building, including features such as chimneys, air conditioning units, and antennas. These possibilities are already known from the INSPIRE Building model, however, it shows that even within generalisation, there is no standard variant that is used.

Here we list seven frequent references for the elevation of the top surface of the LOD1 model of individual buildings, obtained from the INSPIRE Building model and our own research. These references are identical to the ones in Fig. 1, observed in the counter-clockwise manner from the right side of the LOD3 model. For each height reference we assign an internal shorthand for easier referencing in the text.

- **H0**. Height at the roof edges. Because of the roof overhangs, roof edges may have an elevation that is lower than the one of the highest point of the walls, hence this is the lowest possible reference of the top surface.
- **H1**. Height at the roof eaves. This value is typical for terrestrial measurements, as it is usually not visible for airborne acquisition techniques. It may correspond to the reference H0 in the case when there are no roof overhangs.
- **H2**. Height at one third of the height of the roof. The height of the roof may be somewhat ambiguous, depending on the lowest reference of the roof. For instance, if the roof is considered to start at H0 or H1. In this case we consider the total height (from the point H0).
- **H3**. Height at half of the height of the roof. This reference is related to the extrusion coupled with LiDAR point clouds, where it is typical to use the median value of the height of the points within a footprint [4].
- **H4**. Height at two thirds of the height of the roof. This is a case that we have encountered with a 3D GIS company, and it can also be related to the extrusion where superstructures such as dormers and chimneys are present because they elevate the median of the height of the roof points.
- **H5**. Height at the top of the roof. This is a value typical for generalisation from LOD2. It can also be derived from point clouds if necessary.
- **H6**. Height at the top of the construction of the building. This height encompasses the whole construction, and it is usually used with generalisation from LOD3. In case there are no superstructures that extend beyond the top of the roof, the value corresponds to H5.

Each of these find use in specific applications. For instance, in the case H1 the model retains the walls and the building body, but it may completely disregard the roof structure. This may be useful in applications where the roof is not important. Models constructed by abiding by references H5 and H6 may be good in analyses such as visibility [15], [34], but having little value in other applications such as calculating the surface area of walls for marketing purposes [35].

The seven reference heights have been related to the INSPIRE's in Tab. 1. The Table also includes our additions that are not present in INSPIRE, hence, we extend the INSPIRE's references reflecting the additional possibilities, and introduce provisional notations for the new references.

Table 1: List of representations for the height of the top surface of the LOD1 block model. The equal sign means that the reference is re-used from INSPIRE.

Code (§2.3)	Height at	INSPIRE reference	Our reference
H0	Roof edges	`generalRoofEdge`	=
H1	Roof eaves	`generalEave`	=
H2	One third of the roof height	`generalRoof`	`oneThirdRoof`
H3	Half of the roof height	`generalRoof`	`halfRoof`
H4	Two thirds of the roof height	`generalRoof`	`twoThirdRoof`
H5	Top of the roof	`topOfConstruction`	=
H6	Highest point of the building	`highestPoint`	=

3. Experiments with volumes of LOD1 buildings

In the previous sections we have argued that the selection and employment of different modelling variants has a direct effect on the results of a (3D) GIS operation, and that generally there is no good or bad modelling variant because they are all valid within the present CityGML LOD concept and the INSPIRE Building model.

The goal of this section is twofold. First, we introduce a method that determines which is the most suitable geometric representation with respect to a GIS operation, or a set of operations (i.e. use-case). Second, by realising the method with a GIS operation we prove the claim that different geometric variants within LOD1 may have a drastic impact on the result of a GIS operation.

We discuss the operation of the computation of volumes of buildings, which is essential in use-cases such as energy demand estimation [36], [37], determination of property taxes [38], estimation of the population in a given area [16], and in the volumetric visibility analysis of urban environments [39]. We compute the results on the seven variants of LOD1 listed in Tab. 1, and compare the results to the computations on LOD3 models, which for this purpose we consider as ground truth.

3.1 Methodology

The method that we introduce consists of four steps, which are explained below in more detail.

1. *Defining the geometric references.*
   First the method requires to list and define possible geometric variants within an LOD. This is explained in the previous Section. However, the selection of the references to be considered is related to the producers and users, because in some occasions not all representations are

available. For instance, in the case where only airborne measurements are possible, the reference H1 (`generalEave`) is not obtainable and it should not be taken into consideration for the analysis and experiments.

2. *Randomisation of buildings.*
   In this step we generate a large number of different models in the selected geometric variants in a Monte Carlo simulation. Because every building is different, experiments that consider a large number of buildings are required. For instance, an LOD1-H6 representation is different in the case of a building with a chimney, and another building of same dimensions without a chimney. We have developed a method and implemented it in a software prototype that generates random buildings, and their creation in CityGML in multiple LODs and geometric representations. The method, which we have recently introduced in [40], is based on the Monte Carlo method [41] and serves for generating a large number of dissimilar scenarios. The buildings are generated in an automatic and random process where a large number of building parameters, such as building height, size of chimneys, length of roof overhangs, are randomly sampled from a uniform probability distribution function. We support four most common roof types: gabled, flat, shed, and hipped [42]. Each of these has different reference points when it comes to the top of LOD1, so it is important to include more than one roof type. The datasets that were generated for this purpose are described in Sec. 3.2 in more details.

3. *Perform the GIS operation on all instances and on the ground truth.*
   First, for each representation *r*, the volume of a building *b* is calculated: V_r^b. Second, for each building, its ground truth V_{gt}^b is computed. We have implemented this in the Feature Manipulation Engine (FME) by Safe Software Inc.

4. *Evaluation of the differences.*
   In this step we compare the results and compute the errors. This part is more related to the used GIS operation, and the used values may depend on it. For each building *b*, and for each of the used representations, the error in the volume is calculated, along with the relative error:

   $\varepsilon_r^b = V_r^b - V_{gt}^b$
   $\mu_r^b = (V_r^b - V_{gt}^b) / V_{gt}^b$

   The relative error may be a useful indicator to assess the discrepancies because it gives the relative difference between the results, which may be more relevant in the context of a certain use-case since it does not depend on the size of a building. Afterwards, for each of the geometric references two root mean square error (RMSE) values are computed:

   $$\mathrm{RMSE}_{\varepsilon_r} = \sqrt{\frac{\sum_{b=1}^{n}\left(\varepsilon_r^b\right)^2}{n}}$$

$$\mathrm{RMSE}_{\mu_r} = \sqrt{\frac{\sum_{b=1}^{n}\left(\mu_r^b\right)^2}{n}}$$

where n is the number of buildings. As another error metric, we sum the volumes of all buildings for each representation $\sum_b^n V_r^b$, and compare them to the ground truth $\sum_b^n V_{gt}^b$ by computing their ratio.

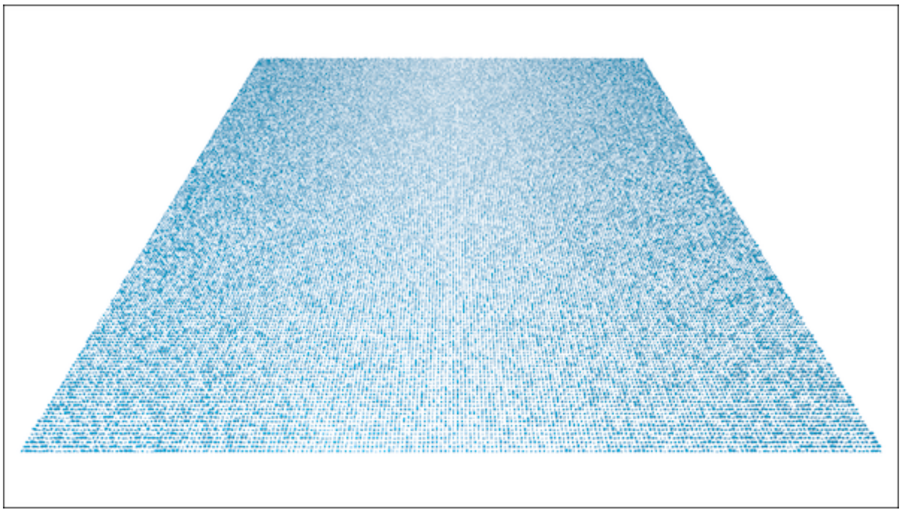
The presented methodology is focused towards the computation of volumes for buildings, but it can be applied to virtually any other (3D) GIS operation.

3.2 Datasets

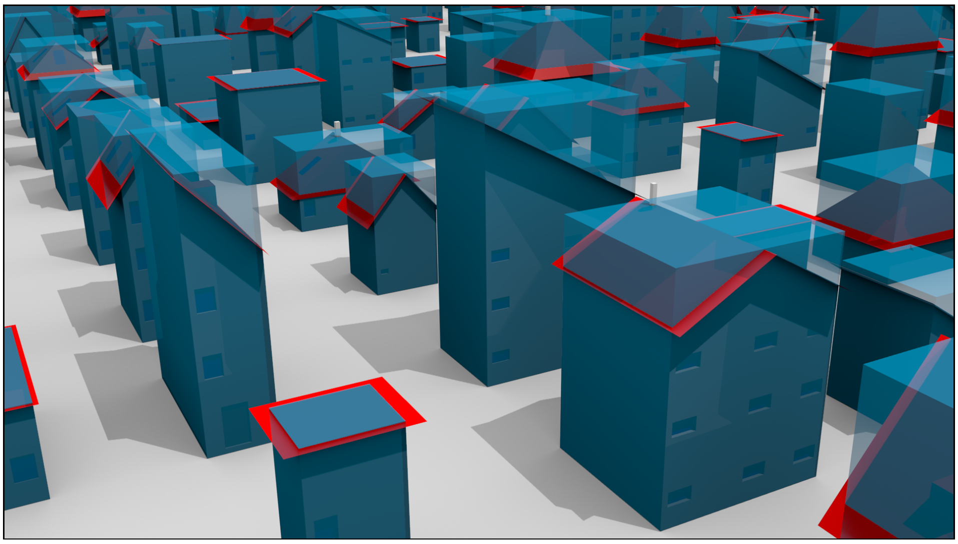
We have generated 40000 buildings with our software prototype ``Random3Dcity'', and represented them in seven LOD1 CityGML datasets, one for each of the geometric variants described in Sec. 2 and Tab. 1. Further, a detailed LOD3 model has been generated as the *ground truth* model for which the reference volume can be calculated. Because this dataset is synthetic and it is generated from a set of known building parameters, we may consider the LOD3 model as the exact representation and therefore use it as a reference value. The buildings being randomly generated, means that there are no two buildings that are the same, and it makes our method statistically unbiased.

Figure 2 shows the visualisation of the randomly generated datasets. In Fig. 2a the whole extent of the dataset LOD1-H5 is shown (in a grid of 200x200 buildings), while the Fig. 2b shows a close-up with two datasets: the LOD3 and the LOD1-H5. In the latter figure, since the top surface of LOD1 represents the top of the roof, notice that some of the chimneys of the LOD3 model protrude the block model.

The magnitude of n = 40000 was determined by running the algorithms multiple times. With this value, the discrepancies between instances were negligible.



(a) One of the datasets (LOD1-H5).



(b) Close-up of the dataset LOD1-H5 with the overlapped LOD3 dataset.

Figure 2: Visualisation of the datasets generated with our engine ``Random3Dcity''.

3.3 Results and their interpretation

We present the results of the computation of volumes in Tab. 2 and in Fig. 3. It can be seen that the LOD1 generated when taking the reference H3 for the height of the top surface, causes the smallest errors in the computation of volumes, therefore we can conclude that is the most suitable for this operation. The errors caused by other references may be too significant for a number of use-cases, raising the importance of the awareness of the used reference.

Table 2: Results of the simulation for the volume computation (n = 40000). The footprint represents the actual position of walls (comparable to the footprint obtained from cadastral sources). We have also computed the data for the ground truth for self-validation purposes.

Model (LOD and top reference)	RMSE [m³]	RMSE [%]	$\sum V_r$ [m³]	$\sum V_r/V_{gt}$ [%]
LOD1 – H0	82.56	25.95	10109678.38	80.11
LOD1 – H1	67.95	20.44	10568599.85	83.74
LOD1 – H2	29.70	9.60	11764718.94	93.22
LOD1 – H3	8.89	3.20	12592239.21	99.78
LOD1 – H4	26.80	8.23	13419759.49	106.33
LOD1 – H5	81.35	24.85	15115676.86	119.77
LOD1 – H6	85.87	25.83	15261200.42	120.92
LOD3 (ground truth)	0.00	0.00	12620500.20	100.00

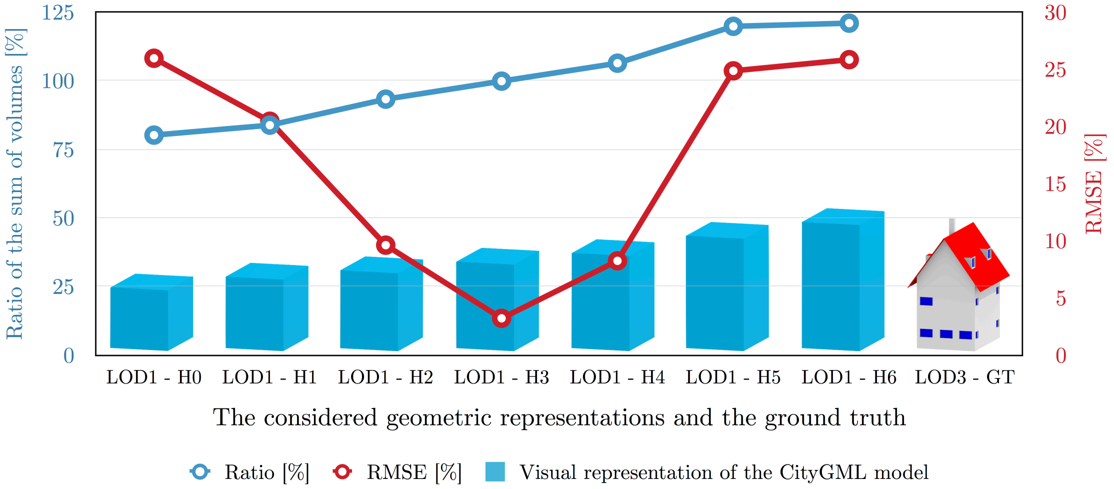
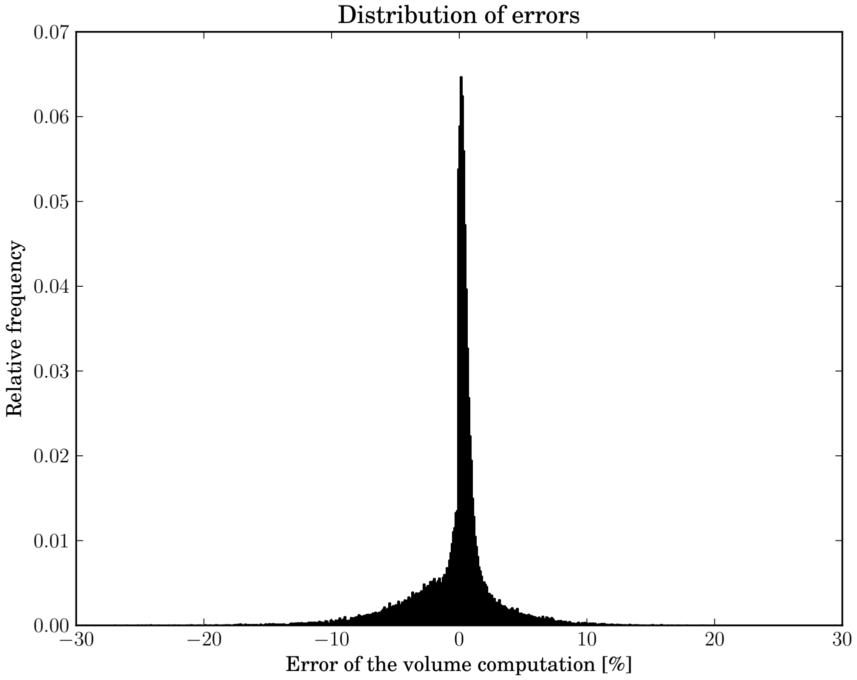


Figure 3: Results of the computation of volumes with the errors depending on the used geometric representation.

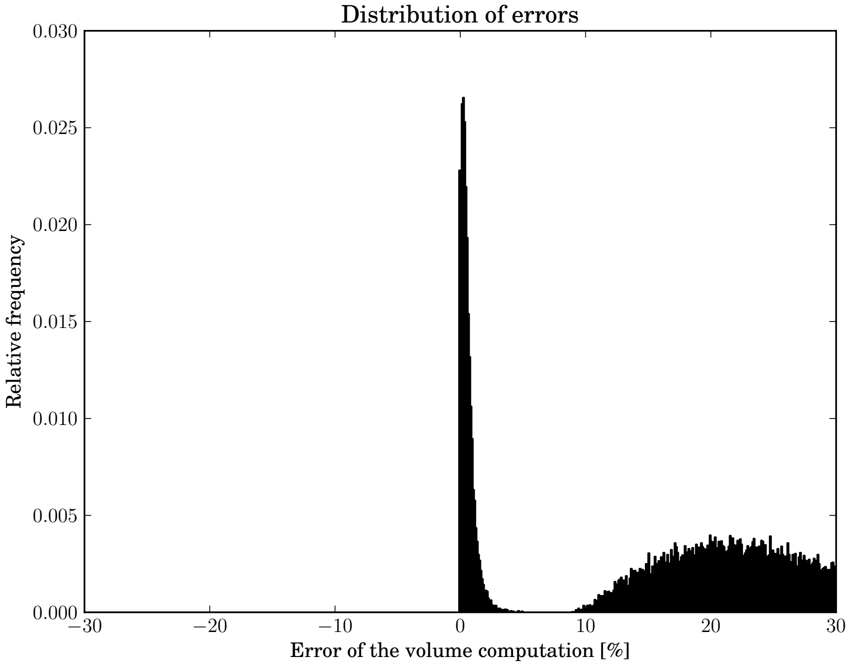
The small difference in error between the references H5 and H6 may be attributed to the fact that not all buildings have chimneys and other superstructures that are higher than the top of the roof, and when they have, in many cases they rather insignificantly increase the height of the block model, leading to a relatively small increase in the volume.

In Fig. 4 we focus on the distribution of individual errors of the volumes within a reference for the better understanding of the deviations. This is conveyed with the histograms of relative errors and it is expressed in percentages. Fig. 4a shows the distribution for H3. The peak at 0 can be explained by the presence of flat roofs, which is the optimal case for this reference. In Fig. 4b the distribution for

H5 is shown, and again a favourable peak at 0 caused by flat roofs is present, but on the right side it also shows the distribution of deviations of non-flat roofs. This difference is particularly manifested in shed roofs because of the higher overhangs.



(a) LOD1-H3



(b) LOD1-H5

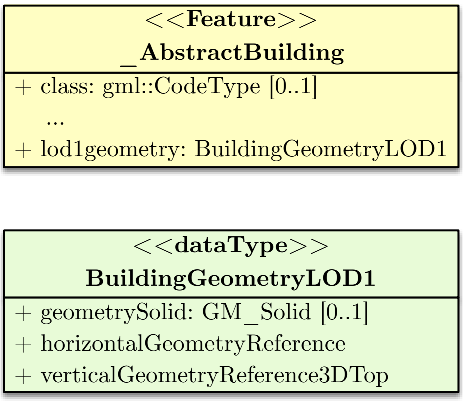
Figure 4: Histogram of the relative errors (μ_r) in the computation of volumes comparing to the ground truth for two references.

4. CityGML implementation

In a recent paper describing the aforementioned interoperable building model of the European Union, Gröger and Plümer [23] state the possibility of introducing the metadata of the geometric representations to CityGML. In this Section we follow their conclusion, and we investigate a possible way to implement the metadata which denote the reference for the elevation of the top surface in LOD1, and also the reference for the horizontal footprint.

The INSPIRE building data specification provides a CityGML Application Domain Extension (ADE) for the Core 3D profile of the INSPIRE building model, but it might be desirable to encode this information in the core of CityGML.

Fig. 5a shows a possible solution through a UML extension of the CityGML data type by adding the attributes describing the references. This solution is strongly influenced by the INSPIRE Data Specification on Buildings, potentially resulting in interoperability. In Fig. 5b we show the implementation by hard-coding the extension as attributes of the tag of the LOD1 geometric representation.


<<Feature>>
_AbstractBuilding
+ class: gml::CodeType [0..1]
...
+ lod1geometry: BuildingGeometryLOD1
<<dataType>>
BuildingGeometryLOD1
+ geometrySolid: GM_Solid [0..1]
+ horizontalGeometryReference
+ verticalGeometryReference3DTop

(a) UML diagram.

```
<cityObjectMember>
 <bldg:Building>
  <bldg:lod1Solid verticalGeometryReference3DTop="topOfConstruction">
   <gml:Solid>
    <gml:exterior>
     <gml:CompositeSurface>
      <gml:surfaceMember>
       <gml:Polygon>
        <gml:exterior>
         <gml:LinearRing>
          <gml:posList>...
```

(b) GML code excerpt.

Figure 5: Implementation of the horizontal and vertical geometric references for the LOD1 model in CityGML.

In this way, the reference applies to all features within a representation. An alternative would be to attach the metadata to each feature separately, however, we do not expect cases of mixing different variants between features within the same dataset to be occurring in practice.

This introductory work is a first step towards the implementation, and therefore it leaves open questions for discussion and future work. For instance, a matter not regarded by INSPIRE is the cardinality between LODs and their geometric references, i.e. should CityGML enable the representation of two equal LODs with different variants. This is currently not possible because of the 1:1 relation between the building feature and its LOD*x* geometry. While this may not be frequent and not feasible, it might be an interesting point of discussion.

5. Conclusions

In this paper we have thoroughly examined the ubiquitous LOD1 block model, and we have focused on the geometric reference for the height of the top surface, as its most prominent ambiguity. We have shown that, while LOD1 is the simplest volumetric form in CityGML, it is surrounded by potential ambiguities because its geometry may be modelled in many variants. This is caused by the shortcoming of the standard when it comes to describing the multiple geometric representations, and because it is not possible to store such information in the metadata. Since the LOD concept of CityGML is intended to reflect the spatio-semantic complexity of a model, the geometric references are not related to the level of detail, and they are rather a closely related category of metadata, that it is important to note because it may influence all downstream applications.

Our contribution is that we have stressed this topic, we have shown with experiments the potentially drastic difference between the variants with respect to a GIS operation (volume computation), and we have explained how to determine the optimal reference with respect an operation. We have discovered in a simulation that, when using block models in the computation of volumes, the variant where the top represents the half of the roof height is the most suitable variant. The framework that we present can be used for determining the best variant in other GIS operations.

We propose adding metadata to CityGML, and we have shown a provisional solution, which may be relevant for the developers of the next version of the standard (v. 3.0 due in 2016; see [43], [44]). Further, we have refined the INSPIRE building model's reference `generalRoof` with three new references, that are especially relevant for extrusion to point clouds, the frequent method for generating LOD1 models.

For future work we plan to simulate a point cloud from a LiDAR survey to find the exact roof median as an alternative to the half of the roof (point H3), to extend this work to LOD2 and the varying representations of the footprints, and to involve more GIS operations and use-cases.

Acknowledgements

We thank Safe Software Inc. for providing us with a licence for FME. Further, we appreciate the information obtained from institutions and companies about their modelling practices which served as input for this paper.

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs. (Project code: 11300)

References

[1] Open Geospatial Consortium, “OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0,” OGC 12-019, Apr. 2012.

[2] T. H. Kolbe, C. Nagel, and A. Stadler, “CityGML–OGC Standard for Photogrammetry,”. Proceedings of the 52nd Photogrammetric Week '09, Stuttgart, Germany, 2009, vol. 9, pp. 265–277.

[3] G. Gröger and L. Plümer, “CityGML – Interoperable semantic 3D city models,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 71, pp. 12–33, Jul. 2012.

[4] H. Ledoux and M. Meijers, “Topologically consistent 3D city models obtained by extrusion,” *International Journal of Geographical Information Science*, vol. 25, no. 4, pp. 557–574, Apr. 2011.

[5] S. U. Baig and A. Abdul-Rahman, “Generalization of buildings within the framework of CityGML,” *Geo-spatial Information Science*, vol. 16, no. 4, pp. 247–255, Dec. 2013.

[6] A. A. Diakité, G. Damiand, and D. Van Maercke, “Topological Reconstruction of Complex 3D Buildings and Automatic Extraction of Levels of Detail,”. Proceedings of the Eurographics Workshop on Urban Data Modelling and Visualisation, Strasbourg, France, 2014, pp. 25–30.

[7] M. El-Mekawy, A. Östman, and K. Shahzad, “Towards interoperating CityGML and IFC building models: a unified model based approach,”. Proceedings of the 5th International 3D GeoInfo Conference. Advances in 3D Geo-Information Sciences.
Lecture Notes in Geoinformation and Cartography, 2011, pp. 73–93.

[8] B. Mao, L. Harrie, and Y. Ban, "Detection and typification of linear structures for dynamic visualization of 3D city models," *Computers, Environment and Urban Systems*, vol. 36, no. 3, pp. 233–244, 2012.

[9] P. J. Morton, "Virtual City Models - A Global Perspective,". EuroSDR/OGC Workshop CityGML in national mapping, Institut national de l'information géographique et forestière, Paris, France, 2013, pp. 1–4.

[10] L. van den Brink, J. Stoter, and S. Zlatanova, "Establishing a national standard for 3D topographic data compliant to CityGML," *International Journal of Geographical Information Science*, vol. 27, no. 1, pp. 92–113, Jan. 2013.

[11] A. Czerwinski, S. Sandmann, S.-M. Elke, and L. Plümer, "Sustainable SDI for EU noise mapping in NRW -- best practice for INSPIRE," *International Journal for Spatial Data Infrastructure Research*, vol. 2, no. 1, pp. 1–18, 2007.

[12] H. R. Ranjbar, A. R. Gharagozlou, and A. R. V. Nejad, "3D Analysis and Investigation of Traffic Noise Impact from Hemmat Highway Located in Tehran on Buildings and Surrounding Areas," *Journal of Geographic Information System*, vol. 4, no. 4, pp. 322–334, Aug. 2012.

[13] A. Strzalka, N. Alam, E. Duminil, V. Coors, and U. Eicker, "Large scale integration of photovoltaics in cities," *Applied Energy*, vol. 93, pp. 413–421, May 2012.

[14] J. M. Bahu, A. Koch, E. Kremers, and S. M. Murshed, "Towards a 3D spatial urban energy modelling approach,". Proceedings of the ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 2013, vol. 2, pp. 33–41.

[15] H. Tomić, M. Roić, and S. Mastelić Ivić, "Use of 3D cadastral data for real estate mass valuation in the urban areas,". Proceedings of the International Federation of Surveyors 3rd International Workshop on 3D Cadastres: Developments and Practices, Shenzhen, China, 2012, pp. 73–86.

[16] INSPIRE Thematic Working Group Buildings, "D2.8.III.2 INSPIRE Data Specification on Buildings – Technical Guidelines," Dec. 2013.

[17] R. San José, J. L. Pérez, and R. M. González-Barras, "3D Visualization of Air Quality Data,". Proceedings of the 11th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'11), Riga, Latvia, 2011, pp. 1–9.

[18] T. Götzelmann, R. Guercke, C. Brenner, and M. Sester, "Terrain-dependent aggregation of 3D city models,". ISPRS workshop on Quality, Scale and Analysis Aspects of Urban City models, ISPRS Archives, Lund, Sweden, 2009, vol. 2, p. 5.

[19] M.-O. Löwner, J. Benner, G. Gröger, and K.-H. Häfele, "New Concepts for Structuring 3D City Models – an Extended Level of Detail Concept for CityGML Buildings,". Lecture Notes in Computer Science. Proceedings of the 13th International Conference Computational Science and Its Applications – ICCSA 2013, Ho Chi Minh City, Vietnam, 2013, vol. 7973, pp. 466–480.

[20] F. Biljecki, H. Ledoux, J. Stoter, and J. Zhao, "Formalisation of the level of detail in 3D city modelling," *Computers, Environment and Urban*

Systems, vol. 48, pp. 1–15, Nov. 2014.

[21] J. Benner, A. Geiger, G. Gröger, K.-H. Häfele, and M.-O. Löwner, "Enhanced LOD concepts for virtual 3D city models,". ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 2013, vol. 2, pp. 51–61.

[22] M. Brasebin, J. Perret, S. Mustière, and C. Weber, "Measuring the impact of 3D data geometric modeling on spatial analysis: Illustration with Skyview factor,". Usage, Usability, and Utility of 3D City Models -– European COST Action TU0801, Nantes, France, 2012, pp. (02001)1–16.

[23] G. Gröger and L. Plümer, "The Interoperable Building Model of the European Union," in *Lecture Notes in Geoinformation and Cartography. Geoinformation for Informed Decisions*, no. 1, A. Abdul-Rahman, P. Boguslawski, F. Anton, M. N. Said, and K. M. Omar, Eds. Johor Bahru, Malaysia: Springer International Publishing, 2013, pp. 1–17.

[24] R. Guercke, T. Götzelmann, C. Brenner, and M. Sester, "Aggregation of LoD 1 building models as an optimization problem," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 2, pp. 209–222, Mar. 2011.

[25] K.-H. Häfele, *CityGML Model of the FJK-Haus*. Institut für Angewandte Informatik (IAI), Karlsruher Institut für Technologie, 2011, pp. 1–9.

[26] S. He, G. Moreau, and J.-Y. Martin, "Footprint-Based Generalization of 3D Building Groups at Medium Level of Detail for Multi-Scale Urban Visualization," *International Journal On ADvances in Software*, vol. 5, no. 3, pp. 378–388, Dec. 2013.

[27] H. Fan and L. Meng, "Automatic derivation of different levels of detail for 3D buildings modeled by CityGML,". Proceedings of the 24th International Cartographic Conference, Santiago, Chile, 2009, p. 10.

[28] Ordnance Survey, "OS MasterMap Topography Layer. User guide and technical specification," Ordnance Survey, Mar. 2014.

[29] S. Oude Elberink, J. Stoter, H. Ledoux, and T. Commandeur, "Generation and Dissemination of a National Virtual 3D City and Landscape Model for the Netherlands," *Photogrammetric Engineering and Remote Sensing*, vol. 79, no. 2, pp. 147–158, 2013.

[30] SwissTopo, "swissBUILDINGS3D. Vereinfachte 3D-Gebäude der Schweiz," Federal Office of Topography (SwissTopo), Sep. 2010.

[31] M. Goetz, "Towards generating highly detailed 3D CityGML models from OpenStreetMap," *International Journal of Geographical Information Science*, vol. 27, no. 5, pp. 845–865, May 2013.

[32] M. Over, A. Schilling, S. Neubauer, and A. Zipf, "Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany," *Computers, Environment and Urban Systems*, vol. 34, no. 6, pp. 496–507, Nov. 2010.

[33] S. He, G. Besuievsky, V. Tourre, G. Patow, and G. Moreau, "All range and heterogeneous multi-scale 3D city models,". Usage, Usability, and Utility of 3D City Models -– European COST Action TU0801, Nantes, France, 2012.

[34] M. Ying, J. Jingjue, and B. Fulin, "3D-City Model supporting for CCTV monitoring system,". International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS Commission IV Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002, vol. 4, p. 4.

[35] F. Albrecht, J. Moser, and I. Hijazi, "Assessing façade visibility in 3D city models for city marketing,". International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 2013, vol. 2, pp. 1–5.

[36] R. Kaden and T. H. Kolbe, "City-wide total energy demand estimation of buildings using semantic 3D city models and statistical data,". ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 2013, vol. 2, pp. 163–171.

[37] D. Perez, J. H. Kämpf, and J.-L. Scartezzini, "Urban Area Energy Flow Microsimulation for Planning Support: a Calibration and Verification Study," *International Journal On Advances in Systems and Measurements*, vol. 6, no. 3, pp. 260–271, Dec. 2013.

[38] R. Boeters, "Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination,". MSc thesis, Delft University of Technology, Delft, the Netherlands, 2013.

[39] D. Fisher-Gewirtzman, A. Shashkov, and Y. Doytsher, "Voxel based volumetric visibility analysis of urban environments," *Survey Review*, vol. 45, no. 333, pp. 451–461, Nov. 2013.

[40] F. Biljecki, H. Ledoux, and J. Stoter, "Error propagation in the computation of volumes in 3D city models with the Monte Carlo method,". Proceedings of the ISPRS/IGU Joint International Conference on Geospatial Theory, Processing, Modelling and Applications, Toronto, Canada, 2014.

[41] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, 2nd ed. Wiley, 2008.

[42] M. Kada, "Scale-dependent simplification of 3D building models based on cell decomposition and primitive instancing,". Lecture Notes in Computer Science. Proceedings of the 8th International Conference on Spatial Information Theory (COSIT 2007), Melbourne, Australia, 2007, vol. 4736, pp. 222–237.

[43] M.-O. Löwner, J. Benner, and G. Gröger, "Aktuelle Trends in der Entwicklung von CityGML 3.0,". Geoinformationen öffnen das Tor zur Welt, 34. Wissenschaftlich-Technische Jahrestagung der DGPF, Hamburg, Germany, 2014, vol. 23.

[44] T. Machl, "Minutes of the International OGC, SIG 3D and TUM Workshop on Requirements for CityGML 3.0," *International OGC, SIG 3D and TUM Workshop on Requirements for CityGML 3.0*, vol. 2. Munich, Germany, pp. 1–28, 21-Jun-2013.

A general data modeling framework for 3D geographic information systems (3D GIS)

Shuqing Zhang [a,b,*], Lihua Liu [a,b], Junyan Zhang [a], and Xiangcong Chen [a,b]

[a] Northeast Institute of Geography and Agroecology, Chinese Academy of Sciences, Changchun 130102, China;

[b] University of Chinese Academy of Sciences, Beijing, China

ABSTRACT: To represent the geometrically complex and diverse three-dimensional world, a flexible and extensible general data modeling framework (GDMF) is established, which integrates two structure types: *discrete* and *function structure*, covering the well-designed boundary description, constructive solid geometry (CSG) and functional representation. Each of the structure types may contain several sub-classes of structure types, for example, discrete structure consists of discrete curved surface (*DCS*), discrete body (*DB*), and so on. These powerful structures can either uniquely or jointly represent an object (or sub-object), thus allowing us to flexibly and effectively capture various geometric regularities. Primary applications show that both natural and man-made spatial objects are effectively and efficiently represented.

Keywords: 3D GIS, data model, data structure, discrete structure, function structure,

1. Introduction

3D geographic information systems (GIS) data model is the abstraction of the 3D reality, and it plays a critical role in the functions of 3D GIS. The core issue of conceptual 3D modeling is representing the geometry [Zlatanova and Tempfli 2000]. During the past decades, many 3D conceptual models have been proposed in the literature. Yet, when dealing with the diverse and complicated three-dimensional world (e.g. man-made buildings possibly differing from culture to culture [Goodchild 2009]), current data models suffer from the following major issues:

Firstly, the design of many current 3D data models is often related to the specific requirements of a particular domain application [Zlatanova et al. 2004], and most models are, hence, application-specific [Zhou et al. 2008]. For example, some models, such as the 3D geo-database topological and geometric model [Mallet 2002] and the Generalized Tri-Prism (GTP) [Wu 2004], are designed for modeling geological objects. Other models, such as the Urban Data Model (UDM)

[Coors 2003] and the semantics-based 3D dynamic hierarchical house property model [Zhu and Hu 2010], intend to model urban environments. These application-specific data models, however, are unable to meet the needs for problems that cut across application domains and spatial domains [Zhou *et al*. 2008, Yuan *et al*. 2011]. Functional emphases for 3D applications can be different as well. In some applications, for example, the efficiency of visualization and/or analysis might be highlighted, whereas the convenience of data input might be preferred for other GIS applications. A data modeling framework that integrates different application-specific data models and satisfies diverse GIS applications is, therefore, desirable [Li and Li 1998]. And this is especially true, along with the implementation of cloud computing in GIS which relies on sharing of resources (including data source).

Secondly, current data models--all belonging to discrete data models--suffer from a severe data redundancy and inefficiency in data representation, due to the primitives used—*node*, *edge*, *face*, and *body/solid*. For instance, the Formal Data Structure (3D FDS) consists of primitives *nodes*, *arcs*/*edges*, and *faces* [Molenaar 1991]; the Tetrahedron-based model (TEN) uses primitives *nodes*, *arcs*, *triangles* and *tetrahedrons* [Pilouk 1996]; and the Simplified Spatial Model (SSM) is composed of primitives *nodes* and *faces* [Zlatanova 2000]. Theoretically, although any complex object can be approximated by these primitives, severe data redundancy may result. This is because that the geometric units of the current primitives are usually much smaller than those perceived by human beings, and accumulation of primitives is often required to represent a spatial entity that is in possess of semantical information. For instance, two triangles might be used to describe a window. Thus a large number of primitives are needed to represent a complex object, let alone to describe massive objects, which would bring about problems for data management, data query and analysis. To clarify, geometric primitives have been extended to a certain degree for the representation of a surface by the standard ISO 19107 Spatial Schema [Herring 2001], but its modeling efficiency is still rather limited. For example, features having the same geometric and semantic properties, e.g. windows in the same floor (or in different floors) of a building, are unable to be represented uniformly. It is necessary to redefine basic elements so as to effectively represent our complex three-dimensional world.

Third, the types of representations are insufficient. In the real world, certain man-made features such as a circular plaza, a domical building, a parabolic cooling tower in a power plant and so on, are perfectly described with theoretical functions. But all current data models belong to discrete data models which are comprised of discrete entities like nodes and triangles, other representations such as functional representations are not considered. On one hand, such a discrete data model could sometimes result in massive volumes of data, for instance, the file sizes of CityGML [Gröger *et al*. 2012] are much larger than those of Building Information Model (BIM) which has volumetric, parametric representations [Berlo and Laat 2011]; on the other hand, accuracy errors also be produced due to the discretization of theoretical functions [Kuttig 2003]. What's more, due to the lack of parametric/functional representation in GIS, a major problem of information lossless transformation exists between GIS and other software (like BIM and CAD). Therefore, to effectively model real-world entities and phenomena and overcome the obstacle of data transformation between GIS and other software, 3D models must be extended to incorporate new representations such as parametric shapes, freeform curves and surfaces, as discussed by [Breunig and Zlatanova 2006, 2011].

To address the above mentioned problems, a flexible and extensible general data modelling framework (GDMF) is proposed, in which multiple well-designed expressions built upon two types of structure (*discrete structure* and *function structure*) are integrated. A spatial object can either uniquely or jointly be described with multiple expressions to effectively capture the various geometric regularities of the object, thus offering powerful representation of 3D spatial data. Note that traditional GIS data models belong to a subset of *discrete structure*. More modeling strategies are hence provided to effectively and efficiently represent the diverse and complicated world, and to meet the diverse demands from a wide range of GIS users and applications.

2. General data modelling framework (GDMF)

2.1 *Basic concepts and naming convention of variables*

Concepts to develop the general data model framework (GDMF) will be defined here.

Facet and **PC-surface**. A ***facet*** is a bounded planar polygon in 3D space, while a parameter curve surface (denoted as ***PC-surface***) refers to a small curved surface that can be defined by a function. A triangle in 3D can also be considered as a facet in this paper.

Sub-object. A spatial object may be dissected into a number of smaller objects that are macro-geometrically distinct and/or semantically different from each other. These smaller objects are called ***sub-objects***. There are four basic shape types (denoted as ***shapeType***) of objects: *point*, *line*, *surface* and *solid*.

Composite object. If an object or sub-object consists of more than one basic shape type, it is called a ***composite object***.

Point :

3D point: $P = POINT\ \{x,y,z\}$

2D point : $P2D=POINT2D\{x,y\}$

Where x, y and z are the coordinates in 2D or 3D space accordingly.

Naming convention of variables. In order to concisely represent our data model, conventions are made as: 1) a lowercase word represents a real value, e.g. *shapeType*; 2) an uppercase word stands for a set, e.g. *POINT*; and 3) a squiggle word denotes a family set (i.e. set of a structure or a tuple), e.g., $\mathcal{OBJ}$. The squiggle words used in this paper and their ordinary forms are presented in a comparative table at the end of this paper.

2.2 *Discrete structure*

Two structure types (*discrete structure* and *function structure*) are defined first, and then the data model of object is presented by integrating the proposed structure type.

In *discrete structure*, an object or sub-object might be composed of a number of smaller objects, or it might be derived from set operation(s) on a set of sub-objects. Two sub-classes of structures are included in *discrete structure*, namely *discrete unit* (*DU*) and *discrete object structur*e which will be detailed hereafter.

2.2.1 *Discrete unit* (*DU*) *structure*

As the major form of *discrete structure*, *discrete unit* (*DU*) *structure* plays an important role in data representation, thus being emphasized here. *DU structure* includes two structure forms: *Discrete curved surface* (*DCS*) and *discrete body* (*DB*).

Discrete curved surface (***DCS***). Being boundary representation, *DCS structure* describes a curved surface with vertices, facets and their explicit relations using various strip types, including such as *triangle fan* (Fig.1-1), *rectangle strip* (Fig.1-2), and *multi-point consecutive strip* (Fig.1-3), *etc*. Each strip type represents compactedly a specific *DCS*. Due to the length limitation, only *triangle fan*, *triangle strip* and *multi-point consecutive strip* are described in detail in this paper hereafter:

(1) ***Triangle fan***. The first vertex and each two neighboring vertices in a data chain form a triangle. In Fig. 1-1, for instance, vertices are stored in order: $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$ (i.e. $\mathcal{P}$). The number of vertices (*np*) and the number of triangles (*ns*) has the following relations,

$$np = ns + 2; \qquad (1)$$

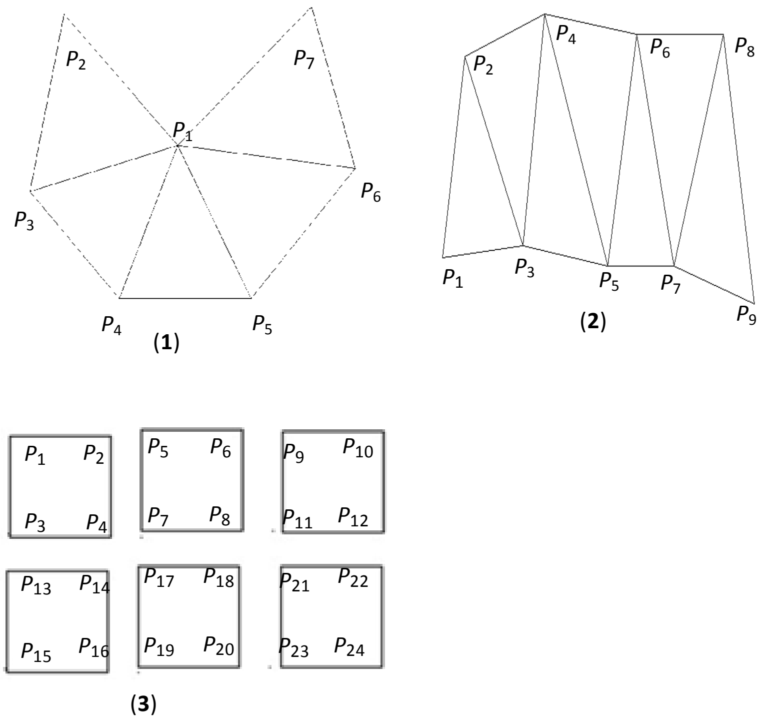

P_1
P_2
P_3
P_4
P_5
P_6
P_7
(1)
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
(2)
P_1 P_2 P_3 P_4
P_5 P_6 P_7 P_8
P_9 P_{10} P_{11} P_{12}
P_{13} P_{14} P_{15} P_{16}
P_{17} P_{18} P_{19} P_{20}
P_{21} P_{22} P_{23} P_{24}
(3)

Fig. 1. Different trip types of *DCS structure*

(1) *triangle fan*, (2) *triangle strip*, and (3) *multi-point consecutive strip*

A triangle fan can be represented as:

$$\mathscr{DCS} = \{stripType, np, \mathscr{P}\} \qquad (2)$$

where *stripType* is "*triangle fan*", $\mathscr{P}$ is the family set of vertices.

(2) ***Triangle strip***. In triangle strip, every three neighboring vertices in a data chain constitute a triangle (Fig. 1-2). The amount of storage space of a triangle strip is the same as that of a triangle fan. A triangle strip can also be represented with Eq. 2, and the relation between *np* and *ns* is the same as Eq. 1.

(3) ***Multi-point consecutive strip*** represents facets (i.e. planar polygons or triangles) that all have the same number of edges, but may separate from each other as shown by Fig. 1-5. This strip type takes the form of:

$$\mathscr{DCS} = \{stripType, ns, np, \mathscr{P}\} \qquad (3)$$

Where $\mathscr{P}=\{\{P_{1,1}, P_{1,2},\ldots,P_{1,np}\}, \ldots, \{P_{i,1}, P_{i,2},\ldots,P_{i,j},\ldots, P_{i,np}\}, \ldots, \{P_{ns,1}, P_{ns,2},\ldots,P_{ns,np}\}\}$ ($1 \le i \le ns$, $1 \le j \le np$), *ns* is the number of facets, and *np* is the number of edges of any facet. Fig. 1-5 illustrates a *multi-point consecutive strip* composed of 6 squares. Windows or doors in a building might be collectively represented with this strip type.

DCS can not only describe a surface object or sub-object, but can enclose a volumetric object or sub-object. Note a planar face is also regarded as a curved surface.

Discrete body (***DB***). A volumetric object or sub-object might be constructed with a number of atomistic volumetric shapes i.e. discrete bodies (*DB*s), each of which is described with discrete data (e.g. vertices), as well as certain geometric parameters of the shape. The definition of *DB structure* thus takes the solid feature's shape property into account. A wide range of solid shapes in possess of "shape semantics", such as *right prism*, *ridge* (Fig. 2), *regular hexahedro*n, *wedge*, etc., have already been utilized in architecture design, and they could then be applied to the establishment of *DB structure*. Once a *DB structure* is defined, it can then serve as a building block to construct a volumetric object or sub-object.

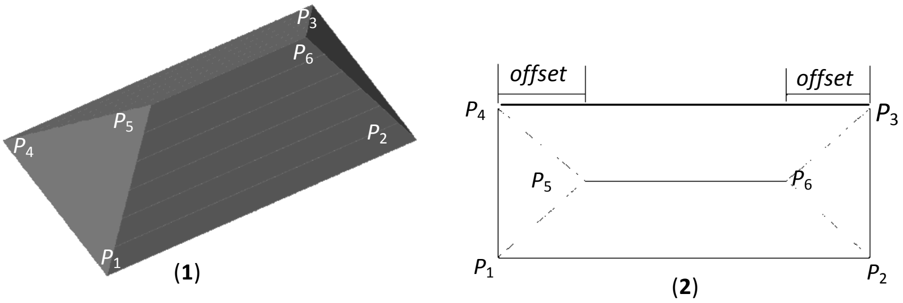


Fig. 2, R*idge* in three dimension (1) and its two-dimensional drawing (2)

Right prism is used to exemplify the definition of a *DB structure* here. Formed by its 2D footprints ($\mathscr{P}2\mathscr{D}$,) and parameters including z_0 (the elevation of the *right prism*'s bottom) and *h* (the absolute height of the *right prism*), a *right prism* is described as:

$\mathscr{DB}$= {*"right prism", np*, {$\mathscr{P}2\mathscr{D}$, z_0, *h*}}, (4)

where *np* is the number of the footprints.

Discrete unit (***DU***) *structure* ($\mathscr{DUS}$). In *DU structure*, all discrete parts of an object or sub-object are all represented with an arbitrary number of *discrete curved surface*s (*DCS*s) or *discrete bodie*s (*DB*s), which is modelled as:

$\mathscr{DUS}$= {*ndu*, $\mathscr{DCS}$(or $\mathscr{DB}$)}, (5)

where *ndu* is the number of *DCSs* or *DB*s, $\mathscr{DCS}$the family set of *DCS*s, and $\mathscr{DB}$ the family set of *DB*s.

2.2.2 *Discrete object structure*

Discrete object structure (denoted as $\mathscr{DISCRETE\_OBJ}$) represents the structure data of an object (or sub-object) composed of more than one sub-objects (including auxiliary sub-objects), e.g., a surface consisting of a set of sub-surfaces. $\mathscr{DISCRETE_OBJ}$is expressed as:

$\mathscr{DISCRETE}\_\mathscr{OBJ}$={*nso*, $\mathscr{SO}$} (6)

Here, *nso* is the number of sub-objects ($\mathscr{SO}$), and $\mathscr{SO}$ the family set of sub-objects' data. Union set operation(s) exists in the sub-objects in a *discrete object structure*.

2.3 *Function structure*

Function structure represents an object or sub-object with a function:

$\mathscr{FUNCTION}$ = {*functionName*, *PARAMETER*}, (7)

where $\mathscr{FUNCTION}$ refers to *function structure*; *functionName* is the name of the function; *PARAMETER* is the set of parameters, in which variables (including operators) might be included.

A large number of functions of any basic shape type, e.g. sphere, cone, hyperboloid, cylinder, etc., could be defined in *function structure*. Basic elements of discrete structure data (e.g. a *DCS* or *DB*) as well as operators can also be regarded as parameters of a function. By incorporating *multi-point consecutive strip* (Eq. 3) as a parameter, a function called "windows (or doors) looping (WDL) function" is presented here, to collectively describe planar windows (or doors) that distribute in the same pattern on a building's different floors.

$\mathscr{WDL}$= {"*window or door looping* ", *UN*, h_0, *nf* , *h*, $\mathscr{DCS}$} (8-1)

={"*window or door looping* ", *UN*, h_0, *nf* , *h*,

{"*multi-point successive strip*", *ns*, *np*, $\mathscr{P}$}} (8-2)

Where, "*window or door looping*" is the function name. *UN* (a *POINT* data type) refers to the unit normal of the surface. h_0 is the elevation of the windows or doors of the floor to start (i.e. start floor), *nf* is the number of the floors, and *h* is the floor height. $\mathscr{DCS}$in Eq. 8-2 is the data of the start floor's windows (or doors) in *multi-point successive strip*.

2.4. Models of object and sub-object

An object ($\mathscr{OBJ}$) of any shape type (e.g. *point*, *line*, *surface*, *solid*, or *composite*) is represented with data header ($\mathscr{H}\_\mathscr{DATA}$) and graphics data ($\mathscr{G}$-$\mathscr{DATA}$). $\mathscr{H}\_\mathscr{DATA}$ comprises the object's serial number (*objectNumber*), the shape type (*shapeType*) and the bounding box (BOX) of the object; while $\mathscr{G}$-$\mathscr{DATA}$ includes the structure type (*structureType*) (e.g. *DCS*, *DB*, etc) and the corresponding structure data ($\mathscr{S}\_\mathscr{DATA}$). A sub-object ($\mathscr{SO}$) is actually the graphics data ($\mathscr{G}$-$\mathscr{DATA}$.)

Thus object ($\mathscr{OJB}$) and sub-object ($\mathscr{SO}$) is respectively modeled as:

$\mathscr{OJB}$ = {$\mathscr{H}\_\mathscr{DATA}$, $\mathscr{G}$-$\mathscr{DATA}$}

= {{*objectNumber*, *shapeType*, *BOX*}, {*structureType*, $\mathscr{S}\_\mathscr{DATA}$}} (9-1)

$$\mathcal{H}\_\mathcal{DATA} = \{objectNumber, shapeType, BOX\} \qquad (9\text{-}2)$$

$$\mathcal{SO} = \mathcal{G}\text{-}\mathcal{DATA} = \{structureType, \mathcal{S}\_\mathcal{DATA}\} \qquad (9\text{-}3)$$

Here, data header is assigned to an object, rather than to a sub-object, while the graphics data of the object is equal to a sub-object. Structure type (*structureType*) of a sub-object is stored outside of its structure data. In the data model, heterogeneous structures (e.g. *DCS, DB, function*, etc), are encoded with structure type (*structureType*), and made uniform as structure data ($\mathcal{S}\_\mathcal{DATA}$), whereas the basic models (i.e. MSB model) of object and sub-object remain unchanged. As the object—structure relationship (a tree-like structure) schema (Fig. 3) demonstrated, an object or sub-object can be directly described with any structure. In this case, the object does not contain sub-objects. Of course, an object can also be constructed with an arbitrary number of sub-objects, thus allowing multiple structures to be used. Note, a tree-like structure is especially suitable for data analysis [Zhang and Zhang 2010].

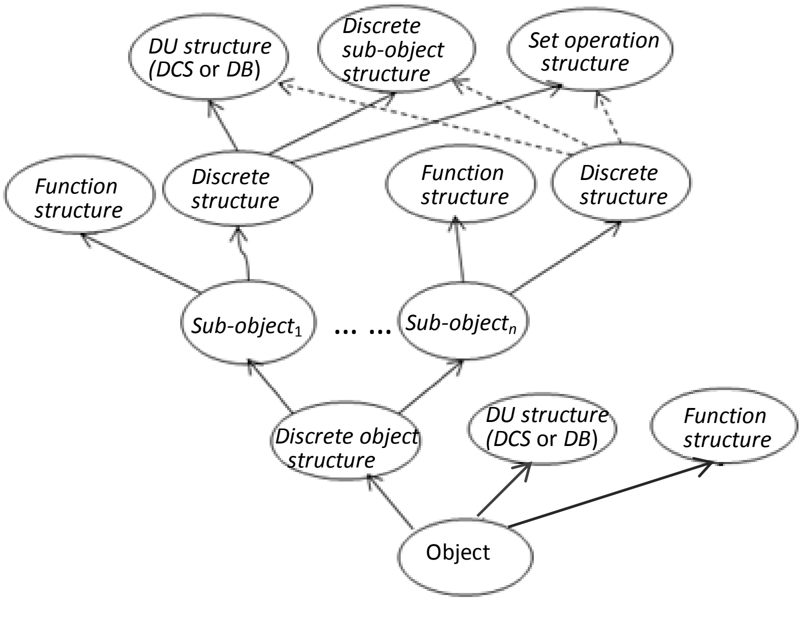


Fig. 3. Object—structure relationship schema

3. Primary applications of GDMF

Two primary application examples are illustratively presented: a mineral ore model (Fig. 4) constructed with *DCS structure* and a relatively complex building (Fig. 5) model with structures of *DCS*, *DB* and *function*.

Example 1, representing an ore body (Fig. 4). Suppose the ore body is directly depicted as an object (i.e. without containing sub-objects) with three *DCS*s, i.e. $\mathcal{DCS}_1$ (a *triangle fan* with 8 points), $\mathcal{DCS}_2$ (a *triangle strip* with 14 points), and $\mathcal{DCS}_3$ (*a triangle fan* with 8 points). The construction of the object can be realized with the following steps :

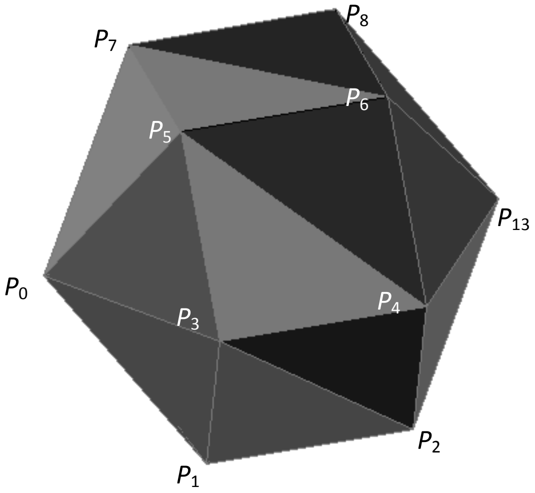


Fig. 4, an ore body model expressed with *DCS*

Step 1: Describe the *DCS*s:

$\mathscr{DCS}$1 ={*“triangle fan”*, 8, $\mathscr{P}$}={ *“triangle fan”,, 8,* P_0, P_1, P_3, P_5, P_7, P_9, P_{11}, P_1}

$\mathscr{DCS}$2={ *“triangle strip”*, 14, $\mathscr{P}$}={ *“triangle strip”, 14,* 101, *8* , P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_1, P_2}

$\mathscr{DCS}$3={ *“triangle fan”*, 8, $\mathscr{P}$}={ *“triangle fan”, 8,*P_{13},P_2,P_4,P_6,P_8,P_{10},P_{12},P_2 }

Step 2: Constructt the graphics data of the ore model using *DU structure* (Eq. 5):

$\mathscr{OBJ}$= {3, {$\mathscr{DCS}$1, $\mathscr{DCS}$2, $\mathscr{DCS}$3}}

={3,{{ *“triangle fan”, 8,* P_0, P_1, P_3, P_5, P_7, P_9, P_{11}, P_1},{*“triangle strip”, 14,* 101, *8* , P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_1, P_2}, {*“triangle fan”, 8,* P_{13}, P_2, P_4, P_6, P_8, P_{10}, P_{12}, P_2 }}

Example 2, Modeling a complex building. A five-storey complex building that has two *regular hexahedron*s (h_1, the main building block; and h_2, the gatehouse), and three *ridge*s (r_1, r_2 and r_3) (Fig. 5) is taken as an example to illustrate the modeling process. What’s more, the building has a door (d) and two windows (w) in the first floor, and 6 windows on each of the other floors. The building can be modeled as a composite object composed of three sub-objects—one for the description of the two *hexahedron*s and the three *ridge*s using *DB structure*, one for the windows and the doors in the first floor using *DCS structure*, and the other for the windows from the second floor to the fifth floor using *function structure*. The detailed modeling process is presented as follows:

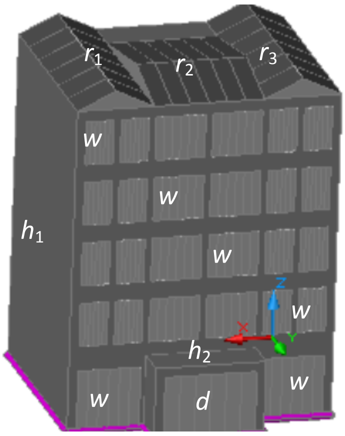


Fig. 5. A five-storey complex building

h_1 and h_2 are two regular hexahedrons; r_1, r_2 and r_3 are *ridges*; *w* and *d* refers to the windows and the doors respectively.

Step 1: Describe the five *DB*s (i.e. h_1, h_2, r_1, r_2 and r_3) as *sub-object*$_1$:

- Define $\mathscr{H}_1$ (structure data of h_1) and $\mathscr{H}_2$ (structure data of h_2) using *right prism*, and describe $\mathscr{R}_i$ ($i \in [1,3]$) (the structure data of r_1, r_2 and r_3 respectively) using *ridge*.
- Represent the *DB*s into a *DU* (Eq. 5):

  $\mathscr{DBS}$= {5, {$\mathscr{H}_1$, $\mathscr{H}_2$, $\mathscr{R}_1$, $\mathscr{R}_2$, $\mathscr{R}_3$}}, where $\mathscr{DBS}$ is the structure data of the *DB*s.
- Describe the *DU* into *sub-object*$_1$ ($\mathscr{GO}_1$) based on Eq. 9-3:

  $\mathscr{GO}_1$= {"*solid*", $\mathscr{DBS}$}, where "*solid*" refers to the shape type.

Step 2: Model the door and the windows of the first floor as *sub-object*$_2$ ($\mathscr{GO}_2$).

- First, represent the windows in the first floor as a *DCS* ($\mathscr{DCS}_1$) and the door as another *DCS* ($\mathscr{DCS}_2$) based on *multi-point successive strip* (Eq. 3). $\mathscr{DCS}_1$, for instance, is expressed as follows:

  $\mathscr{DCS}_1$ ={"*multi-point successive*", 2, 4, $\mathscr{P}$}, where 2 and 4 is the number of windows and the number of points on each window respectively, and $\mathscr{P}$ is the family set of points of the windows.
- Secondly, define the two *DCS*s as a *DU* (Eq. 5):

  $\mathscr{DCS}$= {2, {$\mathscr{DCS}_1$, $\mathscr{DCS}_2$}, where $\mathscr{DCS}$ is the structure data of the *DCS*s.
- Finally, describe the *DU* into *sub-object*$_2$ ($\mathscr{GO}_2$) based on Eq. 9-3:

$\mathscr{GO}_2$= {"*surface*", $\mathscr{DCS}$}, where "*surface*" is the shape type.

Step 3: Model the windows of other floors (i.e. from the second to the fifth floor) as *sub-object*$_3$ ($\mathscr{GO}_3$) using *function structure*.

- First, describe the structure data ($\mathscr{WDS}$) of the windows from the second to the fifth floor based on Eq. 8-2:

  $\mathscr{WDS}$= {"*window or door Looping*", 1, (0.0,-1.0, 0.0), 126.0, 4, 4.0, {"*multi-point Successive*", 6, 4, $\mathscr{P}$}}, where *UN* =(0.0,-1.0, 0.0), h_0 =126.0 m, *nf* =4, *h*=4.0 m, *ns*= 6, *np*=4, and $\mathscr{P}$is the family set of points of windows on the second floor.

- Then, define *sub-object*$_3$ ($\mathscr{GO}_3$) with $\mathscr{WDS}$(the structure data of the windows from second to the fifth floor) based on Eq. 9-3:

  $\mathscr{GO}_3$= {"*surface*", $\mathscr{WDS}$}, where "*surface*" is the shape type.

Step 4: Define the above three sub-objects as an object:

- First, describe the structure data ($\mathscr{DISCRETE\_OBJ}$) of the object based on *discrete object structure* (Eq. 6):

  $\mathscr{DISCRETE\_OBJ}$={3, {$\mathscr{GO}_1$, $\mathscr{GO}_2$, $\mathscr{GO}_3$ }}, where the number of sub-objects is 3.

- Then, construct the model ($\mathscr{OBJ}$) of the object based on Eq. 9-1:

  $\mathscr{OBJ}$ ={{1,"*composite*", *BOX*}, {"*discrete object structure*", $\mathscr{DISCRETE\_OBJ}$ }} , where 1,"*composite*" and *BOX* is the serial number (*objectNumber*), the shape type (*shapeType*) and the data bounding box respectively; "*discrete object structure*" is the structure type (*structureType*).

4. Discussion and conclusion

In this paper, a flexible and extensible general data modeling framework (GDMF) including two structure types (*discrete structure* and *function structure*) is proposed, in which a new structure can be easily integrated by simply presenting its structure module and its ID number. These heterogeneous structures including well-designed boundary representation (B-rep), constructive solid geometry (CSG) and functional representation are then encoded with structure type (*structureType*), and made uniform as structure data ($\mathscr{S\_DATA}$), whereas the basic models (Eq. 9) of object and sub-object remain unchanged. A spatial object or sub-object can now be described with these structures either uniquely or jointly, thus offering a novel modeling strategy using multiple expressions to flexibly and effectively represent the diverse and complicated three-dimensional world. Note, the usage of multiple-expressions does not affect the assignment of thematic semantics.

Our primary application examples show that various geometric regularities, as appeared in the surface in Fig. 4 (natural object) and in the surface and in the solid in Fig. 5 (man-made object), is effectively captured, and data redundancy is significantly reduced. For example, over one third storage space is reduced by *DCS* representation in Fig. 4, compared with traditional B-rep; and the vertices recorded by *right prism* (Eq. 4)—a typical *DB structure*—is almost halved; whereas the *window or door Looping* (WDL) function depicting the windows from second to the fifth floor in Fig.5 takes up only a quarter of the traditional B-rep's storage space. Besides, with the elements (i.e. building blocks) defined by structures, the convenience of data generation can be significantly enhanced, since the generation of a complex object or sub-object is just like piling up building blocks. It should be noted that GDMF would also benefit for data management, data analysis, and data visualization. This related research will be carried out in the future.

ACKNOWLEDGEMENT

This research was under the auspices of the key project of the 12th five-year plan, CAS (KZZD-EW-07-02-003), and the Natural Science Foundation of China (41271196). The authors are overwhelmed with gratitude to professor May Yuan (the University of Oklahoma) for her great work in revising the paper; and we are also very appreciative to the paper review from professor Sara L. McLafferty and Dr. Anand Padmanabhan (UIUC).

Appendix, comparative table of squiggle words and their ordinary forms:

𝒟𝒞𝒮 (*DCS*); 𝒟ℬ (*DB*); ℱ𝒰𝒩𝒞𝒯ℐ𝒪𝒩 (*FUNCTION*); 𝒟𝒰𝒮 (*DUS*); 𝒪ℬ𝒥 (*OBJ*); 𝒮𝒪 (*SO*); ℋ_𝒟𝒜𝒯𝒜(*H_DATA*); 𝒢_𝒟𝒜𝒯𝒜 (*G_DATA*); 𝒮_𝒟𝒜𝒯𝒜 (*S_DATA*); 𝒟ℬ𝒮 (*DBS*); ℋ (*H*); ℛ(*R*);𝒲𝒟ℒ (*WDL*); 𝒟ℐ𝒮𝒞ℛℰ𝒯ℰ_𝒪ℬ𝒥(𝒟*ISCRETE_OBJ*).

References:

Berlo, L.V., and Laat, R.D., 2011. Integration of BIM and GIS:The development of the CityGML GeoBIM extension. In: 5th International 3D GeoInfo Conference. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, Berlin, Germany, 197-210

Breunig, M., and Zlatanova, S., 2006. 3D Geo-DBMS. In: Large-scale 3D Data Integration – Challenges and Opportunities, Taylor & Francis, Boca Raton, Florida, 87-116

Breunig, M., Zlatanova, S. 2011. 3D geo-database research:Retrospective and future directions. *Computers and Geosciences*, 37(7), 791–803

Coors, V., 2003. 3D GIS in Networking environments. *Computers, Environment and Urban Systems*, 27(4), 345-357

Goodchild, M.F., 2009. Geographic information systems and science:today and tomorrow. *Annals of GIS*, 15(1), 3-9.

Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C., and Häfele, K.-H., 2012. OpenGIS city geography markup language (CityGML) Encoding Standard. Project document:OGC 12-019

Herring, J., 2001. The OpenGIS abstract specification, topic 1:feature geometry (ISO 19107 Spatial Schema), Version 5

Kuttig, D., 2003. Potential and limits of functional modeling in the CAD process. *Research in Engineering Design*, 5(1), 40-48

Li, Q.Q., and Li, D.R., 1998. Research on the conceptual framework of the integration of 3D spatial data model. :Acta Geodaetica et Cartographica Sinica, 27(4), 325-330 (in Chinese)

Mallet, J.L., 2002. Geomodeling. Oxford University Press, New York

Molenaar, M., 1991. Formal data structures, object dynamics and consistency rules. In: Digital photogrammetric systems, Karlsruhe, Germany, 262-273

Pilouk, M., 1996. Integrated modeling for 3D GIS. PhD Thesis, Enschede, Netherlands

Wu, L.X., 2004. Topological relations embodied in a generalized tri-prism (GTP) model for a 3D geoscience modeling system. *Computers and Geosciences*, 30, 405-418

Zlatanova, S., and Tempfli, K., 2000, Modeling for 3D GIS: spatial analysis and visualization through the web, *Proceedings of the XIX congress of ISPRS Com*. VI/2, Vol. XXXIII, B4/3, Comm. IV.16-23 July, Amsterdam. pp.1257-1264.

Zlatanova, S., Rahman, A.A., and Shi, W.Z., 2004. Topological models and frameworks for 3D spatial objects. *Computers and Geosciences*, 30(4), 419-428

Zhou, L.C., Lü, G.N., Sheng, Y.H., Xu, H.B., and Wang, H.X., 2008. A 3D GIS's spatial data model based on cell complex. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, Beijing, China, 905-908

Zhu, Q., and Hu, M.Y., 2010. Semantics-based 3D dynamic hierarchical house property model. *Internaltional Journal of Geogrgrapic Information Science*, 24(3), 165-188

Yuan, L.W., Yu, Z.Y., Lou, W., Zhou, L.C., and Lü, G.N., 2011. A 3D GIS spatial data model based on conformal geometric algebra. *Science in China: Earth Sciences*, 54(1), 101-112

Zhang, S.Q., and Zhang, J.Y., 2010. Theoretical analytics of stereographic projection on 3D-objects' intersection predicate. *Internaltional Journal of Geogrgrapic Information Science*, 24(1), 25-46

Automatic Semantic Labelling of 3D Buildings Based on Geometric and Topological Information

Abdoulaye A. Diakité, Guillaume Damiand, and Gilles Gesquière

Université de Lyon, CNRS, LIRIS, UMR5205, F-69622 France

Abstract. The lack of suitable information in 3D models of buildings and cities is still a strong limitation for the increasing number of applications requiring the 3D data. The latter are often obtained from acquisition or modeling processes during which the geometry is well preserved, but the topological and semantic information are lost. We present a new approach to enrich a purely geometric model with topological information. The reconstructed topology combined to the geometry is helpful to several operations like guided building simplification, model correction, etc. In this work we recover the semantic information based on a propagation approach guided by heuristic rules. All the process is automatic and designed such that any user can bring and customize as many rules as needed to supervise the semantic labelling. As example we propose few rules applied to both Building Information Models (BIM) and 3D Geometric Information Systems (GIS) data.

Keywords: Topology; Semantic; Combinatorial Maps; CityGML; BIM.

1 Introduction

Recent progress in the massive 3D acquisition area (photogrammetry, laser scanning, ...) made possible the generation of dense and precise 3D data going from the representation of a simple building to a whole city. It is the case for example in the GIS field where urban model data are obtained thanks to airborne laser points and images, or terrestrial laser scanning. But indoor details of the building are rarely available from such acquisition methods. In the other hand, CAAD[1] tools allow architects to produce models with high level of indoor and outdoor details, leading to very realistic models, used in BIM fields for instance. Several useful applications rely on such type of data to contribute to human well-being (navigation, simulation, etc), involving many different areas of expertise. Due to the different needs of those fields, purely geometric model is clearly insufficient for most of the applications, that require to exchange topological and semantic information to perform analysis.

To face this problem, two standards arise from both GIS and BIM domains, that are respectively the CityGML format from the Open Geospatial Consortium [20] and the Building Smart IFC format [8]. They are mainly semantic-oriented standards, allowing to store all kind of information useful to describe

[1] Computer-Aided Architectural Design

buildings, ranging from their intrinsic components up to their environment. They are more and more used and aim at being central for all expert fields involved in a construction or renovation project. Unfortunately the information stored in such standards are often poor in practice. Indeed only the geometry is more or less completely informed, leading to lack of important information. Thus topology and semantic need to be retrieved to complete information available in the standards.

We propose a new method based on a propagation approach directed by heuristic rules to retrieve the semantic information of the building components (wall, roof, openings, etc). Starting from the geometry of a model presented as a bunch of unconnected polygons, we use the Combinatorial Map (C-Map) data structure [9,18] to subdivide the model into structured cells with their topological relationships. The latter in addition to the geometric properties available allow us to define the rules and to design the propagation process of the semantic labelling among all the components of the model. We tested our method on both BIM and GIS models.

After a study of previous works on this topic, we will first describe the topological formalism behind the method. The semantic labelling based on that topology and the geometry will be then detailed for BIM and GIS data, and the results of the method will be analyzed. Finally a global discussion including the outlooks will be held to propose potential improvement of the work.

2 Related Works

2.1 Existing Approaches

3D Building modelling is an extensively research topic, and even more during this last decade in which major interest grown up from private institutes and local authorities regarding 3D urban models. This is due to the interesting range of applications they offer, e.g. building renovation, cultural heritage preserving, navigation maps, etc. Depending on the applications targeted, the proposed methods in the literature focus more on topological, semantic or geometric aspect of the model.

An important amount of work have been dedicated to topology in both BIM and GIS domains, because topological relationships between the components of the model are essential for the data consistency [15] and also crucial for simulation processes [4, 28]. Several works addressed topological query operations issues [6, 12], while others proposed data structures to handle building models. Combinatorial data structures appear to be very relevant as topological model for buildings. Generalized Maps (G-Maps) were used to represent the topology of indoor scenes reconstructed from 2D plans [16] and to optimize simulation processes (visualization, lighting, etc) [14]. Thomsen et al. [27] also used G-Maps to take advantage of their generic properties from lower to higher dimensions, in addition to cell-tuple structures. The authors proposed the construction of topological model from city data but the process involves considerable user interaction for consistency. Boguslawski and Gold introduced the Dual Half-Edge

(DHE) data structure for modelling building interior using cell complexes [5]. Globally, combinatorial data structures are equivalent in their main skills. As DHE and G-Maps, C-Map offers iterators to navigate through any entity of the cell complex and attributes can be associated to any cell of any dimension.

Methods allowing feature identification and semantic information recovering in building and urban models also take an important place as the two major standards (IFC and CityGML) are mostly semantic-oriented. But the data available in those standards are not always consistent [10]. Furthermore, many techniques were developed to extract particular features depending on the type and the data quality. Bauer et al. [3] extracted features on façades point cloud while Pu and Vosselman [21] proposed the extraction of features like walls, doors, windows, etc, from terrestrial laser scanning. Thiemann and Sester [26] proposed a partitioning of complex building model based on an adaptation of the algorithm of Ribelles et al. [22]. The model is intersected with planes of its boundaries to detect features interpreted using a rule-based decision tree. The method does not seem to deal with indoor details. More recently, Boulch et al. [7] introduced a semantic labelling method on CAD building models based on a constrained attribute grammar with geometry specific predicates on planar 3D primitives. The method deals mainly with surface-oriented models (the identified components are not volumes).

Regarding the geometry, two main topics are leading the researches in urban modelling: building generalization and model repairing. As a 3D city model has a huge amount of polygons, it is of major interest to reduce them for interactive visualization and navigation purposes. This is the reason why many works addresses the building simplification issues [13, 23, 25]. In the other hand, it is quite common to meet invalid geometries and aberrations in city models. This is usually not a problem for visualization, but it is a serious drawback for most of the applications that need to rely on valid geometry to proceed to credible calculations. In that sense, recent works are oriented in repairing common polygons errors in GIS [1, 17].

In our work, the geometry is assumed to be clean enough. And contrary to the methods in the literature, our approach is generic, deals with indoor as well as outdoor details and needs no prior information regarding the model, except its geometry. This can be done thanks to the formalism behind C-Maps that we use as data structure to model the topology. From that topology, in addition to the geometry, we define heuristic rules to semantically identify building components.

2.2 Combinatorial Maps

A C-Map is an edge-centered data structure representing the spatial subdivision of an object of any dimension, by a cellular decomposition. In 3D it describes an object by the mean of 0-cells (vertices), 1-cells (edges), 2-cells (faces) and 3-cells (volumes). The basic element of a C-Map is a *dart* which is a part of an oriented edge plus a part of each *incident* i-cell (two cells are incident if one belongs to the boundary of the other). The darts are linked between them thanks to β_i links,

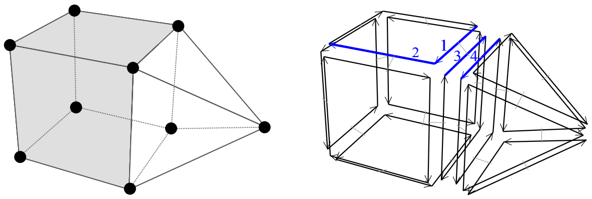


Fig. 1. Example of a 3D C-Map. Left: a 3D object made of two volumes adjacent along a face (at the basis of the pyramid). Right: the corresponding C-Map, with darts represented by arrows (sometimes numbers). Darts 1 and 2 belong to the same face and darts $1, 2$ and 3 belong to the same volume.

allowing the representation of the incidence and *adjacency* relationships binding the cells (two i-cells are adjacent if they share a common incident $(i-1)$-cell).

More precisely, a 3D C-Map is $C = (D, \beta_1, \beta_2, \beta_3)$, with D a finite set of darts, β_1 a *partial permutation*[2] on D, β_2 and β_3 *partial involutions*[3] on D. β_1 of a dart $d \in D$ returns dart $d' \in D$ belonging to the next edge, the same face and the same volume than d. Similarly, β_2 of a dart d gives dart d'' belonging to the other face, the same edge and the same volume than d. Finally, $\beta_3(d)$ returns dart d''' that belongs to the other volume, the same edge and the same face than d. Some constraints are defined on the C-Map to guarantee its topological validity (see [9,18] for more details). Partial permutation and partial involutions allow to represent objects with boundaries: when a dart d is such that $\beta_i(d) = \varnothing$, d is said i-free.

As an illustration, in Fig. 1, we have $\beta_1(1) = 2$, where both darts 1 and 2 belong to the top face of the cube volume. $\beta_2(1) = 3$, where darts 1 and 3 describe the same edge, but belong to different faces of the cube. $\beta_3(3) = 4$ with 3 being a dart of a face of the cube while 4 is a dart of the adjacent volume (the pyramid), and both 3 and 4 describe the same edge and the same face.

The previous notions allow us to describe any cell as a set of darts. For a 3D C-Map, 2-cell(d) is the set of darts that can be reached from a given dart d and using β_1 and β_3 as many times as possible; 3-cell(d) is the set of darts that can be reached from a given dart d and using β_1 and β_2 as many times as possible. Intuitively, since β_i allows to consider the other i-cell containing a given dart, if we use all the β's links except β_i we obtain all the darts belonging to a same i-cell. In Fig. 1-right, the face separating the cube and the pyramid is the set of 8 darts containing darts 3 and 4, and the volume describing the cube is the set of 24 darts containing darts 1, 2 and 3.

A C-Map allows to associate information to any cell through *attributes*. We will denote i-attr(d) the attribute of the i-cell(d). The attributes are used to

[2] A partial permutation f on a set D is a bijection from $D \cup \{\varnothing\}$ to $D \cup \{\varnothing\}$ with $f(\varnothing) = \varnothing$ and s.t. $\forall x, y \in D$, $f(x) = f(y) \neq \varnothing \Rightarrow x \neq y$.

[3] A partial involution g on a set D is a partial permutation on D satisfying $g(x) \neq \varnothing \Rightarrow g(g(x)) = x$.

store the geometry by associating to each 0-cell of the C-Map a 3D point in $\mathbf{R}^3$. Such C-Map, with 3D points associated to the 0-cells is called a Linear Cell Complex (LCC), and is the data-structure used in this work.

In [11], a method allowing to reconstruct the topological description of a building with a 3D LCC is introduced. Authors used this description as basic data structure to extract automatically the different level of details of the building.

3 Topological Formalism

Our approach is targeting two types of building data: BIM and GIS. A fundamental difference between them comes up from their different acquisition methods. The latter lead GIS models to be mainly based on the representation of observable surfaces of the buildings while BIM models are made of volumetric primitives representing the building components [19]. Here we discuss how LCCs are used to describe each type of data and to recover their topology. These descriptions will be used in the next section to propose the heuristic rules of our automatic semantic labelling algorithm.

3.1 BIM models

BIM data are resulting from designers that use CAAD tools to model buildings. They are often very detailed and offer indoor and outdoor details. Despite a visual differentiation between the components of the model, their information are rarely explicitly available and the geometry of the whole model is often stored as a polygonal mesh, with a list of vertices, and face sequences. The topological reconstruction gives us a component-based decomposition of such building model with a full connection network between all the components (see Fig. 2).

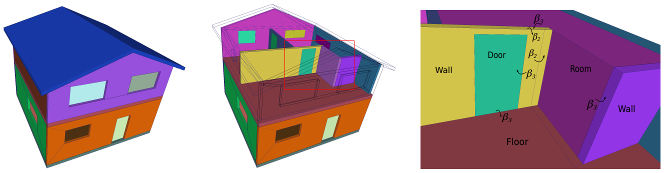


Fig. 2. Component-based topological reconstruction of a BIM model from unconnected 3D polygons. Left: view from outside. Middle: view of the inside components. Right: example of navigation by β_i links, the room volume is filled and one wall is hidden for visual purpose. Each color represents one 3-cell.

Each volume of the 3D LCC represents a meaningful building component (wall, floor, ceiling, roof, door, window, room, etc). Note that rooms are also described in the topological description by property of the reconstruction (a room can be seen as an air volume). This is particularly useful for navigation algorithms where we can retrieve all the volumes adjacent to a room (walls, openings, floors and ceilings). The semantic of each component is stored by associating to each volume of the LCC a 3-attribute containing an *id* giving the type of the component. No more additional information is required by our method.

By properties of LCCs, volumes are subdivided into faces describing the different parts of each component. For example a wall could be described by six faces describing a cuboid. Faces and volumes can be traversed thanks to darts and the different β's links. For example given a dart d belonging to a wall, we can iterate through all the darts of the wall by using a depth search algorithm starting from d and using all the possible β_1 and β_2 links. For each dart d' of the wall, $\beta_3(d')$ (when it exists) gives a dart of a building component adjacent to the current wall. Thanks to the topological description, different algorithms can be proposed to navigate through the parts of the building (see Fig. 2-right).

3.2 GIS models

The GIS data are often obtained by aerial or terrestrial laser scanning, photogrammetry or stereovision methods mainly resulting in 3D surface-based models. The acquisition schemes provide more or less dense point cloud that are processed and meshed to obtain polygonal surfaces. In this work, our data are resulting from aerial techniques sharpened to produce roof and ground surfaces and were extracted from CityGML files. The walls are obtained by extruding faces from the roof boundaries to the ground. This leads to models with average level of details (LoD2) [20].

The topological reconstruction gives us here a surfacic description of the buildings since interiors are not described (see Fig. 3-left). For this reason, contrary to BIM models, meaningful information is now associated with faces. Each face of the 3D LCC represents a part of a building, which could be wall, roof or ground. The semantic is here stored by associating to each face of the LCC a 2-attribute containing an *id* giving the type of the component.

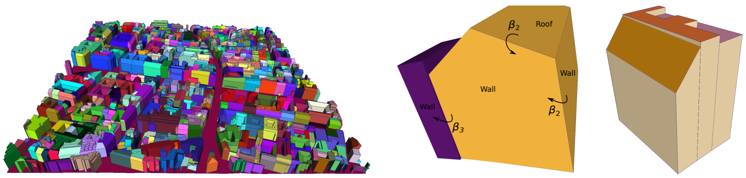


Fig. 3. Left: resulting topological reconstruction on a sample of 3D city model of Paris. Middle: navigation from one cell to another using the β_i links. Right: Example of a building complex composed of 3 volumes.

Note that a building correspond at least to a volume, and at most to a set of connected volumes. Indeed, separated roofs can lead to a volume break-up of the same building. We call *building complexes* such set of volumes supposed to represent a single building (see Fig. 3-right) and we address this issue in the next section by proceeding to a volume clustering followed by a façade extraction.

In the topological description, given a dart d we iterate through all the edges incident to the face F containing d by using β_1 links. For each dart d' of F, $\beta_2(d')$ gives a dart of a face adjacent to F. Thanks to these links, given a face F describing a roof, we can obtain one dart for each face adjacent to F which is a wall or another roof (see Fig. 3-middle).

4 Automatic Semantic Labelling

In this section we introduce an automatic semantic labelling process that entirely relies on heuristic rules based on the geometric properties and on the topological description of BIM and GIS. For each type of data, key features are first detected and labelled, then a propagation approach is adopted to label the remaining cells of the LCC.

The flexibility of the method allows to define as much rules as desired. The strength of our approach is to allow to mix geometrical and topological criteria. This is of significant importance as any expert of any field can define proper rules for general or specific building features. As an illustration of the method, we introduce few rules applied to our data to label common building features. The different propagation approaches will be detailed for BIM and GIS data. In both cases, we consider the 3D LCC $C = (D, \beta_1, \beta_2, \beta_3)$ resulting from the topological reconstruction.

For orientation purpose, we will consider the vector $\boldsymbol{Z}$ as the height axis of the coordinate system of the LCC. A 2-cell is *horizontal* if its normal vector $\boldsymbol{N}$ is collinear to $\boldsymbol{Z}$, and *vertical* if $\boldsymbol{N}$ is perpendicular to $\boldsymbol{Z}$. *null* is used to express non-existent property, e.g. a 3-cell without semantic attribute.

4.1 BIM Models

Our goal is to semantically identify the main components of the building (walls, floors, openings, roof and façade) among the 3-cells of C. We assume to deal with building models in which there is no furniture and where only building component are described, there is no volume describing air spaces. During the topological reconstruction, the air volumes are obtained by duplicating 3-free faces (i.e. faces separating a building component and an air space). The volume with maximal size is the exterior shell of the building (which is not kept in the LCC); all the other air volumes are rooms and are labelled with 3-attr(d) =“room”. Those 3-cells will be the starting point of our semantic labelling propagation. All other volumes are initialized with 3-attr(d) =“null”.

Walls and floors identification: The rooms are directly linked to the walls and the floors surrounding them. We consider all the darts $d \in D$ s.t. 3-attr(d) =“null” and 3-attr($\beta_3(d)$) =“room”.
(1) 3-attr(d) =“wall” if 2-cell(d) is *vertical*;
(2) 3-attr(d) =“floor” if 2-cell(d) is *horizontal*.

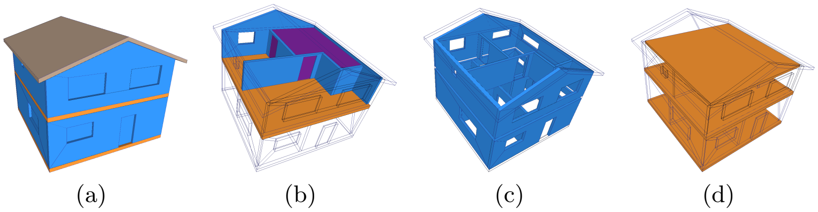


Fig. 4. (a) Result of the walls (blue) and floors (orange) labelling (the unlabelled volumes take random color). (b) Room volume (purple) used to propagate semantic information to surrounding volumes. (c) The identified walls. (d) The identified floors.

Fig. 4 illustrates the result on a simple model. Rule (1) defined for the walls involves that all the volumes having β_3 links with the *vertical* faces of the rooms are categorized as walls. Thus even doors and windows will initially be labelled as so (Fig. 4(a)). But thanks to the propagation approach, this will be corrected in the following steps. Note that with rule (2), floors and ceilings are not differentiated.

Windows and doors identification: At this step of the process, some volumes identified as walls are in fact openings components (doors, windows). Assuming that any door or window of the building has top and bottom surfaces containing at least one horizontal face each and is embedded in a wall volume, we can define adapted rules. We consider each dart $d \in D$ s.t. 3-attr(d) =“wall”. Let d_t and d_b two darts s.t. 2-cell(d_t) and 2-cell(d_b) are the top and bottom *horizontal* faces of 3-cell(d).
(3) 3-attr(d) =“window” if 3-attr($\beta_3(d_t)$) = 3-attr($\beta_3(d_b)$) =“wall”;
(4) 3-attr(d) =“door” if 3-attr($\beta_3(d_t)$) =“wall” and 3-attr($\beta_3(d_b)$) =“floor”.

The volumes are considered as windows if they have their top and bottom faces linked by β_3 to a wall volume (rule (3)), while they are considered as doors if their bottom faces are linked to a floor and their top faces are linked to a wall (rule (4)). Of course several specific configurations could occur in a building model. But these rules are enough in the case of classical models such that the one given in Fig. 5.

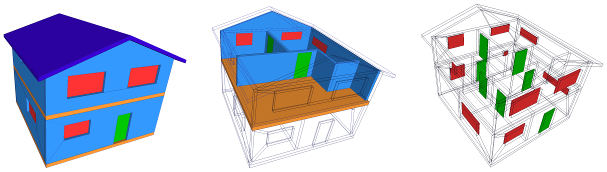


Fig. 5. Left: result of the windows (red) and doors (green) labelling (the unlabelled volumes take random color). Middle: some example of walls used to detect openings. Right: all the identified openings in the model.

Roof and façade identification: For several applications the notion of interior and exterior is of major interest. This is why the façade is a significant information to label, so as the roof, that is actually part of the façade, but here we will label them differently. Considering our topological representation, the room volumes are supposed to fill all the spaces lying inside the building model. Thus any 3-cell is linked by β_3 either to a room or to a component. We denote by D_{hi} the set of darts containing one dart for each 3-cell which is a highest room of C along $\boldsymbol{Z}$. These properties allows to propose the following rules. We consider each dart d, and denote d_t (resp. d_b) one dart of the top (resp. bottom) *horizontal* face of 3-cell(d).

(5) 3-attr(d) =“façade” if 3-attr(d) =“wall” and $\exists d' \in$ 3-cell(d) s.t. d' is 3-free;

(6) 3-attr(d) =“roof” $\big($if 3-attr(d) =“floor”, d_t is 3-free and $\exists d_{hi} \in D_{hi}$
s.t. $\beta_3(d_b) \in$ 3-cell(d_{hi}) (flat roofs)$\big)$;
or $\big($if 3-attr(d) =“null”, $\exists d' \in$ 3-cell(d) s.t.
d' is 3-free and $\exists d'' \in$ 3-cell(d), $\exists d_{hi} \in D_{hi}$
s.t. $\beta_3(d'') \in$ 3-cell(d_{hi}) (tilted roofs)$\big)$.

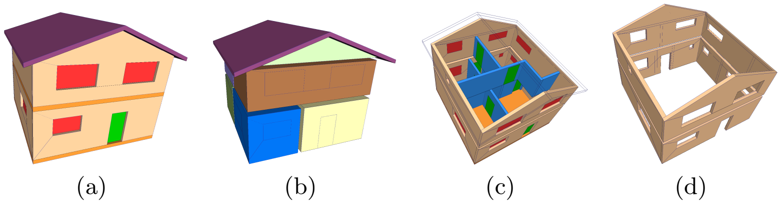


Fig. 6. (a) Result of the roof (violet) and façade (beige) labelling. (b) The roof linked to the highest room (light green). (c) Difference between the inside and outside walls. (d) All the identified walls of the façade.

Rule (5) addresses the walls because they are the only components considered for the façade here (see Fig. 6(d)). But this could be modified easily to include other features. The first part of rule (6) regarding the roof considers that a floor with its top face in the exterior and its bottom face linked to one highest room is a flat roof. But if the concerned 3-cell is part of a pitched roof, no rule defined so far would recognize it, leaving the volume with no semantic attribute (*null*). Thus if such volume has one dart linked by β_3 to one highest room in addition to a 3-free dart, it will be considered as part of a pitched roof by the second part of rule (6) (see Fig. 6(b)). All roofs of a building may not be necessarily at the highest height. Thus for particular cases other rules must be defined.

4.2 GIS Models

In a LoD2 city model there are only few building features interesting to identify: roofs, walls, ground and façades. Our goal here is to extract those features from the model and to label them using the LCC attributes. In a first step, we will identify the 2-cells to consider as roofs before spreading the semantic labelling to walls and ground faces.

Roofs and walls: Contrary to BIMs, at the beginning of this process, $\forall d \in D$, 2-attr(d) = "null". The heuristic rule necessary to identify the roofs is simply based on normal orientation checking. Once identified, the roofs will help us to recognize and label the walls. Let $\boldsymbol{N_d}$ be the normal of 2-cell(d) and $\alpha = angle(\boldsymbol{N_d}, \boldsymbol{Z})$.

(7) 2-attr(d) ="roof" if $|\alpha| \in [0, \frac{\pi}{4}]$;

(8) 2-attr(d) ="wall" $\big($if 2-attr(d) ="null" and 2-attr($\beta_2(d)$) ="roof"$\big)$;
or $\big($if 2-attr($\beta_2(d)$) ="wall" and $|\alpha| = \frac{\pi}{2}\big)$.

In rule (7), the range chosen for α allows to cover flat roofs as well as pitched ones, as illustrated in Fig. 7. It is a criterion dependent of the model, since the inclination of pitched roofs depends on many parameters (geographic position, surrounding environment, local weather, etc). In the other hand, it involves that even the ground surface is labelled as roof, but this will be fixed in the following steps. First part of rule (8) tells that any unlabelled face linked to a roof by β_2 represents a wall. This is a consistent heuristic on our data due to their production process (see Sect. 3.2). Second part of rule (8) is necessary just in case some coplanar faces are not merged, resulting in several 2-cells describing the same planar surface. Because of this, some vertical 2-cells might be unlabelled while their neighbors by β_2 are recognized as walls. Figure 7 shows what we get after this step, i.e a model in which all the 2-cells are either labelled as roof or as wall. We can then proceed to the ground surface labelling.

Ground surface: The ground can be intuitively described as the biggest surface (in area) with the lowest height in the model. It is necessary to several simulation processes (e.g flood simulation). It is unlikely to have a flat and regular ground surface (except for roads or particular installations), thus the ground can be composed of several faces in the model. At this level of the semantic labelling process, those ground patches are tagged as roofs, so we propose rules to correct them. We consider each dart d s.t. 2-attr(d) ="roof".

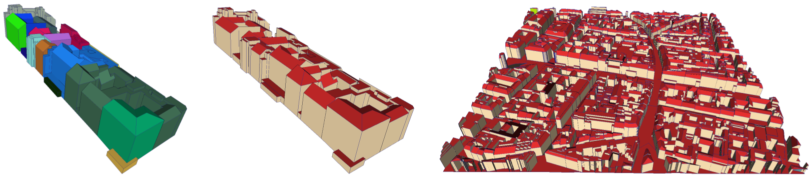


Fig. 7. Example of roofs (in red) and walls (in beige) semantic labelling from building volumes of LoD2. Left: result of the topological reconstruction. Middle: semantic labelling of the left image. Right: semantic labelling on the model shown in Fig. 3. At this step, the ground surface is also marked as a roof.

(9) 2-attr(d) =“ground” $\big($if 2-cell(d) is the biggest face with the lowest height$\big)$;
or $\big($if $\exists d' \in$ 2-cell(d) s.t. 2-attr($\beta_2(d)$) =“ground”$\big)$.

The first part of rule (9) is dependent of the coplanar face merging process applied during the topological reconstruction. Otherwise the assumption of the biggest face will not be usable. The second part of rule (9) is a direct consequence of the first part, since once the main 2-cell of the ground is found, the information is spread to its proper neighbors. Figure 8 shows the result on a district. Thanks to the semantic information, the ground can be isolated from the rest of the model. At the end of this step, we have now enough information to proceed to the façade extraction.

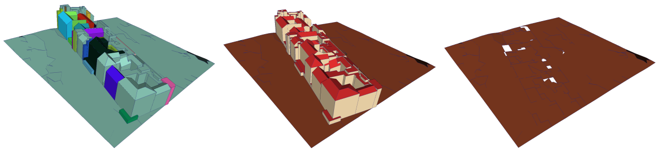


Fig. 8. Ground surface identification. The image at the left shows the resulting topological representation of the initial model. The middle image illustrates the roofs, walls and ground identification and the right image is the isolated ground patch (holes are due to missing polygons on the input data).

Façade Extraction: The term façade is used to define the exterior side of a building, not only the front. With GIS data, the main issue is first to identify building complexes properly. The initial data does not contain enough information to allow a perfect clustering of the volumes in building complexes. But with the enhanced LCC model, rules can be defined to group 3-cells. Initially, each 3-cell belongs to its own cluster. Considering two clusters c_1 and c_2:
(10) c_1 and c_2 are merged if $\exists d_1 \in c_1$ and $\exists d_2 \in c_2$ s.t. $\beta_3(d_1) = d_2$;

$$\begin{aligned}&\text{2-attr}(\beta_2(d_2)) = \text{“roof”},\ \text{2-attr}(\beta_2(d_1)) = \text{“wall”};\\ &\text{and } \exists d' \in \text{2-cell}(\beta_2(d_1)) \text{ s.t. 2-attr}(\beta_2(d')) = \text{“roof”};\\ &\text{and if the height of 2-cell}(\beta_2(d_1)) \leq \mathit{dist},\\ &\qquad\quad \text{where } \mathit{dist} \text{ is a predefined distance.}\end{aligned}$$

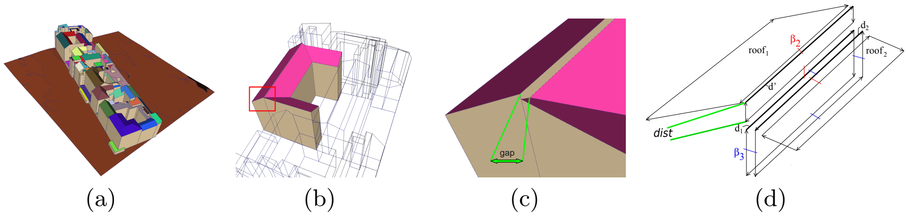


Fig. 9. (a) Initial clustering of building volumes, each roof color represent a building complex. (b) and (c) Example of gap between two building, detected thanks to the rule. (d) C-Map representation of (c), to illustrate the rule.

This rule allows to group the buildings when roofs are separated by a small gap which could result from either inaccuracy issues during the data acquisition, or architectural design (as illustrated in Fig. 9).

The gap is filled by the wall face extruded from the higher roof ending up in two 3-cells sharing a common face. The rule describes the topological configuration of the wall above that common face and just consists in measuring its height to compare it to a threshold named *dist* in the rule definition (Fig. 9(d)). Such gaps can be detected and the corresponding volumes clustered in the same building complex. A proper threshold can directly help to solve the case of buildings designed with roofs of different height levels.

Now the façade is identified on the formed building complexes. Similarly to BIM models, the notion of exterior (represented by 3-free darts) and interior is used to identify the proper cells. Each dart d is considered:

(11) 2-attr(d) =“façade” $\big($if d is 3-free$\big)$;
or $\big($if d and $\beta_3(d)$ do not belong to the same cluster$\big)$.

Those two simple rules are enough to access to all the 2-cells of the façade. As illustrated in Fig. 10 where a building complex made of three volumes is presented, extracting the façade means identifying the inner faces of the complex. Once they are known, it is possible to remove them. They are characterized by two darts linked by β_3 and belonging to the same building complex.

5 Implementation details and Limitations

All our algorithms based on C-Maps/LCC were implemented using the Computational Geometry Algorithm Library (CGAL [24]) and we used the Open Asset

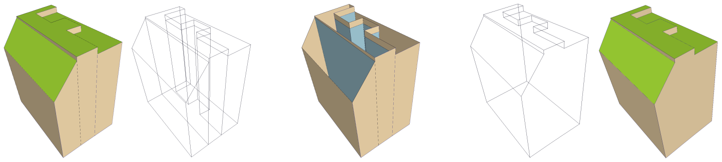


Fig. 10. Façade extraction on building complex. Left: the original complex (roof in green) in filled and wire-frame version. Middle: the faces inside that are not part of the façade (in blue). Right: the resulting façade.

Import Library (Assimp [2]) to parse the input data (Collada, Obj, etc). The complete process of semantic labelling, including the topological reconstruction and façade extraction were executed in 1.02s on the BIM model in Fig. 2 (836 faces), 1.23s on the GIS model in Fig. 7 - left (1481 faces), 6.19s on the model in Fig. 8 (4352 faces) and 224.65s on the model of Fig. 7-right (55378 faces). Note that several optimizations could speed up our method. The computer used is a laptop with an Intel Core i7-2760QM 2.40 GHz and 8GiB of RAM.

An important constraint of the method we presented relies on the quality of the input data. It is well known that 3D models resulting from GIS acquisition methods are subject to common artifacts (gaps, holes, polygon permeation, etc). While our approach is strongly related to topology, those defects lead to topological errors limiting the capacities of the automatic semantic labelling. BIM models are not safer since they are resulting from architects who mainly care about the visual aspect only. This leads to models in which it is frequent to meet inconsistent geometries, created just for hiding or producing a visual effect. In the other hand, the richness of the details is still very challenging for automatic feature identification. Although we apply pre-processing algorithms (e.g coplanar faces simplification, ill faces correction or removal, points adjustment, etc) before the topological reconstruction and the semantic labelling, we do not pretend to proceed to model correction. But solving the correctness issues of the data will be probably of great benefit for all the applications using them. We also adopt a margin of error ϵ for all the computations, as the numerical data of the models are exposed to rounding issues. Depending on the fixed ϵ the method can fail where the errors are significant (e.g important gap between two faces supposed to share an edge).

6 Conclusion and Outlooks

We presented a framework to automatically retrieve the semantic information of a building or a city model thanks to the combination of geometry and topology. Starting from the geometry described by a bunch of unconnected polygons, the topological links between the vertices, the edges, the faces and the volumes are reconstructed first. Then heuristic rules are proposed based on the relationships between the components. Features of interest are labelled through a propagation approach. The strength of the approach is in its adaptability to BIM or GIS data in addition to its flexibility in creating proper rules for the models. The method were tested on synthetic and real data, and offered interesting results.

Some improvements can still be brought to the method to perform better results. For example it would be interesting to combine cadastre information to the GIS data for an easier and accurate identification of building complexes. In the other hand, it can be of major interest to set a correction tool to heal the input data often containing inconsistent information. We also plan to allow users to locally apply in a model the rules they define and to investigate the automatic semantic labelling of more detailed models, e.g LoD3 for GIS or furnished model for BIM. There is still a long way to go to get a fully automatic and reliable interpretation of the different building elements, but we believe that our method is a promising approach.

References

1. N. Alam, D. Wagner, M. Wewetzer, J. von Falkenhausen, V. Coors, and M. Pries. Towards automatic validation and healing of citygml models for geometric and semantic consistency. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-2/W1:1–6, 2013.
2. Assimp Project. Open asset import library, 2014.
3. J. Bauer, K. Karner, K. Schindler, A. Klaus, and C. Zach. Segmentation of building models from dense 3d point-clouds. In *27th Workshop of the Austrian Association for Pattern Recognition*, pages 253–259, 2003.
4. V. Bazjanac. Space boundary requirements for modeling of building geometry for energy and other performance simulation. In *CIB W78: 27th International Conference*, 2010.
5. P. Boguslawski and C. Gold. Rapid modelling of complex building interiors. In Thomas H. Kolbe, Gerhard Knig, and Claus Nagel, editors, *Advances in 3D Geo-Information Sciences*, Lecture Notes in Geoinformation and Cartography, pages 43–56. Springer Berlin Heidelberg, 2011.
6. A. Borrmann and E. Rank. Topological Operators in a 3D Spatial Query Language for Building Information Models. In *Proc. of the 12th ICCCBE*, 2008.
7. A. Boulch, S. Houllier, R. Marlet, and O. Tournaire. Semantizing Complex 3D Scenes using Constrained Attribute Grammars. *Comp. Graph. Forum 2013,*, 32(5):33–42, 2013.
8. Building SMART International. Industry Foundation Classes (IFC), IFC4, 2013.
9. G. Damiand and P. Lienhardt. *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing.* A K Peters/CRC Press, 2014.
10. S. Daum and A. Borrmann. Efficient and robust octree generation for implementing topological queries for building information models. In *Proc. of the EG-ICE Workshop on Intelligent Computing in Engineering*, pages 370–385, 2012.
11. A. A. Diakité, G. Damiand, and D. Van Maercke. Topological Reconstruction of Complex 3D Buildings and Automatic Extraction of Levels of Detail. In *Eurographics Workshop on Urban Data Modelling and Visualisation*, pages 25–30. Eurographics Association, 2014.
12. C. Ellul and M. M. Haklay. Using a b-rep structure to query 9-intersection topological relationships in 3d gis - reviewing the approach and improving performance. In

3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, pages 127–151. Springer Berlin Heidelberg, 2009.

13. H. Fan and L. Meng. A three-step approach of simplifying 3d buildings modeled by citygml. *International Journal of Geographical Information Science*, 26(6):1091–1107, 2012.
14. D. Fradin, D. Meneveaux, and P. Lienhardt. A Hierarchical Topology-Based Model for Handling Complex Indoor Scenes. *Computer Graphics Forum*, 25(2):149–162, June 2006.
15. G. Gröger and L. Plümer. How to achieve consistency for 3d city models. *GeoInformatica*, 15(1):137–165, 2011.
16. S. Horna, G. Damiand, D. Meneveaux, and Y. Bertrand. Building 3D indoor scenes topology from 2D architectural plans. *In GRAPP*, pages 37–44, 2007.
17. H. Ledoux, K. Arroyo Ohori, and M. Meijers. A triangulation-based approach to automatically repair GIS polygons. *Computers & Geosciences*, 66:121–131, 2014.
18. P. Lienhardt. N-Dimensional Generalized Combinatorial Maps and Cellular Quasi-Manifolds. *Int. J. Comput. Geometry Appl.*, 4(3):275–324, 1994.
19. C. Nagel, A. Stadler, and T. H. Kolbe. Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3d models. In *Academic Track of Geoweb 2009 Conference, Vancouver*, 2009.
20. Open Geospatial Consortium. City Geography Markup Language (CityGML) Encoding Standard, version 2.0.0, 2012.
21. S. Pu, G. Vosselman, and Commission Vi. Automatic extraction of building features from terrestrial laser scanning. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 33–39, 2006.
22. J. Ribelles, P. S. Heckbert, M. Garland, T. Stahovich, and V. Srivastava. Finding and removing features from polyhedra. In *Proceedings of DETC*, volume 1, pages 1–10, 2001.
23. M. Sester. 3D visualization and generalization. In *Photogrammetric Week*, volume 7, pages 03–09, 2007.
24. The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.4 edition, 2000.
25. F. Thiemann. Generalization of 3D Building Data. In *Part 4, GeoSpatial Theory, Processing and Applications*, pages 286–290, 2002.
26. F. Thiemann and M. Sester. Interpretation of building parts from boundary representation. In *Proceedings of the 1st International Workshop on Next Generation 3D City Models, Bonn.* Whittles Publishing, 2005.
27. A. Thomsen, M. Breunig, E. Butwilowski, and B. Broscheit. Modelling and managing topology in 3d geoinformation systems. In *Advances in 3D Geoinformation Systems*, pages 229–246. Springer, 2008.
28. C. Van Treeck and E. Rank. Analysis of building structure and topology based on graph theory. In *Proceeding of the 10th International Conference on Computing in Civil and Building Engineering. Weimar*, 2004.

3D MODELLING AND SIMULATION OF GROUNDWATER USING 3D TEN AND VOLUMETRIC DATA MODEL

Ishaku Bashir Yakubu, Izham Mohammad Yusoff and Main Rindam

Abstract

Access to clean water has been demand of every society for domestic, agriculture and industrial uses. However, with increase in population, adverse weather events and pollution have rendered the surface water inaccessible to meet human needs. Groundwater serve as alternative for surface water, but it exploration have been hindered by complex geological structure. Many researchers have attempted to model and visualize the hydrogeological process for sustainable groundwater exploration using DEM. However, these approaches implementation of DEM in visualization of hydrogeological process in 3D are not true representation of the subsurface geology; thus, affecting the reliability and application. In this study, 3D TEN data structure was used to build electrical resistivity data in 3D and volumetric object was used to visualize groundwater flow in 3D by implementing 3D TEN data model. The movement and direction of groundwater flow were defined by implementing 3D steady state of Darcy's law from Lagragian approach. The Simulation was run on different geological structure based on porosity and permeability over a time frame of 1 hour, 2hours, 4hours, and 8 hours interval. The study show possibility of modelling resistivity survey data in 3D and water yield capacity of an aquifer is a function of overburden material of the aquifer. The data mining techniques adopted in this paper has proof the efficiency of 3D data models implementation in a heterogeneous geological structure for 3D modelling, simulation and visualization of hydrogeological processes for sustainable groundwater exploration This study will help stake holders and groundwater resource managers to maximize resources to meet water needs of the society. 3D modelling by implementing 3D TEN and Octree data model has proving to be an effective tools for modelling of resistivity data and visualization of groundwater flow in near reality to meet societal water needs.

1.1 Introduction

Access to clean and quality water have been the demand of every society with it provision solemnly responsibility of the government. In recent times, increase in population growth has triggered demand for water for agriculture, domestic and industrial uses. Groundwater exploration becomes imperative due to unreliable nature of surface water (i.e. anomalous weather); vulnerability to pollution and uneven spatial distribution of river networks in a given locality.

In recent times, Geographical Information System (GIS) and Vertical Electrical Sounding (VES) have been integrated in groundwater survey and exploration by many field workers including but not limited to: [1-7]. However, VES and GIS groundwater exploration models, results are often in two dimension (2D) thus, affecting reliability and generalization [8, 9]. In addition, as increase population continue to exert pressure on water, there is need to develop a model that is capable of visualizing hydrogeological processes in three-dimensions (3D) to help policy makers meet a realistic and sustainable provision of water for the teaming population.

The advancement of computer graphics has enable the development of many 3D GIS model for groundwater exploration i.e. [10-14] among several others. 3D conceptual frame work model have been developed with limitation of *in-situ* measurements. Although the existing 3D models on groundwater exploration has provided basic and indicate the needs for

3D modelling and visualization of hydrogeological process, the lack of *in-situ* data have often affect the reality representation of the subsurface geology in reality. Other 3D modelling application such as TOUGH 2 [15], MODFLOW [16] and computer intelligence to predict the potential water yield of a given location. However, aspect of groundwater flow such as geological medium, porosity and well establish network of void in different geological formation have not been incorporated into existing the model.

Though advancement in computer graphics as well as development of computer with high processing ability has enable 3D GIS practitioners to express different phenomenon in 3D; as dynamic and rigid field using different 3D GIS data models, lack of specific data model for groundwater survey and exploration has affected the representation of subsurface geology in 3D [9]. [17] has classified 3D GIS data model into surface, volume and octree based data model which has enabled 3D GIS practitioners to represent and visualized objects as well as events in 3D. Scholars such as [18] and [19] utilized 3D GIS data models in simulation and visualization of excess over line flow and pollution dispersion in 3D. Despite this advancement, aspect of 3D modelling and simulation from *in-situ* measurement and absent of unique data model that fit into 3D groundwater modelling and simulation has received less attention in previous studies.

This paper therefore adopt 3D Tetrahedron Network (TEN) data model and volumetric soft geo-object for representation and visualization of *in-situ* data generated from VES into 3D regional representation of the subsurface geology addressing issue of specific data model suitable for 3D groundwater modelling and simulation. 3D TEN is most suitable for representing complex geological formation as a network of tetrahedron, facet and facile. The ease with which 3D TEN data model can be manipulated while preserving 3D properties [20, 21] necessitates it adoption. To manipulate and process VES data, tessellation is adopted to enable the tilting of curve tin subsurface as measured from electrical resistivity responses as one or more geometric shapes.

2.1 Design and Modelling of VES data using 3D TEN

3D TEN is an extension of 2D Triangular Irregular Network (TIN) [20]. Objects in 3D TEN are described in Figure 1(a) by connected but non overlapping tetrahedral made up of four vertices, six edges, and four faces [20]. All real-world objects are irregular and 3D, they can adequately be represented using 3DTEN[22]. For efficiency, data may be processed into a primitive form as points, lines or surface topology. The 3D TEN represents 3D objects with a tightly arrayed but non overlapping tetrahedrons as an extension of TIN and can represent 3D objects with complex spatial topological relations [10, 20]. The characteristics attributes of 3D TEN data structure have made it efficient in modelling of an object consisting of volume and rigid properties [20].

3D TEN as internal data structure for this research because of it computational efficiency [23], well defined array of triangle, ease of representation, easy maintenance for visualization using network of non-overlapping tetrahedron [21] and flexibility of representing complex object. 3D TEN advantage in modelling of 3D object according to [20] is the simplest data structure that can be reduced to point, line, volume representation to enabled the definition and partitioning of real world object. Due to the outlined advantages, it can be manipulated and volume computation of subsurface lithology for this study. The partition ability of 3D TEN enable modelling of non overlapping volume in a given space [20].

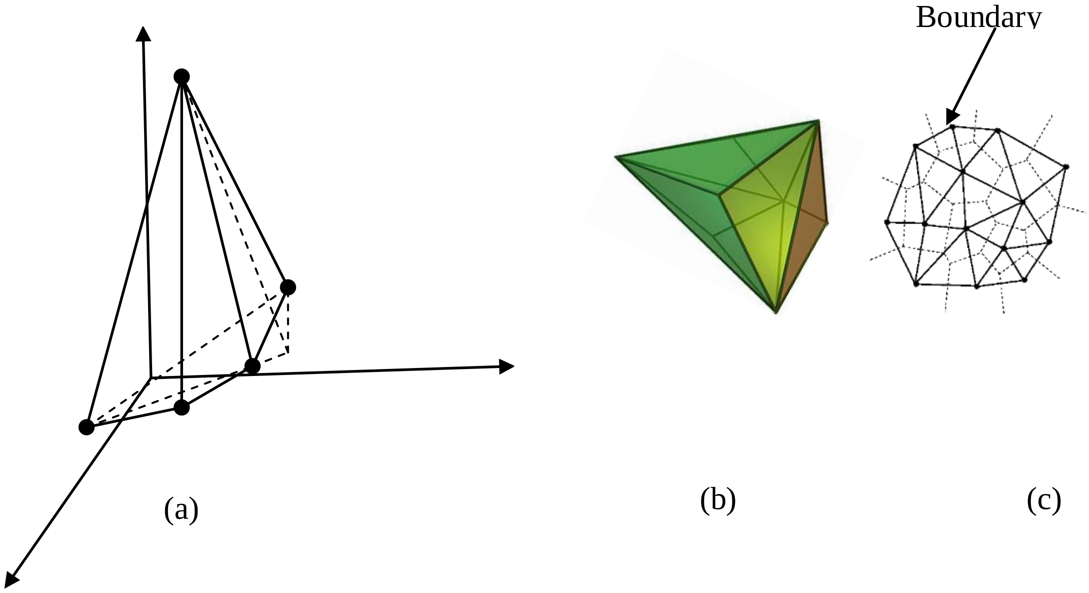


Figure 1: 3D TEN model and Delaunay triangulation.

Figure 1(b) is a 3D TEN build from network of tetrahedron to enable the 3D representation of the subsurface geology as deduced from VES interpretation. A series of mesh is created through Delaunay triangulation in Figure 1(c) of the generated 3D TEN containing several other geometric criteria related to interpolation accuracy. A constrained Delaunay triangulation was developed to relax the Delaunay property suitable to preserve edges, while enjoying optimality properties similar to those of a Delaunay triangulation. Therefore, from 3D modelling point of view, the interpreted VES results can now be modelled into 3D while groundwater flow within each of the modelled geological formation can be simulated as a dynamic field.

3.1 3D Groundwater Flow Model Based on Darcy's Law

For any model describing visualization of groundwater flow, the flow rate must be express proportional to nature and characteristics of geological formation that constitutes layer [24]. This allow for the modelling and simulation of groundwater flow within a confined geological formations. Although available 3D data model can be applied in 3D modelling of groundwater the nature, processes and existing theories that govern groundwater movement needs to be incorporated into the 3D modelling and visualization process to allow for reality representation.

Thus, [9] outlined two approach of describing changes in field: Eulerian methods that describe changes of field at a fixed location at a series of points and Lagrangian methods which describe changes which occur along a fluid element confined at a series of points leading to change in shape size and volume. For this research, Lagrangian approach was adopted to visualized water flow rate confined within a geological layer at a series of point.

Darcy law of groundwater flow described the 3D movement of water confined in a given soil medium. Such that the steady flow rate is given as

$$\frac{\partial(k_x \frac{\partial h}{\partial x})}{\partial x} + \frac{\partial(k_y \frac{\partial h}{\partial y})}{\partial y} + \frac{\partial(k_z \frac{\partial h}{\partial y})}{\partial z} = 0 \qquad (1)$$

Where $\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}$ and $\frac{\partial h}{\partial z}$ are changes in flow volume along X, Y and Z and k is the porosity of the geological medium through which water flow. For the 3D modelling and simulation of groundwater the following equation were adopted to enable visualization of groundwater flow using Lagragian approach:

$$\begin{array}{ccc} a_x & a_y & a_y \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ & \vdots & \\ n_x & n_y & n_y \end{array} \qquad (2)$$

Where a, b, c,....................n are geological layers and x, y, z are dimensions of the layers respectively.

Equation (2) shows the predicted function of a dynamic point in 3D Delaunay triangulation [9]. This equation is modified to represent the net inflow/outflow of water in a geological layer at a point within 3DTEN. Based on the developed equation, volumetric geo objects can be model in 3D to simulate groundwater flow within a given layer.

3.2 Groundwater Field Simulation using volumetric Soft Geo-object

The aim of groundwater field simulation using soft geo-object as a volumetric data model is to 1) allow for the representation of water elements with distinct volume proportional to the particle sizes of the geological layer; 2) to visualize in near reality the available quantity for water for exploration.

A GIS geodata base was created containing information regarding the different geological and hydrological units of the subsurface as well as information related to the porosity of each respective layer as deduced from resistivity analysis. However, the technical aspect of dynamic field simulation involve the designing and simulation of soft geo-object to represent water element confined into 3D TEN representation of geological layer using 3D MAX software. The rationale behind this operation is to allow implementation of real time rendering and dynamic field simulation in a given space.

Space is represented as a set of objects with spatial properties or as a set of locations with properties referred to as a field [25] that varies as a function of time. To enable simulation of groundwater, a 3D steady state equation of Darcy's law of groundwater flow was adopted derived from 2D field. A 2D field (f) is defined as a singled-value function of a location in 2D space: $f(x,y)$ [9]. Thus, extending 2D field in to 3D field we have $f(x,y,z)$. Integrating time into field change at a given point within a given geological layer, the field function becomes $f(x(t),y(t),z(t))$. Therefore, the rate of change in groundwater flow in 3D field at a given geological layer is defined by the following sets of equations:

$$f(a) = k(\frac{\frac{\partial h}{\partial x} + \frac{\partial h}{\partial y} + \frac{\partial h}{\partial z}}{\partial t}) = 0 \qquad (3)$$

$$f(b) = k(\frac{\frac{\partial h}{\partial x} + \frac{\partial h}{\partial y} + \frac{\partial h}{\partial z}}{\partial t}) = 0 \qquad (4)$$

$$f(c) = k(\frac{\frac{\partial h}{\partial x} + \frac{\partial h}{\partial y} + \frac{\partial h}{\partial z}}{\partial t}) = 0 \qquad (5)$$

$$(d) = (\frac{\frac{\partial h}{\partial x} + \frac{\partial h}{\partial y} + \frac{\partial h}{\partial z}}{\partial t}) = 0 \qquad (6)$$

Where a, b, c and d are the delineated geological layers ∂h is the change in flow within a geological formation, $\partial x, \partial y, \partial z$ are the instantaneous points within the geological units ∂t is the simulation time in hours and k is the porosity if the geological layer..

3.3 3D volume visualization of groundwater as volumetric geo-object

Groundwater visualization confined within a given geological layer is defined by the nature, characteristics and the properties that constitute the layer. Such that the volume of water visualize depend on these properties and time as defined by equation 3-6. The water element is considered as a dynamic field confined within a rigid space i.e. the subsurface. After processing the water element appears as collection of many small particles that increases with volume over time. The water element has attribute of shape, sizes, colour, speed, time and direction as associated with volumetric geo-objects.

Water is simulated as geo-object using a particle system as basic elements with a life span proportional to time. For this research work, the simulation of water flow was design as a mesh shown in Figure 2. As time increases, groundwater flow volume tend to increase proportional to particles size of the geological units through which it flow and the mesh moves with the fluid flow hence, the positions of the mesh elements change constantly over time [9].

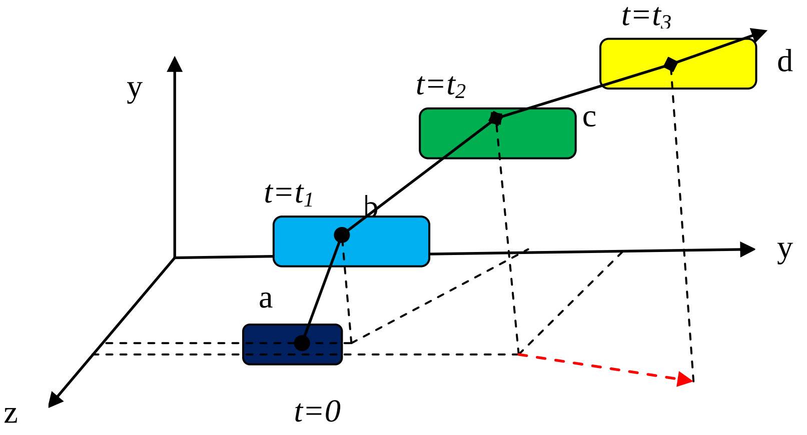


Figure 2: Simulation method implemented.

4.1 3D Modelling and Simulation of Groundwater Flow Using 3D TEN and Volumetric Geo-object

4.2 The Study Area

The Bida basin located between longitudes 4˚45’E to 6˚10’E and latitudes 8˚47’N to 9˚10’N in Niger state of Northern Nigeria. Bida basin is drained by two major rivers, namely; river Chanchaga and Gbako, and some smaller drainage that empty into Niger River. It has an area extend of 81.79 Km^2. The basin has been selected for the 3D modelling and simulation of groundwater due it importance in terms of water needs, development due to population growth and urbanization of surrounding communities.

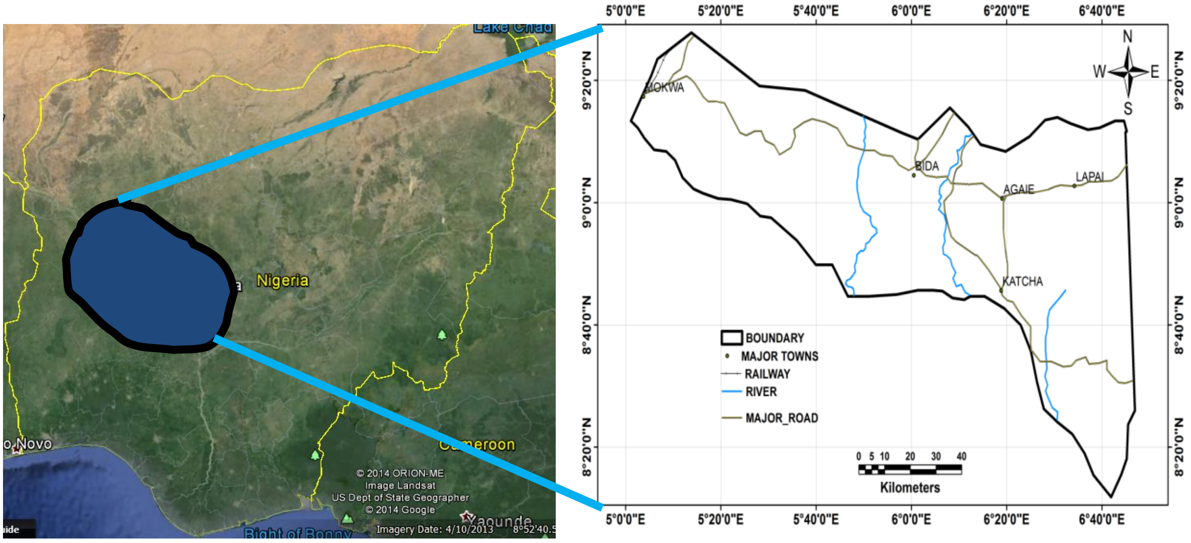


Figure 3: Location of Bida Basin

In this study, the procedure adopted in 3D modelling and simulation of groundwater using 3D TEN and volumetric data model involves the following steps: (1) acquisition of subsurface information using VES and recording the coordinate of each VES station; (2) analysis of VES data by constraining the delineated VES layer into available borehole logs to obtained corresponding geology type; (3) development of GIS maps of the delineated layes thickness and groundwater potential map; (4) processing of the thickness information in GIS data base into 3D using 3D TEN data structure; and (5) simulation of groundwater flow within the modelled 3D layer using volumetric geo-objects to represent the water layer. The movement of water element within modelled layer is governed by 3D steady state of Darcy’s law and the design of the flow field treated using Lagrangian approach which describes changes along a fluid element confined at a series of points leading to change in shape size and volume. The generated data is stored in ArcGIS and commercially available 3D modelling software.

A geological map of Niger state in scale of 1:600,000 was used to extract map of Bida Basin and production of thematic layer maps of different geological formations and groundwater potential map. Groundwater flow volume was generated based on subsurface porosity of the geological layer and computed from Darcy’s groundwater infiltration model [26].

4.3 Computation of Layer Thickness and Groundwater Potential Map

The first phase of layer thickness computation is the use commercially available res1d.exe software with user defined startingr model. Based on the resistivity survey data the program iterate and find the optimal layer resistivity and thickness Table 1. The delineated layers as deduced from the computer iteration are held in a GIS geo-data base for spatial analysis. An Inverse Distance Weight (IDM) interpolation is adopted for interpolation of the delineated layer. Figure 4 show the spatial distribution of the thickness of the delineated layers. The raster grid obtained from the interpolation is superimposed on the 3D TEN while varying the height segment to obtained 3D structure of the modelled layers. Based on the resistivity survey, a groundwater potential map (GWP) Figure 5 is produced by assigning weight based on the geological formation that characterised the delineated aquifer layer[1, 27] and from computed flow volume using Darcy's law

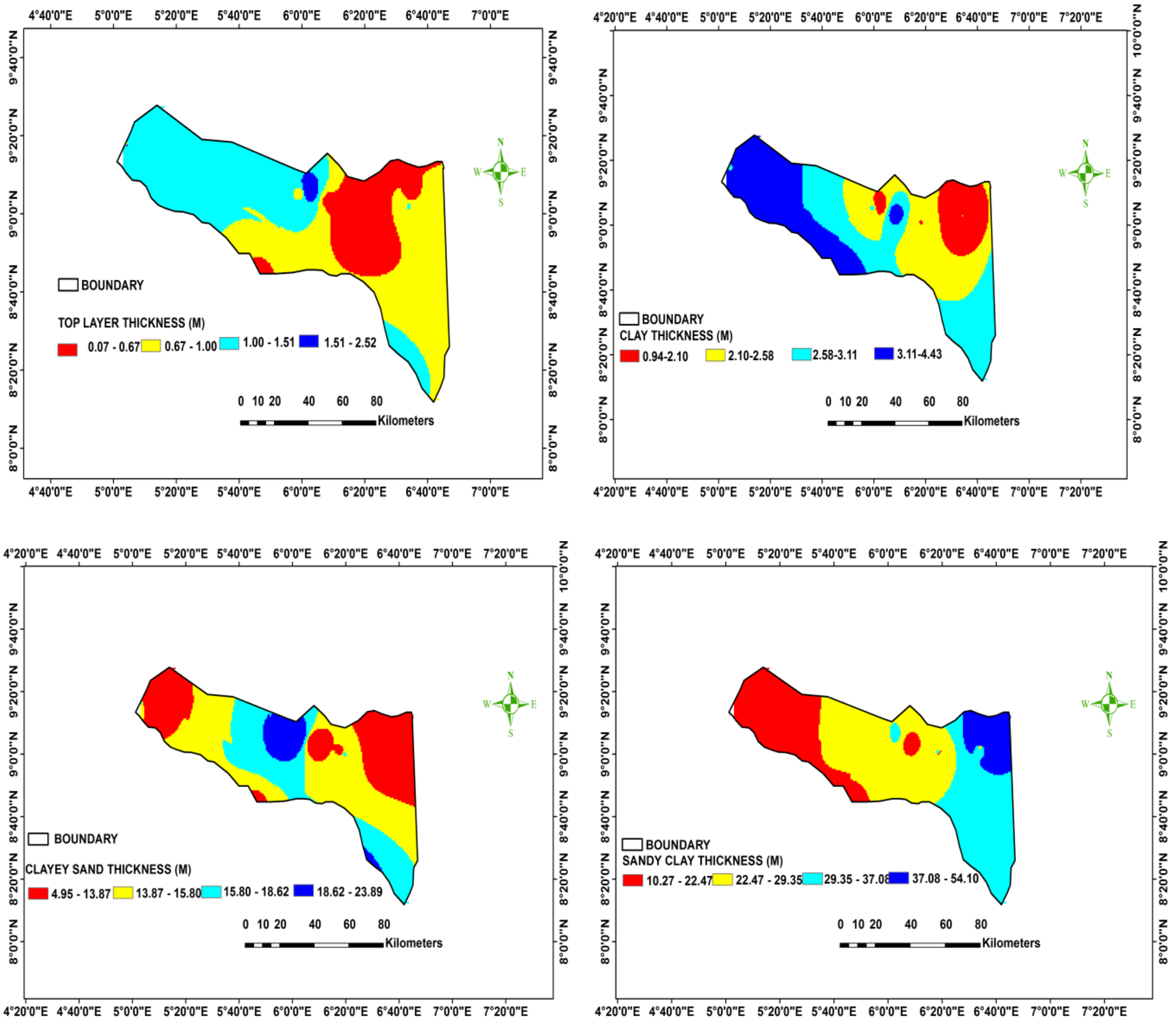


Figure 4: Subsurface layer thickness

Table 2: Resistivity survey results.

VES ID	Easting	Northing	Subsurface Parameters							
			LR=LAYER RESISTIVITY(Ω)				LT=LAYER THICKNESS (M)			
			LR 1	LT 1	LR2	LT2	LR3	LT3	LR3	LT4
VES 1	5.47293	10.4031	1091.23	1.00	388.31	9.09	165.31	11.78	829.09	22.03
VES 2	6.73793	7.80871	1832.38	0.53	194.08	1.14	564.32	12.17	209.91	16.72
VES 3	6.30906	9.01081	1140.01	0.50	168.07	1.90	65.10	11.24	34.22	37.23
VES 4	6.31679	9.00608	1157.62	0.08	1092.29	2.34	202.54	14.79	565.58	8.45
VES 5	6.3226	9.00131	1731.93	0.05	766.42	2.41	101.66	17.56	408.19	28.35
VES 6	6.57461	9.06956	1369.39	0.36	910.21	1.44	280.63	13.66	75.92	54.73
VES 7	6.63328	9.43645	1169.16	0.40	225.84	2.25	868.44	4.18	94.29	62.44
VES 8	6.56937	9.04867	1369.04	0.81	909.92	2.44	910.21	13.66	75.91	54.73
VES 9	7.09263	8.93816	1146.92	1.00	590.70	8.59	1087.02	11.86	498.76	27.56
VES 10	6.56759	9.03734	2637.37	1.13	683.86	0.70	646.85	9.84	137.35	19.17
VES 11	5.06117	9.28963	1131.32	0.54	403.9	2.66	54.28	10.76	339.81	13.52
VES 12	5.06826	9.29931	1473.78	0.54	623.44	2.87	146.41	7.71	35.54	16.77
VES 13	5.06996	9.29434	1212.99	2.12	327.07	2.11	347.84	3.99	602.58	20.74
VES 14	5.05471	9.29434	193.87	2.17	35.01	6.49	29.96	23.02	157.31	13.30
VES 15	5.05448	9.28552	1308.77	1.56	284.03	4.16	502.51	22.17	543.55	20.30
VES 16	6.13867	9.05756	1308.39	0.52	126.50	3.75	33.18	8.69	334.34	16.95
VES 17	6.03333	9.10065	1156.37	2.55	348.24	1.32	412.45	22.87	256.32	35.00
VES 18	6.20944	8.30323	1341.26	2.45	243.65	3.74	365.56	34.89	345.50	45.89
VES 19	5.99226	9.08782	1289.68	0.67	367.07	2.67	301.34	23.89	156.67	22.56
VES 20	5.76230	8.72456	1124.32	0.54	253.11	4.50	48.28	12.84	46.78	18.45

The GWP map computed is used to identify high and low volume of groundwater areas. The GWP map forms a basis through which 3D simulation of groundwater flow is based to determine areas of low and high potential.

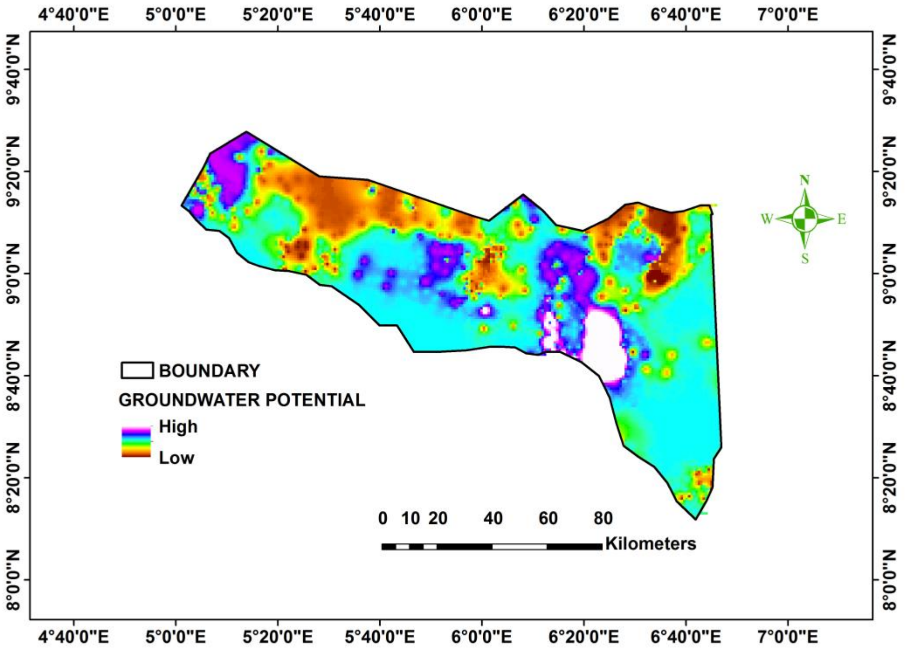
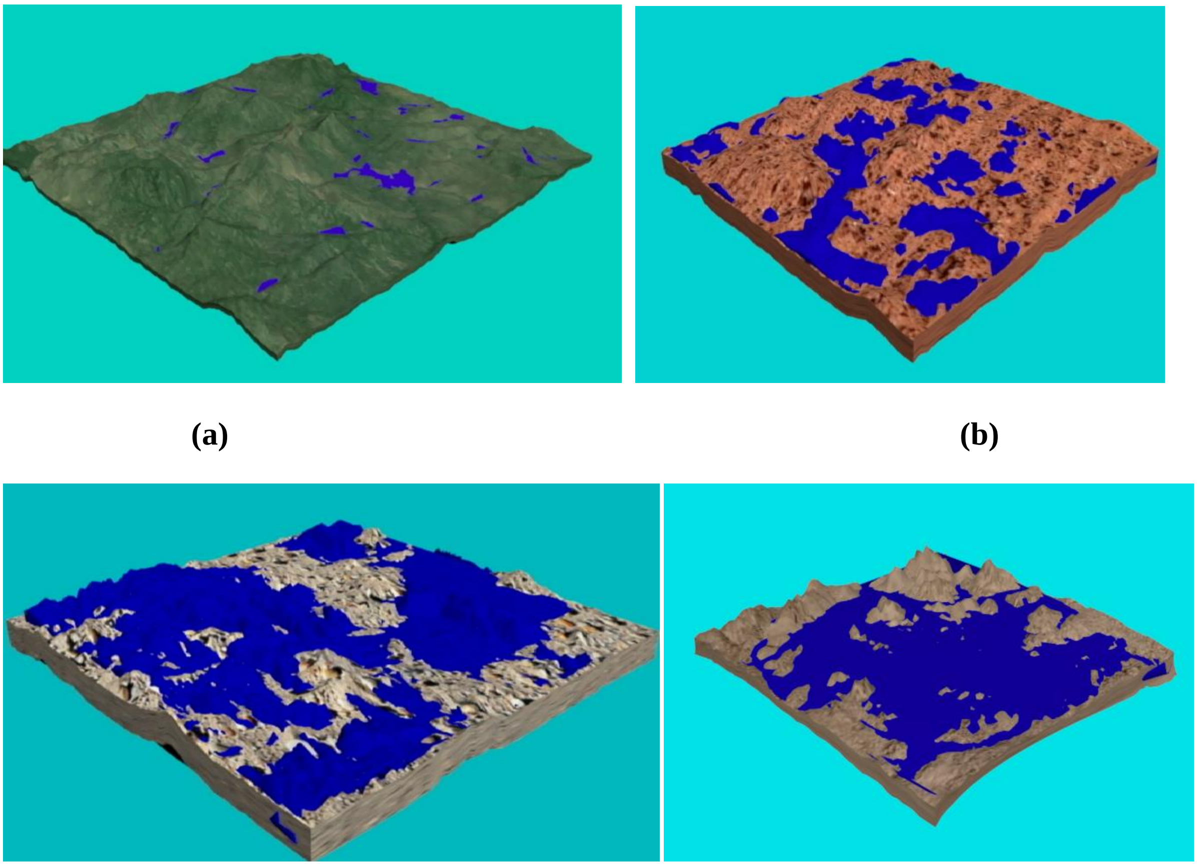


Figure 5: GWP map of Bida Basin

5.0 3D Dynamic Simulation of Groundwater from Darcy’s Law

From equation (3), (4) (5) and (6), the total volume of water generated within the top soil is for 1 hour, 2 hours, 4hours and 8 hours is 17461228.07; 870614.04; 435307.02; and 348245.61 per meter cube (Km^3) respectively. Clay layer has lowest water flow rate due to nature of it pore spaces thus yield volume tend to be low and slow and the following values were recorded 330833; 82708.33; 41354.17; and 33083.33 Km^3 over the same time frame. The clayey sand layer has water volume of 2369.52; 29586.91; 590773.81; and2363095.24 indicating increase in porosity due to transition from clay to sand. The last layer is the sandy clay layer with flow volume of 249685.53; 312106.91; 624213.84; and 2496855.35 per Km^2 marking layer with the highest water yield potential and moderate flow rate due to alternating sandy clayey and sand formation.

Based on the computed volume (i.e. as a function of geological composition and porosity of the subsurface layer), simulation was run to visualise movement of water at 1 hour, 2 hours, 4 hours and 8 hours interval. The implication of the simulation results on time frame is vast. However, for the purpose of this research, the visualization help in determining the likely yield of an aquifer based on the prevalent geology that is marked as aquifer during groundwater exploration survey. Scholars such as [28-30] opined that; though the nature and geology of an aquifer is important in water yield potential but geology of vadose zone influence aquifer as it determine the amount of water delivered to the aquifer from infiltration and hydrogeological circulation.



(a) **(b)**

(c) **(d)**

Figure 6: 3D dynamic simulation using volumetric soft geo-object visualized at 1 hour, 2 hours 4hours and 8 hours over (a) top soil, (b) clay, (c) clay clayey sand and (d) sandy clay.

Figure 6 is the simulated result showing continuous increase in volume of water (represented in blue) as the particle sizes of the geological layer increases. Such that top soil which hydrogeological movement is in response to surface elevation retained little water over time in response to surface slope. However, clay soil with tighter void between it particles characterized by high water retention capacity low water transmission hence less water it seen indicating that possibilities of low water yield within clay aquitard. The clay-clayey sand unit containing mixture of clay and sand appear to have a relative high volume of water compare to clay layer. However, some spot of undulating compacted material appears created by clay which forms an aquitard within this layer hence having little or no trace of water after the simulation. The sandy clay layer simulation has higher volume of water compare to other of geological formation due to large void between it particles allowing for easy movement of water within this layer hence high water yield potential. The sandy clay layer is capable of yielding the highest volume of water for sustainable exploration.

The compact material within these geological layers has resulted to change in size, shape, volume, speed and rate in groundwater movement within the subsurface. By implication and application of this simulation on groundwater exploration is: for any reliable groundwater exploration project nature, type and degree of compacted material exert influence on the likely yield of water since the flow direction is likely to be modified in response to compact materials.

6.1 Conclusions

This paper report the integration of geophysical data from electrical resistivity survey by processing, modelling using 3D TEN, simulation of groundwater using volumetric geo-object and visualization of groundwater flow computed and defined by Darcy's law of groundwater movement. The delineated layers from resistivity survey were interpolated and converted to raster data sets which are super imposed on the 3D TEN data structure to obtain thickness of geological layer proportional to the ones delineated. The modelled 3D geological layers are simulated using Lagragian approach which defined dynamics of a mesh as a function of time within a given 3D field. The approach is this work has addressed the issue suitable and adaptable data model for 3D modelling and simulation of groundwater which has receive less attention in previous research.

3D modelling and dynamic simulation of groundwater flow from integration of VES data within GIS is important by providing better understanding of hydrogeological processes on a regional scale. Detailed knowledge is presented about the potential of a given geological formation to yield reasonable amount of water available for large exploration. 3D dynamic simulation and visualization has provide platform through which policy makers and other stake holders in groundwater exploration can understand the complexity of geology formation on groundwater exploration projects through visualization. Therefore, the results obtained in this research will immensely benefit relevant agencies such as Ministry of water resources, River Basin Development Authority and the government in policy formulation and execution of groundwater exploration projects in Bida Basin.

REFERENCES

1. Adiat, K., M. Nawawi, and K. Abdullah, *Application of Multi-Criteria Decision Analysis to Geoelectric and Geologic Parameters for Spatial Prediction of Groundwater Resources Potential and Aquifer Evaluation.* Pure and Applied Geophysics, 2013: p. 1-19.
2. Babiker, I.S., et al., *A GIS-based DRASTIC model for assessing aquifer vulnerability in Kakamigahara Heights, Gifu Prefecture, central Japan.* Science of the Total Environment, 2005. **345**(1): p. 127-140.

3. Babiker, I.S., et al., *Assessment of groundwater contamination by nitrate leaching from intensive vegetable cultivation using geographical information system.* Environment International, 2004. **29**(8): p. 1009-1017.
4. Balakrishnan, P., A. Saleem, and N.D. Mallikarjun, *Groundwater Quality mapping Using Geographic Information System (GIS):A Case Study of Gulbaga City, Kornataka, india.* African Journal of Environmental Science and Technology 2011. **5**(12): p. 1069-1084.
5. Gossel, W., A.M. Sefelnasr, and P. Wycisk, *A GIS-Based flow model for Groundwater Resources Management in Development Areas in Eastern Sahara Africa*, in *Applied Groundwater Studies In Africa. IAH Selected Papers on Hydrogeology*, S.M.A. Adelana and A.M. MacDonald, Editors. 2008, CPRPress/Balkena: Leiden The Netherland. p. 43-46.
6. Khan, H.H., et al., *GIS-based impact assessment of land-use changes on groundwater quality: study from a rapidly urbanizing region of South India.* Environmental Earth Sciences, 2011. **63**(6): p. 1289-1302.
7. Srivastava, P.K., M. Gupta, and S. Mukherjee, *Mapping spatial distribution of pollutants in groundwater of a tropical area of India using remote sensing and GIS.* Applied Geomatics, 2012. **4**(1): p. 21-32.
8. Izham, Y.M. and M. Rindam, *3D GIS urban runoff mechanism: A new perspective using volumetric soft geo-object.* Geografia: Malaysian Journal of Society and Space, 2012. **8**(5): p. 124-139.
9. Beni, H.L. and J.P. Mostfavi, *Dynamic Field Process Simulation within GIS: The Vornoi Approach.* The International Archives of the Photogrametry. Remote Sensing and Spatial Information Science, 2008. **Vol. XXXVII**(Part B2): p. Pp. 891-898.
10. Tao, J., *3D GIS Integrated model algorithm based on block model*, in *IEEE Pasific-Asia Workshop on Computer Inteligence and Industrial Application*2008, IEEE computer society. p. 315-319.
11. Ross, M., M. Parent, and R. Lefebvre, *3D geologic framework models for regional hydrogeology and land-use management: a case study from a Quaternary basin of southwestern Quebec, Canada.* Hydrogeology Journal, 2005. **13**(5-6): p. 690-707.
12. Nury, S.N., et al. *Three Dimensional (3D) Aquifer Visualization for Sustainable Water Management.* in *ERE: Environmental Research Event*. 2009.
13. Nasib, *3D Ground-Water Flow Modeling Based on GIS*, in *GEO, Track GIS* 2012.
14. Holger, K., et al., *Rigorous 3D geological models as the basis for groundwater modelling*, A.M.G.S.o. America, Editor 2007, Illinois State Geological Survey: Colorado, USA.
15. Asahina, D., et al., *Hydro-mechanical model for wetting/drying and fracture development in geomaterials.* Computers & Geosciences, 2014. **65**: p. 13-23.
16. Trambauer, P., et al., *On the validity of modeling concepts for the simulation of groundwater flow in lowland peat areas–case study at the Zegveld experimental field.* Hydrology and Earth System Sciences, 2011. **15**(9): p. 3017-3031.
17. Gong, J., P. Cheng, and Y. Wang, *Three-dimensional modelling and application in geological application exploration engineering.* Computer and Geosciences, 2004. **30**(4): p. Pp. 391-404
18. Izham, M.Y., M.U. Ujang, and A. Abdul-Rahaman, *3D Dynamic Simulation and Visualization for GIS Based Infiltration Exess Overland Flow Modelling*. Lecture Notes On Geoinformation and Catography, ed. W. Cartwright, et al. 2008.
19. Lin, H., et al., *A Virtual Geographic Environment for a Simulation Of Air pollution Dispersion in the Pearl River Delta (PRD) Region*. 3D Geo-Information Science ed. C. William, et al. 2009: Springer-Verlag Berlin Heidelberg. 3-13.

20. Abdul-Rahman, A. and M. Pilouk, *Spartial Data Modelling for 3D GIS*. 2007, Verlag Berling Heidelberg: Springer
21. Zlatanova, S., A. Rahman, and M. Pilouk, *3D GIS: current status and perspectives.* International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 2002. **34**(4): p. 66-71.
22. Lemmon, C.J., *Boundary mapping and its application to geographic routing*, 2010, James Cook University.
23. Penninga, F., P. van Oosterom, and B.M. Kazar, *A tetrahedronized irregular network based dbms approach for 3d topographic data modeling*, in *Progress in Spatial Data Handling*. 2006, Springer. p. 581-598.
24. Harbaugh, A.W., *MODFLOW-2005, the US Geological Survey modular ground-water model: The ground-water flow process*. 2005: US Department of the Interior, US Geological Survey Reston, VA, USA.
25. Worboys, M.F. and M. Duckham, *GIS: a computing perspective*. 2004: CRC press.
26. Fetter, C.W. and C. Fetter, *Applied hydrogeology*. Vol. 3. 2005: Wiley New Jersey.
27. Chenini, I., A.B. Mammou, and M. El May, *Groundwater recharge zone mapping using GIS-based multi-criteria analysis: a case study in Central Tunisia (Maknassy Basin).* Water resources management, 2010. **24**(5): p. 921-939.
28. Adiat, K.A.N., M.M.N. Nawawi, and K. Abdullah, *The integrated use of Landsat TM and SRTM images as a reconnaissance investigation tools in Groundwater prospect evaluation, - A Case study of Kedah State, Malaysia*, in *Map Asia & ISG 2010*2010: Kuala Lumpur, Malaysia.
29. Dar, I.A., K. Sankar, and M.A. Dar, *Remote sensing technology and geographic information system modeling: an integrated approach towards the mapping of groundwater potential zones in Hardrock terrain, Mamundiyar basin.* Journal of Hydrology, 2010. **394**(3): p. 285-295.
30. Elewa, H.H., et al., *Determining groundwater protection zones for the Quaternary aquifer of northeastern Nile Delta using GIS-based vulnerability mapping.* Environmental Earth Sciences, 2013. **68**(2): p. 313-331.

POINT CLOUD DATA SEGMENTATION TO DETECT CURBS IN URBAN ENVIRONMENTS

B. Rodríguez-Cuenca [a, *], M.C. Alonso[a], S.García-Cortés [b], C. Ordóñez [b]

[a] Department of Physic and Mathematics, Alcalá University, Campus Universitario Ctra. Madrid-Barcelona, km. 33,600, 28871, Alcalá de Henares, Madrid, Spain - borja.rodriguezc@edu.uah.es
[b] Department of Mining Exploitation, University of Oviedo, Escuela Politécnica de Mieres, C/ Gonzalo Gutiérrez Quirós, 33600, Mieres, Asturias, Spain - sgcortes@uniovi.es

Commission II, WG II/2

KEY WORDS: Curbs, Road boundaries, 3D point cloud, Segmentation, Feature extraction

ABSTRACT

Non-manual curb detection is an important issue in road maintenance, 3D urban modeling, and autonomous navigation fields. This paper is focused on the segmentation of curbs and street boundaries using a 3D point cloud registered by a mobile laser scanner (MLS) system. This method provides a solution based on the projection of the measured point cloud on the XY plane. Over that plane, a segmentation algorithm is carried out based on morphological operations to determine the location of street boundaries. The proposed method is valid in both straight and curved road sections and applicable both to laser scanner and stereo vision 3D data due to the independence of its scanning geometry.

This proposed method has been successfully tested with two datasets measured by different sensors. The first dataset corresponds to a point cloud measured by a TOPCON sensor in the Spanish town of Cudillero. That point cloud comprises more than 6,000,000 points and covers a 400-meter street. The second dataset corresponds to a point cloud measured by a RIEGL sensor in the Austrian town of Horn. That point cloud comprises 8,000,000 points and represents a 160-meter street. The method provides success rates in curb extraction of over 85% in both datasets.

1. INTRODUCTION

Laser scanner sensors and stereo vision systems provide fast and accurate three-dimensional information of objects, buildings, and landscapes without maintaining direct contact with the measured objects. This information is useful in several remote sensing applications like digital terrain model generation [1], 3D city modelling [2], and feature extraction [3]. Laser scanner sensors can be placed on aerial (aerial laser scanners, ALS) or terrestrial platforms (terrestrial laser scanners, TLS). TLS can be categorized into two types: static and dynamic. Static TLS data collection is carried out from base stations: A sensor is fixed in a base station, from which the point cloud is sensed. Dynamic TLS or mobile laser scanner (MLS) sensors are installed in a mobile platform. MLS sensors have a navigation system based on global navigation satellite systems (GNSS) and inertial measurement units (IMU). This work is focused on segmenting 3D point clouds produced by MLS sensors to determine existing curbs in urban environments. An accurate method for determining the location of curbs, road boundaries, and urban furniture is crucial for several applications, including 3D urban modeling and developing autonomous navigation systems [4, 5]. Moreover, accurate and automatic detection of cartographic-entities saves a great deal of time and money when creating and updating cartographic databases [6]. The current trend in remote sensing feature extraction is the development of methods as automatic as possible. The aim is to develop algorithms that can obtain accurate results with the least possible human intervention in the process. It is difficult to create a fully automatic method of determining the location of and extracting every piece of urban furniture in a city for the following reason: Urban furniture has a heterogeneous typology; every city, and almost every street, has its own typical furniture. Most works on feature extraction have proposed semi-automatic methods in which the user must control a few settings for accurate detection. The authors of this work have developed a semi-automatic method to detect curbs and road boundaries through segmenting the measured point cloud. We attempted to minimize the number of thresholds. In the current method, the user must control two settings that depend directly on features of the studied area: the height of the curb and the point density in the curb's vertical wall. The paper is organized as follows: Section 2 summarizes the previous studies related to ours; Section 3 shows the proposed method to segment the point cloud; and in Section 4 the results obtained in two study cases performed with different MLS sensors are detailed. Finally, our conclusions and future lines of work are described in Section 5.

2. RELATED WORKS

Many applications for TLS have been reported since the appearance of these systems. The 3D modeling of buildings and indoor areas [7, 8], the geometry verification of tunnels [9], the detection of urban furniture and pole-like objects [10], and the modeling and reconstruction of 3D trees [11] are some of the applications for which TLS sensors have been used. Additionally, several applications for point clouds detected via MLS sensors exist in the current literature. They have been used in applications such as vertical wall extraction [12], façade modeling [13], building footprint detection [14], and the extraction of pole-like objects, such traffic signs, lamp posts, or tree trunks [10].

The amount of the point clouds data provided by laser scanning systems is extremely large, composed by (x, y, z) coordinates and additional information such as intensity, Global Positioning System (GPS) time, or the scanning angle of a million points. The analysis and processing of these data is computationally complex. Hence, in order to reduce the processing times and the complexity of the datasets, the point clouds are often simplified before an algorithm is used for feature extraction, mapping, or decision making. In some cases, the point clouds are segmented into several clusters that are individually analyzed and classified [15]. In [16] a segmentation method based on the difference of normals of a point and its neighborhood applied in a multi-scale approach is proposed. The difference of normals algorithm is efficient for segmenting a large 3D point cloud into various objects of interest at different scales such as cars, road curbs, trees, or buildings. In other works, the 3D point cloud is divided into other smaller clouds formed by slices of the original to reduce the amount of data. In [17], as a previous step in the detection of existing street curbs, the measured point cloud is divided into several road cross sections using the GPS time data. Another option to make the point cloud more manageable is to decompose the measured data in a 3D voxel grid. In [18], a method is presented that performs a 3D scene analysis from streaming data. This procedure consists of a hierarchical segmentation formed by multiple consecutive segmentations of the point cloud. The original point cloud is sparsely quantized into infinitely tall pillars; each pillar is quantized into coarse blocks and each block contains a linked list of its occupied voxels. In this work, voxels are considered the atomic unit that object categories are assign to. The original point cloud is projected into a 2D raster image that represents the XY surface. Thus, the 3D information is reduced to a 2D raster in which image processing techniques can be applied to determine the location of the target features.

The work presented in this paper is devoted to curb detection from MLS point clouds. In the current literature, there are many studies related to this issue. Some of them use as input data point clouds obtained from stereo vision and recently, several authors have focused on the detection of road markings, lines, and road sides in straight and curved areas based on data obtained by stereo cameras [19, 20]. There are also some methods in the current literature used to detect curbs and roadsides based on point clouds measured with TLS and MLS sensors. In [21], a method to detect curbs using 3D scanner sensor data was presented. The detection process starts with the voxelization of the point cloud and the separation of those points that represent the ground. Later, candidate points for curbs are selected based on three spatial variables: height difference, gradient value, and normal orientation. Using a short-term memory technique, every point located in a voxel whose vertical projection is in the road is considered a false positive and is deleted. Finally, the curb is detected by adjusting a parabolic model to the candidate points and performing a RANSAC algorithm to remove false positives. The performance of the method depends on the correct selection of the thresholds for each of the three variables used. This method provided a detection rate (completeness) of about 98% in two studied datasets.

Weiss and Dietmayer [22] automatically determined the position of lane markings, sidewalks, reflection posts, and guardrails by a vertically and horizontally automotive laser scanner data. Curb detection applies a third-order Gaussian filter to sharpen the vertical distance profile, which defines the shape of the curb. This profile is divided into sections with a certain width, forming an accumulative histogram. Candidate curbs are found through a histogram-based algorithm that searches those slots of the histogram that are candidates to represent curbs and guardrails. Because not every candidate is a valid curb, the locations of real curbs are determined by analyzing the heights, slopes, and interruptions of every polygon.

Belton and Bae [23] proposed a method to automatize the identification of curbs and signals using a few steps. The rasterization of the 3D point cloud into a 2D grid structure allows each cell to be examined separately. First, the road is extracted; then, cells that are adjacent to the road are likely to contain curbs. Points in these cells are used to determine the vertical plane of the curb, from which a 2D transversal section is calculated. The top and the bottom of the curb are determined as the points that are furthermost above and below the line defined by the two furthermost points in the 2D section. This procedure has several limitations. The proposed method would not provide good results detecting concave and non-horizontal roads; furthermore, the method could provide poor results for shorter or curved curbs due to confusing edges with other points of the studied profile.

Yang et al. [17] carried out edge detection by dividing the measured point cloud into 2D sections using the GPS time at which every point was registered. They applied a moving window to these 2D sections to detect the roads and road boundaries. Curbs were detected by analyzing the elevation and shape change in the moving windows studied. They also presented a method to detect curbs in occluded parts of the cloud, but some problems in areas with irregular shapes were detected. The value of the parameters and the length of the moving window are critical to the performance of the proposed method.

A recent work by Hervieu and Soheilian [24] describes a method to extract curbs and ramps, as well as reconstruct lost data in areas hidden by obstacles in the street. A system for the reconstruction of road and sidewalk surfaces is also proposed. They adjust a plane to a group of points from the cloud and compute the angular distance between the normal vector and the z vector. After that, a prediction/estimation model is applied to detect road edges. The procedure requires the user to manually select the curb direction, which is not always easy. This method could fail in curved or occluded sections. To solve this problem, they propose a semi-automatic solution in which the user must choose some points of the non-detected curb to reconstruct these sections.

In [25], Kumar et al. developed a method to detect road boundaries in both urban and rural roads, where the non-road surface is comprised of grass and soil and the edges are not as easily defined by slope changes alone. A 2D rasterization of the slope, reflectance, and pulse width of the detected point cloud is carried out. Gradient vector flow and a balloon model are combined to create a parametric active contour model, which allows the road boundaries to be determined. Roadside detection is carried out using a snake curve, which is initialized based on the navigation track of a mobile van along the road section. The snake curve moves using an iterative process until it converges on the roadsides, where the minimum energy state is located. This method has been tested in straight sections and provided good results, but its performance in curved sections is unknown. The procedure is computationally complex, which could make the detection process too slow.

Apart from the methods described in the current literature, there are other solutions for curb detection in commercial software packages [26]. But their technical details could not be found in the literature. These solutions are not fully automatic; users must provide some initial information to the software.

3. METHOD

The method proposed in this work uses 3D point coordinates (x, y, z) of the cloud data measured by MLS as inputs. The output of the procedure is a new 3D point cloud formed by those points that belong to a curb. The proposed method for segmenting MLS data consists of four consecutive steps, as shown in Figure 1.

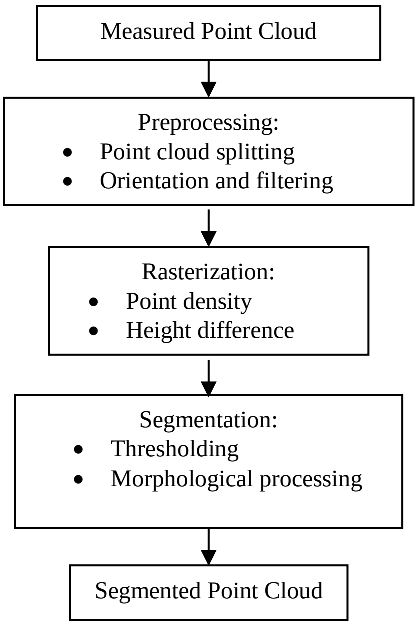
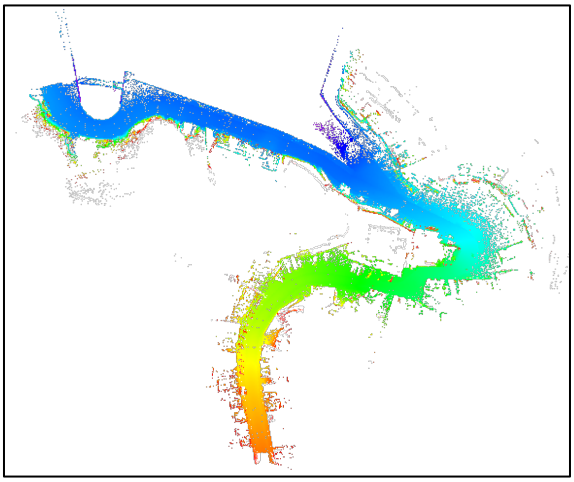


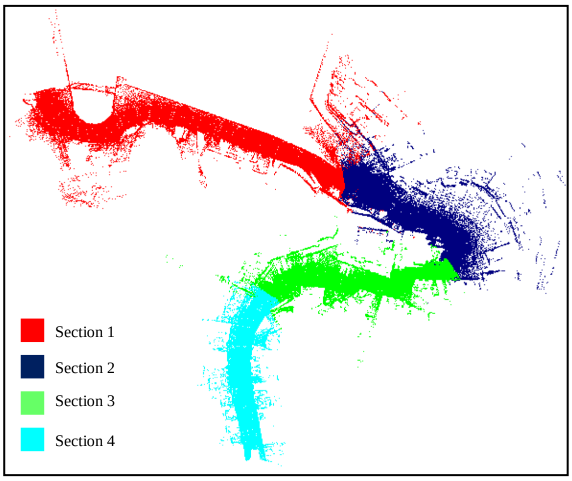
Figure 1 – Flow chart of the proposed method

3.1. Preprocessing: The 3D point clouds data are difficult to process due to the large amount of information corresponding to millions of points. The work with these raw MLS data could involve a significant amount of time and computational effort. Thus, in this first stage, some methods to facilitate the handling of the point cloud are carried out. Preprocessing is divided into three main steps: a point cloud splitting, a coordinate system change, and a filtering phase.

3.1.1. Point cloud splitting: We divide the measured point clouds into slices and process them individually. In the case of Test Site 1, the original point cloud is divided into four slices, each formed by 1.5 million points (Figure 2). The following steps of the proposed method are applied on every slice separately.



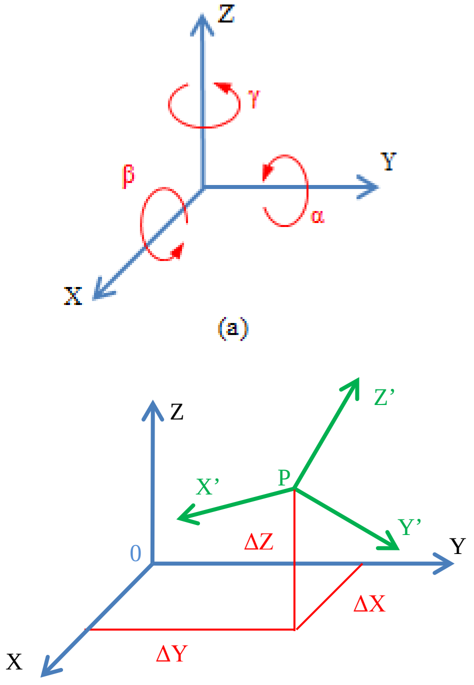
(a)



(b)

Figure 2 – (a) original MLS point cloud in dataset 1 and (b) slices in which the original point cloud is divided

3.1.2. Orientation and filtering: In this step, a coordinate system change is carried out, moving from a global coordinate system, in which the MLS measures the point cloud, to a local coordinate system. The point clouds registered by the MLS sensor are properly geo-referenced in a global reference system by means of a navigation system and an inertial measurement unit (IMU), which provides coordinates within a global frame to every point. The point clouds used in this work are geo-referenced using UTM projection in Zone 29 for dataset 1 and UTM projection in Zone 33 for dataset 2, and WGS84 ellipsoidal heights in both cases. MLS datasets are usually quite large, and managing them is computationally difficult and expensive work. In order to ease and speed up every operation in the segmentation procedure, the original coordinate system is



(b)

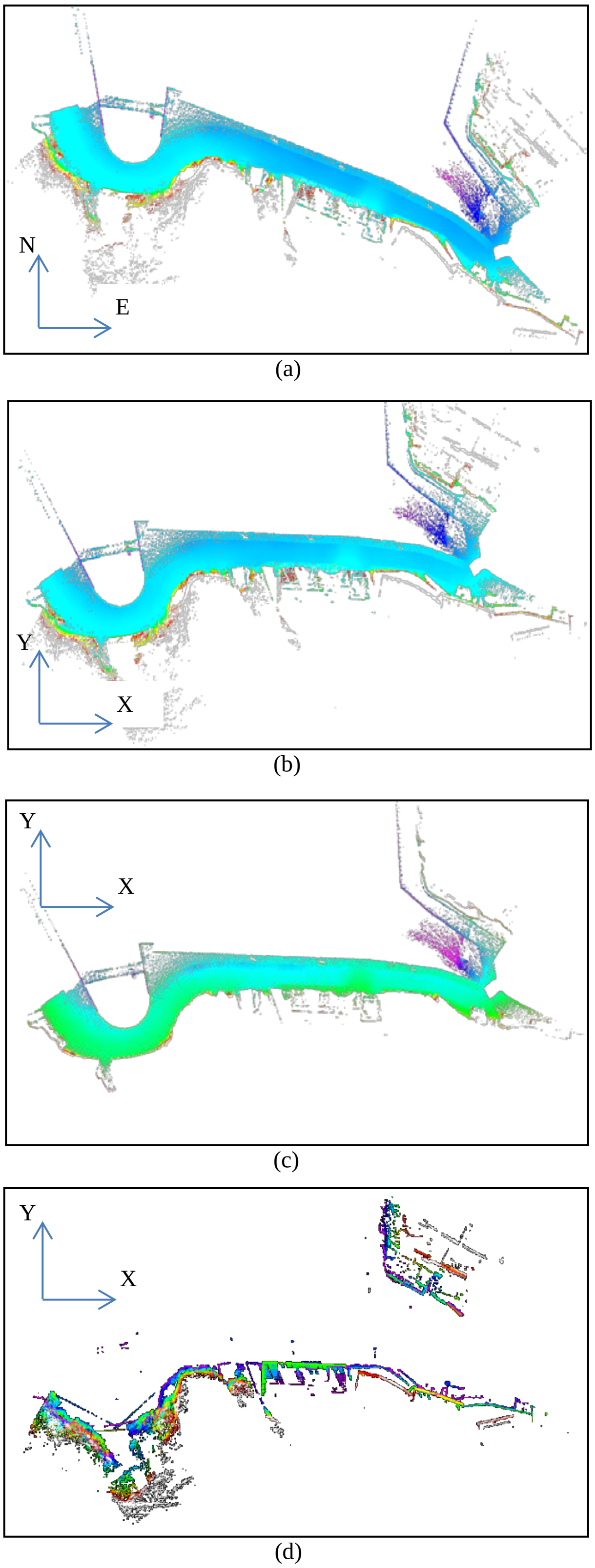
Figure 3 – (a) (ΔX, ΔY, ΔZ) translation and (b) (β, α, γ) rotation angles for the coordinate system change

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos\gamma * \cos\beta & \sin\gamma * \cos\beta & -\sin\beta \\ \cos\gamma * \sin\beta * \sin\alpha - \sin\gamma * \cos\alpha & \sin\gamma * \sin\beta * \sin\alpha + \cos\gamma * \cos\alpha & \cos\beta * \sin\alpha \\ \cos\gamma * \sin\beta * \cos\alpha + \sin\gamma * \sin\alpha & \sin\gamma * \sin\beta * \cos\alpha - \cos\gamma * \sin\alpha & \cos\beta * \cos\alpha \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} \quad (1)$$

transformed by means of a translation and three rotations into a local Cartesian coordinate system whose origin is located at the beginning of the MLS trajectory and being the x-axis coincident with the average direction of the vehicle. The coordinate system change is carried out through equation (1), in which (x, y, z) and (X, Y, Z) are the 3D coordinates of local and global systems respectively, (ΔX, ΔY, ΔZ) are the three components of the translation (Figure 3(a)), and β, α, and γ represent the rotation angles about the x, y, and z axes, respectively (Figure 3(b)).
This procedure is carried out individually for every section in which the original point cloud was divided in the former step. Once the coordinate system is changed, a filtering process is then carried out to reduce the point cloud size. Every point located under the GPS antenna has a negative height, and it is assumed that those points that represent curbs are below the antenna. For this reason, only the points with local negative z-coordinate values are kept, and the points located over the GPS antenna height are removed. Fig. 4 shows every step of the preprocessing stage on the slice 1 of Test Site 1.

3.2. Rasterization: In this step, to simplify the analysis and reduce the computing cost of segmentation, the 3D point cloud is projected in a 2D grid that represents the XY surface, moving from a 3D cloud to a 2D raster image. Moreover, in a raster image, it is possible to apply image analysis techniques to detect those pixels that contain points that belong to a roadside. The efficiency of the detection process will depend on the pixel size of the created image. The cell size also depends on the point density of the point cloud: the higher the point cloud’s density, the higher the image resolution can be, but this can also require more computation resources. In any case, the grid spacing must be large enough to contain a significant number of points, but small enough to allow only a small number of salient features in each cell [23]. An empirical rule has been created to determine the optimum cell size of the grid for the rasterization of every point cloud. This rule depends on the distance between consecutive scans in the central part of the detected point cloud. Thus, the cell size must be between 4 and 5 times larger than the distance between consecutive scans. We have found that these values allow the proper detection of curbs with different scan densities and curb widths. For each pixel, two values are calculated and saved as digital values (DVs) in the grid: the difference between the highest and the lowest point of all points contained in the studied cell (resulting in an image similar to a normalized digital surface model (nDSM), and the number of points contained in every cell. Thus, the rasterization step provides two images: one with DVs representing the height difference between the highest and the lowest points (referred to as a nDSM from now on) and another in which each pixel’s DV is the number of points contained within it (referred to as image density from now on). Normally, the intensity of the laser reflected pulse is also available in MLS data, but we do not consider that information relevant for this work.

3.3. Segmentation: This step consists of segmenting the rasterized 3D point cloud in order to select those pixels of the image that are candidates to represent a curb. It is divided into two stages: thresholding and morphological operations. These two stages provide a binary 2D image; the points of the measured cloud that are contained within the 1-value pixels are extracted and considered curbs.



(a)

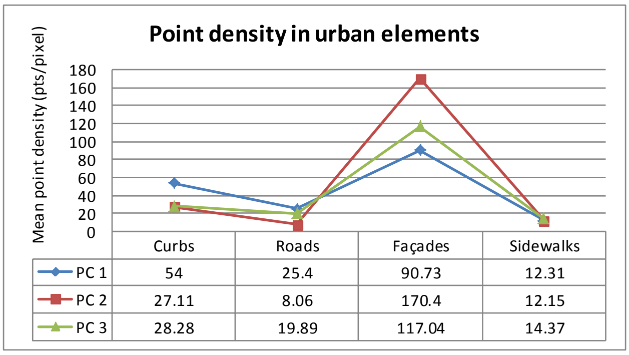
(b)

(c)

(d)

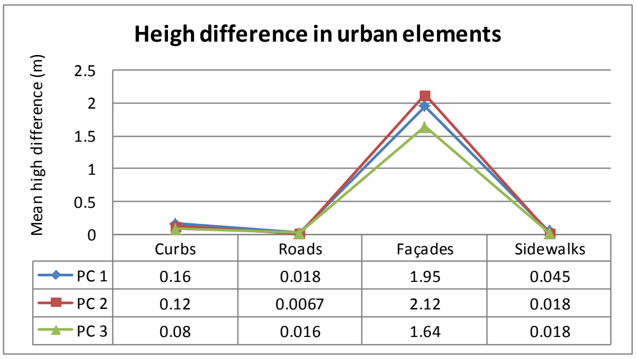
Figure 4 - (a) point cloud corresponding to slice 1 in Test Site 1 in UTM coordinate system, (b) after the coordinate system change, (c) filtering in negative and (d) positive heights

3.3.1. Thresholding: There are several features in a point cloud data that can be used to segment the cloud or for feature extraction issues, such as the normal and the curvature or the reflected laser intensity among others. For curb segmentation, two features of the real curb feature were taken into account: the difference between the highest and the lowest point of a curb and the point density in these elements. Curbs have a certain height (normally higher than 5 cm and lower than 25 cm) and a higher point density than horizontal surfaces due to their vertical face. Figures 5 and 6 shown the value of the considered features for four different urban elements in the negative height point cloud: curbs, roads, façades, and sidewalks. Surfaces that are orthogonal to the laser pulses (façades and curbs) show a higher density than those that are nearly parallel to the laser pulses (roads and sidewalks) [27]. As can be seen in Figure 6, the point density in curbs is higher than in horizontal elements like roads and sidewalks, but lower than higher surfaces like façades. Furthermore, the height difference in curbs is much lower than other vertical elements as the façades, but a bit higher than horizontal classes.



	Curbs	Roads	Façades	Sidewalks
PC 1	54	25.4	90.73	12.31
PC 2	27.11	8.06	170.4	12.15
PC 3	28.28	19.89	117.04	14.37

Figure 5 – Point density of four urban elements in three different point clouds



	Curbs	Roads	Façades	Sidewalks
PC 1	0.16	0.018	1.95	0.045
PC 2	0.12	0.0067	2.12	0.018
PC 3	0.08	0.016	1.64	0.018

Figure 6 – Height difference in four urban elements in three different point clouds

Those pixels in the nDSM image whose digital values fall between two established thresholds (Hmin and Hmax in Eq (2)) and whose density image DV is higher than a minimum threshold (Dmin) (Eq (3)) are selected as curb-candidate pixels. This information is saved in a binary image in which pixels are labeled with 1 if they represent curb-candidate pixels and 0 otherwise.

$$\text{Hmin} < \text{nDSM}[i,j] < \text{Hmax} \tag{2}$$

$$\text{Density image } [i,j] > \text{Dmin (points/pixel)} \tag{3}$$

Pixels corresponding with the background, (valued 0) will not be taken into account in the following step of the detection method. There will be more pixels, in addition to those curbs, which fulfill the conditions imposed and should be eliminated via the following process.

3.3.2. Morphological operations: A morphological opening is applied to the binary image obtained in the former step of the method. This morphological operation is conceptually composed of two phases: erosion and dilation [28]. The erosion operation will first remove those isolated pixels that do not represent a curb, but satisfy the conditions established in the thresholding step. During the dilation operation, every pixel that is 4-connected with the candidate pixels of the original binary image is added to the curb candidate set of pixels also labeled as 1.

An example of the performance of the opening operation can be seen in Fig. 8. Dilation is applied to an image in which a linear structure and three isolated pixels are presented (Fig. 8(a)). The linear element represents a curb, and the isolated pixels correspond to those satisfying the rules of the thresholding step, but not representing a roadside. By applying the morphological operator, this salt and pepper effect is removed (Fig. 8(b)) and the linear structure of the candidate pixel region grows (Fig. 8(c)). The morphological processing is applied two times over the binary image to remove those groups of pixels that do not represent a curb. The first time the morphological operation is applied in a 5x5 window and the second time in a 3x3 window. The results of every step of the method can be seen in Figure 7.

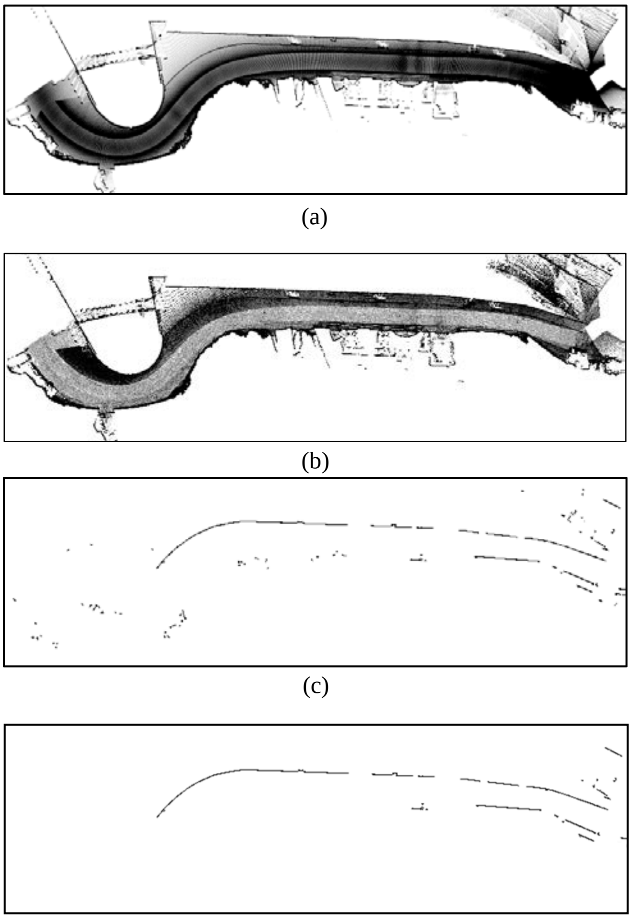

(a)
(b)
(c)
(d)

Figure 7 – Results obtained in every step of the proposed method for slice 1 of dataset 1: (a) and (b) are the density and the nDSM image, and (c) and (d) are the results of the thresholding and the morphological operations, respectively

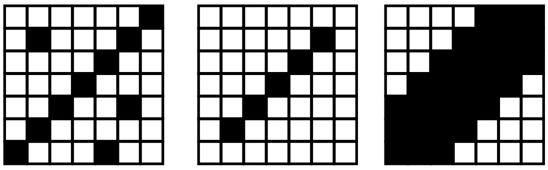
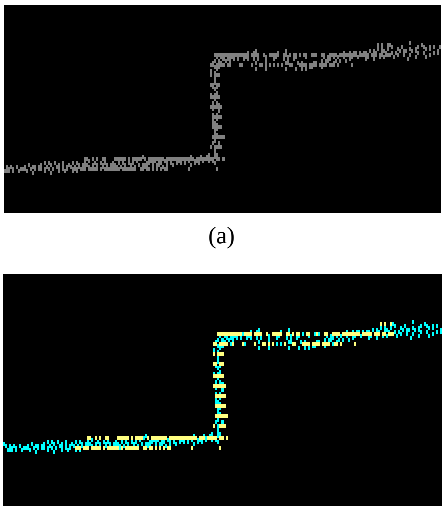


Figure 8 - (a) synthetic image representing a linear structure and three isolated pixels, (b) represents the image (a) once the erosion is applied and (c) is the result of the dilation opeartion

3.4. Segmented point cloud: The final goal of this method is the segmentation of the existing curbs in the point cloud. Thus, once the curb detection in the binary image is carried out, it is time to move from the 2D raster image to the 3D point cloud. The new point cloud is just composed by those points belonging to the negative Z-value cloud and whose (x, y) coordinates are within the 1-value pixels of the binary image obtained in the former step. The segmented point cloud is in the local coordinate system. In order to combine the measured point cloud and the segmented one, the original global coordinate system is recovered and both clouds are superimposed (Figure 9).



(b)

Figure 9 – (a) profile of a curb in the measured point cloud and (b) segmented curb: the detected curb in yellow and the background points in blue

4. RESULTS

4.1. Test Site 1

4.1.1. MLS Sensor

To determine the accuracy of the proposed method, two test sites were selected, and one dataset was acquired by the IP-S2 Compact + system (Fig. 10), produced by Topcon Inc. One advantage provided by the MLS is the possibility of generating a geo-referenced point cloud combining laser scanner data with IMU and a GPS installed in the mobile platform. The IP-S2 incorporates a dual frequency GNSS receiver, an IMU and connection to external wheel encoders, which receive odometry information. These three systems provide a highly accurate 3D position for the vehicle. The IP-S2 Compact + scanner is equipped with 5 laser scanners that collect 150,000 points per second at a range of 40 m, with a vertical field of view of 360º. It is also equipped with a panoramic camera that delivers 360º spherical imagery. The system specifications are shown in Table 1.

Table 1. Manufacturer specifications for the IP-S2 Compact + scanner

Operation range	0.7-80 m
Field of view	190º per scanner – 360º for 5 scanner option
Absolute accuracy	45 mm
Pulse repetition rate (PRR)	IEC class 1 eye-safe
Scan frequency (SF)	100 kHz
Angular resolution	0.5º-1º
Heading (IMU)	0.0667º
Roll and pitch (IMU)	0.025º

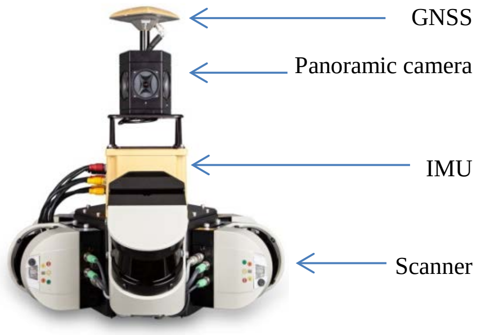


Figure 10 – IP-S2 Compact + system used to measured the point cloud used in Test Site 1

4.1.2. Test Site 1

The point cloud from Test Site 1 was measured in Cudillero, a town in the north of Spain. This is a typical fishermen village with low houses and very narrow streets, sometimes only one-way roads and very tight sidewalks. This point cloud covers a 400-meter long street with both straight and curved sections (Fig. 12). The presence of building stone walls, trucks, and other obstacles, such as fences or bar terraces, makes the roadside detection more difficult. The measured point cloud comprises more than 6 million points. The values of the parameters for the test site 1 are listed in Table 2.

Table 2 – Algorithm settings used in the test sites 1

Pixel size	20x20 cm
Δh (Hmin and Hmax)	0.05 < nDSM < 0.20
Point density (Dmin)	20 points/pixel

Results: For a visual analysis of the results obtained in Test Site 1, (Fig. 12(a)) the detected curbs have been superimposed on an ortho-image of the studied area in Test Site 1 for visual analysis. The result of the curb edge segmentation method is thoroughly shown in three details: A, B and C. Detail A represents a road section with a sidewalk on one side and a stone wall on the other side, with no sidewalk. Detail B is a curved section of a street, and Detail C is focused on a street with sidewalks on both sides of the road. For each detail, three images are shown: a street-level view (Fig. 12(b), (e) and (h)), the original point cloud of the detail (Fig. 12(c), (f) and (i)) and the result of the segmentation, in which curbs are represented in yellow and the background in blue (Fig 12(d), (g) and (j)). It can be seen that the proposed method correctly detected the existing curbs both in straight and curved sections.

Reference data: in order to evaluate the accuracy of the curb detection, a manual extraction of the road boundaries from the original point cloud was carried out. It was performed by digitizing the observed road borders from the point cloud as the ground truth data. For Test Site 1, the ground truth has a length of almost 520 meters. The evaluation of the results was carried out by comparing the curbs extracted by the proposed method with the previously compiled ground truth. This was performed using three indices commonly used in the evaluation of road detection: completeness (Eq. (8)), correctness (Eq. (9)), and quality (Eq. (10)) [29, 30].

$$Completeness = \frac{length\ of\ matched\ reference}{length\ of\ reference} = \frac{TP}{TP+FN} \quad (4)$$

$$Correctness = \frac{length\ of\ matched\ extraction}{length\ of\ extraction} = \frac{TP}{TP+FP} \quad (5)$$

$$Quality = \frac{length\ of\ matched\ extraction}{length\ of\ extracted+unmatched\ reference} = \frac{TP}{TP+FP+FN} \quad (6)$$

where TP (true positive) represents the length of the curb detected that matches the reference roadside, FP (false positive) represents length of the detected curbs that do not match the ground truth, and FN (false negative) represents the total length of the undetected curbs that exist in the ground truth.

The proposed method identified 525.1 meters as curbs from the Test Site 1 data; 35.8 meters from such amount represent FP caused by low vegetation and elements with geometry similar to those of curbs, such as isolated stones, car bottoms, and stair steps; 489.3 meters of the detected curb matched the ground truth curb, and 30.2 meters belonging to the ground truth were not detected due to the occlusion of the curb by low vegetation or the existence of other elements, such as pedestrians or bins, over the curb (Fig. 14(a)). In these cases, is not possible to achieve the detection because of the change in the curb geometry. The precision indices of our method for Test Site 1 are summarized in Table 3. It can be seen that completeness and correctness are both above 90% (94.2 and 93.2, respectively) and the quality value is 88.11%.

Table 3 – Accuracy of the detection method at Test Site 1

Test Site 1	Data present curbs
Algorithm detected (AD)	525.1 m
User detected (UD)	519.5 m
False positive (FP)	35.8 m
False negative (FN)	30.2 m
True positive (TP = AD-FP)	489.3 m
Evaluation indices	
Completeness	94.2%
Correctness	93.2%
Quality	88.11%

4.2. Test Site 2

4.2.1. MLS Sensor

The point cloud used in Test Site 2 was measured by a Riegl VMX-250 (Figure 11). This system integrates two LIDAR sensors VQ-250 and an IMU/GNSS unit, and it is deployed on the rear part of a van. The VQ-250 are rotational sensors that acquire points with a 360º field of view on planes set, in this case, at 45º to the horizontal and 45º to the trajectory (i.e. driving direction). The system specifications are shown in Table 4.

Table 4. Manufacturer specifications for the RIEGL VMX-250 sensor

Operation range	1.5-200 m
Field of view	80º per scanner
Absolute accuracy	5mm (σ)
Pulse repetition rate (PRR)	IEC/CDRH class 1 eye-safe
Scan frequency (SF)	150 kHz/sensor
Angular resolution	0.5º-1º
Heading (IMU)	0.015º
Roll and pitch (IMU)	0.005º

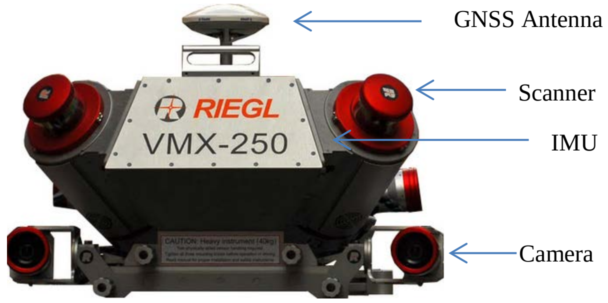


Figure 11 – Riegl VMX-250 system used to measured the point cloud used in test site 1

4.2.2. Test Site 2

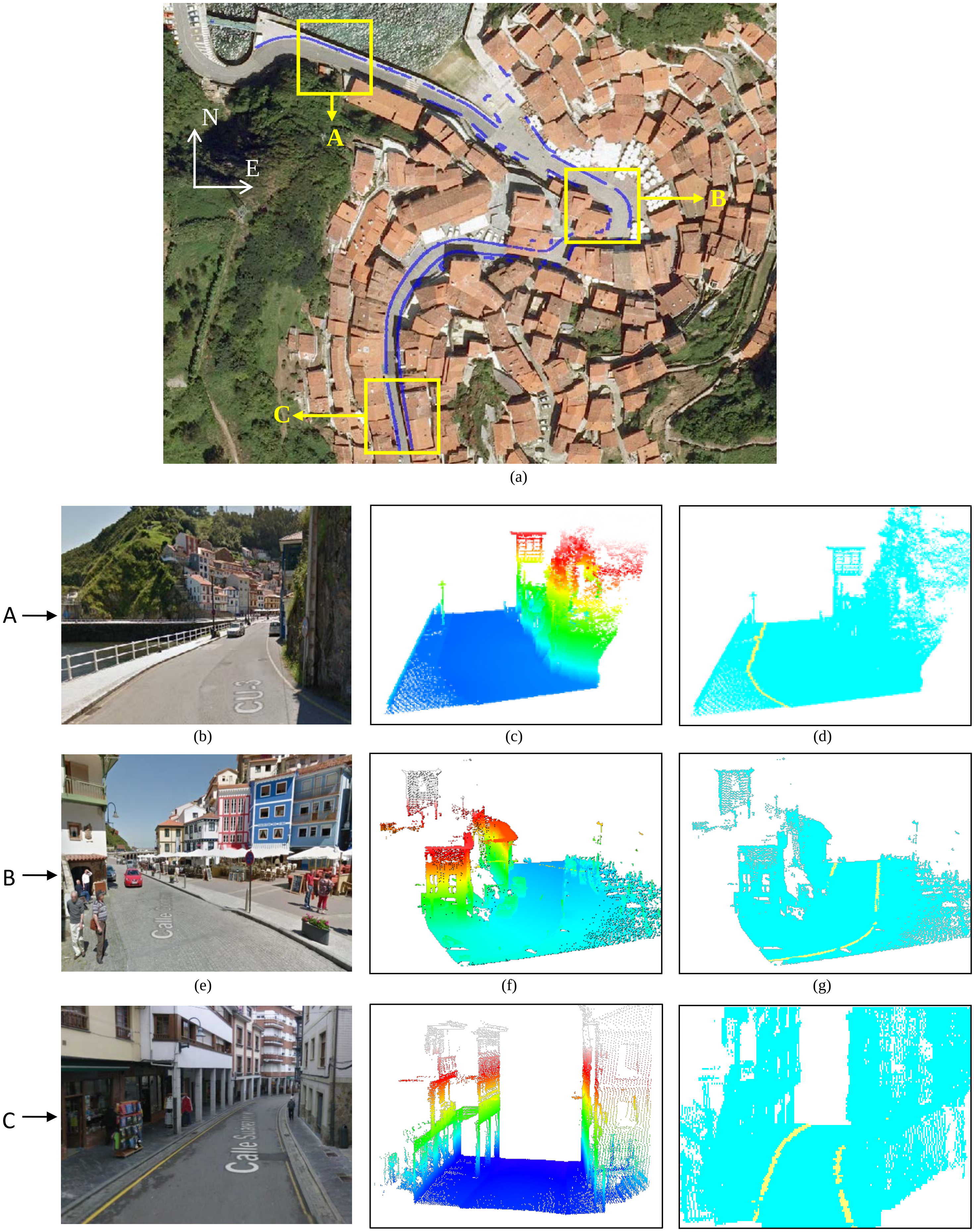
Test Site 2 corresponds to a 200-meter long street section in Horn, a city in the northeast of Austria (Figure 13(a)). This is a typical urban area that has a road with structured road boundaries in the form of curbs and ramps at crosswalks and garages. One side of this street is reserved for parking, which creates shadows in the point cloud, making the detection of curbs challenging. The point cloud corresponding to Test Site 2 consisted of more than 11.5 million points. The values of the parameters used for the test site 2 are listed in Table 5.

Table 5 – Algorithm settings used in the test sites 2

Pixel size	5x5 cm
Δh (Hmin and Hmax)	0.1 < nDSM < 0.20
Point density (Dmin)	20 points/pixel

Results: The segmented point cloud that represents curbs was vectorized and superimposed on an ortho-image of the studied area (Fig. 13(b)). The outcome of the proposed method is shown in detail in Details A and B. Detail A (Fig. 13(c)–(e)) represents a section of the street in which there is a curved curb and a sidewalk occluded by the shadow of a parked car. The proposed method correctly determines the curved section, but it is not possible to determine the curb in the occluded sidewalk due to the lack of information. Detail B (Fig. 13(f)–(h)) is focused on a zebra crossing; in this road section, the method correctly determined the existing curbs, but not the ramps to the zebra crossing due to the different height of these elements.

Reference data: as in Test Site 1, a ground truth was created by digitizing the observed curbs from the point cloud to evaluate the accuracy of the curb segmentation. The ground truth has a length of more than 220 meters. The completeness, correctness,



(a)

(b)

(c)

(d)

(e)

(f)

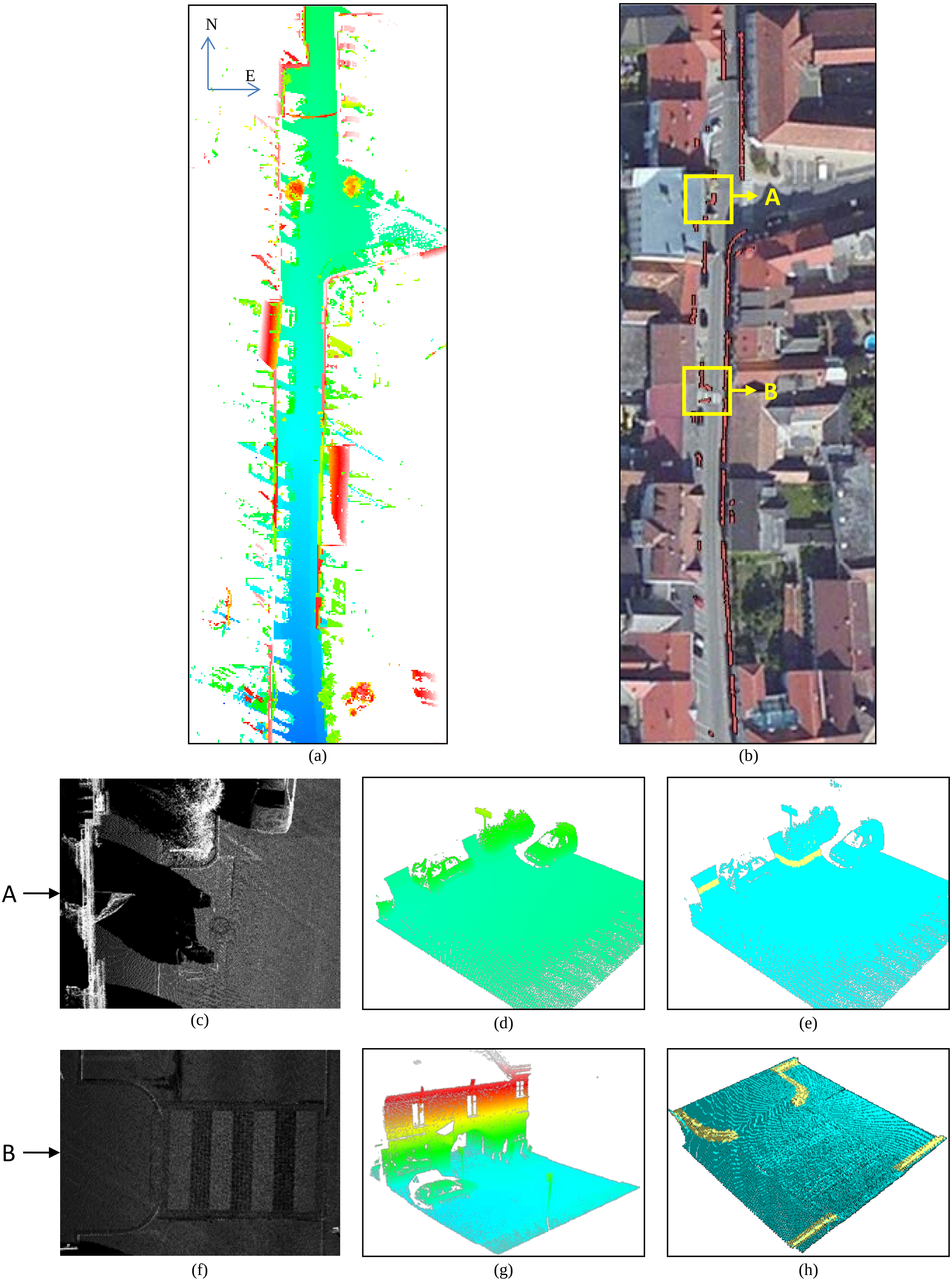
(g)

(h)

(i)

(j)

Figure 12 – (a) The segmented point cloud (in blue color) overlapping an aerial image of Test Site 1 with three detail views. For every detailed view, street-level imagery (first column) is shown, as well as the original point cloud (second column) and the result of the proposed method (third column). Detailed view A is shown in (b), (c), and (d); Figures (e), (f), and (g) correspond to detailed view B, and C is represented in (h), (i), and (j).



(a) (b)

(c) (d) (e)

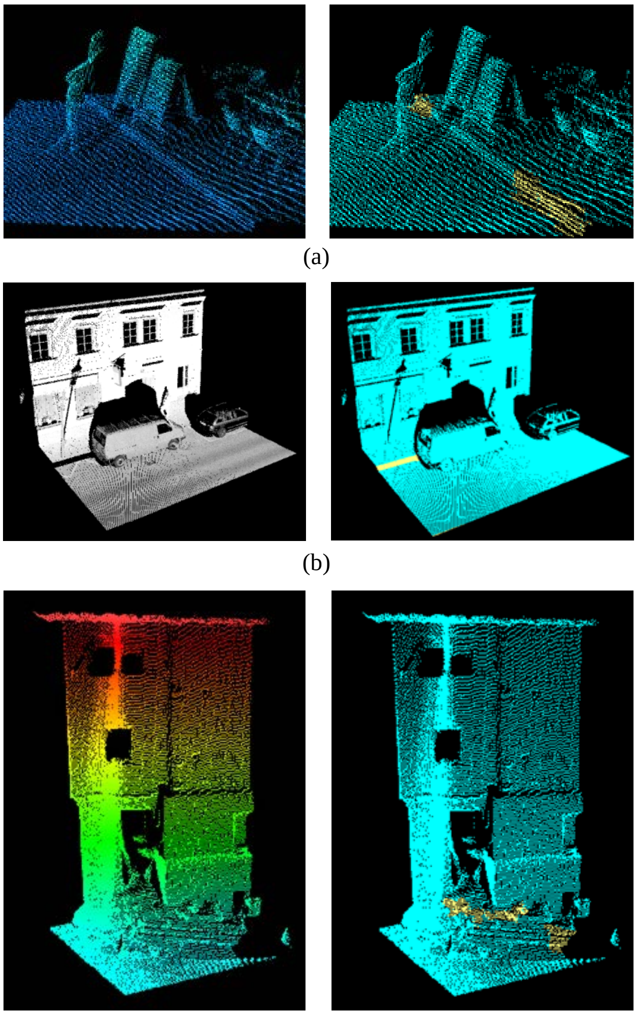
(f) (g) (h)

Figure 13 – (a) Point cloud measured by a RIEGL sensor used as Test Site 2, (b) ortho-image of the studied area overlapped by segmented curbs with two details. For both details, a plan view of the intensity lidar data (first column), the original point cloud (second column), and the result of the proposed method are shown with the curbs in yellow and the background in blue (third column)

and quality indices were obtained by comparing the segmented curbs with the ground truth, and the results are summarized in Table 6. The proposed method labeled 231.61 meters as curbs from the Test Site 2 data. Among them, 21.4 meters represent false positives caused by elements with geometry similar to curbs, such as car bottoms and stair steps (Fig. 14(c), (d)). 14.55 meters of curbs present in the reference data were not detected by the proposed method due to the short distance existing between parked cars (Fig. 14(b)) or other elements over the curbs (Fig. 14(a)). More than 210.21 meters of the detected curb matched the ground truth curb. The parameters used to measure our method's accuracy are shown in Table 6. The proposed method achieved a completeness of 93.52%, a correctness of 90.76%, and a quality of 85.42%.

Table 6 – Accuracy of the detection method at Test Site 2

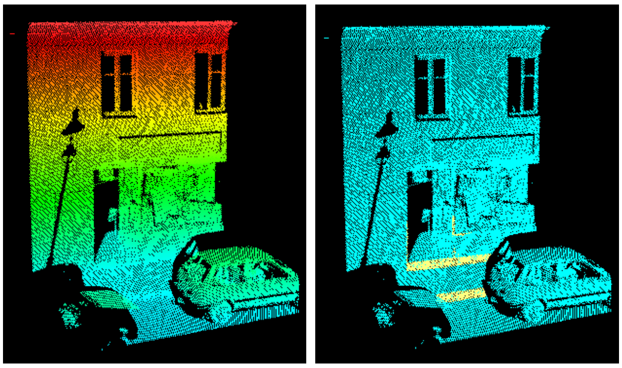
Test Site 2	Data present curbs
Algorithm detected (AD)	231.61 m
User detected (UD)	224.76 m
False positive (FP)	21.4 m
False negative (FN)	14.55 m
True positive (TP = AD-FP)	210.21 m
Evaluation indices	
Completeness	93.52%
Correctness	90.76%
Quality	85.42%



(a)

(b)

(c)



(d)

Figure 14 – (a) and (b) represent false negative cases. In (a), the curb is not detected due to an element over it; in (b), the short curb between cars is removed in the morphological operation procedure. (c) and (d) exemplify two false positives; in both cases some elements such as stair steps or benches, with more similar geometry than curbs, are wrongly determined.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, a method to detect curbs from MLS cloud data was presented. This method begins with a coordinate system transformation, moving from a global coordinate system to a local one. In order to simplify the calculations involved in the procedure, a rasterization over the XY plane was carried out, passing from the 3D data to a 2D image. Two features were considered in the rasterization: the number of points contained in every cell, and the height difference between the highest and the lowest point for each cell. Different image processing techniques such as thresholding and an opening morphological operation were applied to determine the location of curbs in the image.

The method was tested in two datasets measured by different MLS sensors, both corresponding to urban environments. The results obtained show completeness and correctness indices higher than 90% and a quality value around 85% in both test sites. From these results, it is possible to conclude that the proposed method can be useful: (1) for curb detection in straight and curved road sections, and (2) for MLS data and stereo vision point clouds, due to its independent scanning geometry. However, it is still difficult to deal with occluded curbs in shadowed areas and false positives caused by elements with similar properties than the curbs. In the near future, other variables will be incorporated to enhance this method, bridging to decrease the false positives rate, curb edge detection, and estimate the location of road boundaries when they are occluded in the point cloud.

Acknowledge

The authors would like to thank the Spanish Ministry of Science and Innovation for financial support, Project No. CGL2010-15357/BTE and Project No. BIA2011-26915. We would also like to thank Topcon Inc. and RIEGL Inc. for providing data.

References

[1] R. Wack and A. Wimmer, "Digital terrain models from airborne laser scanner data-a grid based approach," *International Archives of Photogrammetry, Remote Sensing and Spatial*

Information Sciences, vol. XXXIV, pp. 293-296, 2002.

[2] R. C. Tse, C. Gold, and D. Kidner, "3D City Modelling from LIDAR Data," in *Advances in 3D Geoinformation Systems*, P. van Oosterom, S. Zlatanova, F. Penninga, and E. Fendel, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 161-175.

[3] G. Priestnall, J. Jaafar, and A. Duncan, "Extracting urban features from LiDAR digital surface models," *Computers, Environment and Urban Systems,* vol. 24, pp. 65-78, 2000.

[4] R. Schmidt, H. Weisser, P. Schulenberg, and H. Goellinger, "Autonomous driving on vehicle test tracks: overview, implementation and results," in *Proceedings of the IV IEEE Intelligent Vehicles Symposium*, 2000, pp. 152-155.

[5] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, H. Bae, T. Brown, D. Demitrish, J. Struble, and M. Darms, "Autonomous driving in urban environments: boss and the urban challenge," *Journal of Field Robotics,* vol. 25, pp. 425-466, 2008.

[6] X. Ding, W. Kang, J. Cui, and L. Ao, "Automatic extraction of road network from aerial images," in *International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA)*, 2006, pp. 220-223.

[7] X.-j. Cheng, H.-f. Zhang, and R. Xie, "Study on 3D laser scanning modeling method for Large-Scale history building," in *International Conference on Computer Application and System Modeling (ICCASM)*, 2010, pp. V7-573-V7-577.

[8] D. Gonzalez-Aguilera, A. L. Muoz, J. G. Lahoz, J. S. Herrero, M. S. Corchon, and E. Garcia, "Recording and Modeling Paleolithic Caves through Laser Scanning," in *International Conference on Advanced Geographic Information Systems & Web Services (GEOWS)*, 2009, pp. 19-26.

[9] R. Argüelles-Fraga, C. Ordóñez, S. García-Cortés, and J. Roca-Pardiñas, "Measurement planning for circular cross-section tunnels using terrestrial laser scanning," *Automation in Construction,* vol. 31, pp. 1-9, 2013.

[10] C. Cabo, C. Ordoñez, S. García-Cortés, and J. Martínez, "An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 87, pp. 47-56, 2014.

[11] H. Park, S. Lim, J. Trinder, and R. Turner, "3D surface reconstruction of Terrestrial Laser Scanner data for forestry," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2010, pp. 4366-4369.

[12] M. Rutzinger, S. Oude Elberink, S. Pu, and G. Vosselman, "Automatic extraction of vertical walls from mobile and airborne laser scanning data," presented at the The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Paris, 2009.

[13] M. Rutzinger, B. Höfle, S. Oude Elberink, and G. Vosselman, "Feasibility of Facade Footprint Extraction from Mobile Laser Scanning Data," *PFG Photogrammetrie, Fernerkundung, Geoinformation,* vol. 2011, pp. 97-107, 2011.

[14] K. Hammoudi, F. Dornaika, and N. Paparoditis, "Extracting building footprints from 3D point cloud using terrestrial laser scanning at street level," presented at the ISPRS/CMRT09, Paris, 2009.

[15] B. Douillard, J. Underwood, V. Vlaskine, A. Quadros, and S. Singh, "A Pipeline for the Segmentation and Classification of 3D Point Clouds," in *Experimental Robotics*. vol. 79, O. Khatib, V. Kumar, and G. Sukhatme, Eds., ed: Springer Berlin Heidelberg, 2014, pp. 585-600.

[16] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of Normals as a Multi-scale Operator in Unorganized Point Clouds," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, 2012, pp. 501-508.

[17] B. Yang, L. Fang, and J. Li, "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 79, pp. 80-93, 2013.

[18] H. Hanzhang, D. Munoz, J. A. Bagnell, and M. Hebert, "Efficient 3-D scene analysis from streaming data," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 2297-2304.

[19] J. Ruyi, K. Reinhard, V. Tobi, and W. Shigang, "Lane detection and tracking using a new lane model and distance transform," *Machine Vision and Applications,* vol. 22, pp. 721-737, 2011.

[20] R. Labayrade, J. Douret, and D. Aubert, "A multi-model lane detector that handles road singularities," in *IEEE Intelligent Transportation Systems Conference (ITSC '06)*, 2006, pp. 1143-1148.

[21] G. Zhao and J. Yuan, "Curb detection and tracking using 3D-LIDAR scanner," in *IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 437-440.

[22] T. Weiss and K. Dietmayer, "Automatic Detection of Traffic Infrastructure Objects for the Rapid Generation of Detailed Digital Maps using Laser Scanners," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 1271-1277.

[23] D. Belton and K. Bae, "Automating post-processing of terrestrial laser scanning point clouds for road feature surveys," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. XXXVIII, pp. 74 - 79, 2010.

[24] A. Hervieu and B. Soheilian, "Semi-automatic road/pavement modeling using mobile laser scanning," presented at the City Models, Roads and Traffic - CMRT13, Antalya, Turkey, 2013.

[25] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "An automated algorithm for extracting road edges from terrestrial mobile LiDAR data," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 85, pp. 44-55, 2013.

[26] VirtualGrid. (2014). Available: http://vrmesh.com

[27] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data," *Remote Sensing,* vol. 2, pp. 641-664, 2010.

[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*: Pearson Education, 2011.

[29] C. Heipke, H. Mayer, C. Wiedemann, and O. Jamet, "Evaluation of Automatic Road Extraction," *International Archives of Photogrammetry and Remote Sensing,* vol. 32 (3-2W3), pp. 47-56, 1997.

[30] J. Hu, A. Razdan, J. C. Femiani, C. Ming, and P. Wonka, "Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 4144-4157, 2007.

A GIS DATABASE ENGINE FOR LINKING CALCULATION AND STORAGE

Chen Wu [a], Qing Zhu [a, b,*], Weiping Xu [a], Yeting Zhang [a], Xiao Xie [a], Feng He [a], Yan Zhou[c]

[a] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, P.R.China - wuc_oct17@126.com, (zhuqing, zhangyeting)@lmars.whu.edu.cn,
yuqiexing@gmail.com, xiexiao1229@163.com, gis_hf@whu.edu.cn
[b] National-local Joint Engineering Laboratory of Spatial Information Technology for High-speed
Railway Running Safety, Southwest Jiaotong University, P.R.China
[c] School of Resources and Environment, University of Electronic Science and Technology of China,
P.R.China - zhouyan_gis@uestc.edu.cn

KEY WORDS: Real-Time Application, GIS Database Engine, MMDB, DRDB, RDBMS, NoSQL

ABSTRACT:

The fast-paced, constantly changing world produces massive data, which poses challenges for GIS. The volume, velocity, variety and value of big data challenge the data management ability of GIS. Traditional GIS database engine limits its functional to the data organization of geographic features in Static GIS or managing time-varying geospatial data in Temporal GIS. It offers offline manner to store and manage spatial and temporal information of geographical entity, which cannot satisfy the time-sensitive and dynamic processes which demand real-time information and relevant data reaches responders prior making intensive calculation and critical decisions, due to the restricted ability and imbalance of communication, calculation and storage. This paper designs a new collaborative DRDB and MMDB GIS database engine linking calculation and storage for real-time application. Based on functional segmentation, this engine consists of three parts that respectively cover the need of real-time accessing, composite calculating and massive storage which greatly promotes the integration among calculation, communication and storage. Finally, as a typical real-time application of geospatial data storage and calculation, suspect tracking based on video surveillance in public security is illustrated and proves the validity of the proposed GIS database engine.

* Corresponding author.

1 INTRODUCTION

It is becoming an urgent demand for government and institute to rapidly and accurately perceive and decode dynamic changes in complex geographical environment and make valid guidance for emergency prediction, assessment and response in public safety or disaster monitoring (Kwan et al., 2005, Laurini et al., 2005). For example, china builds the large-scale video monitoring network, covering multi-level city, which dynamically accesses nearly TB-scale real-time video data per day and stores totally PB-scale video files which construct important basic data base for the national public safety and emergency management system of science and technology (Li et al., 2014). The big data is the solid foundation of critical decision. These real-time applications are time-sensitive and dynamic processing, demanding real-time information and relevant data reaches responders prior making intensive calculation and critical decisions (Alamdar et al., 2014). However, the ability of calculation and storage is separated and imbalanced which makes the real-time accessing and decision making not seamlessly coupling. Therefore, the challenge emerges how to link intensive calculation and storage of massive real-time data such accessible but overwhelming dynamic data, including readings from large-scale distributed instruments and massive data stream generated by domain computation models. GIS as a powerful tool of geospatial information science is expected to be adapted to all these real-time data involvement and data-driven application possibilities (Zerger et al., 2003, Al-Sabhan et al., 2003). More specifically, GIS database engine long-term focuses on relatively static mode of organizing, managing and analysing historical data needs to be shifted to a far more dynamic process of accessing, calculation, simulation and decision making (Ulusoy, 1995).

Traditional GIS database engine limits its functional to the data organization of geographic features in Static GIS or managing time-varying geospatial data in Temporal GIS (Yuan, 1996, Langran, 1989, Ozsoyoglu et al., 1995). The engine offers offline manner to store and manage spatial and temporal information of geographical entity, mapping one “snapshot” of geographical world into computer databases by geospatial data persistence, followed by further development and integration of on-demand application functions operated on these database record (Erwig et al., 1999). However, it is not fit for real-time application because: (1) storing on the on-disk database, the problem of I/O storage bottleneck is becoming acute which cannot satisfy the storage efficiency of real-time data. For example, the commercial GIS database engines, such as ArcSDE, SuperMap SDX+, MapGIS, use relational database/ or object-relational database as the data storage layer, including Oracle, SQL Server, and DB2 etc. (2) real-time application requires not only accessing real-time state of related objects and detecting events, but also aggregating related real-time and historical information, for calculation, geo-simulation and decision making especially in continuously changing processes. For example, ArcGIS GeoEvent Processor is an extension for ArcGIS for Server. It can accommodate multiple streams of data flowing continuously through filters and processing steps that defined (Mollenkopf, 2013, Artz, 2013). But the processing data is wholly confined to the real-time accessing data, cannot autonomously load related historical data or knowledge library from the archive memory to make composite analysis which is more valuable. In conclusion, the existing GIS database engine fails to store dynamic update and support real-time processing of observation data acquired from the increasingly changing environment. Therefore, this paper designs a GIS database engine which is segmented to three modules that respectively cover the need for real-time accessing, composite calculating and massive storage, aiming at satisfying the demand of real-time application.

2 FUNCTIONAL REQUIREMENTS OF GIS DATABASE ENGINE TO SUPPORT REAL-TIME APPLICATION

The monitoring of natural and social process is necessary. When an exception occurs, the government can make an emergency response, such as flooding, volcanic eruptions, landslides, group incidents and so on. Usually sensor data are regularly sent to control centre, stored in a huge database with big volume, and displayed with animated demonstration when needed. Decision-makers can follow the phenomenon monitoring and schedule relevant data from the database or data warehousing when event triggers. Therefore, the data accessing, storing and calculation are splitted. However, with the development of real-time application, the emergency response must be in real-time. It means that the change detection and analysis should be conducted when data is streaming in. Data accessing, storing and calculation must be linked together. Only do like this, transactions can be fulfilled with deadline. Traditional GIS database engine get used to deal with static geographic features in Static GIS or manage time-varying geospatial data in Temporal GIS. It is not fit for real-time application for the limitation on real-time accessing and real-time calculation. In this chapter, we analyse characteristics that real-time application have and list the functionalities that a GIS database engine should provide.

2.1 Real-time Data Accessing

For real-time application, time is a critical factor. In the past, we were dealing with applications in which the time of storing or updating information was not a very critical factor. In real-time application, whatever the type of the sensor is, information must be collected in real-time, and be available to a lot of potential users immediately. Especially when a crisis occurs, the measuring periodicity is accelerated, which multiplies the transmitted data. Let’s take flood monitoring as an example. Sensors are distributed along the river to regularly measure relevant parameters, such as chemical, physical, biological and water height. If the tide is rising, the periodicity of collecting information will be increased from every hour to every minute. The consequence could be a sort of congestion in data arrival, and the system must be able to manage this sudden increase within critical specified times (Stonebraker et al., 2005, Laurini et al., 2005). So the system should increase the importance of time that all transactions of data accessing must be committed very rapidly and reduce the loss of data.

For low-latency real-time applications, interfacing with a traditional database will add excessive latency and overhead to the application. To achieve low latency, a system should be able to perform data accessing without having a costly storage operation. Traditional database systems offer general data management functionality and are designed to handle applications on static data, ranging from Online Transaction Processing (OLTP) to data warehousing. It can reliably store large, finite data-sets. It stores data on disk and indexed before any query processing takes place. The storage, indexing, and ad hoc query execution all adds a great deal of unnecessary latency to the real-time accessing that are unacceptable for data streaming in. For example, committing a database record requires a disk write and a log record which are time-consuming. For real-time applications, it is not acceptable to conduct such a time-intensive operation. In addition, observation data contains much redundant or useless information, so pre-processing is necessary. So it is not necessary to store all the data before pre-processing. Furthermore, intelligent service is triggered by the event extracted from observation data, so change detection should be processed in real-time. All these procedures should be operated real-time.

For low latency accessing, the observation data should be stored and processed entirely in main memory. Main memory databases (MMDB) have higher data throughput than traditional database, as they can avoid disk writing for most operations, if given sufficient memory (Garcia-Molina, 1992). The real-time data stored in MMDB can directly conduct data compression processing or event extraction without disk data exchange. In a word, storing real-time data in MMDB, on one hand brings MMDB’s high efficiency into full play; on the other hand excellently support the pre-processing before further utilization.

2.2 Massive Data Storing

The volume of real-time data is growing exponentially. For example, HD camera produces 3.6GB spatial-temporal data per hour and the amount of data in a city can reach PB level every day. China has over 20 million cameras throughout the country and the amount of data will reach EB level (Li et al., 2014). An international famous high resolution satellite acquires a scene of image per 5 minutes. The image of each scene can reach more than 5GB and the satellite can produce 1.4TB data per day.

As explained in chapter 2.1, the key idea to store the real-time data is the main memory. Although MMDB gives a good solution to real-time data accessing, its drawback is obvious. MMDB is suited to handle real-time business logic processing, but it’s fragile and its volume is limited. So saving the data to disk or other reliable memory regularly is necessary. In addition, in order to ensure ACID (atomicity, consistency, isolation and durability) properties of transactions, when transaction is submitted, the operations must be recorded in logs. Therefore, the real-time data after processing must be synchronized to disk database based on the time period or event driven.

Traditional storage system cannot meet the need of storing large-scale data. Relational database makes sure that all transactions are absolutely reliable by ACID to meet the requirements of reliable performance. But the database performance, scalability and concurrency are bottlenecks (Lu et al., 2014, Li et al., 2014). Distributed database is becoming increasingly important given the favourable price-performance characteristics of low-cost commodity clusters, for example NoSQL(Not Only SQL). NoSQL database tolerates inconsistency to improve reading and writing performance, high concurrency and takes a method to achieve eventual consistency. This feature can remove 40-60% of the transaction and improve the performance a lot. Furthermore, no relationship between the data makes it easy to extend that greatly improves the scalability in the architecture level (Han et al., 2011). Therefore, using NoSQL to store the massive real-time data can not only exploit its advantage of high efficiency to the full, but also take advantage of scalability.

2.3 Comprehensive Calculating

Events of interest depend partly on real-time data and partly on history. Some real-time applications desire to compute some sort of analysis starting from a past point in time and then seamlessly continue with the calculation on live data. This capability requires switching automatically from historical to live data, without manual intervention. In traditional database, data is always stored before it is queried. But in real-time application, the latest stream data are still in the processing pipe. The conventional method cannot integrate real-time and stored data for calculation seamlessly. If one algorithm can works well on historical data now, how it can be seamlessly expanded to real-time data or the mixture of historical and real-time data to make comprehensive calculation without application modification is necessary for real-time application. Therefore, it is expected that the database engine will be as ubiquitous for processing real-time data or for processing stored data. A solution will satisfy the requirement if the streaming process container MMDB is a logical extension of traditional database. The scope of integrated database can be either a real-time stream or a stored data in traditional database. Main Memory Database abandons the traditional way that Disk-Resident Database uses because it assumes that all data stores in memory without I/O. It redesigns database architecture and makes corresponding improvements in data cache, fast algorithms, and parallel operation etc. In order to shield users from the inherent complexities of dealing with streaming data or stored data, it’s essential to redesign the data model, programming interface, index and analytic capabilities to meet the need of seamlessly scheduling.

Calculation depends on different source of data. If change has been detected, it will trigger an event and schedule relevant data to make a comprehensive analysis. The relevant data may include real-time data, historical data, rule base and so on. These data are distributed in different container. In the past, the systems are passive which wait to be told what to do by command control and information transmission. Passive systems require applications to continuously poll for conditions of interest. Unfortunately, polling mode results in additional overhead on the application, and additional latency, because the polling interval is added to the processing delay. Active systems avoid this overhead by incorporating built-in event/data-driven processing capabilities. After triggering an event, the active system will load the relevant rule by the event type, and schedule all the data related together according to the detail rule described and make comprehensive analysis. All the data is scheduled autonomously that avoid blocking for external events, thereby facilitating low latency.

3 GIS DATABASE ENGINE COLLABORATING DRDB AND MMDB

In this chapter, we design a new GIS database engine oriented real-time application. The engine facilitates the coordination and implementation of real-time accessing, massive storage and comprehensive calculation. It combines the advantages of DRDB (Disk-Resident Database) and MMDB (Main Memory Database) to greatly satisfy the low latency of real-time application. The integrated organization of DRDB and MMDB effectively address the performance and agility requirements of stream processing applications, not only be extremely efficient in real time stream processing, but also shield the users from the inherent complexities of dealing with streaming data and underlying physical resources.

3.1 Architecture of GIS database engine for Real-time Application

According to the functional requirements of real-time application analysed in Chapter 2, we divide the GIS database engine into three basic modules: Real-Time accessing module (RA), Comprehensive Storing module (CS) and Autonomous Loading module (AL). Real-Time accessing module is responsible for accessing real-time observation data and detecting event. It makes full use of the characteristic of MMDB that have higher data throughput than traditional database without disk writing for most operations. Establishing a MMDB can not only support streaming data, but also support online calculating and change monitoring.

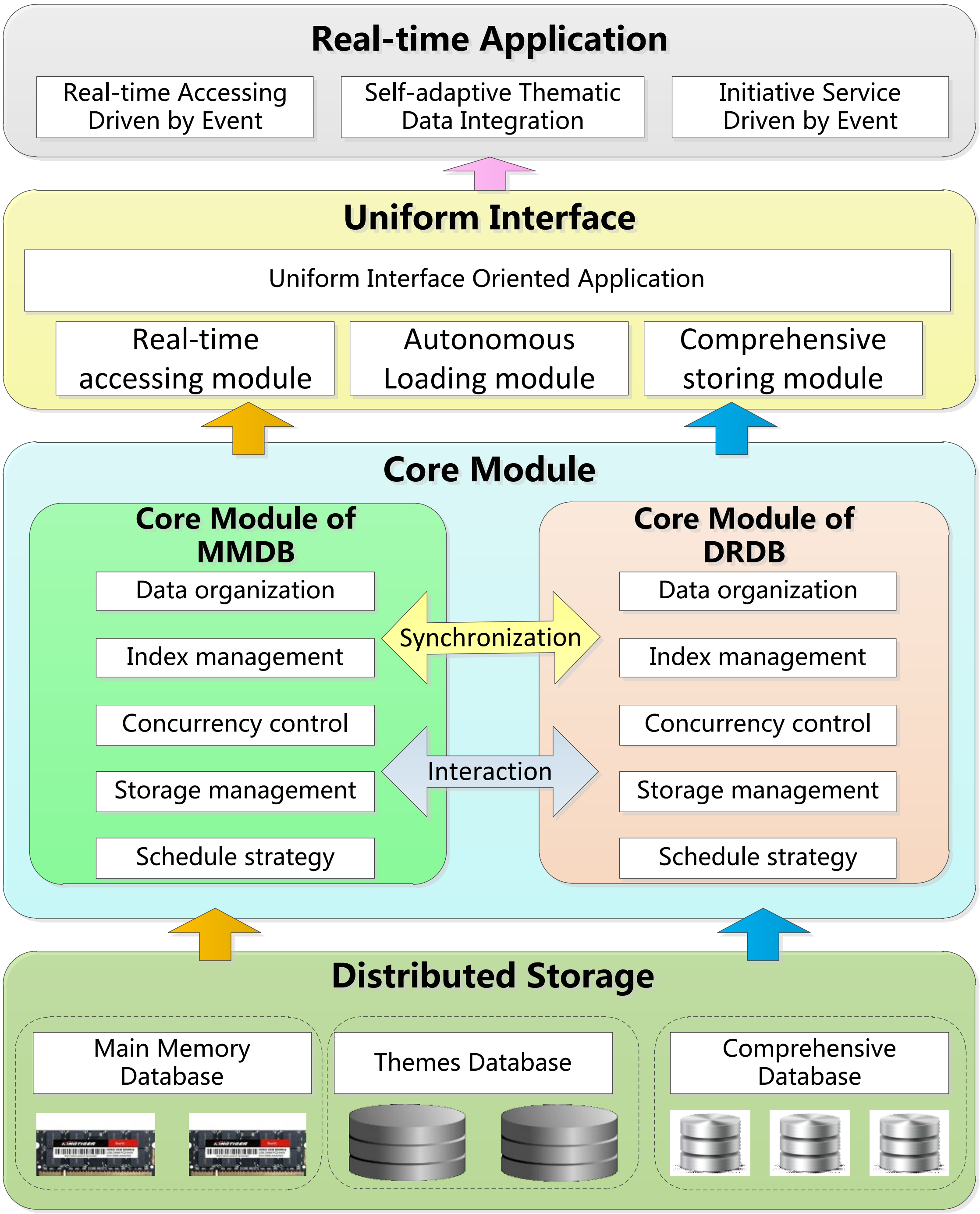


Figure 1. Architecture of GIS Database Engine

Comprehensive Storing module stores the ever-increasing real-time data and rule base and so on. It's a big container to gather all the data that real-time application needs. CS uses NoSQL database to store massive data, because NoSQL database has high performance of writing and reading and the advantage of scalability. Autonomous Loading module is in charge of loading the rule sets, crawling historical data related from CS and scheduling real-time data from RA for comprehensive calculation. It requires querying from different databases and tables, so the ability of correlative query is essential. It also need to shield inherent complexities of dealing with streaming data or stored data. AL takes advantage of the correlative query, relational integrity constraints and high frequency updates of RDBMS, establishing themes database oriented efficient information service. Based on the three basic modules, we establish a uniform interface oriented application. It guarantees the encapsulation and independency during the stages of accessing, loading and storing and makes the user transparent to the bottom implementation which is more effective for programming.

A general structure of GIS database engine is illustrated in Figure 1. The three module division can greatly support the real-time application, such as real-time accessing driven by event, self-adaptive thematic data integration and initiative service driven by event without functional conflict. RA, CS and AL are functional divisions, the implementation of these modules are the management of MMDB and DRDB. Therefore, according to the storage type, the engine includes the core module of MMDB and DRDB. Core module of MMDB contains data organization, index management, concurrency control, storage management and schedule strategy. MMDB abandons the traditional way that Disk-Resident Database uses because it assumes that all data stores in memory without I/O. The data processing speed in main memory database is much faster than the traditional database, generally more than 10 times. This characteristic makes MMDB have ability of huge throughput in transaction-intensive applications. So it needs to redesign database architecture and make corresponding improvements in data model, index, data cache, calculation algorithms, and parallel operation etc. At the same time, core module of DRDB also contains data organization, index management, concurrency control, storage management and schedule strategy. In this engine, the DRDB contain NoSQL database and RDBMS, so the module should collaborate with these two databases well. MMDB is fragile and its volume is limited. So saving the data to disk or other reliable memory regularly is necessary. Therefore, there is a synchronization module to synchronize real-time data in MMDB to disk database based on the time period or event driven. AL may crawl historical data related from CS and schedule real-time data from RA for comprehensive calculation, so there need an interaction between MMDB and DRDB, so the interaction module is responsible for this function.

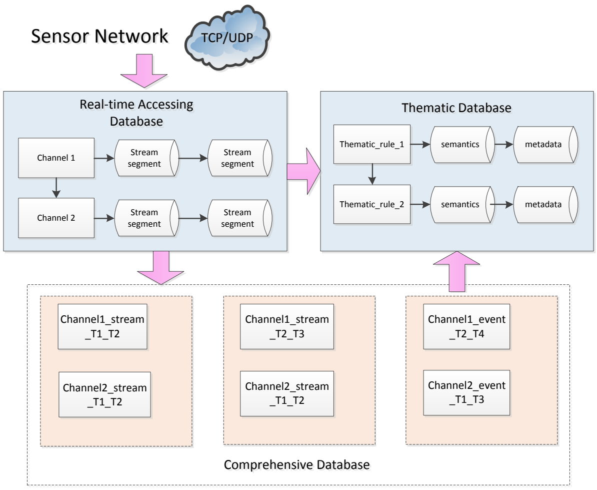


Figure 2. The distribution of Storage

3.2 Hierarchical and Distributed Data Storage

By distributing different types of data in different data containers and scheduling collaboratively, the engine can finally meet the time-critical requirements of GIS for real-time applications. How to distribute data to different storage container is necessary. Several kinds of real-time systems can be considered according to the type of temporal constraints. Usually, we speak about hard real-time when transactions must be fulfilled with hard deadline, for instance, in bank trading applications, constraints can be of millisecond magnitude. In flood monitoring, the phenomena under monitoring are slower, and we call it soft real-time. Therefore, the combination of temporal constraint and database performance is the solution to meet the requirements of real-time application. From the perspective of database itself, different database has different characteristics that suit for certain kind of data. For example, MMDB can satisfy the high efficiency and low latency of real-time accessing. NoSQL database can fit the massive storage of increasingly historical observation data. RDBMS can meet the need of correlation query, relational integrity constraints and high frequency updates for comprehensive calculation.

We divide the real-time storage to three part, real-time accessing database, comprehensive database and themes database. The storage distribution is illustrated in Figure 2. The sensor network directly connect real-time accessing database composed by MMDB. Every sensor link has a data channel in database and manages stream data separately. After online processing and change monitoring, real-time accessing database will synchronize historical data to comprehensive database composed by NoSQL databases. The ancient streaming data are distributed to guarantee the high efficiency of concurrent scheduling. Themes database based on RDBMS stores the semantic information, operation rule of business application, and the metadata of concrete data. Semantic information is the inner attributes of Geo-Object such as position, time, state, topological relation and so on. It is the description of concrete data and is usually the entry point of query which is more valuable. Correlative query of semantic information is popular and complex, so RDBMS perfectly matches the integrity constraint and correlative query of this kind of application. When user conducts business query, it will correlative query from different tables according to typical semantic data, then schedule the rule data, and then finally query the concrete data by object identifier. Therefore, the themes database has to keep integrity constraint of semantics and correlative queries.

3.3 Interaction between Calculating and Storing

Some real-time applications desire to calculate starting from a past point in time and then seamlessly continue with the calculation on live data. This capability requires switching automatically from historical to live data, without manual intervention. However, the live data stores in real-time accessing database, and the historical stores in comprehensive database. These two databases are splitted and uncorrelated. How can the applications know where to schedule the data required?

In chapter 3.1, we describe that the new GIS database engine has a uniform interface that shields users from the inherent complexities of dealing with data streams and physical resources. In the programming level, how to implement this target? As we know, MMDB has its unique index, such as AVL tree or T tree (Lu et al., 2000, Choi et al., 1996). DRDB has its unique index, such B tree. Index greatly increases the scheduling efficiency. If we can build a uniform index over the MMDB and DRDB, then we can get the data we want by the exclusive object identifier. The index of MMDB is based on the pointer address and operate memory block because data are all stayed in main memory. The index of DRDB is based on the disk address and operates disk bock. So it is difficult to merge the two indexes together. However, we can additionally build a mixed index that is based on the MMDB and DRDB. From this new index, we can know where the data is, in MMDB or DRDB. So when we schedule a data, we can query from this mixed index at first to locate the general range of the data, such as in MMDB or DRDB. Then we use the built-in index in MMDB or DRDB to query the exact data. The structure of mixed index is illustrated in Figure 3.

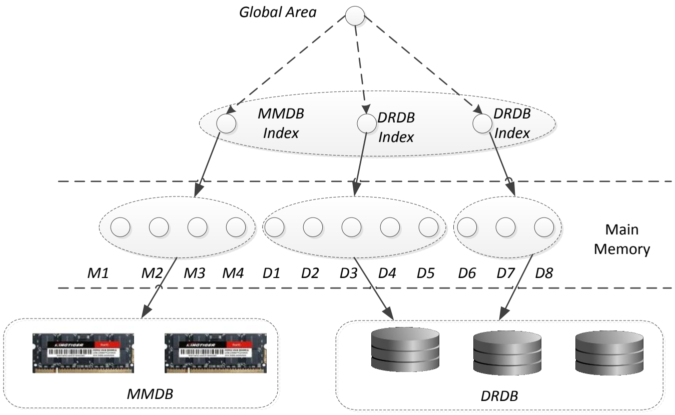


Figure 3. The Structure of Mixed Index

The mixed index is the bridge between MMDB and DRDB. Considering the frequent synchronization and interaction between MMDB and DRDB, the mixed index will update frequently. So the mixed index stays in memory permanently that greatly reduces the time cost in disk data exchange. In addition, the robustness of mixed index is the key link to maintain the normal calculation of real-time application, so the mixed index should have several synchronous backup, for example “master-slave” mode of MMDB.

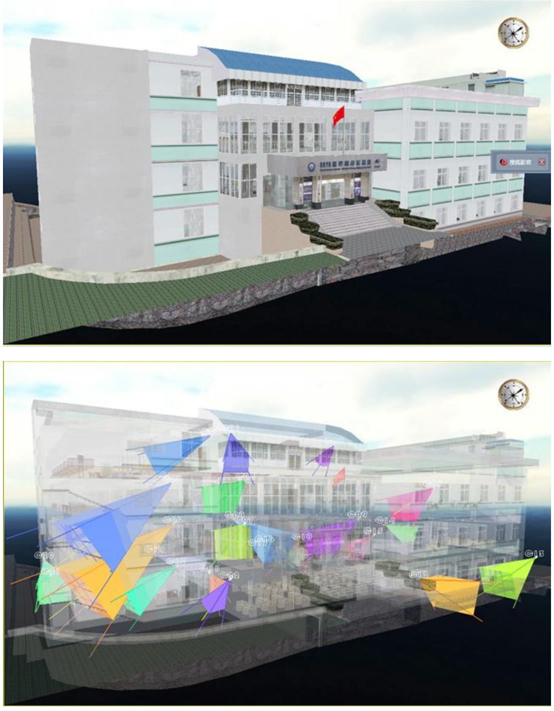


Figure 4. The Distribution of Cameras

4 A CASE OF REAL-TIME APPLICATION IN PUBLIC SECURITY: SUSPECT TRACKING BASED ON VIDEO SURVEILLANCE

Video Surveillance is one of the most typical and complex real-time application in daily life. With the trend of complex street network and indoor space, how to schedule the related video data from thousands of cameras and track path of suspicious person is becoming complicated. The GeoVideo, mapping the video frames to geographic space, promotes the security monitoring to a spatial perspective and provides a high feasible solution to this problem. Besides, video, as a single data source, cannot fully demonstrate the context of the scene or event, collaborating judgment fused with different type of monitoring data, knowledge library or spatial-temporal pattern can greatly increase the scientificity and practicality of decision making. In this chapter, the new GIS database engine proposed can be used for video surveillance in a 3 floors office building with totally 26 cameras indoor and outdoor.

The distribution of cameras is illustrated in Figure 4. The top picture is the full view of the office building. The bottom picture is in the transparent mode which demonstrates the position of cameras and the view frustum of each camera illustrated with different colour. In the application of monitoring suspicious person, RA can be used to addresses real-time observation of people within a busy environment, leading to a description of their actions and interactions, including moving object detection and tracking, object classification, human motion analysis, and activity understanding (Lee et al., 2000, Tian et al., 2005). It extract moving object, foreground, background and global object motion estimation, then map each object with enhanced semantic description and reduce redundancy object. These pre-processing can support fast retrieval and analysis for latter operations. As demonstrated in Figure 5, the suspect wears red cloth has been marked up by yellow rectangle and his position in real world has been synchronized in the 3D scene. Then the AL retrieves the indoor structure of 3D scene from CS to calculate and analyse possible trajectory according to spatial topology and accesses the real-time monitoring data along the path from RA (Kwan et al., 2005). For example, if the suspect walks up the stairs, AL will autonomously loads the camera data on the other side of the stairs according to the topological connectivity. If the person enters to an office, the AL will schedule the list of staff working in this office from CS to check whether he belongs to this office. If the man is a stranger and no staff stays at this office, AL will send a warning to the security office. At last, after the warning is cancelled, the characteristics of suspicious person, track and behaviours will be stored in CS as a case for further analysis. As demonstrated in Figure 6, the yellow line is the moving trajectory of suspect in first floor. In short, this GIS database engine well solves the need of real-time accessing massive observation data and dynamically aggregating related

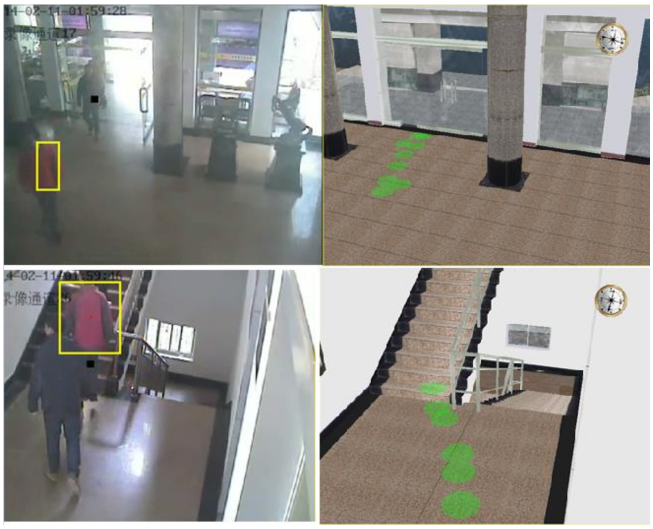


Figure 5. Suspect Monitoring in Real World and in 3D Scene

information for comprehensive calculating in real-time application.

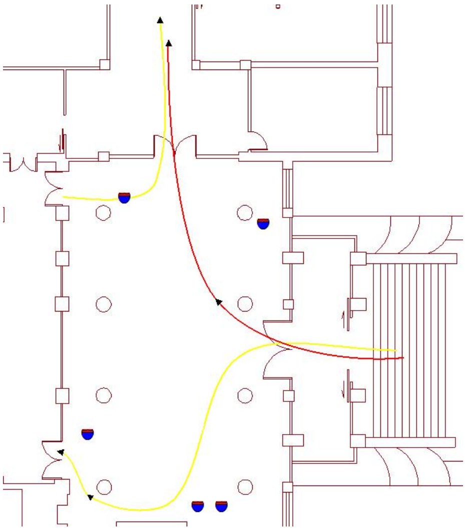


Figure 6. Moving Trajectory of Suspect

5 ACKNOWLEDGEMENTS

The research for this paper was funded by the National Technology Support Plan of China (863 Program, No. 2012BAH35B02), the National Basic Research Program of China (973 Program, No. 2011CB302306) and the National Natural Science Foundation of China (No. 41101354).

6 REFERENCE

Alamdar, F., Kalantari, M., Rajabifard, A., 2014a. An evaluation of integrating multisourced sensors for disaster management. International Journal of Digital Earth, pp.1-23.

Erwig, M., Gu, R. H., Schneider, M., Vazirgiannis, M., 1999a. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. GeoInformatica, 3(3), pp.269-296.

Li, D.R., Yao, Y. Shao, Z.F., 2014a. Big Data in Smart city. Geomatics and Information Science of Wuhan University, 39(6), pp. 631-640.

Mollenkopf, A., 2013. Using ArcGIS GeoEvent Processor for Server to Power Real-Time Applications. Esri International Developer Summit. Esri Inc.

Zerger, A., Smith, D. I., 2003a. Impediments to using GIS for real-time disaster decision support. Computers, environment and urban systems, 27(2), pp.123-141.

Al-Sabhan, W., Mulligan, M., Blackburn, G. A., 2003a. A real-time hydrological model for flood prediction using GIS and the WWW. Computers, Environment and Urban Systems, 27(1), pp.9-32.

Li, Q.Q., Li, D.R., 2014a. Big Data GIS. Geomatics and Information Science of Wuhan University, 39(6), pp. 641-644.

Kwan, M. P., Lee, J., 2005a. Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. Computers, Environment and Urban Systems, 29(2), pp.93-113.

Lu, F., Zhang, H.C., 2014a. Big Data and Generalized GIS. Geomatics and Information Science of Wuhan University, 39(6), pp. 645-654.

Laurini, R., Servigne, S., Noel, G., 2005. Soft real-time GIS for disaster monitoring. Geo-information for Disaster Management, pp. 465-479.

Stonebraker, M., Çetintemel, U., Zdonik, S., 2005. The 8 requirements of real-time stream processing. ACM SIGMOD Record, 34(4), pp.42-47.

Artz, M., 2013. The New Age of Real-Time GIS. ESRI. http://blogs.esri.com/esri/esri-insider/2013/04/01/(1 May, 2013).

Lee, L., Romano, R., Stein, G., 2000. Introduction to the special section on video surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 745.

Tian, Y. L., Lu, M., Hampapur, A., 2005. Robust and efficient foreground analysis for real-time video surveillance. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. I, pp. 1182-1187).

Han, J., Haihong, E., Le, G., Du, J., 2011. Survey on NoSQL database. In: 6th international conference on Pervasive computing and applications, Port Elizabeth, South Africa, pp. 363-366.

Ulusoy, Ö., 1995a. Research issues in real-time database systems: survey paper. Information Sciences, 87(1), pp.123-151.

Ozsoyoglu, G., Snodgrass, R. T., 1995a. Temporal and real-time databases: A survey. IEEE Transactions on Knowledge and Data Engineering, 7(4), pp. 513-532.

Lu, H., Ng, Y. Y., Tian, Z., 2000. T-tree or b-tree: Main memory database index structure revisited. In: Proceedings of 11th Australasian Database Conference, Canberra, Australia, pp. 65-73.

Choi, K. R., Kim, K. C., 1996. T*-tree: a main memory database index structure for real time applications. In: Proceedings of Third International Workshop on Real-Time Computing Systems and Applications, Seoul, South Korea, pp. 81-88.

Garcia-Molina, H., Salem, K., 1992a. Main memory database systems: An overview. IEEE Transactions on Knowledge and Data Engineering, 4(6), pp.509-516.

Langran, G., 1989a. A review of temporal database research and its use in GIS applications. International Journal of Geographical Information System, 3(3), pp.215-232.

Yuan, M., 1996. Temporal GIS and spatio-temporal modeling. In: Proceedings of Third International Conference Workshop on Integrating GIS and Environment Modeling, Santa Fe, NM.

MODELLING URBAN CHANGES USING TIME SERIES OF VIRTUAL 3-D CITY MODELS

J. Kaňuk[a, *], M. Gallay[a], J. Hofierka[a]

[a] Institute of Geography, Faculty of Sciences, Pavol Jozef Šafárik University, Jesenná 5, 040 02 Košice, Slovakia - (jan.kanuk, jaroslav.hofierka, michal.gallay)@upjs.sk

Commission II, WG II/2

KEY WORDS: urban landscape changes, 3-D city model, 3-D GIS, time-series

ABSTRACT:

Assessing urban changes requires sophisticated methods and tools for a better understanding of processes which affect various components of the city structure. Recent developments in geospatial technology have provided new methods for 3-D data collection that can be effectively used in mapping and analysis of urban areas represented by virtual 3-D city models. These models, with various levels of detail, can replace the traditional 2-D representation of urban areas. Besides the improved visual representation of the landscape including the vertical dimension, the 3-D city models allow for enhanced assessment of the complex morphological changes occurring in a city and express the spatial relationships and processes affecting the landscape. In this paper, we propose a new, retrospective approach to the temporal analysis of urban changes using a time series of virtual 3-D city models derived from the commonly available data sources, such as orthoimagery and city plans. The approach is based on detection of individual changes in the city structure, such as buildings, trees and open areas. To demonstrate the applicability of the approach we have used the study area from the city of Prešov, Slovakia. This area has experienced a rapid transformation from agricultural and semi-natural landscape to the largest residential area in the city. We have developed a series of 3-D city models from the 1960s period up to the most recent state and analysed various aspects of the landscape changes in this part of the city.

1. INTRODUCTION

Contemporary cities represent the kind of environment in which the vertical dimension and volume of objects strongly influence human activity and natural phenomena. From the point of view of analysing spatial data in a geographic information system (GIS) the traditional 2-D modelling approach has limitations (Jones, 1989; Lange, 2001; Abdul-Rahman & Pilouk, 2007; Kolbe et al. 2007). The three-dimensionality (3-D) of urban space has been addressed by using virtual 3-D city models. Generating a digital database for 3-D landscape modelling with a high level of detail is a complex task. The financial and time costs to develop a detailed 3-D city model using these methods are very high, although alternative cost-effective approaches were also proposed (e.g., Rau & Cheng, 2013). A plethora of published research exists on applications of 3-D city models, e.g. in urban planning (Ranzinger & Gleixner, 1997), geology volume 3-D modelling (Qi et al., 2007), visibility and view with regard to the building height (Yu et al., 2007; Yasumoto et al., 2011), photovoltaic potential of buildings (Hofierka & Kaňuk, 2009), noise modelling (Law et al., 2011), route planning and optimal placement of facilities within buildings and between them (Thill et al., 2011), wind comfort in the streets (Janssen, Blocken, & van Hooff, 2013), real estate business (Rau & Cheng, 2013), a 3-D cadastre for taxation purposes and managing the ownership (Billen & Zlatanova, 2003; Guo et al. 2013), and many others.

The review of 3-D GIS technology and applications indicates that the progress in 3-D mapping and 3-D spatial analysis of urban environment has been focused on the present-day stage of cities. The virtual 3-D city models can also be applied in spatiotemporal analysis of processes and phenomena in urban landscape, which experiences more dynamic changes of the land morphology, use, and cover over time than in rural areas (Acevedo & Masuoka 1997; Krpo, 2012). Such a 4-D analysis can be achieved with a time series of 3-D models representing the past stages of the landscape assuming reliable data sources exist for 3-D reconstruction of the past stages of the city. This is more the case of the modern history of cities dating back several hundred years and is relatively well-documented in plans, maps, photographs, or technical documentation. If a systematic 3-D data collection with laser scanning or photogrammetry is

[*] Corresponding author. jan.kanuk@upjs.sk

performed over a certain period of time, it is possible to develop a time series of 3-D data representing various stages of the urban landscape. However, the problem arises if the modern data collection methods had not been available at earlier stages of the city, perhaps with the exception of aerial photography.

Benefits of the virtual 3-D city models even for older time periods can be exploited by adopting a novel methodology, which is presented in this paper as a retrospective approach. We developed a methodological framework for constructing time series of virtual 3-D city models as a new tool for spatiotemporal analysis of urban landscape. The applicability of the presented approach is also demonstrated in a case study using a 3-D solar irradiance model, and assessment of morphological and functional changes of an urban area in Slovakia.

2. STUDY SITE

The methodological framework of generating a time series of virtual 3-D city models is presented as a case study of the Sekčov area, which is an integral part of the city of Prešov in Eastern Slovakia. Prešov is the third largest city in the country and has over 91,300 inhabitants and the average population density of almost 1,300 people per square kilometre, according to the Statistical Office of the Slovak Republic (December 31st , 2012). The historical development of the city is thoroughly discussed in Kohlmayer (1980) and Matlovič (1998). The most striking development was determined by political and economic circumstances after the World War II when the former Czechoslovakia was re-established and further influenced by the Soviet Union under the socialist political regime until 1989. It was similar to many other post-communist cities affected by the socialist planning (e.g. Kotus, 2006). There had been a fivefold increase of urban fabric since 1945 up to the end of the 20th century (Matlovič, 1998). The industrialisation naturally induced development of the third sector activities and stimulated increase in housing construction, mainly blocks of flats.

The study area delineated comprises the Sekčov urban settlement which is now the largest residential area within Prešov providing home for more than 27,000 people (2012). Altogether, almost 30,000 people live in the delineated square of 2 by 2 kilometres. The area extends over the flat alluvial plain surrounded by fluvial terraces of the Torysa and Sekčov rivers. The terrain altitude ranges from 240 to 280 metres above the mean sea level. Originally swampy meadows on the alluvial plain of the river Sekčov were selected for their close vicinity to the city centre. The landscape was completely transformed into a highly populated and mainly residential area.

3. METHODS AND DATA

The period of the marked landscape changes since the second half of the 20th century is well-recorded by geospatial data sources. Hence, it was possible to reconstruct particular stages of the Sekčov area as a series (6 time horizons, Fig. 1) of virtual 3-D city models of the present and its earlier stages. In order to generate the models the retrospective approach was applied. The key step in this methodology was to reconstruct the model of the present-day stage of landscape for which the most accurate data sets were used (subcentimetre accuracy). The models of the past stages were derived from this model by modifying or omitting particular features according to various types of auxiliary data. The virtual 3-D city model of each stage of the landscape consists of three main data set layers: 3-D model of buildings, land cover model, and digital elevation model.

3.1. 3-D model of buildings

The initial step in generating the time series of the virtual 3-D models of buildings was the construction of the present-day stage (valid for 2010). The locations and floor plans of the buildings were generated from data obtained by electronic tachymetry supplied as the 2-D technical geodatabase by the Municipal Office of the Prešov City. The data were the most accurately measured type of data set used in the study (subcentimetre accuracy). The vertical dimension of the buildings and related facilities (e.g., chimneys, towers) was measured with a laser distance meter (submetre accuracy). The main objective was to capture the third level of detail (LoD3) (Flick, 1996). In this way, the virtual 3-D city model of the Sekčov area was generated based on the 3-D multipatch shapefile format (ESRI, 2008) and it comprised almost 1,000 3-D objects. For the solar irradiance modelling, however, the building layer had to be converted to polygonZ features. The study area was mainly agricultural and contained only few buildings before 1950. No major reconstruction took place here as evidenced by the historic cadastre maps and land use planning documentation. For that reason, it was possible to reconstruct the earlier stages from the current stage of the city. The 3-D building models for particular time period were omitted from the virtual 3-D city model of the current stage depending on their existence according to the historical records. Each 3-D building model was assigned attributes such as function, height, date of construction. The exact date of building construction was ascertained from the date of building approval recorded in the archive data of the Municipal Office of Prešov, Kohlmayer (1980) and Matlovič (1998). Modifications, destruction or changes of general shape of the buildings were not identified in the historical records for the area.

3.2. Land cover model

The land cover model time series was generated using the 2-D technical geodatabase of the city of Prešov, and by manual interpretation and vectorization of orthoimagery and historical maps. The following land cover categories were considered in the models: permanent grass land, arable soil, forests, gardens, roads, railways, water streams, continuous urban fabric. The classes were represented as vector polygons. The present-day land cover was reconstructed with regard to the highest possible realism using a high resolution orthoimagery and the technical geodatabase. The main goal was to express the landscape character before accurately locating the land cover features. Together with buildings, trees make a significant 3-D appearance of the landscape at the level of scale taken. Hence, the trees were handled separately as points represented by 3-D symbols with defined tree type and height. By this means, individual trees mainly within the urban fabric were exactly located while, for continuous coverage, the trees were randomly located to represent the real appearance of the landscape as close as possible. Earlier stages of the land cover were reconstructed using historical maps, ground level photographs (from 1950 - present), and interviewed local inhabitants. Due to generalisation and map scale issues, many features were represented as symbols (e.g., roads) in the historical maps while their spatial extent was reproducible only from the most current geodatabase and orthoimagery. In such cases, the existence of the feature was important and the spatial extent was estimated. In other cases, some features were not displayed due to the same cartographic reasons and we assumed their presence in the landscape as, for example, shrubs along the river or trees in the urban areas (parks, gardens, etc.).

3.3. Digital elevation model (DEM)

The current terrain surface of the area was represented by a gridded DEM derived from the ground surveyed point heights and contour lines (40,580 points). The present-day DEM was modified according to the records found in older technical maps. In case when symbols were used to mark terrain features such as road cuts or embankments, the original terrain was estimated and intentionally modified. The DEM was interpolated into a 1 meter grid using the v.surf.rst module (Neteler & Mitasova 2008).

4. RESULTS AND APPLICATIONS

4.1 Modelling morphological and functional changes

The generated time series of virtual 3-D urban landscape for the Sekčov area is the main output of the presented research. Transformation of the landscape can be analysed by defining the time period by the user. In order to demonstrate the applicability of the proposed methodology and the 3-D time series, 6 time horizons representing distinctive periods of development were selected. Each time horizon was derived from the previous stage of the model using a set of available data as described above. The resulting time series of the 3-D city models shows the transformation of the study site from agricultural and semi-natural landscape to a major residential area in the city. By this means, a different perspective of the urban changes is communicated to non-specialists or the general public who may find reading 2-D maps more difficult.

The time series in Figure 1 visualizes the change of the landscape form 1950s. This process started in 1969 and the construction of the first blocks of flats began in 1978. Besides the massive blocks, the construction involved public facilities for education, health care, and entertainment, small shops, hypermarkets, restaurants, etc. This process has continued until now, but the original plan has not been fully achieved because of political and economic changes after 1989. The original terrain parameters required specific preparation works. The river Sekčov was relocated into an artificial, 5 km-long channel. Its bottom is about 2.5 meters below the original terrain. The original channel was filled up. The swampy areas were meliorated and intermittent water streams were diverted into the main stream of Sekčov. These changes also affected the original land cover and other landscape features, such as hydrologic regimes, soil and microclimate. The development of the residential area had 3 major stages. It began in the southern part of the area in 1978 and ended in 1989. The next stage (1989-1999) included blocks of flats with a complete technical infrastructure such as gas, power and water supply, sewerage, roads. Since 2000 the area has been filled up with various commerce objects such as shops, services, and so on. The recent development is concentrated in new, previously empty zones. We can also find very important functional changes inside several buildings (Fig. 2). Many residential buildings used to be state-owned before 1989, later former renters were allowed to buy flats and become owners. Moreover, some of these blocks of flats have lost their purely residential function and have become multifunctional buildings.

Solar irradiance

Recent developments in 3-D GIS technology provided new powerful tools to assess spatial distribution of solar radiation and also to select properly the location of thermal collectors or photovoltaic panels commonly used as a source of renewable energy in urban areas. Hofierka & Zlocha (2012) developed the v.sun module of GRASS GIS for such analysis over 3-D urban surfaces. To demonstrate the applicability of the retrospective approach in 3-D city modelling, we have applied this module to the study area. Direct solar irradiance received by building surfaces was calculated for 15 December, 1 pm. Figure 3 demonstrates the solar energy conditions in 1999 (Fig. 3a, 4b) and their change after constructing new blocks by 2010. As can be seen from comparing, the lower parts of the older blocks are the most affected where the newly built blocks cast their shadow. Even though the figure shows solar irradiance for a particular moment of the annual solar energy distribution, the results indicate that the residential area containing large blocks of flats has a very different irradiance situation affecting the energy balance of walls and availability of direct solar radiation throughout the year.

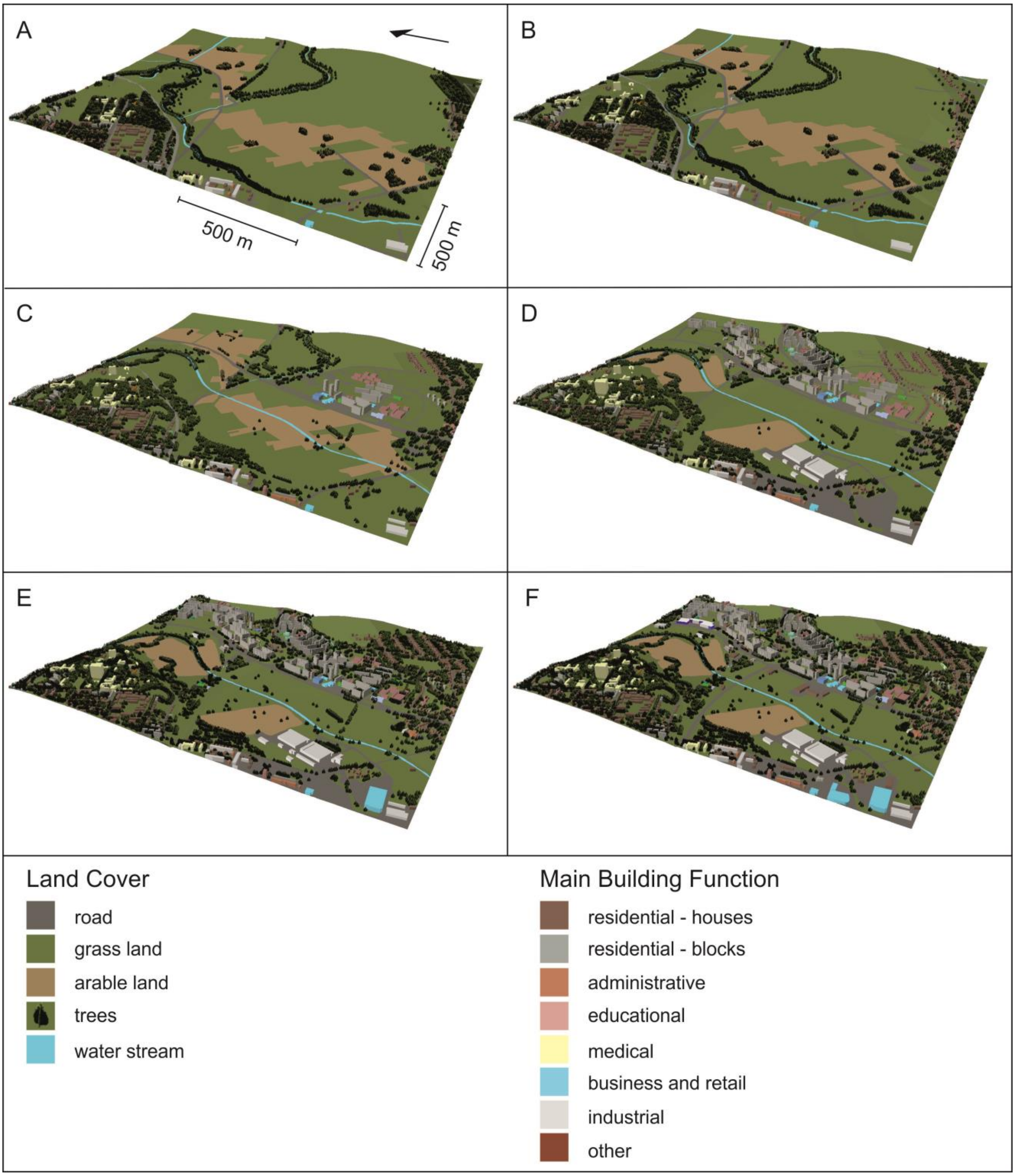


Fig. 1. Time series of virtual 3-D city models representing the Sekčov area in (a) 1959, (b) 1969, (b) 1979, (d) 1989, (e) 1999, (f) 2010.

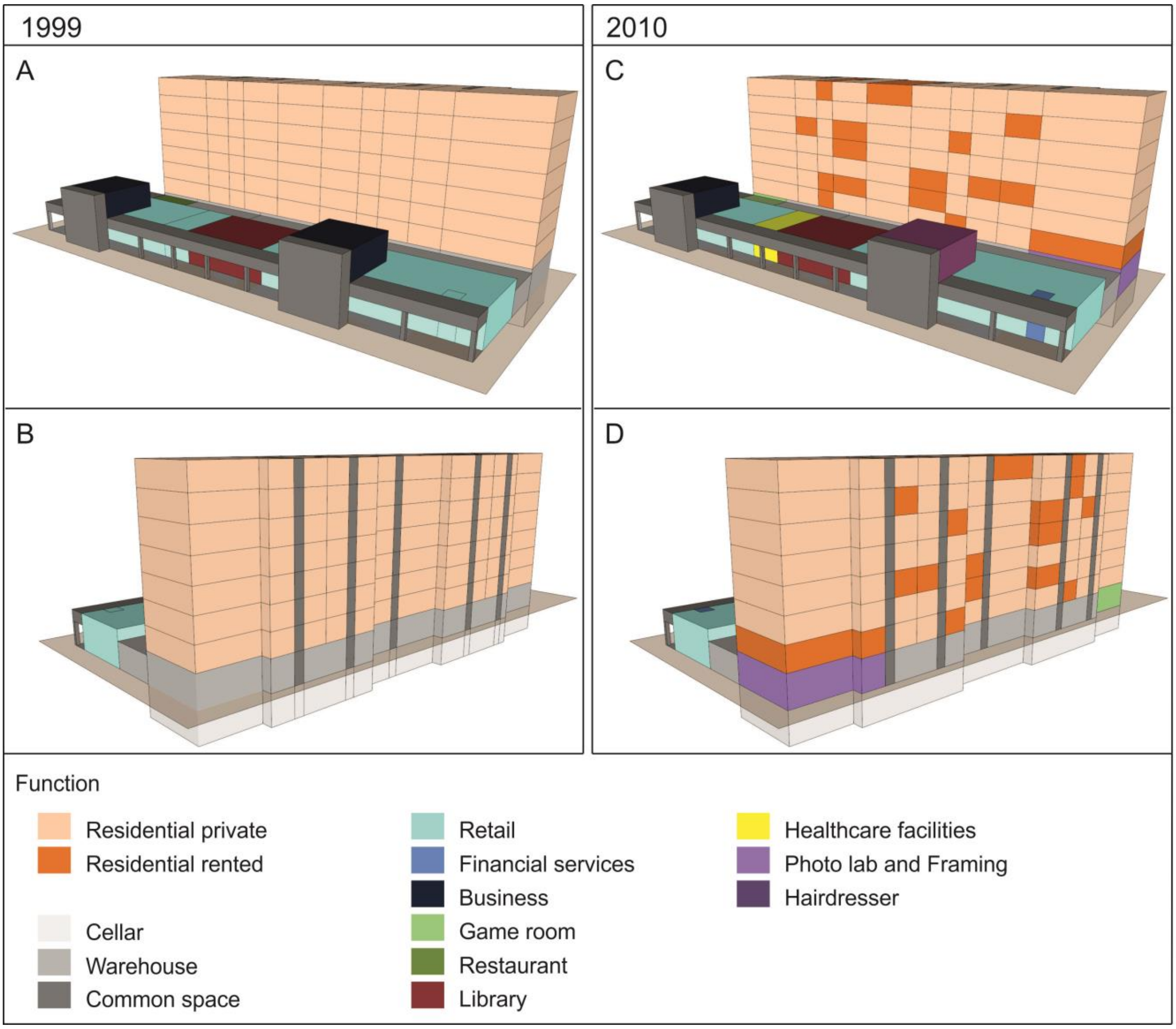


Fig. 2. Functional changes within a single block of flats of the Sekčov area demonstrated as 3-D data structure for 1999 (a, b) and for 2010 (c, d) viewed from the front and back.

5. CONCLUSION

In this study, we demonstrated a new, retrospective approach to the temporal analysis of urban changes using a time series of virtual 3-D city models derived from commonly available data sources, such as orthoimagery and city plans. The method is based on backtracking the changes in the city components from the most recent state mapped with high accuracy using modern mapping methods (digital photogrammetry, ground surveying). Using various auxiliary data this present-day city model was modified to reflect the previous stages of the city. The method provides higher positional accuracy with clear benefits in lower production costs and more accurate results in land changes detection and the temporal analysis of urban changes. The use of 3-D city models in a 3-D GIS environment also provides new possibilities in analysing spatial relationships and processes affecting the dynamics of a city. We also explored potential of the time series in modelling distribution of solar radiation based on 3-D GIS approach. The applied methodology to construct the virtual 3-D city time series is most suitable for modelling the history of urban areas, e.g. for the purposes of urban planning, as a communication tool for demonstrating the impact of past decisions on the following stage of urban landscape. The retrospective approach is applicable especially in areas where the buildings were progressively build on a green field or the initial urban pattern was densified. This is the case of many towns and cities in the socialist urban planning. If older structures had been demolished and replaced by newer construction, relevant archive technical documentation or photogrammetric imagery could have been used. This fact gives a motivation for future work. The realm of 3-D modelling of urban landscape and its change in time can focus on development of GIS tools enabling assessment of the 4-D change similar to its 2-D analogy in map algebra.

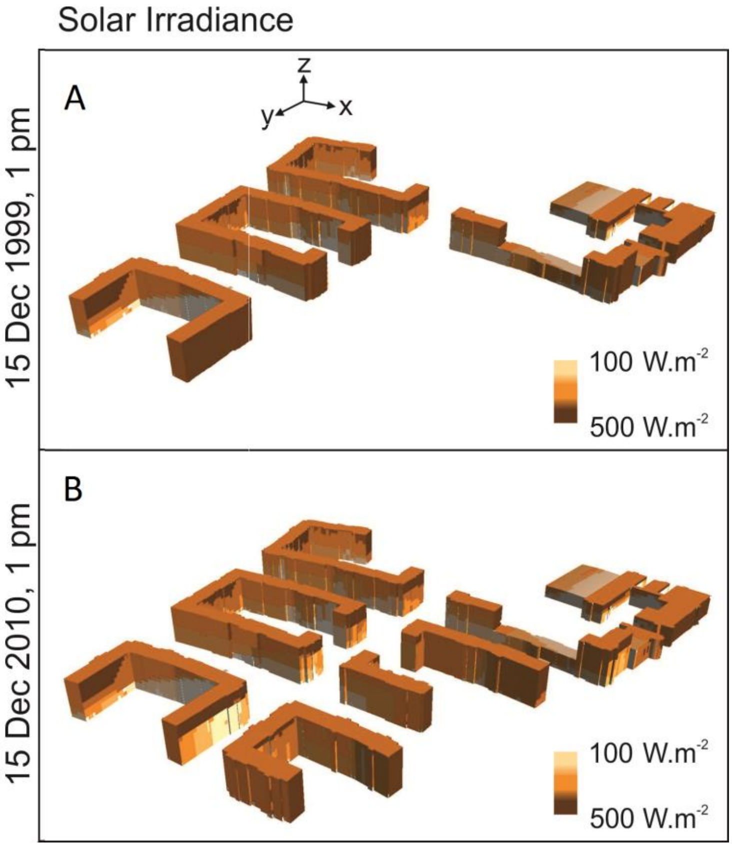


Fig. 3. Direct solar irradiance of 3-D buildings calculated by the v.sun module (Hofierka & Zlocha, 2012) for the north-eastern part of the Sekčov area at 1 pm, 15 December 1999 (a), and 15 December 2010 (b). The scale is given by the axes arrows of 20 m length as showed in (a).

6. REFERENCES

Acevedo, W., & Masuoka, P. (1997). Time-series animation techniques for visualizing urban growth. *Computers & Geosciences*, 23(4), 423-435.

Abdul-Rahman, A., & Pilouk, M. (2007). *Spatial modelling for 3-D GIS*. Heidelberg, Germany: Springer-Verlag.

Billen, R., & Zlatanova, S. (2003). 3-D spatial relationships model: A useful concept for 3-D cadastre? *Computers, Environment and Urban Systems*, 27, 411–425.

ESRI (2008). The Multipatch Geometry Type: an ESRI white paper. http://www.esri.com/library/whitepapers/pdfs/multipatch-geometry-type.pdf (30 June 2014)

Flick, S. (1996). An object-oriented framework for the realization of 3-D geographic information systems. In Rumor, M., McMillan, R., & Ottens, H. F. (Eds.) *Proceed. of the Second Joint Europ. Conf. and Exhib. on Geog. Inform. (Vol. 1): From Research To Application Through Cooperation* (Barcelona, Spain), pp. 187-196. Amsterdam, The Netherlands: IOS Press.

Guo, R., Lin, L., Shen, Y., Ping, L., Biao, H., & Renrong, J. (2013). Developing a 3D cadastre for the administration of urban land use: A case study of Shenzhen, China. *Computers, Environment and Urban Systems*.

Hofierka, J., Kaňuk, J. (2009): Assessment of Photovoltaic Potential in Urban Areas Using Open-Source Solar Radiation Tools. *Renewable Energy*, 34(10), p. 2206-2214.

Hofierka, J., Zlocha, M. (2012): A New Solar Radiation Model for 3-D City Models. *Transactions in GIS*, 16(5), p. 681-690.

Janssen, W.D., Blocken, B., & van Hooff, T. (2013). Pedestrian wind comfort around buildings: comparison of wind comfort criteria based on whole-flow field data for a complex case study. *Building and Environment*, 59(1), 547-562.

Jones, C. B. (1989). Data structures for three-dimensional spatial information systems in geology, *International Journal of Geographical Information Systems*, 3(1), 15-31.

Kohlmayer, V. (1980). *Vývojové tendencie urbanistickej štruktúry Prešova* (Development tendency of the Prešov City urban structure). (Unpublished doctoral dissertation). Faculty of Architecture, Slovak Technical University, Bratislava, Slovakia.

Kolbe, T.H., Gröger, G., & Plümer, L. (2005). CityGML – Interoperable access to 3-D city models. In Oosterom, Zlatanova, Fendel (Eds.), *Proceed. of the Int. Sympos. on Geo-inf. for Disaster Management*, 21. – 23. March in Delft, Springer Verlag, http://www.ikg.uni-bonn.de/fileadmin/sig3-D/pdf/Gi4Dm_2005_Kolbe_Groeger.pdf. (30 June 2014)

Kotus, J. (2006). Changes in the spatial structure of a large Polish city – The case of Poznań, *Cities*, 23(5), 364-381.

Krpo, A. (2012). 3D video models of possible growth across Auckland. http://shapeauckland.co.nz/3d-video-models-of-possible-growth-across-auckland/ (30 June 2014)

Lange, E. (2001). The limits of realism: perceptions of virtual landscapes. *Landscape and Urban Planning*, 54(1–4), 163-182.

Longley, P., Goodchild, M.F., Maguire, D., & Rhind, D. (2005). *Geographic information systems and science*. Wiley: UK.

Matlovič, R. (1998). Geografia priestorovej štruktúry mesta Prešov (Geography of spatial structure of the city of Prešov). *Geografické práce*, 8(1), 261 p.

Neteler, M., & Mitasova, H. (2008). *Open Source GIS: A GRASS GIS Approach.* Third Ed. The International Series in Engineering and Computer Science: Volume 773, Springer, New York.

Qi, M., Zhang, B.-L., Liang, G.-H., Wang, J., & Cai, X.-P., (2007). 3-D modeling and visualization of geology volume based on geophysical field data. *Data Science Journal*, 6, 652–657.

Rau, J-Y., & Cheng, C-K. (2013). A cost-effective strategy for multi-scale photo-realistic building modeling and web-based 3-D GIS applications in real estate. *Computers, Environment and Urban Systems*, 38(3), 35-44.

Ranzinger, M., & Gleixner, G. (1997). GIS datasets for 3-D urban planning. *Computers, Environment and Urban Systems*, 21(2), 159-173.

Thill, J.-C., Dao, T.H.D. & Zhou, Y. (2011). Traveling in the three-dimensional city: applications in route planning, accessibility assessment, location analysis and beyond. *Journal of Transport Geography*, 19(3), 405-421.

Yasumoto, S., Jones, A.P., Nakaya, T., & Yano, K. (2011). The use of a virtual city model for assessing equity in access to views. *Computers, Environment and Urban Systems*, 35(6), 464-473.

Yu, S., Han, S., & Chai, C. (2007). Modeling the Value of View in High-Rise Apartments: A 3-D GIS Approach. *Environment and Planning B: Planning and Design*, 34(1), 139–153.

7 . ACKNOWLEDGEMENTS

This work was supported by the Slovak Research and Development Agency under the Grant No. APVV No. 0176-12 and Ministry of Education of the Slovak Republic under the Grants VEGA No. 1/0272/12 and 1/0473/14.

A PERSPECTIVE VIEW-BASED APPROACH TO DETERMINE THE GEOSENSOR COVERAGE IN 3D VECTOR ENVIRONMENTS

A. Afghantoloee[a], S. Doodman[a], F. Karimipour[a, *], M.A. Mostafavi[b]

[a] Department of Surveying and Geomatic Engineering, College of Engineering, University of Tehran, Tehran, Iran
(a.afghantoloee, s_doodman, fkarimipr)@ut.ac.ir
[b] Center for Research in Geomatics, Department of Geomatics, Laval University, Quebec, Canada
mir-abolfazl.mostafavi@scg.ulaval.ca

Commission II, WG II/2

KEY WORDS: Wireless Sensor Networks, Coverage Determination, Sensor Deployment, 3D Vector Models, CityGML

Geosensors are tiny and ingenious devices that collect data about their nearby area, and are capable of communicating with each other. They are usually deployed in a wireless network to monitor and collect physical and environmental information such as motion, temperature, humidity, pollutants, and traffic flow in a given area. The information is, then, communicated to a processing center, where they are integrated and analysed for different applications (Ghosh and Das, 2006).

A geosensor covers only a certain region, which depends on the sensing and communicating range (limited by signal amplitude) as well as the environment conditions such as visibility (limited by obstacles). The total area covered by a Wireless geoSensor Network (WSN) is obtained from the union of the regions covered by individual sensors. Therefore, efficient deployment of geosensors in a WSN is an important issue that affects the coverage as well as communication between sensors (Argany *et al.*, 2011). Several optimization methods (i.e., global or local, deterministic or stochastic, etc.) have been proposed to detect and eliminate coverage holes and hence increase the coverage of geosensor networks. Some methods use general optimization techniques, while some others consider the problem as a geometric issue and use tools from computational geometry (Karimipour *et al.*, 2013). The key point of all deployment optimization algorithms, is determining the coverage of an individual geosensor.

The geosensor coverage estimation methods can be classified into: (i) the methods that consider a raster environment (Akbarzadeh *et al.*, 2013; Argany *et al.*, 2012; Cortés *et al.*, 2004), which are limited by the spatial resolution; and (ii) the methods that model the environment as a vector dataset (Ghosh and Das, 2006; Guvensan and Yavuz, 2011; Ma *et al.*, 2009; Wang and Cao, 2006, 2011), which have been mostly proposed for 2D spaces and do not consider the earth topography and human-made obstacles. On the other hand, sensing model (i.e., binary or probabilistic, omnidirectional or directional, etc.) affect significantly the coverage estimation..

In this paper, we propose an approach to determine the coverage of a geosensor with directional sensing model in a 3D vector environment. To obtain the exact coverage of a geosensor, a flat plane, called *perspective plane*, perpendicular to the direction of the sensor with a certain pre-defined distance to the sensor is considered. The polygons are projected on the perspective plane according to the perspective geometry. The sensing region of the sensor is, then, projected as a circle on the perspective plane, called *perspective circle* (Figure 1).

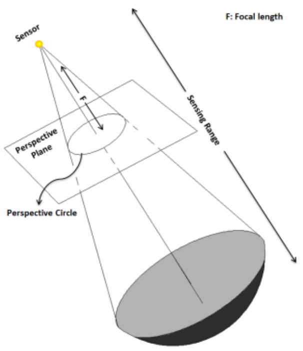


Figure 1. The perspective plane and perspective circle

* Corresponding author

As illustrated in Figure 2, the polygons projected on the 2D perspective plane are classified into:

- The polygons that fall entirely within the perspective circle and thus are located in the sensor's field of view (e.g., polygons **A** and **B**);
- The polygons that are totally located outside the perspective circle, i.e., are out of the sensor's field of view.
- The polygons that intersect with the perspective circle, i.e., partially fall within the sensor's field of view (e.g., polygon **C**). This fraction is extracted through computing their intersection with the perspective circle.

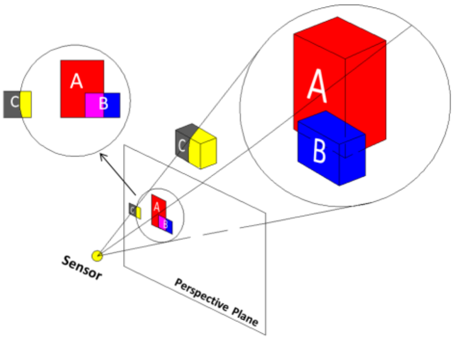


Figure 2. Classifying the projected polygons on the perspective plane respect to their position in the perspective circle

Having the above information extracted for each polygon, it must be checked against other polygons to specify the portion that is visible by the sensor, i.e., is not covered by other polygons (see polygon **A** in Figure 2). The pseudo-code of the proposed approach is as follows:

```
Input:  A set of polygons $\boldsymbol{P}$={$\boldsymbol{P}_1$, …, $\boldsymbol{P}_n$} that model a 3D environment
        A geosensor node $\boldsymbol{S}$ and its position, direction, sensing range and field of view
Output: The region sensed by the sensor $\boldsymbol{S}$
Sort the polygons {$\boldsymbol{P}_1$, …, $\boldsymbol{P}_n$} descending based on their distance to the sensor
Create an empty list $\boldsymbol{L}$
For each sorted $\boldsymbol{P}_i$
    If $\boldsymbol{P}_i$ is located in front of $\boldsymbol{S}$ and within the sensing range of it
        $\boldsymbol{PP} \leftarrow$ Projection of $\boldsymbol{P}_i$ on the perspective plane of $\boldsymbol{S}$
        $\boldsymbol{V} \leftarrow$ The portion of $\boldsymbol{PP}$ that fall within the perspective circle of S
        For each elements $\boldsymbol{l}_i$ in $\boldsymbol{L}$
            $\boldsymbol{l}_i = \boldsymbol{l}_i - \boldsymbol{V}$
        Insert $\boldsymbol{V}$ into $\boldsymbol{L}$
Transfer $\boldsymbol{L}$ to 3D space
```

The proposed approach was applied on the *peaks* function as a simulated 3D space, and a 4-meter-high geosensor with 90° of field of view in both vertical and horizontal directions was considered (Figure 3). Figure 4 illustrates the perspective view and the area covered by the sensor.

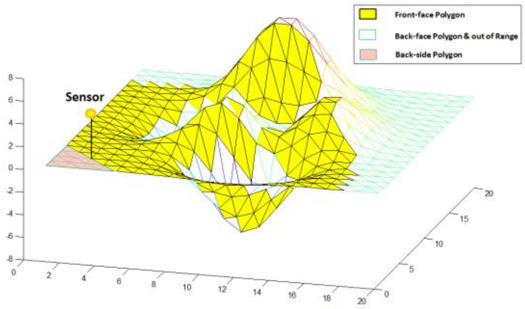
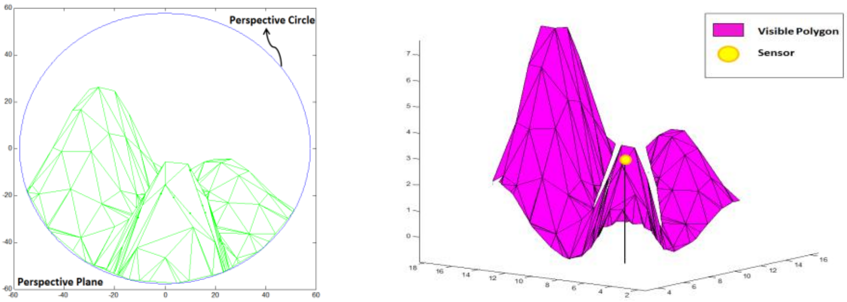


Figure 3. The sample CityGML dataset



(a) (b)

Figure 4. Computation of the coverage of the geosensor considered in Figure 3: (a) The perspective view; (b) The area covered by the geosensor

A CityGML dataset was also used as another case study (Figure 5), where a geosensor with direction of [1, 1, -1] and 90° of field of view is considered. In CityGML, a 3D vector model consists of objects, which *per se* are composed of a series of polygons, which are related to either topography of TIN model or components of objects in a 3D space such as buildings. Figure 6 illustrates the steps to determine the area covered by the sensor.

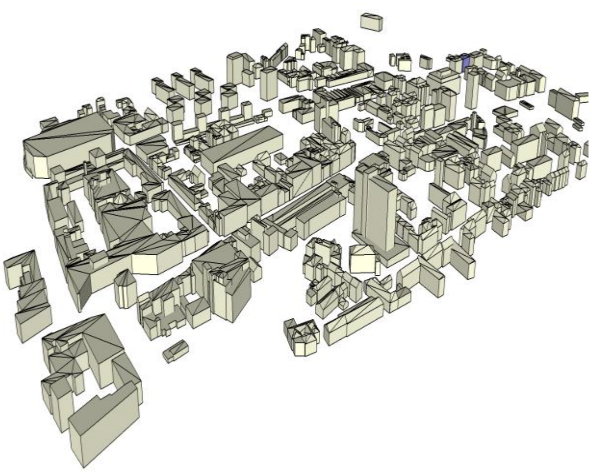
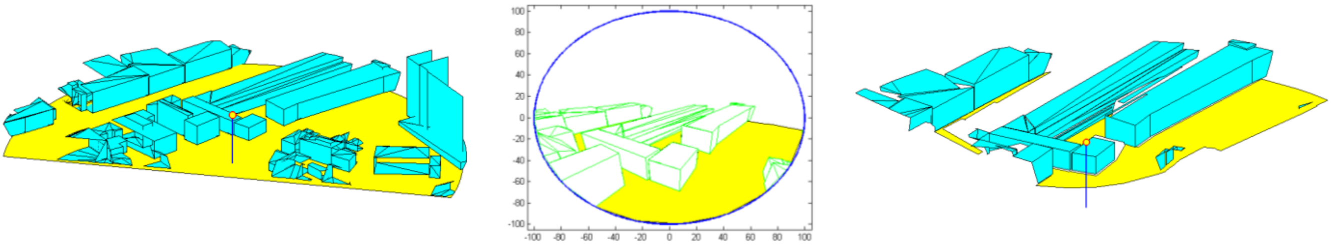


Figure 5. The sample CityGML dataset



(a) (b) (c)

Figure 6. Determining the area covered by the geosensor illustrated in Figure 5: (a) The polygons located within the sensing range of the sensor and their normal vector is toward the sensor; (b) The perspective view of the polygons visible by the sensor; (c) 3D presentation of the polygons visible by the sensor

For both case studies, the results of coverage estimation using 2D/3D and raster/vector models of the environment are computed and compared, in terms of speed and accuracy.

References

Akbarzadeh, V., Gagne, C., Parizeau, M., Argany, M., Mostafavi, M.A., 2013. Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. IEEE Transactions on Instrumentation and Measurement 62, 293-303.

Argany, M., Mostafavi, M.A., Akbarzadeh, V., Gagné, C., Yaagoubi, R., 2012. Impact of the quality of spatial 3d city models on sensor networks placement optimization. GEOMATICA 66, 291—305.

Argany, M., Mostafavi, M.A., Karimipour, F., Gagné, C., 2011. A gis based wireless sensor network coverage estimation and optimization: A voronoi approach. A Voronoi Approach. Transacton on Computational Sciences Journal 14, 151-172.

Cortés, J., Martínez, S., Karatas, T., Bullo, F., 2004. Coverage control for mobile sensing networks. IEEE Transaction on Robotics and Automation 20, 243-255.

Ghosh, A., Das, S.k., 2006. Coverage and connectivity issues in wireless sensor networks, Wireless, and sensor networks: Technology, applications, and future directions.

Guvensan, M.A., Yavuz, A.G., 2011. On coverage issues in directional sensor networks: A survey. Ad Hoc Networks 9, 1238-1255.

Karimipour, F., Argany, M., M.A. Mostafavi, 2013. Spatial coverage estimation and optimization in geosensor networks deployment, Wireless sensor networks: From theory to applications, pp. 59-83.

Ma, H., Zhang, X., Ming, A., 2009. A coverage-enhancing method for 3d directional sensor networks. IEEE Communications Society, 2791-2795.

Wang, Y., Cao, G., 2006. Movement-assisted sensor deployment, IEEE Infocom (INFOCOM‘04), pp. 640-652.

Wang, Y., Cao, G., 2011. On full-view coverage in camera sensor networks, Proceedings of INFOCOM 2011, IEEE, pp. 1781-1789.

Fixating on Adequate Landmarks

P. Viaene[1], K. Ooms[1], P. Vansteenkiste[2], M. Lenoir[2], P. De Maeyer[1]

[1]Ghent University, Geography Department; Krijgslaan 281 (S8), 9000 Ghent, Belgium
[2]Ghent University, Department of Movement and Sport Sciences; Watersportlaan 2, 9000 Ghent, Belgium

{Pepijn.Viaene, Kristien.Ooms, Pieter.Vansteenkiste, Matthieu.Lenoir, Philippe.DeMaeyer}@UGent.be

Abstract

Landmarks play an important role in facilitating indoor navigation. This role could be enhanced further by incorporating them in wayfinding aids such as maps and route instructions. This requires, however, an adequate and workable criterion that can be used for the identification of these objects or structures. Based on the results of a study in which participants navigated through a building while thinking aloud and wearing a mobile eye tracker, it can be concluded that uniqueness is a valid criterion and, furthermore, an important measure that defines the quality of a landmark. Consequently, unique landmarks were most used to base wayfinding decisions on, and decision points lacking in unique objects caused great wayfinding difficulties. As a result, object landmarks were fixated on most since these objects varied much more in appearance than structural landmarks.

Keywords: Navigation, Building, Eye Tracking, Think Aloud, Spatial Awareness

1 Introduction

Navigation is the competence of planning and following a route to get from one's current location to a chosen destination. This leads to the understanding that spatial knowledge about the environment is essential to permit navigation, otherwise the first phase of navigating (i.e. planning a route) would be impracticable. Therefore, apart from wayfinding tools such as maps, the cognitive model of the environment in which one wishes to navigate is central in our ability to navigate as it represents all known spatial relations between objects and/or locations (Iaria and Barton 2010). Essential for the construction of this mental map is the selection and use of landmarks (Siegel and White 1975; Golledge and Montello 1998), which are prominent and identifiable elements in an environment (external to man) that enable an observer to locate himself and to set objectives like reaching a destination or selecting an optimal route (Sorrows and Hirtle 1999). As a result, various studies have highlighted the importance of landmarks in the context of wayfinding and navigation – outdoor as well as indoor –, because they are essential in structuring spatial knowledge (Siegel and White 1975; Golledge and Montello 1998).

In extreme situations, this close relationship between landmarks, mental maps and wayfinding performances is demonstrated by people who suffer from topographical disorientation, which is the inability to orient oneself and leads to immense navigational difficulties. This disorder is related to the capability of constructing a cognitive model of the environment and results in problems to identify and use landmarks (Iaria & Barton 2010). More common, however, is that a lack of clues (i.e. landmarks), due to poor signage and a complex architectural design, results in an environment that negatively influences wayfinding performances (Raubal & Egenhofer 1998).

The complex architectural design of indoor environments (i.e. buildings) often cause an even higher risk on losing orientation than outdoors (Brunner-Friedrich and Radoczky 2006), which stresses the need of landmarks as they can help significantly, as Gärling et al. (1983) stated, “to achieve the important practical goal of facilitating orientation”. In addition, buildings are in most cases characterised by the third dimension (i.e. floor levels). This vertical dimension renders navigation more difficult and has an impact on the wayfinding performances because the directional information is more complex (Soeda et al. 1997; Jeffery et al. 2013).

Hence, indoor navigation can benefit from implementing landmarks in various ways. For example, the representation of landmarks on maps may increase the readability and clearness as landmarks act as point of correspondence between different forms of spatial knowledge (e.g. reality, wayfinding tools (such as maps) and the cognitive model of the environment) (Presson and Montello 1988). Furthermore, landmarks are beneficial for the formulation of route instructions as they serve as point of reference (Lovelace et al. 1999). For example, route instructions containing landmarks as descriptive features are rated as highly effective (Hund and Padgitt 2010).

However, the integration of landmarks in wayfinding aids such as route instructions and maps is hampered because there is much uncertainty about which salient objects can serve as adequate landmarks and how these objects can be detected (Sefelin et al. 2005). One of the main difficulties of identifying landmarks lies in the fact that landmarks are traditionally defined by their function (as put forward above) and not by their appearance. Although some have tried to quantify the saliency based on features that contribute to the visual, semantic and structural saliency in line with the landmark theory of Sorrows and Hirtle (1999) (e.g. Raubal and Winter (2002) and Fang et al. (2011)), this remains a problematic affair. Moreover, the weights that have to be attributed to these features are difficult to determine as they are highly dependent on the perception and individual preferences of the observer, which are influenced by gender, age, social and cultural background, experience, familiarity with the environment and intentions (Raubal 2001). This was also put forward by Peponis et al. (1990) who stated that, apart from the theoretical point of view of cognition, the vague concept of a landmark, which merges various general and idiosyncratic features, interferes with the understanding and use for practical applications of landmarks. Therefore, this study proposes to evaluate objects on another criterion, namely that of being unique as it is more easily to determine and, moreover, is in line with the characteristics of landmarks put forward by research on (spatial) cognition.

Apart from the perceptual saliency of an object, Stankiewicz and Kalia (2007) proposed two other properties of a landmark, namely being informative and being persistent. The notion of being informative was defined by Stankiewicz and Kalia (2007) as the fact that less common objects provide more information about the location of an observer than common objects. This is similar to the feature of being unique. Uniqueness of an object (with respect to the whole route) was also used by Elias (2003) as a reliability criterion for the selection of landmarks, because false landmarks (i.e. identical of very similar objects) can mislead the navigator as landmarks are associated with specific wayfinding actions. However, Elias (2003) only focussed on the automatic derivation of landmarks from existing databases based on this criterion and did not verify if the selected objects would also have been used in natural wayfinding behaviour. Furthermore, the study was situated in an outdoor environment and while outdoor landmarks can be very diverse (e.g. a shop, a church, an apartment building, a monument), there is less variety in terms of type and number of distinctive and salient landmarks indoors (Mast et al. 2012). Hence, the objective of this study is to determine which landmarks are considered most salient while navigating in a building and if they can be characterised by the concept of uniqueness.

2 Study Design

In an exploratory study, twelve participants, who were all in their twenties, completed a route in a complex building twice while wearing a head mounted eye tracking device (iViewX HED (SMI)). The participants knew that the study dealt with indoor navigation and were informed in advance about the course of the experiment. Furthermore, they received a movie ticket as incentive to participate. As they were not familiar with the route, the participants followed the experimenter during the first completion. The second time, they traversed the route independently and the experimenter only intervened if they were lost. After the experiment, the participants were asked to complete a questionnaire in which their experience with the building was assessed. Due to technical problems with the eye tracker, the recordings of three participants were excluded from the analysis.

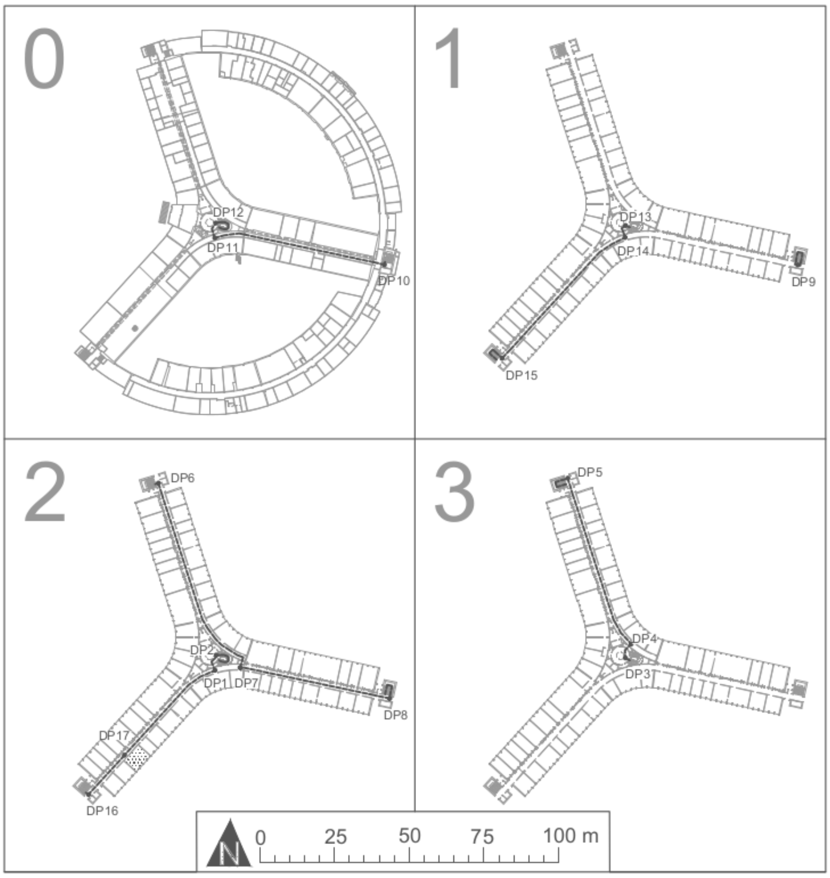


Fig. 1. Illustration of the route (dashed line) across the different floor levels with indication of the decision points (DP).

The route itself had a total length of 440 meters and covered four floor levels (see Figure 1). Decision point 17 is the start and end point of the route. The building was built around 1976 and has a very traditional design. Finally, no additional objects were placed along the route as part of this study. The objects present along the route were mostly related to the scientific departments housed in the building (e.g. posters, geologic samples). In addition, signs, fire protection equipment, sculptures and cabinets are present. Figures 2 and 3 give an idea of the design of the building.

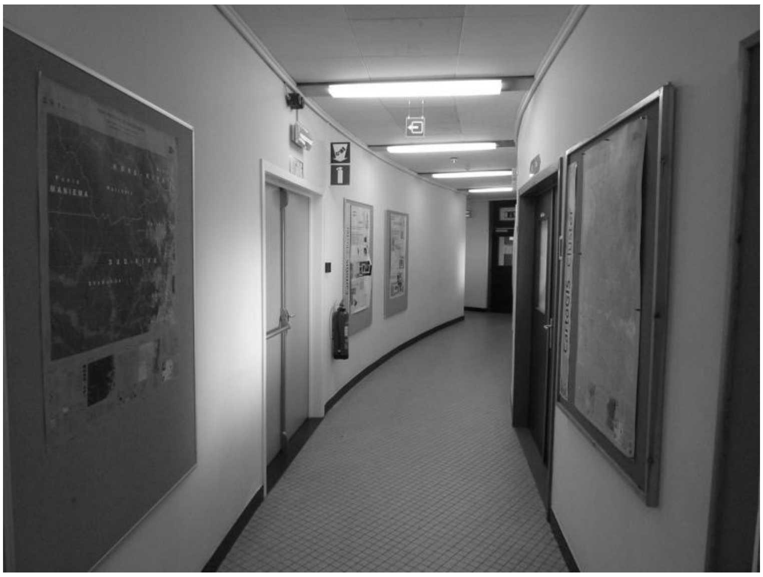


Fig. 2. Picture taken at decision point 1. Participants are expected to turn left trough the double door.

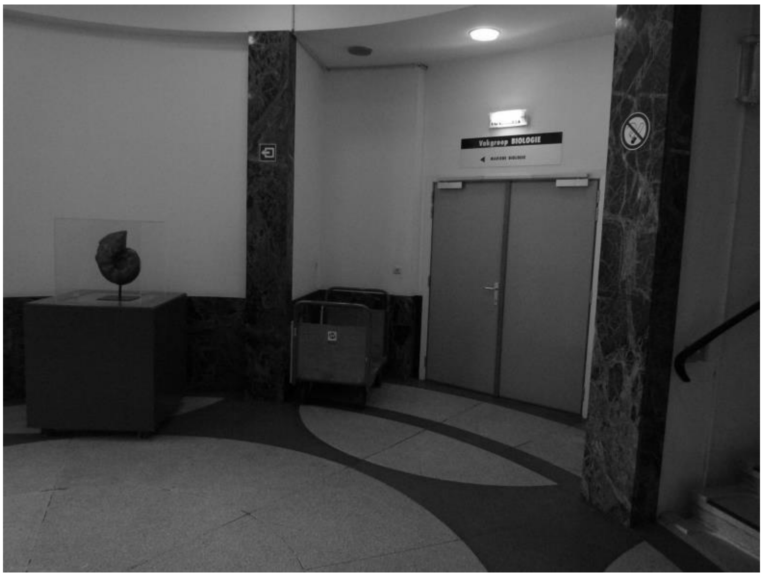


Fig. 3. Picture taken at decision point 12. Participants are expected to go up the stairs on their right-hand side.

As the first objective is to identify the landmarks on which participants relied for the successful completion of the navigational task, it is important that the participants relied on these landmarks to a large extent. First of all, the design of the experiment encouraged the use of landmarks, since an important function of landmarks is that they enable a navigator to retrace a route (Frankenstein et al. 2012). Furthermore, all participants were completely unfamiliar with the test environment. Consequently, they were supposed to mainly implement the simplest wayfinding strategy of guidance. This strategy implies that participants rely on beacon knowledge whereby a landmark is associated with the action of approaching or avoiding it (Wang et al. 2014). In addition, the cognitive model should have been largely based on the use of landmark knowledge as it was in the first stages of its development (Golledge & Montello 1998, Siegel & White 1975). Finally, they could only rely on the information provided by the environment as no additional wayfinding aids, such as maps, were given.

In order to identify the salient objects that were potentially used as landmarks, a method is needed that can give more insight in the cognitive processes of the observer. This, in the assumption that during a navigational task the cognitive processes are related to the cognitive model of the environment in which landmarks form an important part. In this study, we opted for eye tracking. The reason to track the eye movements in order to learn more about the selection of landmarks is based on the eye-mind hypothesis, which states that during a task certain aspects can be analysed to examine cognitive processes because eye fixations are closely related to the human ability to encode spatially distributed visual stimuli. These aspects include the locus of the eye fixation and its duration. The locus indicates the element that is being processed internally even if subjects are not consciously aware of this and the duration is related, but not necessarily identical, to the time needed to encode and to operate on that element (Just & Carpenter 1976). Furthermore, a landmark is eye-catching, as saliency is a main property of a landmark and represents the degree in which a landmark distinguishes itself in its environment (Caduff & Timpf 2008, Stankiewicz & Kalia 2007)

However, formulating conclusions about the cognitive processes solely based on eye movements is not without danger. The relation between the locus of the eye fixation and the selective attention is not that straightforward, because people can on the one hand extract information by peripheral vision and on the other hand focus on a point without picking up information. The validity of eye movements depends on the type of task undertaken and the significance of the peripheral vision during that task (van Gog et al. 2009, Williams & Davids 1997). In order to clarify specific fixations, another method, which is more commonly used to study cognitive processes related to (indoor) wayfinding (e.g. (Hölscher et al. 2006)), will also be applied, namely the think aloud method. Many authors encourage or see benefits in the combination and interaction of verbal protocols and eye tracking data (Elling et al. 2012, 2011, Gerjets et al. 2011, van Gog et al. 2009, Williams & Davids 1997). Therefore, the participants were asked to share their thoughts aloud during both completions of the route. To stimulate verbalising, a small exercise on thinking aloud (in which they were asked to tell something about themselves and to solve a fairly simple mathematical problem aloud) was conducted before the experiments began. Before the beginning of the first completion of the route, the participants were asked to apply the think aloud method during this completion as follows: "*Describe all things that come to mind while completing the route, ranging from visual stimuli and scents to sounds and feelings related to the route and the navigational task*". Before the second traversal, they were instructed as follows: "*Apply the think aloud method in the same way as during the first completion but with specific attention to the objects that helped you to retrace the route or gave confirmation that you are on the correct route*". Furthermore, the participants were explained that there is no such thing as a wrong verbalisation and that all thoughts during the navigational task are valuable.

The eye fixations, recorded during eighteen recordings, were analysed by transferring them to a reference image containing 25 landmark categories with the help of BeGaze 3.4. Only the fixations at decision points were transferred, because we only focus on landmarks near decision points as Michon and Denis (2001) stated that landmarks are most needed at these locations. Furthermore, we believe that they are most relevant at decision points having the construction of route instructions in mind. In this study, a decision point was defined as a location where a change of direction took place and where multiple directional possibilities were present. The verbal protocols were analysed with the help of the software package Elan EUDICO Linguistic Annotator (version 4.6.2).

3 Results and Discussion

3.1 Structural versus Object Landmarks

As participants were completely unfamiliar with the test environment, it is expected that semantics will play a minor role and that the main features that contribute to the overall saliency of a landmark will be visual and structural. Therefore, it was examined for each decision point whether participants fixated more on object landmarks, which are objects that are independent of the building's structure (Stankiewicz and Kalia 2007), or on structural landmarks. After examining the average fixation count, average fixation time and fixation with the longest duration, it is clear that object landmarks were mostly fixated on at decision points.

Participants fixated longer and more frequently on structural landmarks, namely staircases, at only three out of seventeen decision points. Two of these locations were the only two decision points (2 and 12) along the route where the participant needed to go up the staircase in the central hallway. Both the staircase as the hallway have a very characteristic shape, which might explain the preference for structural elements at these locations. For the third location (15), a possible explanation might be that, although test persons passed almost identical staircases on other floors, there were very little object landmarks. Moreover, identical objects (i.e. emergency exit sign, emergency plan, radiator and fire alarm) were present numerous times along the route.

These findings are also reflected in the verbal protocols. Participants rarely referred to a structural element (i.e. corridor or staircase). Referrals to corridors were only made after entering that corridor (as most corridors were screened off by doors). Most participants mentioned the central hallway, but not when they were actually in that hallway. For example, some participants stated at the very beginning of the route that they had to proceed to the central hallway, but at the central hallway they referred to other objects (e.g. signs revealing the departments that are housed in the adjacent corridors) to make a navigational decision. Finally, staircases were mentioned more often and especially going up or down the stairs. However, no differentiation was made between different stairs and most verbalisations only described the action itself. Staircases were not mentioned in a way that explained why the action was taken at that location.

3.2 Structural Landmarks

With respect to the structural elements, the category "corridor" was fixated on most and longest. Although there were six decision points where a change of floor level occurred (via a staircase), participants looked more at staircases at only two of these decision points. One of them is the third location mentioned in the previous paragraph (15). The importance of the corridor as structural landmark might be explained by a study of Buchner and Jansen-Osmann (2008), which demonstrated that metric information associated to a hallway segment (i.e. length) is an important element of route knowledge.

However, these findings are not reflected in the verbal protocols. Except for decision points 2 and 12, participants never mentioned the corridor itself at decision points where a staircase had to be ascended or descended. An explanation might be found in the manner in which the fixations were assigned to the category "corridor", which will probably have led to an overestimation of the importance of hallways. This category contains all fixations on floor, ceiling and bare walls that cannot be attributed to another category. It is questionable whether a person detects the existence of a corridor in this way. As a result, it is not sensible to draw conclusions related to the most used structural landmark.

3.3 Object landmarks

Pointing out the most salient category of object landmarks is not that straightforward. Categories that were fixated on most at each decision point depending on the eye measure are listed in Table 1. Doors were focussed on most (eight out of seventeen decision points). Except for two of these locations, participants fixated most on the door that one has to take to complete the route correctly. At the remaining locations, the most fixated objects varied considerably.

When looking at the specific objects that attracted fixations for a specific category, it appears that in most cases the maximum attention may be related to a single object (e.g. display box (2), big plant (8), goods lift (9), old wooden information panel (10)). In a next step, the uniqueness of these objects is examined.

Table 1. Overview of the most fixated landmark categories for each decision point.

DP	max. AF* count	max. AF* time	max. fixation time
1	door (route)	door (route)	door (route)
2	other	other	route indicator
3	route indicator	route indicator	route indicator
4	door (route)	door (route)	door (route)
5	window	window	window
6	door (route)	other	other
7	door (route)	door (route)	door (route)
8	ornament	ornament	ornament
9	elevator	elevator	elevator
10	poster	poster	poster
11	door (other)	door (other)	door (other)
12	door (other)	door (other)	door (other)
13	route indicator	route indicator	other
14	door (other)	door (route)	door (route)
15	window	window	route indicator
16	door (route)	door (route)	door (route)
17	door (route)	door (route)	poster

* Average Fixation

3.4 Uniqueness of selected Landmarks

In order to test the criterion of uniqueness, it was verified whether the objects related to the most fixated on categories are unique. An object was defined as unique when no similar objects were present along the route or if the similar objects could have been differentiated based on one clear feature. For example, along the route there were many double brown doors, but only one was composed with two windows. The result is that eleven out of seventeen objects can be considered unique. The other six seem to obstruct the idea of identifying landmarks based on their uniqueness. However, the notion of uniqueness as proposed in the introduction indicated that false (i.e. not unique) landmarks may lead to wrong navigational decisions. Therefore, the lack of unique landmarks at decision points should have been reflected in the wayfinding performances of the participants. We now turn to the answers given in the questionnaire and the decision points that caused most difficulties.

3.4.1 Questionnaire

Although most participants stated in the questionnaire that they did not perceive wayfinding difficulties in their daily life, both in- and outdoors, route finding (and as a consequence the navigational task) was perceived as problematic in these circumstances. The average rating of complexity on a five-point scale was four. Furthermore, all participants disagreed with the proposition that the route was easy to reconstruct. Moreover, all participants made a wrong turn at least once. The questionnaire also inquired about the reason for this. All but one person made remarks that can be traced back to two causes. First, the different segments of the route are very similar. Second, there were not enough orientation points at decision points. One person even stated that he “chose the wrong landmarks in the beginning of the route”. Finally, based on a given definition, “a landmark is a distinct object in an environment that can be referred to unambiguously and can be used in route instructions”, participants attributed an average of four to the statement that there were a lot of landmarks present along the route. In contrast, the posing that there were a lot of *useful* landmarks was attributed with a lower average score of three on a five-point scale. On one hand, this may indicate that not all landmarks were perceived equally qualitative. On the other hand, this could mean that the landmarks were not always available at the needed locations (i.e. decision points).

3.4.2 Decision Points

Overall, wrong wayfinding decisions were made at ten out of seventeen decision points. The location that created most difficulties was decision point 11, where four participants got lost. This is followed by decision points 1 and 8 where three people were mistaken, and decision point 5 that caused problems for two persons. The other five locations did not raise difficulties for more than one person.

Decision points 1 and 11 are in fact very similar: a navigator has to leave a corridor towards the central hallway through a grey double door. In both cases there is a room, which can be accessed by a brown door opposite to this grey door. Moreover, the grey doors are accompanied by the same objects: evacuation signs, a fire extinguisher and an emergency light. Noteworthy is that there is another location along the route that resembles these decision points and where five participants were mistaken, making it the location along the route where most participants made a wrong decision. This location is situated between decision points 6 and 7 and was not classified as a decision point, because no change of direction took place. In other words, the objects present at these locations function as false landmarks as there were no unique elements available and resulted in wrong wayfinding decisions.

In addition, decision points 5 and 8 are also alike: participants had to go down a floor level by using the staircase at the end of the corridor. However, there were unique objects present at these locations, respectively a wooden pallet (category "other") and a large plant (category "ornament"). At decision point 8, participants mostly fixated on this unique object, but this was not the case at decision point 5. An explanation might be related to the wayfinding performances of the participants. Persons who did not experience wayfinding difficulties at decision point 5 (all but two) fixated most on the category "other". For example, their maximum fixation time for this category was 1358.6 ms compared to 499.7 ms for the two participants that got lost at that location. The latter fixated six times as much on doors and three times as much on the staircase. If only people who took a correct navigational decision at that location were taken into account, then the category "other" would have been the most fixated category and, as a result, the criterion of being unique would apply as well.

The same reasoning can be applied for decision point 8. The three participants who made a wrong wayfinding decision fixated on average much more on walls and staircases, while the remaining participants fixated much more on the category "ornament", as shown in Table 2. In other words, people who fixated on the unique objects made the correct navigational decision.

This could be explained by the fact that the selection of landmarks is highly individual. For example, potential landmarks are interpreted based on the background knowledge of an observer, which results in the fact that specific (types of) objects are often related to specific locations (Frankenstein et al. 2012). Therefore, the pallet and the plant, although unique and noticeable, might be judged as unsuited to serve as landmark by some participants. However, based on the eye tracking results it cannot be said that the wrong group "chose" another landmark to which the wayfinding action is associated. Instead, the results indicate that they scanned the environment (stairs, walls and (other) doors), perhaps looking for more clues. Therefore, the fact that they did not fixate on the plant or on the pallet can rather be interpreted as an indication that these persons did not select a landmark at all to be associated with that specific location and the action related with it. An explanation might be that the speed in which landmark knowledge is obtained depends on the observer. For example, a study of Hegarty et al. (2006) indicated that there are large differences between participants when asked to point to and to estimate the distance to landmarks after a short acquaintance with the environment.

Table 2. Overview of landmark categories that were fixated on at decision point 8 by participants who made the correct and wrong wayfinding decision at this location.

DP8	Average Fixation Count		Average Fixation Time [ms]		Maximum Fixation Time [ms]	
Category	correct	wrong	correct	wrong	correct	wrong
door	0.7	1.5	144.7	319.7	479.2	819.7
corridor	2.2	6.0	308.0	1101.8	879.3	2697.0
floor	0.2	0.3	25.0	66.4	219.8	398.5
wall	2.0	5.7	283.1	1035.4	879.3	2298.5
stair	2.3	5.3	403.0	879.3	1840.0	1800.2
Signs	0.2	0.5	41.6	99.8	399.6	259.3
pictogram	0.2	0.3	41.6	76.5	399.6	259.3
floor plan	0.0	0.2	0.0	23.3	0.0	139.7
people	1.2	0.5	269.7	123.2	1378.9	419.6
other	0.1	0.0	13.3	0.0	159.7	0.0
ornament	3.7	0.7	674.4	113.2	1857.9	339.6
elevator	1.0	2.3	184.8	489.7	838.5	1817.8
heating	0.3	0.7	55.0	116.6	300.2	619.7
poster	0.3	0.3	43.2	179.8	459.4	559.4
fire	0.2	0.7	14.9	96.3	119.6	258.5
window	0.8	0.5	138.2	79.9	459.3	279.6

Finally, at the remaining decision points where the most fixated objects are not considered unique, namely locations 9, 15 and 16, there were no unique objects present. As a result, participants could not have focussed on such items. Surprisingly, few participants got lost at these decision points. However, verbal protocols indicated that most participants based their navigational decisions on knowledge obtained at an earlier stage or decision point. For example, several participants stated at decision point 8 (by referring to the plant) that they had to go down two floor levels. Similarly, at the end of the route most participants stated that they were on the first floor and that they had to go to the second floor. Therefore, they knew that they had to go up one floor level at location 15 and enter the hallway at decision point 16.

4 Conclusions

Starting from the problem that landmarks lack a workable identification measure, especially indoors, this study made an effort to provide evidence for a new criterion, namely that of being unique, for the selection of landmarks in order to promote the integration of these cognitive important elements in wayfinding aids and route instructions. The results show that the criterion of uniqueness can be considered essential for the quality of a landmark as landmarks lacking the characteristic of being unique often lead to wrong navigational decisions. As a result, object landmarks, which can vary more in their appearance compared to structural features, were fixated on most and considered best to base wayfinding decisions on. Following, the presence of decision points without unique objects resulted in a complex route and challenging navigational task. By approaching the notion of a landmark from a different point of view, the study has proposed a more simplified identification criterion (i.e. uniqueness), which is in line with earlier research conducted by Elias (2203) and does not conflict with the primary functions and characteristics of landmarks. Finally, this study is one of the first to detect potential landmarks with the use of an eye tracking device during a real navigational task indoors. By doing so, the authors hope to have contributed to the implementation of landmarks in wayfinding applications and to the research on indoor landmarks as a whole.

5 Future Research

Very little research has been conducted with respect to the relation between eye fixations and the use of landmarks. Although the authors did not find contradictions between eye fixations and verbal protocols, further research has to be conducted on this topic in order to fully exploit the potential of eye tracking measures within the domain of indoor navigation and wayfinding.

Furthermore, as Stankiewicz and Kalia (2007) noted, persistence is also an important characteristic of landmarks. However, the most fixated objects were not always found equally persistent. This may be attributed to the fact that participants were aware of the limited time interval in which the experiment was conducted, though it is believed more research will shed more light on this topic.

Acknowledgments

This research would not have been possible without the financial support of the Special Research Fund (“Bijzonder Onderzoeksfonds” (BOF)) of Ghent University. Furthermore, the authors would like to thank the Department of Movement and Sport Sciences for the use of their mobile eye tracking device.

6 References

Brunner-Friedrich B, Radoczky V (2006) Active Landmarks in Indoor Environments. In: Bres S, Laurini R (eds) Vis. Inf. Inf. Syst. Springer-Verlag, Berlin, pp 203–215

Buchner A, Jansen-Osmann P (2008) Is Route Learning More Than Serial Learning? Spat Cogn Comput 8:289–305. doi: 10.1080/13875860802047201

Caduff D, Timpf S (2008) On the assessment of landmark salience for human navigation. Cogn Process 9:249–67. doi: 10.1007/s10339-007-0199-2

Elias B (2003) Determination of Landmarks and Reliability Criteria for Landmarks. Fifth Work. Prog. Autom. Map Gen. Paris. Paris, pp 1–12

Elling S, Lentz L, de Jong M (2012) Combining Concurrent Think-Aloud Protocols and Eye-Tracking Observations: An Analysis of Verbalizations and Silences. IEEE Trans Prof Commun 55:206–220. doi: 10.1109/TPC.2012.2206190

Elling S, Lentz L, de Jong MDT (2011) Retrospective Think-Aloud Method: Using Eye Movements as an Extra Cue for Participants ’ Verbalizations. Proc. SIGCHI Conf. Hum. Factors Comput. Syst. - CHI ’11. CHI 2011, Vancouver, BC, Canada, pp 1161–1170

Fang Z, Li Q, Zhang X, Shaw S (2011) A GIS data model for landmark-based pedestrian navigation. Int J Geogr Inf Sci 26:1–22. doi: 10.1080/13658816.2011.615749

Frankenstein J, Brüssow S, Ruzzoli F, Hölscher C (2012) The language of landmarks: the role of background knowledge in indoor wayfinding. Cogn Process 13:165–170. doi: 10.1007/s10339-012-0482-8

Gärling T, Lindberg E, Mäntylä T (1983) Orientation in buildings: effects of familiarity, visual access, and orientation aids. J Appl Psychol 68:177–186.

Gerjets P, Kammerer Y, Werner B (2011) Measuring spontaneous and instructed evaluation processes during Web search: Integrating concurrent thinking-aloud protocols and eye-tracking data. Learn Instr 21:220–231. doi: 10.1016/j.learninstruc.2010.02.005

Golledge RG, Montello DR (1998) A new Framework for Understanding the Acquisition of Spatial Knowledge in Large-Scale Environments. In: Egenhofer MJ, Golledge RG (eds) Spat. temporal Reason. Geogr. Inf. Syst. Oxford University Press, New York, New York, USA, pp 143–154

Hegarty M, Montello DR, Richardson AE, et al. (2006) Spatial abilities at different scales: Individual differences in aptitude-test performance and spatial-layout learning. Intelligence 34:151–176. doi: 10.1016/j.intell.2005.09.005

Hölscher C, Meilinger T, Vrachliotis G, et al. (2006) Up the down staircase: Wayfinding strategies in multi-level buildings. J Environ Psychol 26:284–299. doi: 10.1016/j.jenvp.2006.09.002

Hund AM, Padgitt AJ (2010) Direction giving and following in the service of wayfinding in a complex indoor environment. J Environ Psychol 30:553–564. doi: 10.1016/j.jenvp.2010.01.002

Iaria G, Barton JJS (2010) Developmental Topographical Disorientation: a newly discovered cognitive disorder. Exp brain Res 206:189–196. doi: 10.1007/s00221-010-2256-9

Jeffery KJ, Jovalekic A, Verriotis M, Hayman R (2013) Navigating in a three-dimensional world. Behav Brain Sci 36:523–43. doi: 10.1017/S0140525X12002476

Just MA, Carpenter PA (1976) Eye fixations and cognitive processes. Cogn Psychol 8:441–480. doi: 10.1016/0010-0285(76)90015-3

Lovelace KL, Hegarty M, Montello DR (1999) Elements of Good Route Directions in Familiar and Unfamiliar Environments. Spat. Inf. theory. Cogn. Comput. Found. Geogr. Inf. Sci. Springer-Verlag, Berlin, Germany, pp 65–82

Mast V, Jian C, Zhekova D (2012) Elaborate Descriptive Information in Indoor Route Instructions. Proc. 34th Annu. Meet. Cogn. Sci. Soc. Austin, TX: Cognitive Science Society., Sapporo, Japan, pp 1972–1977

Michon P, Denis M (2001) When and Why Are Visual Landmarks Used in Giving Directions ? In: Montello DR (ed) Spat. Inf. theory. Cogn. Comput. Found. Geogr. Inf. Sci. Springer-Verlag, Berlin, Germany, pp 292–305

Peponis J, Zimring C, Kyung Chou Y (1990) Finding the building in wayfinding. Environ Behav 22:555–590.

Presson CC, Montello DR (1988) Points of reference in spatial cognition Stalking the elusive landmark. Br J Dev Psychol 6:378–381.

Raubal M (2001) Human wayfinding in unfamiliar buildings: A simulation with a cognizing agent. Cogn Process 2:363–388.

Raubal M, Egenhofer MJ (1998) Comparing the complexity of wayfinding tasks in built environments. Environ Plan B Plan Des 25:895–913.

Raubal M, Winter S (2002) Enriching Wayfinding Instructions with Local Landmarks. In: Egenhofer MJ, Mark DM (eds) Geogr. Inf. Sci. Second Int. Conf. GIScience 2002. Springer-Verlag, Berlin, Germany, pp 243–259

Sefelin R, Bechinie M, Müller R, et al. (2005) Landmarks: yes; but which?: five methods to select optimal landmarks for a landmark-and speech-based guiding system. Proc. 7th Int. Conf. Hum. Comput. Interact. with Mob. devices Serv. ACM Press, Salzburg, Austria, pp 287–290

Siegel AW, White SH (1975) The development of spatial representations of large scale environments. In: Reese HW (ed) Adv. child Dev. Behav. Academic Press, New York, New York, USA, pp 9–55

Soeda M, Kushiyama N, Ohno R (1997) Wayfinding in Cases with Vertical Motion. Proc MERA 97 Int Conf Environ Stud 559 – 564.

Sorrows M, Hirtle S (1999) The nature of landmarks for real and electronic spaces. In: Freska C, Mark DM (eds) Spat. Inf. theory. Cogn. Comput. Found. Geogr. Inf. Sci. Springer-Verlag, Berlin, Germany, pp 37–50

Stankiewicz BJ, Kalia AA (2007) Acquisition of structural versus object landmark knowledge. J Exp Psychol Hum Percept Perform 33:378–390. doi: 10.1037/0096-1523.33.2.378

Van Gog T, Kester L, Nievelstein F, et al. (2009) Uncovering cognitive processes: Different techniques that can contribute to cognitive load research and instruction. Comput Human Behav 25:325–331. doi: 10.1016/j.chb.2008.12.021

Wang L, Mou W, Sun X (2014) A Development of Landmark Knowledge at Decision Points. Spat Cogn Comput 14:1–17. doi: 10.1080/13875868.2013.784768

Williams AM, Davids K (1997) Assessing cue usage in performance contexts: A comparison between eye-movement and concurrent verbal report methods. Behav Res Methods 29:364–375.

A multi-step transformation process for automatically generating indoor routing graphs from existing semantic 3D building models

Aftab Ahmed Khan [*], Andreas Donaubauer, Thomas H. Kolbe

Chair of Geoinformatics, Technische Universität München, Germany

(aftab.khan,andreas.donaubauer,thomas.kolbe)@tum.de

Abstract. IndoorGML, a draft standard of the Open Geospatial Consortium (OGC), defines an information model for indoor space based on the requirements of indoor navigation. IndoorGML allows to represent, manage, and store different infrastructures of the indoor environment in primal (volumetric and boundary geometries) and dual (graph model) spaces along with semantic information. Furthermore, it provides a sound mathematical framework to derive, use, and manage parallel and hierarchical graph structures (layers) based on the different contextual considerations for the purpose of indoor navigation and information services. IndoorGML is not tightly coupled with a specific type of semantic 3D building model. Instead, existing standards for semantic 3D building models from the Building Information Modeling (BIM) and Topography Information Modeling (TIM) domains, namely the Industry Foundation Classes (IFC) and the City Geography Markup Language (CityGML) can be used in combination with IndoorGML. IndoorGML provides a unique platform for existing 3D semantic building models to integrate, manage, and to extend their horizon of applications with the other indoor thematic context spaces, e.g., sensor space. Therefore, there is a need to investigate the potential of integrating these different semantic building models with the IndoorGML model. This investigation goes beyond the conversion from one schema to the other; it also includes the concept of automatically deriving correct navigation structures for the indoor navigation with different types of locomotion.

In this work, we describe a multi-step transformation process to automatically generate IndoorGML datasets from existing indoor building model data given in either IFC or CityGML LoD4. Moreover, we address semantic transformations, geometric transformations, topologic analyses, and spatial reasoning in order to derive navigation structures for the different types of locomotion. We tested our methods with a complex public building. In addition to the description of our conceptual work, this paper documents the lessons we learned from this test.

1 Introduction

CityGML and IFC are two well-known semantic models from the Topography Information Modelling (TIM) (GIS), and Building Information Modelling (BIM) domains.

adfa, p. 1, 2011.

Both semantic 3D building models represent and manage semantic, geometry, and topology information through different approaches, e.g., CityGML uses boundary representations to represent building geometry while IFC mainly uses volumetric and parametric approaches. In recent years, many researchers tried to integrate both models to take benefit from the respective other area of specialization. Most of these integrations or transformations aim at translating a dataset from one schema to the other (El-Mekawy 2012, Isikdag and Zlatanova 2009).

The new draft OGC indoor modelling standard, i.e., IndoorGML, facilitates the representation, storage, and management of primal (volumetric and boundary representation) and dual spaces (graph models) of different indoor thematic contextual spaces based on the requirements of indoor navigation (IndoorGML 2014a). In addition, for indoor navigation and information services, IndoorGML provides the opportunity to manage and integrate multiple as well as hierarchical graph models.

In order to use existing semantic 3D building models either modelled according to IFC or CityGML for the representation of topographic space in IndoorGML, the 3D building models need to be both abstracted to graph models and transformed into volumetric and boundary geometries including their semantic information. This transformation requires to take care of the correct topology apart from other transformations' requirements, such that the correct navigation structures can be derived. Therefore, unlike the traditional works to translate from one information model to the other, in our case, there is a need to investigate semantic transformations, geometric transformations, topology analysis and spatial reasoning with the objective to derive correct navigation structures for indoor navigation. As integral part of these transformations, there is a need to apply algorithms for creating subspaces of the topographic space, taking into account different locomotion types, namely walking, driving and flying.

In order to fulfill these requirements and in order to achieve a high level of automation in the transformation process, we designed a multi-step transformation process to automatically generate IndoorGML datasets from indoor building models either represented in IFC or CityGML LoD4. In the remainder of this paper, we describe this new multi-step approach. Rest of the paper is organized as follows; section 2 discusses related work, and section 3 presents the generic approach and transformation steps to achieve an IndoorGML model of the main building of Technische Universität München given in either IFC or CityGML. Furthermore, section 4 describes the requirements and results of computing subspaces for different locomotion types. In section 5 we draw conclusions regarding the transformation steps and deriving subspaces.

2 Related work

2.1 From IFC to CityGML LoD4

Many researchers address interoperability and interaction between IFC and CityGML models, which are two prominent semantic models in the thematic areas BIM and TIM (3D GIS) respectively. IFC is an international standard for AEC data exchange and representation. It is designed with the prime objective to represent building objects with geometrical and semantic information (BuildingSmart, 2014). On the other hand, CityGML is an OGC standard for the representation and exchange of 3D urban objects, including buildings (Kolbe, 2009). A number of publications and projects have focused on the integration of IFC and CityGML (Isikdag and Zlatanova 2009, De Laat and van Berlo 2011). Some researchers give attention to transformation of data from IFC to CityGML (De Laat and van Berlo 2011), whereas others focus on extending CityGML with regard to conceptual requirements for converting CityGML to IFC models (Nagel et al. 2009). There is also work on bidirectional transformation between CityGML and IFC using a unified building model (El-Mekawy et al. 2011). Most of the work on transformation of datasets from IFC to CityGML focuses on transformation of geometry and semantics from one representation to the other data model. However, in our case we are also interested in deriving detailed navigable graph structures according to the different locomotion types. Therefore, we focus on a detailed representation of a building model and use an elementary approach to convert 3D building models represented in IFC with semantic, topologic, and geometric information into CityGML and then to IndoorGML in order to achieve correct navigation structures (graphs).

2.2 From CityGML LoD4 or IFC to IndoorGML

CityGML is a well-known OGC standard to store, exchange, and represent urban objects. The main features of CityGML include multi-scale modeling, i.e., five Levels of Detail (LoDs) to represent a city from regional down to interior building level, modules that contain semantic modelling for different thematic areas, definition of classes and relations for the relevant topographic objects in cities. CityGML models objects with respect to their geometrical, topological, semantic, and appearance properties. Especially interesting for indoor navigation are CityGML LoD4 models since they represent interior structures of building, e.g., room, lamps, table, pillars, stairs, etc. with Opening, Room, Building Furniture, and Building Installation classes.

While CityGML defines a detailed representation of the semantic, geometric, and topology information of indoor 3D building at LoD4, (Becker et al. 2009a, Becker et al. 2009b) address the requirements and key concepts related to indoor navigation in indoor space. A proposal was forwarded by Nagel et al. (2010) to have a new standard, i.e., IndoorGML, for indoor space representation based on these requirements and concepts. IndoorGML allows to represent and exchange indoor space information that is essential to develop and implement indoor navigation systems. IndoorGML represents geometric and semantic properties of indoor space but they differ in the space representation from CityGML and IFC. Normally, it is recommended to use IndoorGML in combination with other standards particularly for the representation of indoor subdivisions, where a subspace represented in a subgraph externally references a common indoor building model represented in any other standard, e.g., CityGML (IndoorGML 2014a, IndoorGML 2014b). Therefore, it is considered as a complementary standard to CityGML or IFC to support indoor navigation services.

In our case, we intend to subdivide the indoor space according to different locomotion types. Based on physical constraints of the different locomotion types the navigable spaces can differ. These different geometric navigable models representing navigable spaces for different locomotion types cannot be represented in a common data model using external reference feature of IndoorGML. Thus, we have to create the indoor subspace models of buildings in IndoorGML. The subspace models in IndoorGML will be sublayers of the main topographic layer (representing the building model), furthermore, to make these subspaces coherent with the main topographic layer we consider it important to convert the building model represented in CityGML to IndoorGML. In the following, we present a detailed transformation of each feature type of a public building represented in CityGML LoD4 into IndoorGML for the purpose of computing subspaces.

2.3 Driving routing graphs according to different locomotion types

A great deal of research has been carried out on deriving navigation structures (abstracted graph models) for different locomotion types from 3D building models. In contrast to our work, most papers focus on a single type of locomotion only, e.g., walking or driving. The network graphs extracted from floor plans of the buildings make these graphs only navigable for the locomotion types which are dependent on floor surfaces of the building. However, the process of indoor route planning for different types of locomotion depends on a network graph that has to be extracted from the 3D building model. Hence, these methods (Sahlemariam et al. 2008, Tsetsos et al. 2005, Stoffel et al. 2007, Dudas et al. 2009, Goetz and Zipf 2011, Lertlakkhanaku and Soyoung 2009, Lin et al. 2013, Steuer 2013) do not take into consideration the free space in indoor environment. 3D free space has the same importance as floor surfaces representing the navigable space for specific locomotion types. For example, for a flying vehicle the floor surface can be non-navigable but the free space is important to be navigable for its navigation. It shows that a navigable floor surface of a room does not define the whole room to be navigable for flying objects like UAVs. Therefore, there is a need to represent and extract the network graph from the free space and other parts of interior environment separately to decide about their navigability. Some researchers (Goetz and Zipf 2011, Dudas et al. 2009) consider users or user groups for their indoor navigation and they define a profile for each user by defining his/her physical capabilities and preferences. Furthermore, the network model extracted from the main topographic model of the building is filtered (subgraphed) based on the user's profile. In contrast to our work, these approaches do not represent the actual geometric navigable space for the user because the subgraph representing navigable space for the user is computed from a supergraph. In our work, we are interested in the computation of the actual geometric navigable spaces for the different locomotion types considering their physical constraints. The actual geometrical navigable space is reflected by subgraphs of the main topographic model using IndoorGML. In this work, we are considering the conceptual constraint model presented in (Khan and Kolbe, 2012) for each locomotion type. Most of the previous research papers give the same preference to physical and temporal constraints of locomotion type but in our case we consider physical constraints to be the base, taking precedence on temporal requirements. So, we define the different subspaces based on physical constraints of the locomotion type and generate the different graphs from subspaces automatically.

3 Generating IndoorGML datasets from semantic 3D building models

The general concept of generating IndoorGML datasets from different semantic 3D building models either represented in IFC or CityGML LoD4 and determining navigation structures according to the different locomotion types based on their specific navigating constraints is illustrated in Fig. 1.

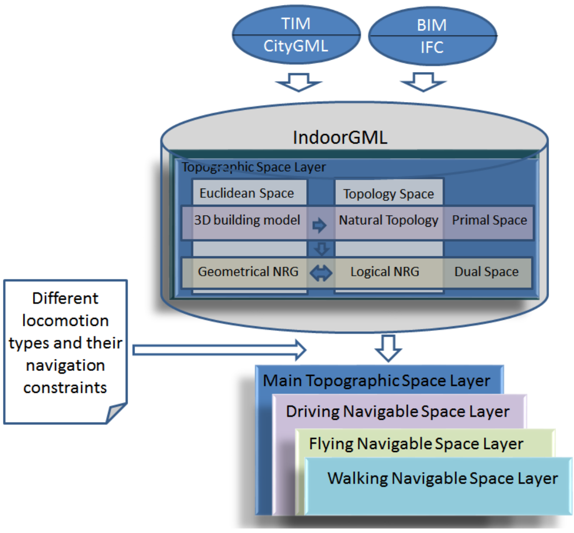

TIM
CityGML
BIM
IFC
IndoorGML
Topographic Space Layer
Euclidean Space
Topology Space
3D building model
Natural Topology
Primal Space
Geometrical NRG
Logical NRG
Dual Space
Different locomotion types and their navigation constraints
Main Topographic Space Layer
Driving Navigable Space Layer
Flying Navigable Space Layer
Walking Navigable Space Layer

Fig. 1. Generating IndoorGML datasets from TIM and BIM sources and determining navigation structures according to the different locomotion types within IndoorGML.

The Multilayered Space-Event Model (MLSEM) is a framework defined by Becker et al. (2009a) which provides not only the method to abstract or to form graph geometries (Node Relation Graph (NRG) Lee 2004) from primal space (volumetric objects e.g. representing topographic space) but also defines a link between those graph models with other graph models representing different contextual thematic spaces of indoor environment for use in indoor applications, e.g., linking an indoor topographic layer with an another layer representing sensor covering area for route planning. IndoorGML, which is based on the MLSEM concepts, is not tightly coupled with a specific type of semantic 3D building model. Instead, existing standards for semantic 3D building models from the Building Information Modeling (BIM) and Topographic Information Modeling (TIM) domains, namely the Industry Foundation Classes (IFC) and the City Geography Markup Language (CityGML) can be used in combination with IndoorGML. In a simple case, transforming IFC or CityGML to IndoorGML just means to create references between nodes of the (manually created) Network Relation Graph (NRG) (Lee 2004) representing the topographic indoor space and the corresponding IFC *IFCSpace* or CityGML *Room* objects. As our intention is to automatically create subspaces of the indoor space described by the IFC or CityGML data and to automatically derive the NRG from these subspaces taking into account the constraints defined by different types of locomotion, the transformation process from IFC or CityGML to IndoorGML is a complex task.

In order to reduce complexity and to allow the existing semantic 3D building models to be represented both according to IFC and to CityGML, we divided this transformation task into multiple subtasks which are grouped into two main steps as shown in Fig. 2: in step 1, IFC data is semantically and geometrically transformed to CityGML LoD4 and the topology is analyzed; in step 2, CityGML LoD4 data is semantically, and geometrically transformed to IndoorGML. We investigated the transformation process from parametric representation to Boundary Representation (BRep) as required both by CityGML and IndoorGML. In the semantic transformation, we focused on transforming the maximum amount of the semantic information related with each indoor object following the schema rules of the IFC source and the CityGML target object. Whereas in topology analyses, we investigated the requirement to have correct topological relations of indoor building model's objects with their connected geometries, e.g., connected door and room geometries must correctly touch each other, there must be no overlap and they must determine boundary geometry. As IFC allows for a user to model a semantic 3D building in many different ways (Nagel et al. 2009), flexibility in the transformation to CityGML is required. We account for this requirement by using a standard spatial ETL tool, FME workbench in our case, for the implementation of step 1.

In the second step, the transformation from CityGML to IndoorGML has relatively fixed rules for the semantic and geometric transformation. Here, the focus of investigation was to transform boundary geometries from CityGML to volumetric space objects in IndoorGML including their semantic information, e.g., a multisurface room feature is translated into a room solid with its boundary geometries, i.e., interior wall surfaces, etc. Besides the transformation from CityGML to IndoorGML the third step of the overall transformation procedure deals with subspacing of topographic space and deriving the NRG for different locomotion types.

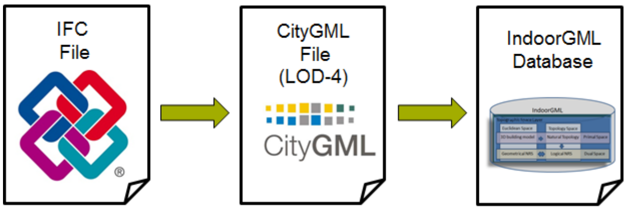


Fig. 2. 3D building model's transformation from IFC to CityGML LoD4 and then to IndoorGML.

3.1 Transformation from IFC to CityGML

The basic concepts and related work to transform from IFC to CityGML are discussed in detail in (Isikdag and Zlatanova 2009). In addition to these concepts, prior to the transformation step we need a detailed representation of the building model particularly for the identification of the navigable spaces, e.g., stairs, ramp, etc. and checking the topological relations between building elements so we can get navigation graph structures for different locomotion types.

The *IFCSpace* class defines all volumes and areas that are bounded by different building elements. For example, in Fig. 3 a room contains stairs. The whole space within that room including the stairs is represented as *IFCSpace*. As we need to compute the subspaces for different locomotion types and since for a specific type of locomotion the stairs are non-navigable (e.g., when using a wheelchair), whereas for another type of locomotion it is navigable (e.g., a walking person). Therefore, there is a need to represent the space above the stairs separately. Furthermore, all steps of the stairs may have different areas and properties. Therefore, each stair step has to be considered individually and, thus, the space above each stair step should have an individual representation (see Fig. 4). If a step is determined as non-navigable for some types of locomotion then the space above it will also be nonnavigable. The same approach is applied on each building element or area where its navigability is represented, e.g., free space above a ramp, free space within circular stairs, etc.

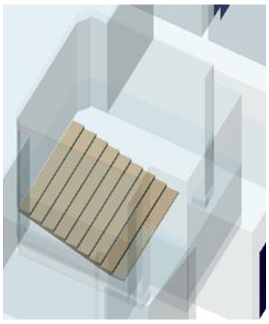


Fig. 3. *IFCSpace* representation of a room.

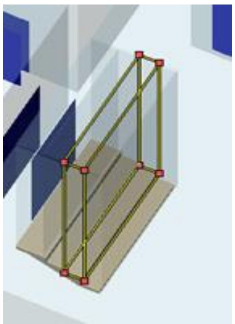


Fig. 4. Space representation above stairs

In the next step, all elements, e.g., *IFCStairs* and *IFCWall* objects, in the building model are checked whether they overlap the *IFCSpace*. If they overlap they are deduced from the *IFCSpace* to ensure that they have only a topological touch relation with the *IFCSpace*. *IFCOpeningElements*, which fill the void spaces in walls are checked for their topological relationships with the *IFCWallStandardCase* through the relation *IfcRelVoidsElements* and *IFCWallStandardCase* relation with the *IFCSpace* is checked through *IfcRelSpaceBoundary*, to be in touch relation and should not have an overlap or gap with the *IFCSpace*. Normally a door or window element fills an *IFCOpeningElement*, in this case we ignore the door or window geometry and consider the geometry of the *IFCOpeningElement* for the transformation because the former overlaps the later. Moreover, we provide simple conversion steps through which the transformation from IFC data into CityGML can be achieved. The conversion from IFC to CityGML is carried out in the following steps given in Table 1.

Table 1. Semantic mapping and transformation steps from IFC to CityGML dataset.

IFC Elements	Transformation details	CityGML Feature Types
IFCOpening-Element	Checking the relation *IFCRelFillsElement* of *IFCOpeningElement* with the *IFCDoor* or *IFCWindow* element; then the properties of *IFCDoor* or *IFCWindow* are attached to the respective *IFCOpeningElement. IFCOpeningElement* is converted into Door or Window *MultiSurface* geometries in CityGML as shown in Fig. 5.	Window MultiSurfaces/ Door MultiSurfaces
IFCSpace	*IFCSpace* geometry, which often is a parametric geometry in IFC is converted into boundary representation geometry and translated into a R*oom* feature (LoD4Solid) in CityGML as shown in Fig. 6.	Room
IFCSpace	*IFCSpace* is converted into *multiSurfaces*. Based on the height and relative altitude of *IFCSpace* the decision about each surface is taken, whether it is a *CeilingSurface* or a *FloorSurface*. If the height is between specific thresholds then it is tagged as *InteriorWallSurface*. Furthermore, *Window* and *Door* surfaces are deduced from *InteriorWallSurfaces* as shown in Fig. 7.	Floor -Surface, Ceiling -Surface, Interior-WallSurface
IFCWall	*IFCWall* is converted into *multisurfaces*. The *multisurfaces* are translated to *WallSurfaces* in CityGML, which represent the exterior shell of the building and have no connection to the *Room* feature type as shown in Fig. 8.	Wall -Surfaces
IFCStairs, IFCBeam, IFCColumn	The *IFCStairs*, *IFCBeam*, and *IFCColumn*, are translated into multisurface boundary geometries in CityGML. Moreover, IFC elements, which are within a specific room are transformed into *IntBuildingInstallation*. (Currently in our transformation process, IFC elements, e.g., *IFCBeam* and *IFCColumn* which extend over more than one room or crossing the boundary to the exterior are transformed into *BuildingInstallation* (in future we will rectify this drawback and will transformed into IntBuildingInstallation).	IntBuilding-Installation,

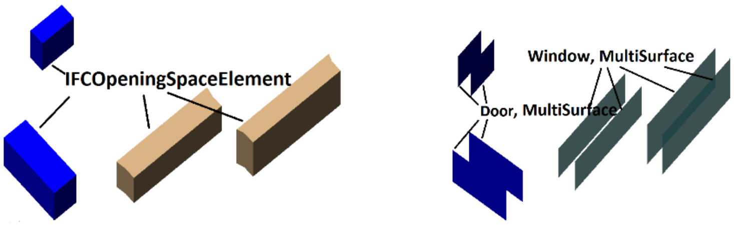


Fig. 5. Transformation of *IFCOpeningElement* to *Window or Door MultiSurfaces*.

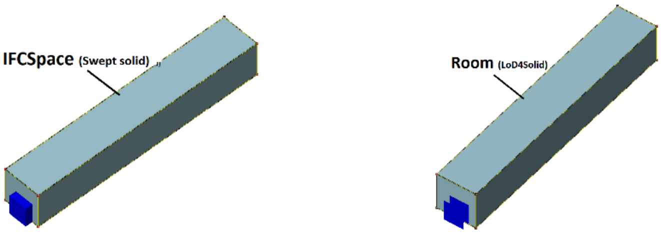


Fig. 6. Transformation of *IFCSpace* to *Room* feature type.

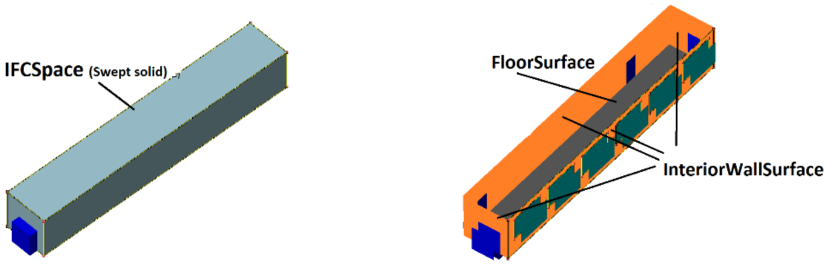


Fig. 7. Transformation from *IFCSpace* to *InteriorWallSurfaces*, *CeilingSurfaces* (not shown here), and *FloorSurfaces.*

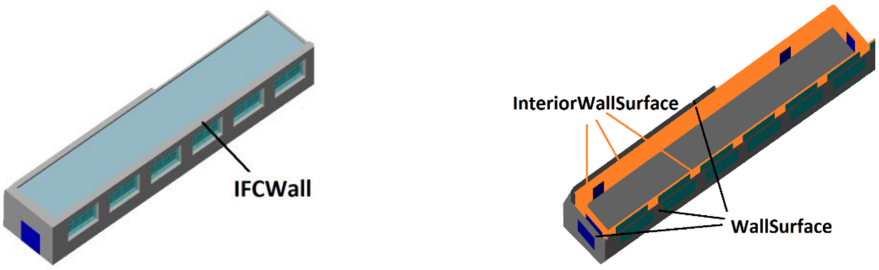


Fig. 8. Transformation from *IFCWall* to *WallSurface*.

IFCSlab objects in the IFC model are not converted to CityGML. They are only used to compute the ground surface and the roof surface of building in CityGML.

The transformation steps discussed in Table 1 were tested for different datasets and the results were found correct. The result of TUM main building's transformation from IFC dataset (shown in Fig. 9) to CityGML is shown in Fig.10.

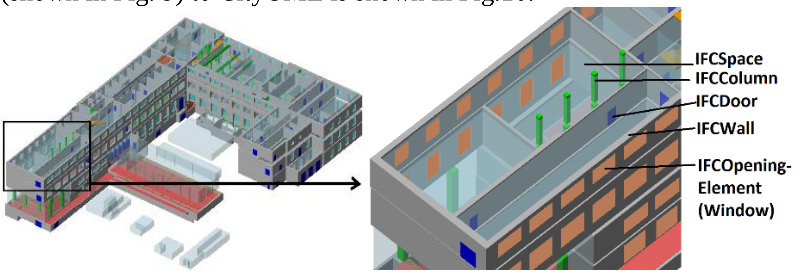


Fig. 9. (A) A part of the 3D model of TUM main building represented in IFC (left). (B). Detail view of a room and a corridor (right).

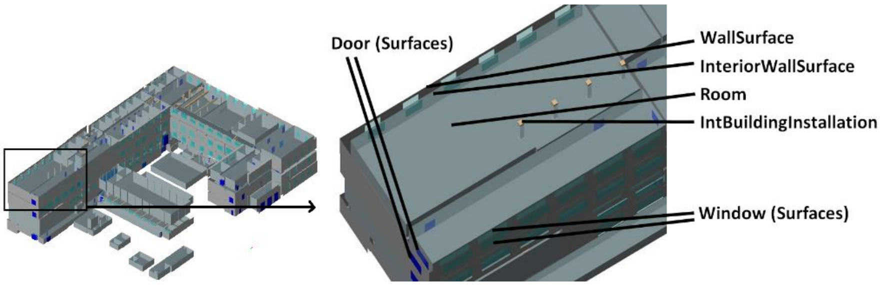


Fig. 10. A part of the 3D building model of TUM represented in CityGML after transformation (Ceiling Surfaces are removed for visualization purpose) (left). Detail view of a room and a corridor (right).

3.2 Transformation from CityGML to IndoorGML

IndoorGML is defined as an independent data model from the different approaches to building modelling, e.g., CityGML or IFC (IndoorGML 2014). Therefore, the main topographic space layer in IndoorGML can be represented using the input from a 3D building model that is represented either in CityGML or IFC or from any other information model describing the interior structure of a building. As discussed in section 2.3 we need to translate a 3D building model represented in CityGML into IndoorGML to obtain the main topographic space layer and to be able to compute subspaces according to different locomotion types. The details about CellSpace, CellBoundary, the structure model of each space layer, and the integration of multilayers can be found in (Becker et al. 2009a, IndoorGML 2014).The transformation mappings and steps to transform between the elements of CityGML LoD4 and IndoorGML are explicated in Table 2.

Table 2. Transformation mappings between conceptual CityGML and IndoorGML classes (continued on next page).

CityGML Feature Types	Transformation details	IndoorGML Elements
Room	*Room* geometry having *GM_Composite-Surface* and *GM_MultiSurface* is enforced to be a closed volume and translated into a Solid in IndoorGML	CellSpace
Door	*MultiSurfaces* representing a single Door are converted into a closed volume (Solid) in IndoorGML	CellSpace
Window	*MultiSurfaces* representing a single Window are converted into a closed volume (Solids)	CellSpace
Door as a Surface	A surface representing a Door is translated into a 3D boundary geometry in IndoorGML.	CellBoundary
Window as a Surface	A surface representing a Window is translated into a 3D boundary geometry in IndoorGML.	CellBoundary
Interior-WallSurface	An *InteriorWallSurface* representing the boundary surface of a room in CityGML is translated into a 3D boundary geometry (*CellBoundary*) of the incident room *CellSpace* in IndoorGML.	CellBoundary
FloorSurface	A *FloorSurface* representing the boundary surface of a room is converted into a 3D boundary geometry (*CellBoundary*) of the incident room CellSpace.	CellBoundary
Ceiling -Surface	A *CeilingSurface* representing the boundary surface of a room is converted into a 3D boundary geometry (*CellBoundary*) of the incident room CellSpace.	CellBoundary
Closure -Surfaces	Objects sealed using *ClosureSurfaces* are converted into a closed volume (Solid) in IndoorGML. Simultaneously, surfaces are converted into 3D boundary geometries of objects.	CellSpace and CellBoundary
BuildingFurniture, BuildingInstallation, IntBuildingInstallation	*BuildingFurniture, BuildingInstallation*, and *IntBuildingInstallation* represented by *MultiSurfaces* are converted into closed geometries (Solid) in IndoorGML.	CellSpace

WallSurface, *RoofSurface*, and *GroundSurface* objects are treated as outer *CellSpace* objects in IndoorGML and their geometries are not translated. Furthermore, each feature type in the CityGML LoD4 3D building model is translated into either a *CellSpace* or a *CellBoundary* geometry in IndoorGML with all the related attributes as described by table 2. Afterwards, the dual space geometries including state geometries (nodes) and transition geometries (edges) representing *CellSpaces* and *CellBoundary* in primal space respectively are computed to generate a space layer based on the MLSEM's method.

The steps defined in table 2 were implemented as a FME workspace and tested on of TUM main building model to translate from CityGML to IndoorGML. The resulting IndoorGML model is shown in Fig. 11.

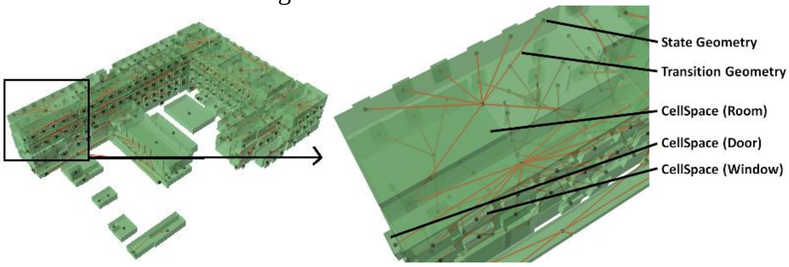


Fig. 11. Excerpt of the 3D model of TUM main building represented in IndoorGML (Space-Cell, State geometries, and Transition geometries) after transformation from CityGML model (left). Detail view of a room and a corridor (right).

4 Deriving routing graphs for different locomotion types

4.1 Subspacing approach

After having derived the IndoorGML building model either from IFC or CityGML, in next step, we compute the routing graphs for the different types of locomotion based on their specific navigating physical constraints. For each type of locomotion, i.e., flying, driving, and walking we consider an example based on its common usage in indoor environment. Those include Unmanned Aerial Vehicle (UAV), wheelchair, and a walking person respectively. The indoor navigation constraints of each locomotion type are based on the locomotion type's constraints model defined in Khan and Kolbe (2012).

In the field of robotics path planning, the mapping from work space to configuration space to determine a safe route for a rigid object resemble to a route for a point through the configuration space map. This approach has withdrawn the requirement for 2D or 3D collision detection and simplifies the path planning problem to finding a line that connects the start and target configurations avoiding the unsafe space. It also distinguishes the work space into three categories based on two solid objects which cannot overlap: obstacle configurations, in which objects will overlap; safe or free configurations, in which no overlap occurs and contact surface configurations, in which two or more objects touch each other (Lozano-Perez, 1983). This method is not specific to the robotics but also has been applied in the areas of construction, auto mechanics, etc. (Wise and Bowyer, 2000). Considering the simplicity, accuracy, and application of this approach in different fields we intend to compute the navigable spaces for the locomotion types through configuration space mappings. In a 3D environment we considered the generalized geometric models of a flying object, a walking person, and a wheelchair as 3D sphere and cylinders respectively along with their specific navigating physical constraints. The computation of the configuration space mapping was carried out based on Minkowaski's sum method (Varadhan and Manocha 2006).

The decision to determine a specific element of the indoor space as navigable or non-navigable for the given locomotion type is taken by considering the physical navigating constraints of the locomotion type and spatial information (semantic, geometric, and topology information) of the element. The indoor space element, which is determined as non-navigable, will determine obstacle space around it to be deducted from the free space.

4.2 Example scenario

Consider a 3D building model containing a corridor and a room that contains four columns. The representation of building elements in CityGML and corresponding representation in IndoorGML are presented in Fig. 12 (A) and Fig. 12 (B) respectively. The extraction of a network model from the building as main topographic space layer in IndoorGML is shown in Fig. 13 (A). Most of the methods compute the navigable subspace for the locomotion type using constraints of the indoor space at the graph level. For example, the navigable space for the wheelchair shown in Fig. 13 (B) is computed considering its capabilities and constraints of the indoor space from the network model shown in Fig. 13 (A). The decision of the navigability of each element of building, e.g., a door shown in Fig. 13 (A) is taken after considering its spatial properties, i.e., length and width. If the length and width of the door is greater than the length and width of the wheelchair, then the door is considered to be navigable.

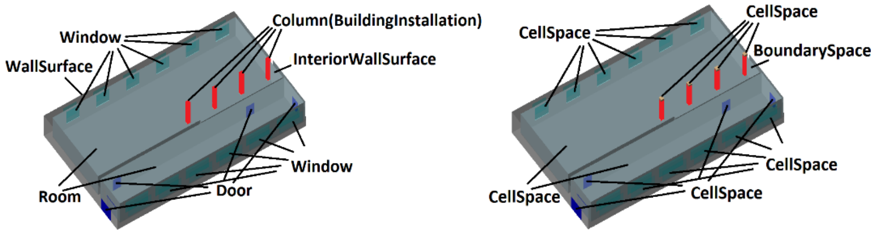


Fig. 12. (A) 3D building model in CityGML (left) (B) 3D building model in IndoorGML (right)

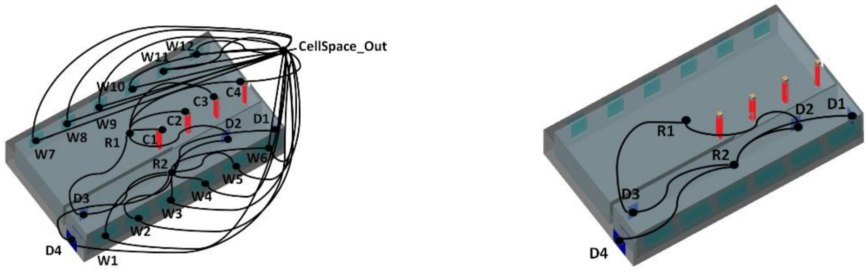


Fig. 13. (A) In IndoorGML, the main topographic layer of the 3D building model (left). (B) Navigable subspace computed based on network model according to wheelchair navigation (right).

The network model and the subspace building model representing the navigable space for the wheelchair shown in Fig 13 (B) is not enough precise for approximating the reasonable navigable space. Because, there are other locomotion types (e.g. flying) which may require precise or the detail geometric indoor navigable space so they avoid collision with the obstacles (e.g. column) located in the room. Therefore, we have to compute the actual navigable space after deducing the obstacle space (non-navigable space created by obstacles).

The physical constraints of the wheelchair are considered to determine the obstacles according to the constraints model defined in Khan and Kolbe (2012). For example, considering a wheelchair and its navigation constraints, a decision to determine navigability of the specific element is taken after considering all its properties. If there is a free space element from indoor space then we consider a collection of constraints for a wheelchair which need to be fulfilled to declare the free space navigable for the wheelchair. In this example, as a first step, we consider the *ScaleGeometryRelatedConstraint* of the locomotion type, according to this constraint it needs volume of 1 meter cubic or more free space to navigate. Furthermore, we have to consider more constraints as given in (Khan and Kolbe 2013), some of them are shown as example in Fig. 14 and they are combined through *complexlocomotionconstraint* operator “and”. So they all need to be fulfilled to determine free space navigable for the wheelchair. The next constraint is the *DirectionalGeometryRelatedConstraint,* which requires the wheelchair to have a surface to be held on or the free space must have a floor surface. Once that constraint is fulfilled, the free space element is checked for *NotConsiderConstraint,* whether the indoor element is “Window” in this case as it is not window so it become irrelevant to be fulfilled, otherwise if it is window then it is determined as non-navigable. Then in the next step, the *TopologicalGeometryRelatedConstraint* is considered which emphasizes that the free space must fulfill the requirement to be navigated “within” with the geometry of locomotion type. If the free space has enough space to contain locomotion type within then that free space element is navigable otherwise it determines as non-navigable. In further realization of constraints of the wheelchair on free space element of indoor space, the *CapacityContraints* are considered that include *CrossThrough and PassOn,* in this case, the wheelchair is evaluated if it has the capacity to cross through free space and pass on floor surface of the free space, the free space is computed as navigable otherwise non-navigable.

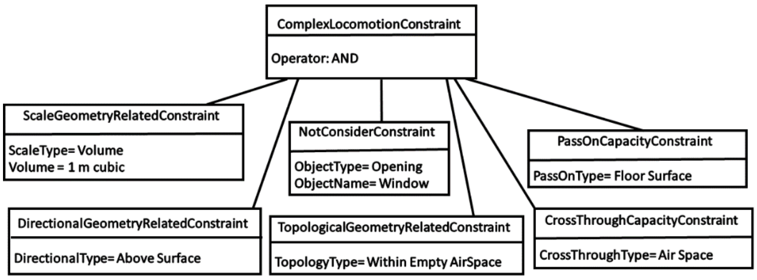

ComplexLocomotionConstraint
Operator: AND
ScaleGeometryRelatedConstraint
ScaleType= Volume
Volume = 1 m cubic
NotConsiderConstraint
ObjectType= Opening
ObjectName= Window
PassOnCapacityConstraint
PassOnType= Floor Surface
DirectionalGeometryRelatedConstraint
DirectionalType= Above Surface
TopologicalGeometryRelatedConstraint
TopologyType= Within Empty AirSpace
CrossThroughCapacityConstraint
CrossThroughType= Air Space

Fig. 14. An example of a complex locomotion constraint for computing navigable subspace for a wheelchair formed by aggregating subconstraints.

In this example, after considering constraints from Fig. 14 we determined the free space and door spaces as navigable for the wheelchair. Furthermore, considering other constraints from constraints model of the locomotion type we declare columns, windows, and walls of the room as non-navigable. The non-navigable spaces (e.g. columns, windows, and walls) will determine obstacle spaces based on Minkowski's sum as shown in Fig. 15 (A) in pink color. The actual navigable space is determined after deducing the obstacle space as shown in Fig. 15 (A) in green color. Furthermore, the route graph for the wheelchair is formed using the IndoorML method (Poincare duality) from the actual navigable space as shown in Fig. 15 (B).

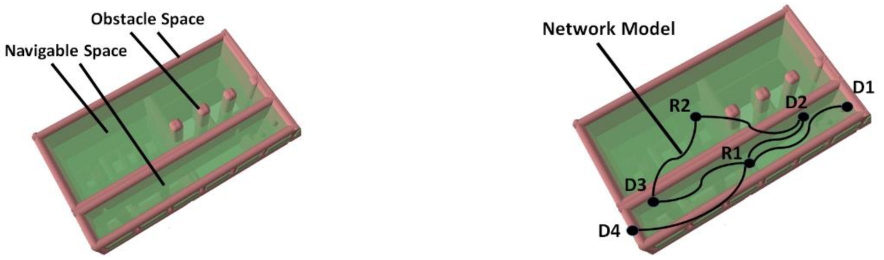


Fig.15. (A). Actual Navigable spaces after deducting obstacle spaces according to the wheelchair shown in green color (left). (B). Network model of the corresponding navigable space (right).

The difference between navigable space that is computed for a wheelchair through graph based approaches and free or safe navigable space which is computed through configuration space (Lozano-Perez 1983) approach can be observed in Fig. 13 (B) and Fig. 15 (B) respectively. The navigable space computed using configuration space is more precise particularly giving geometric details of non-navigable space around the columns which is not possible to represent through graph based approaches.

4.3 Discussion

We demonstrate in section 4.2 why it is important to derive geometric subspaces for the different locomotion types in contrast to many other approaches (Meijers et al. 2005, Stoffel et al. 2007, Dudas et al. 2009, Lertlakkhanaku and Soyoung 2009, Petrenko et al. 2014) where subspaces are computed only on a graph model level. The navigable space that is computed through graph based approaches, in essence, used only some geometric position (centroid of the object) and connection information between spatial objects (topological graph). The semantic information (e.g. types of spaces, and properties of building components) and the actual geometry of the object have not been considered yet. In contrast, the subspacing we carry out through the configuration space approach uses fully geometric and semantic information from a semantic 3D building model. In addition, if there are obstacles within an indoor space (e.g. column), the methods based on the graphs will fail or be not precise enough for approximating the reasonable navigable space, which may limit the path planning in many route planning applications.

From the brief discussion and comparison above it is apparent that it is necessary to compute the accurate subspaces at the geometric level for the given locomotion type and to extract the network models from the navigable space.

4.4 Implementation

We tested our methods on the complex TUM main building model, the semantic 3D model of which is available as an IFC dataset as shown in Fig. 9. The IFC dataset is translated into a CityGML dataset and further into IndoorGML as showed in Fig. 10 and Fig. 11 respectively. To compute the subspaces according to different locomotion types the obstacles for the given locomotion type are determined based on their constraint models. After determining the obstacles the unsafe regions around each obstacle are computed based on Minkowski sum (Barki et al. 2009), e.g., for the wheelchair the obstacle space is shown in Fig.16. The network graphs are extracted from the actual navigable spaces (calculated after deducing the obstacle spaces) to represent the subspaces for the given type of locomotion. Fig. 17 and Fig. 18 show the resulting network models for the navigable spaces for UAVs and wheelchairs respectively. The results of the subspacing of the main topographic space according to different locomotion types show that they significantly differ from each other.

The process of transformation is implemented using the ETL tool FME Workbench (FME 2014) and the implementation of subspacing by a Java program in combination with Oracle spatial DBMS and Esri ArcObjects (Oracle 2014, ArcObjects 2014). The automation and proof of concept on error-prone real-world data show that our steps of transformation are simple to implement and easy to adjust to deal with the flexibility of IFC input data.

Furthermore, our experience shows that normally the dataset, e.g., IFC 3D building models, provided by Construction Engineering and Design community have several topological issues, e.g., geometries are overlapping or have unnecessary gap to directly generate CityGML or IndoorGML models. Therefore, before using those models, we recommend to perform topology checks on the building elements as we did as part of the transformation process (see section 3.1).

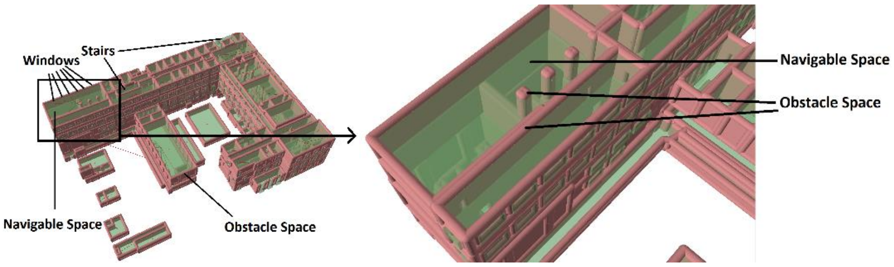


Fig. 16. Obstacle spaces (Unsafe regions) around obstacles (in pink) and navigable spaces for the wheelchair (in green) (left). Detail view of a room and a corridor (right).

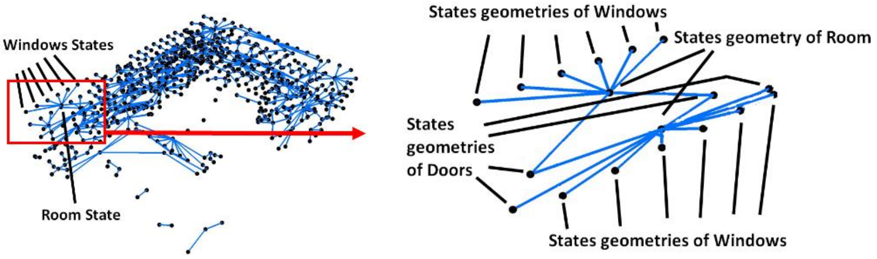

Windows States
Room State
States geometries of Windows
States geometry of Room
States geometries of Doors
States geometries of Windows

Fig. 17. The dual space of the UAV navigable subspace layer (left). Detail view of a room and a corridor (right).

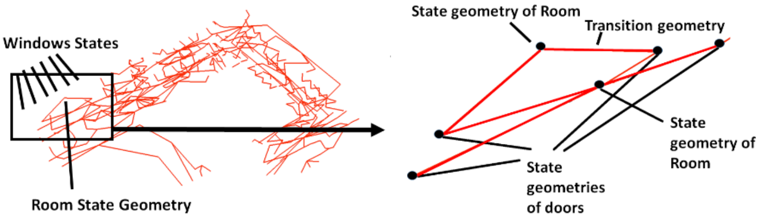


Fig. 18. The dual space of the wheelchair navigable subspace layer (state geometries of windows and stairs spaces are missing (non-navigable)) (left). Detail view of a room and a corridor (right).

5 Conclusions

We investigated a multi-step transformation process and demonstrated that IndoorGML datasets can be automatically derived from existing semantic 3D building models structured according to IFC or CityGML to support indoor navigation for different types of locomotion.

Dividing the overall transformation process into the two steps “IFC to CityGML LoD4” and “CityGML LoD4 to IndoorGML” has major advantages as follows: First, the workflow allows the source data to be structured either according to IFC or CityGML. Second, the transformation procedure from IFC to CityGML can be kept quite flexible accounting for the high degree of flexibility offered by IFC for structuring building models whereas the CityGML-to-IndoorGML transformation has fixed and simpler transformation rules. Furthermore, we presented the subspacing approach and demonstrated it for a public building using IndoorGML taking into account different locomotion types. The subspaces are computed using the real 3D geometry based on configuration space method and then network models are extracted. The subspaces created at the geometric level are more precise and consider semantic and geometric information of 3D building model making our approach different from other approaches.

The detailed representation of the 3D building model’s elements (e.g. detail representation of stairs free space) and their topology checking support to extract correct and detailed graphs for indoor navigation. Overall, the automation of the transformation process and the subspacing to support different types of locomotion for the indoor navigation for a public building show that our methods simplify the process and help to avoid manual errors and demonstrated the feasibility of the approach.

6 References

1. ArcObject (2014), ESRI ArcObjects: http://resources.esri.com/help/9.3/arcgisengine/java/doc/b0a96bd8-fc78-4573-9a70-e108cf6a4580.htm, Accessed 14 June 2014
2. Barki H, Denis F, Dupont F (2009) Contributing vertices-based Minkowski sum computation of convex polyhedral. Computer-Aided Design 41(7):525-538
3. Becker T, Nagel C, Kolbe T H (2009) (a) A Multilayered Space-Event Model for navigation in indoor spaces. In: Lee J, Zlatanova S (eds) 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, p 61-77
4. Becker T, Nagel C, Kolbe T H (2009) (b) Supporting contexts for indoor navigation using a multilayered space model. In: Tenth international conference on mobile data management: systems, services and middleware, Taipei, May 2009. IEEE, p 680-685
5. BuildingSMART: http://www.buildingsmart.org/. Accessed 13 June 2014
6. De Laat R, van Berlo L (2011) Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In: Kolbe T H, König G, Nagel C (eds) Advances in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, p 211-225
7. Dudas P M, Ghafourian M, Karimi H A (2009) ONALIN: ontology and algorithm for indoor routing. In: Proceedings of the Tenth International Conference on Mobile Data Management: Systems, Services, and Middleware (MDM09), Taipei, May 2009. IEEE, p 720-825
8. El-Mekawy M, Ostman A B, Hijazi I (2012) An evaluation of ifc-citygml unidirectional conversion. International Journal of Advanced Computer Science and Applications 3(5):159-171
9. El-Mekawy M, Östman A, Shahzad K (2011) Towards interoperating CityGML and IFC building models: a unified model based approach. In: Kolbe T H, König G, Nagel C (eds) Advances in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, p 73-93
10. FME (2014), Safe softwares: www.safe.com, Accessed 13 June 2014
11. M, Zipf A (2011). Formal definition of a user-adaptive and length-optimal routing graph for complex indoor environments. Geo-Spatial Information Science 14(2):119-128
12. IndoorGML (2014)(a): Open Geospatial Consortium (OGC) IndoorGML draft. OpenGIS specification. OGC's document no. OGC 14-005r1, Version. v.0.9.0.
13. IndoorGML (2014)(b): IndoorGML: www.indoorgml.net, Accessed 13 June 2014
14. Isikdag U, Zlatanova S (2009) Towards defining a framework for automatic generation of buildings in CityGML using building Information Models. In: Lee J, Zlatanova S (eds) 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, Springer Berlin Heidelberg, p 79-96
15. Khan A A, Kolbe T H (2012) Constraints and their role in subspacing for the locomotion types in indoor navigation. In: Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Nov 2012. IEEE, p. 1-12
16. Khan A A, Kolbe T H (2013) Subspacing based on connected opening spaces and for different locomotion types using geometric and graph based representation in Multilayered Space-Event Model. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Vol II-2/W1p, p. 173-185

17. Kolbe T H (2009) Representing and exchanging 3D city models with CityGML. In: Lee J, Zlatanova S (eds) 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, Springer Berlin Heidelberg, p 15-31
18. Lee J (2004) A spatial access-oriented implementation of a 3-D GIS topological data model for urban entities. GeoInformatica 8 (3):237-264
19. Lertlakkhanaku J, Soyoung B (2009). GongPath Development of BIM based Indoor Pedestrian Navigation System. In: Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, Seoul, Aug 2009. IEEE, p 382-388
20. Lin Y H, Liu Y S, Gao G, Han X G, Lai C Y, Gu M (2013) The IFC-based path planning for 3D indoor spaces. Advanced Engineering Informatics 27(2):189-205
21. Lozano-Perez T (1983) Spatial planning: A configuration space approach. Transactions on Computers IEEE 100(2):108-120
22. Meijers M, Zlatanova S, Pfeifer N (2005) 3D Geo-information indoors: structuring for evacuation. In: Proceedings of the 1st International ISPRS/EuroSDR/DGPF-Workshop on Next Generation 3D City Models (EuroSDRBonn), Bonn, Germany, 21–22 June 2005, p. 6
23. Nagel C, Stadler A, Kolbe T H (2009) Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3D models. In: ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Vol XXXVIII-3-4/C3
24. Nagel C, Becker T, Kaden R, Li K, Lee J, Kolbe T H (2010) Requirements and space-event modeling for indoor navigation. Tech. Rep. OGC 10-191r1, Open Geospatial Consortium, Discussion Paper.
25. Oracle database 11g: www.oracle.com, Accessed 14 June 2014
26. Petrenko A, Sizo A, Qian W, Knowles A D, Tavassolian A, Stanley K, Bell S (2014) Exploring mobility indoors: an application of sensor-based and GIS systems. Transactions in GIS 18:351–369, doi: 10.1111/tgis.12102
27. Steuer H (2013) High precision 3D indoor routing on reduced visibility graphs. In: Krisp**,** Jukka M. (ed) Progress in Location-Based Services*,* Lecture Notes in Geoinformation and Cartography Springer*,* Berlin*,* p 265-275
28. Stoffel E P, Lorenz B, Ohlbach H J (2007) Towards a semantic spatial model for pedestrian indoor navigation. In: Rolland C, Trujillo J, Yu E, Zimlanyi E, (eds) Advances in Conceptual Modelling-Foundiations and Applications, Lecture Notes in Computer Science, vol 4802, Springer, Berlin, p 328-337
29. Sahlemariam Y, Ahn S, Ko H (2008) Context based pathfinder for personalized indoor navigation. Master thesis. Korea Institute of Science and Technology.
30. Tsetsos V, Anagnostopoulos C, Kikiras P, Hasiotis P, Hadjiefthymiades S (2005) A human-centered semantic navigation system for indoor environments. In: Proceedings of the International Conference on Pervasive Services *ICPS'05*, July 2005. IEEE, p 146-155
31. Varadhan G, Manocha D (2006) Accurate Minkowski sum approximation of polyhedral models. Graphical Models 68(4):343-355
32. Wise K D, Bowyer A (2000) A survey of global configuration-space mapping techniques for a single robot in a static environment. The International Journal of Robotics Research 19:762

RFID-Based Mobile Positioning System Design for 3D Indoor Environment

Emrullah Demiral[1], Ismail Rakip Karas[1], Muhammed Kamil Turan[2], Umit Atila[1]

[1]Department of Computer Engineering, Karabuk University, Karabuk, Turkey

emrullahdemiral@gmail.com

ismail.karas@karabuk.edu.tr

umitatila@karabuk.edu.tr

[2]Department of Medicine, Karabuk University, Karabuk, Turkey

kamilturan@gmail.com

Abstract In this study, an RFID based indoor positioning system has been proposed. In the system, while RFID readers have been considered to be mobile, RFID tags have been attached on fixed positions inside building. Performance of various types of readers and tags on indoor positioning has been investigated and most appropriate tag/reader couple has been used. In the experiments of this study, geographical proximity approach has been used. As the results of tests performed on three different model proposed for indoor positioning, it has been shown that best rate for position estimations without error have been obtained from third model with the rate of approximately 76% and in the worst case, position estimation error has been obtained 2 meters.

Keywords: RFID, Indoor, Positioning, Mobile, Geographical Proximity

1 Introduction

The number of high, complex and wide spread buildings increases with each passing day in modern cities of today's world. These buildings with dozens of floors, hundreds of corridors, rooms and passages are almost like a city in terms of their complexity and number of people occupied. Depending on the size and complexity of buildings many problems arise which need to be solved. One of the leading problems is developing personalized navigation applications for performing safe and quick evacuation of buildings in case of emergency. Beside, guiding visually impaired people, security, visitor tracking, finding addresses, service organizations and guiding tourists are some other application areas. In widely used 2D applications such as car navigation systems, location of a user is determined using Global Positioning System (GPS). However, GPS cannot be used directly in indoor spaces and we need accurate indoor positioning systems.

There are many technologies and systems proposed for indoor positioning. RFID technology is a step ahead from other current technologies in terms of accuracy and other advantages. Today, there are some RFID readers and tags which are plug and play on smart phones. These portable RFID readers and tags which can be easily found in market are not widespread because of some disadvantages such as short read ranges, causing mobile devices run out of charge quickly and making attached mobile device grow in size. Beside, with the advancement of technology it is expected that RFID readers and tags are placed inside mobile devices in production just like integrated WIFI adapters and this advancement will provide a more effective use of RFID in the near future. Hence, mobile phones will be able to use RFID based indoor positioning systems and serve various personalized services.

1.1 Positioning Systems for Indoors

There are several technologies addressed for indoor positioning. These are Ultrasound, RFID, Bluetooth, WLAN, Pseudo-GPS and Infrared. There are major disadvantages of using these systems except RFID such as requirement of direct line of sight between receiver and transmitter in Ultrasound and Infrared technologies, delays in data transmission and limited bandwidth in Bluetooth and signal reflection and varying signal strength due to the dynamic network structures in WLAN. However, there is no need of direct line of sight between receiver and transmitter in RFID technology. RFID offers the opportunity of working in any environment and interaction between receiver and transmitter is fast, but we should keep in mind that RFID readers are expensive (Lemieux 2009, Candy 2008). Advantages and disadvantages of indoor positioning technologies are summarized in Table 1.

Table 1 Indoor positioning technologies (Lemieux 2009, Candy 2008)

Technology	Advantages	Disadvantages	Accuracy
Ultrasound	-Simple and cheap equipment -Precision measurement possibilities	- The necessity of the establishment of the recipients in every room -Negative influence of a high density-tone. -Receiver and transmitter need to see each other directly.	a few centimeters when it is placed frequent enough
RFID	-The transmitter and receiver don't need to see each other directly.	-Passive tags have low precision than active tags. -RFID Readers are expensive.	1 cm-2 meters. (Depending on

	-Opportunity to work in all types of environments. -Fast interactions. -Passive tags are cheaper and smaller than active tags and it doesn't need batteries.		the placement of RFID readers and tags)
Bluetooth	-Any kind of Bluetooth device can be monitored (cell phone, mp3 player, laptop) -Variable reading distance (1 m. vicinity) -Relatively cheap for small scale installations.	-Cost effective for large scale installation -Limited bandwidth. -Transmission delay. -The possibility of a maximum of seven sub-links corresponds to main connection.	2-15 meters.
WLAN (Wifi)	-The possibility of use the infrastructure of IEEE 802.11. -Low cost.	-Poor performance for multi-layered and very dense areas. -Due to signal reflection and dynamic network structure variable signal strength.	1-3 meters. (in range of 50 meters)
Pseudo-GPS		-Necessity of establishment of small satellites similar to GPS satellites into-building -Mobile devices have got GPS receiver -High cost	About 1 meter
Infrared	-Compact. -Low power consumption.	-Sensitive to daylight. -Receiver and transmitter need to see each other directly. -Installation and maintenance is cost.	5-10 meters.

1.2 RFID Technology

RFID is a term used to define wireless non-contact use of radio frequency electromagnetic fields to transfer identification data of an object for the purposes of identifying and tracking (Khong and White 2005). The data is stored in tags which is an electronic data storage device (Transponder tag) like smart cards. On the other hand, unlike smart card systems the power needed for both tags and transferring data between reader and tag is provided by use of non-contact electromagnetic field. A reader is required to receive data from a tag (Finkenzeller 2003). A reader loads energy to its antenna in order to make it transmit radio signals for activating tags and receiving data from tags. An activated tag transmits its data (Khong and White 2005). The antenna provides communication between reader and tag and some properties differ such as frequency range which affects the performance of the system depending on the shape and size of the antenna (Dziadak et al. 2009).

RFID tags can be either active or passive according to the power source. An active tag has its own power source generally obtained from a battery and this kind of tag transmits its ID periodically. A passive tag gets its power from the signal of the reader (Manish and Shahram 2005).

Two fundamental components of an RFID system are reader and tag. Beside this, antennas, computers and database systems are used in order to make system more effective. Components of an RFID system are shown in Figure 1. Another important issue is the frequency range of the reader. Available frequencies are LF (Low Frequency), HF (High Frequency) and UHF (Ultra High Frequency) (L. Wang et al. 2007). In addition, frequency ranges such as SHF (Super High frequency) or microwave can be used. Properties of radio frequencies are shown in Table 2.

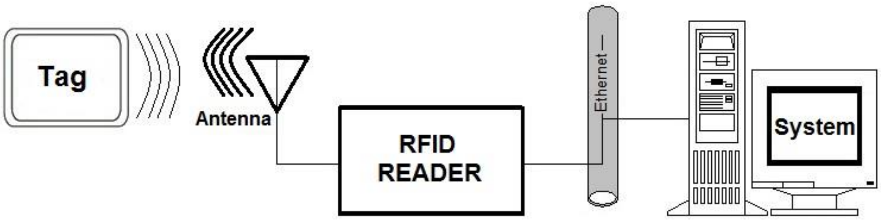


Fig. 1 Components of an RFID System

Table 2 Properties of Radio Frequencies (Era Build 2006, N. Li and B. Becerik Gerber 2011)

	Low Frequency(LF)	**High Frequency(HF)**	**Ultra High Frequency(UHF)**	**Microwave**
Frequency Range	125-135 kHz	13.56 MHz	400-960 MHz	2.45-5.8 GHz
Read Range	<0.5 meter (passive)	<1.0 meter (passive)	<10 m. (passive) >10 m. (active)	>100 m (active)
Standards	ISO 11784/5, 14223, 18000–2	ISO 14443, 15693, 18000–3	ISO 18000–6/7, EPCGen1 and 2	ISO 18000–4/5
Impact of Metal/liquid	Very Low	Low	High	High
Data Transfer Speed	Low	Average	High	High
Multiple Readings	-	50 tags/sec.	150 tags/sec.	-
Usage	Farming, Security, Beverage Factory	Medicine Industry, Healthcare Sector	Production, Logistics and Construction Sectors	Military, Shipping, Airlines

2 RFID Based Positioning Techniques

2.1 Distance Estimation

In this approach, position of a target is estimated based on the geometric features of triangle. Determining the position of a target is possible with two ways. One of them requires measuring the angle between the unknown position and at least two reference points. Intersection of lines drawn according to these angles points the estimated location. The other one requires knowing the distances between the target and three reference points in order to estimate the target position.

While determining the positions with the methods mentioned above, several measured features are used related to communication between readers and tags such as received signal strength (RSS), time of arrive (TAO), time difference of arrive (TDOA), received signal phase (RSP) and angle of approach (AOA).

2.2 Scene Analysis

Scene analysis approach consists of two main steps. First of all, environmental information (finger print) is gathered. Then location of the target is estimated by comparing the finger print set and online measurements. Generally RSS based finger prints are used. First of two fingers print based methods is the k-Nearest Neighbors (kNN) also known as (radio wave mapping) and the second one is probability method. Nearest neighbors method is realized using first RSS measurement on stations with known locations in order to create RSS database defined as radio wave map. Later in the online phase, k nearest matches is searched in the signal space produced by existing RF devices using RSS measurements directed to target points. Later, mean squared error is applied on the selected adjacent points to estimate the target point's location. In the probability approach, location of the target is determined assuming that target point has n possible locations and there is a signal strength vector observed by experimental probability and Bayes formulation in the online phase. Therefore, the location with highest probability is selected. Typically, probability methods consist of calibration, active learning, error prediction and tracking phases.

2.3 Proximity

This approach depends on the distribution of receiver/transmitter. It is assumed that point is in the same location with receiver/transmitter when the target enters the domain of receiver/transmitter. If more than one target is detected, it is assumed that target is in the same location with the receiver getting the strongest signal. This approach is fundamental and easy to apply. In addition, accuracy is proportional to the size of cells.

3 Materials and Method

3.1 Determination of the Reader and Tags to be used in the Study

Basic components of the RFID systems are RFID readers and RFID tags. The target that is required to be determined of its location could be an RFID reader or an RFID tag. In other words, the tags could be fixed and the readers could be in motion, and vice versa. In our study, since the requirement of equipment independent buildings (eg. in emergency situations), a system which the RFID readers are in motion and passive tags are attached to predetermined points on the surfaces of building has been suggested. Usages and performances of various types of readers and tags have been investigated. Considering the usability and functionality, ATID AT 870 model has been preferred for the mobile RFID Reader. On the other hand, 14 different types of passive tags have been tested in various aspect. Their performances have been compared regarding to the range of reading, independency from aspect of reader, and response time. After these detailed tests, TE34 Gain model passive tag has been preferred.

3.2 Settlement Patterns of the Tags

The tags have been attached in three different design models to the application area. In the first model, the tags have been consecutively attached within one meter intervals as a straight line to the ceilings of corridors and a room in application area. Height of the ceiling is 320 cm. Figure 2 shows a corridor having tags plugged on ceiling.

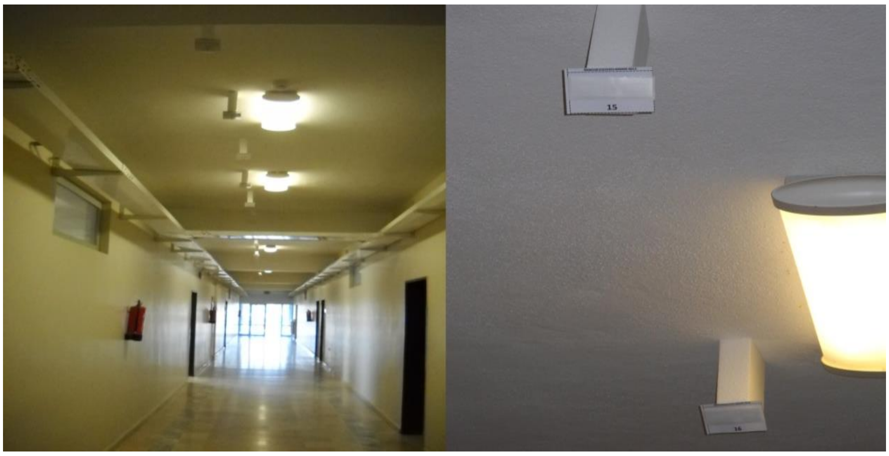


Fig. 2 Image of the corridor having tags plugged on ceiling in which Model 1 has been installed

In the second model, the tags have been attached up to the 220 cm height on the surfaces of two opposite walls of a room. Figure 3 shows the room in which Model 2 has been installed.

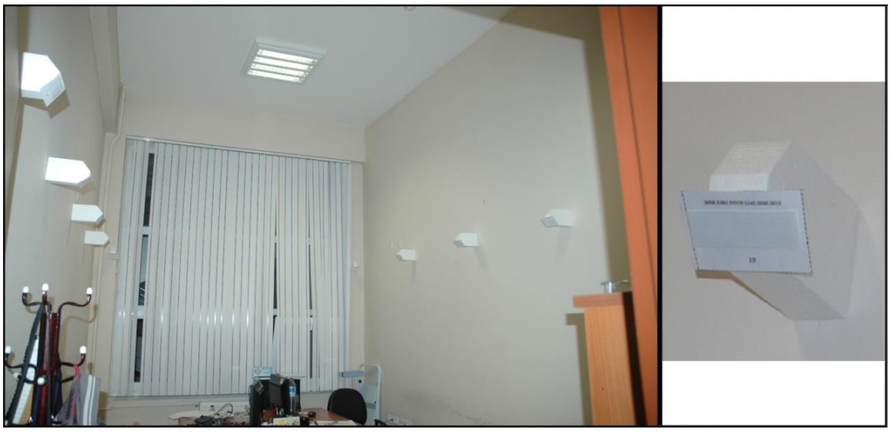


Fig. 3 Image of the room in which Model 2 has been installed and close view of tag on wall

In the third model, the tags have been attached to the ceiling of a room which has 4 m x 4 m dimension. 16 different tags have been placed to the centers of 1 m dimensional quadratic cells. Height of the ceiling is 250 cm. Figure 4 shows the room in which Model 3 has been installed.

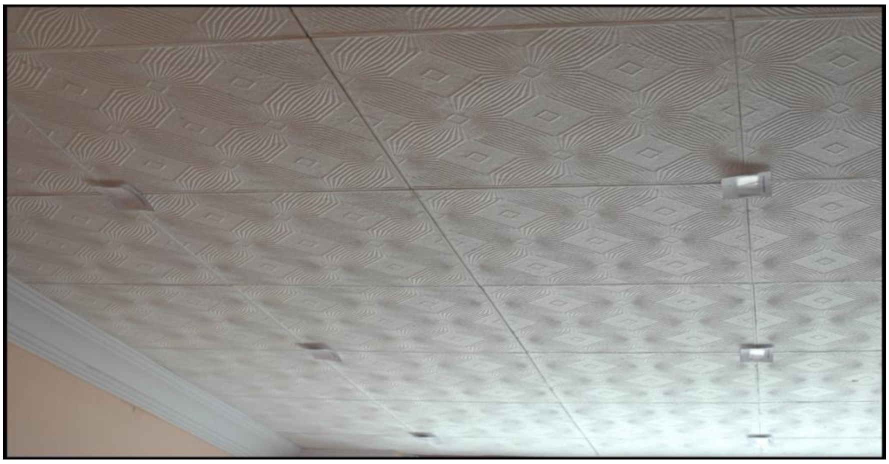
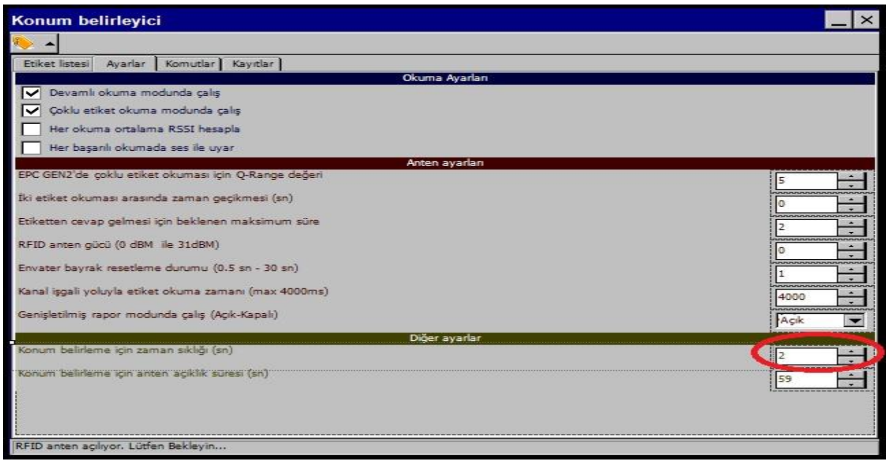


Fig. 4 Image of the room in which Model 3 has been installed

3.3 Developed Software

In the study, a software for the mobile RFID devices has also been developed. Geographical proximity approach has been used in this program. In every two seconds, RFID reader receives data from the tags within the reading range. Receiving data period also can be set by the user in "Configuration" tab (Figure 5).



Okuma Ayarları

☑ Devamlı okuma modunda çalış
☑ Çoklu etiket okuma modunda çalış
□ Her okuma ortalama RSSI hesapla
□ Her başarılı okumada ses ile uyar

Anten ayarları

EPC GEN2'de çoklu etiket okuması için Q-Range değeri	5
İki etiket okuması arasında zaman geçikmesi (sn)	0
Etiketten cevap gelmesi için beklenen maksimum süre	2
RFID anten gücü (0 dBM ile 31dBM)	0
Envater bayrak resetleme durumu (0.5 sn - 30 sn)	1
Kanal işgali yoluyla etiket okuma zamanı (max 4000ms)	4000
Genişletilmiş rapor modunda çalış (Açık-Kapalı)	Açık

Diğer ayarlar

Konum belirleme için zaman sıklığı (sn)	2
Konum belirleme için anten açıklık süresi (sn)	59

RFID anten açılıyor. Lütfen Bekleyin...

Fig. 5 Configuration (Ayarlar) Tab

At the end of every period, the tag which has the strongest signal is determined by the software based on collected data. Then the predefined coordinates of this tag are sent to the user. The software has been tested in three different design models as mentioned above.

4 Experimental Results

4.1 Model 1: Placement of Tags as a Straight Line

In the first design model, the height of ceiling is 320 cm and the length of corridor is approximately 30 m (Figure 6). The program has been tested in the test area including corridor and classroom and results given below has been obtained. In the test phase, each point with installed tags are considered as a node. The position of a person is accepted equal to the position of a tag (node) with the strongest signal received. The node number is passed to the user as the position data. Node numbers are set through the corridor from 1 to 31 incremented by one and from 32 to 35 in the classroom. Distance between nodes installed through the corridor is 1 meter.

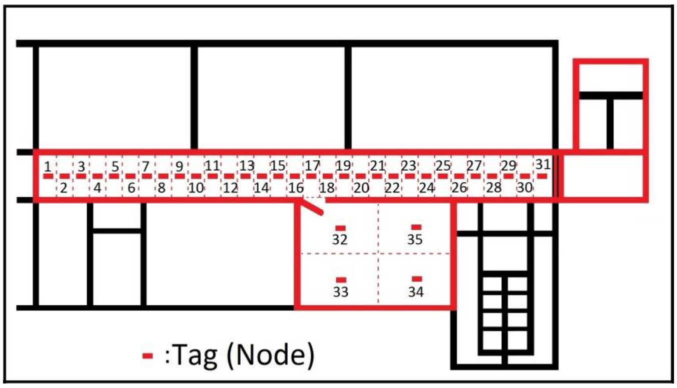


Fig. 6 The points of tags which are attached in study area (shown on the floor plan)

Different results have been obtained in cases such as the carrier person is stable or in motion. Firstly, the case in which the carrier person is stable in a point has been tested. For this purpose, we have waited in the point 2 for 1 minute. During this time, direction of the reader's antenna has been set towards the nodes with increasing node number through the corridor. Time period for reading has been set to be 2 seconds. In other words, position estimation is repeated every 2 seconds.

The position of the person is accepted as one of three points which have numbers 3, 4 and 5 just after point 2 and one of these points is estimated as the person's position. It is thought that direction of the antenna affects the quality of signal and results have been obtained like this. Another factor that affects the signal is thought to be human body. A person who blocks the signal by standing between reader and tag makes the signal quality worse.

While the person is stable, the position accuracy has been obtained between 1 m and 3 m depending on the direction of reader's antenna (ie horizontal or vertical) (Table 3).

Table 3 Positioning test results for Model 1 while the person is stable

Error Value (meters)	Explanation	Results	Percentage (%)
3	Estimated position is in front of the actual position 3 m.	5	16,66
2	Estimated position is in front of the actual position 2 m.	13	43,33
1	Estimated position is in front of the actual position 1 m.	12	40
0	Estimated position is equal to actual position.	0	0
Total		30	

While in motion, the test has been performed in rooms and corridors during four minutes in average walking speed (aprx 6 m/s). During this time, direction of the reader's antenna has been set towards the direction of the person's motion. Time period for reading has been set to be 2 seconds same as previous test. There has been taken estimation data for 112 points during the test of the program. Error rates for these measurements are given in Table 4.

Table 4 Positioning test results for Model 1 while in motion

Error Value (meters)	Explanation	Results	Percentage (%)
-6	Estimated position is behind the actual position 6 meters.	1	0,89
-5	Estimated position is behind the actual position 5 meters.	3	2,68
-4	Estimated position is behind the actual position 4 meters.	0	0
-3	Estimated position is behind the actual position 3 meters.	4	3,57
-2	Estimated position is behind the actual position 2 meters.	10	8,93
-1	Estimated position is behind the actual position 1 meter.	12	10,71
0	Estimated position is equal to actual position.	51	45,54
1	Estimated position is in front of the actual position 1 m.	22	19,64
2	Estimated position is in front of the actual position 2 m.	9	8,04
Total		112	

As seen in Table 4, in the worst case, position estimation error has been obtained 6 meters. In 4 measurements out of 112 which approximately correspond to 3% of total, position estimation error has been observed between 4 and 6 meters. In the case that position estimation error has been considered ± 1 meter, 85 measurements out of 112 which approximately correspond to 75% of total has provided the criteria. Beside, in the case that position estimation error has been considered ± 2 meters, 104 measurements out of 112 which approximately correspond to 93% of total has provided the criteria (Table 5). The position accuracy depends on the walking speed in this model.

Table 5 Position estimation errors and percentages

Position Estimation Error (meters)	Estimation Count	Percentage (%)
±1	85/112	75,89
±2	104/112	92,85

4.2 Model 2: Placement of tags to Side Walls Oppositely

In the second model, the tags have been attached up to the 220 cm height on the walls (Figure 2). The dimension of the room is 3 m x 5 m. Unlike Model 1, tags are installed on wall not on ceiling. The height of ceiling in Model has been set 320 cm. In Model 2, height of tags attached on wall is 220 cm. In this model, distance between reader and tag is reduced 1 meter compared to previous model. It is aimed to improve accuracy of RSSI by reducing the distance. On the other hand, there have been performed tests in two different periods. Period with 2 seconds which has been applied for Model 1 has also been applied to Model 2. Beside, 1 second period has also been tested.

We have used the room shown in Figure 7 for the test with motion. Tags attached on the same wall have been labeled with the same letter. Tags have been given consecutive numbers. There has been attached totally 12 passive tags in the room. Each point with attached tags is considered as a node. The position of a person is accepted equal to the position of a tag (node) with the strongest signal received. In this model, an area on the room plan is shown to the user for indicating position instead of passing node number to user. This process is performed by the "Map" tab in the program. The area in which user can get signal from tag labeled with B5 is shown in the program interface as seen in Figure 8. Size of an area is set to be 1,5 m x 1m according to the position of a tag or 1,5m x 0,5 m for corners of the room.

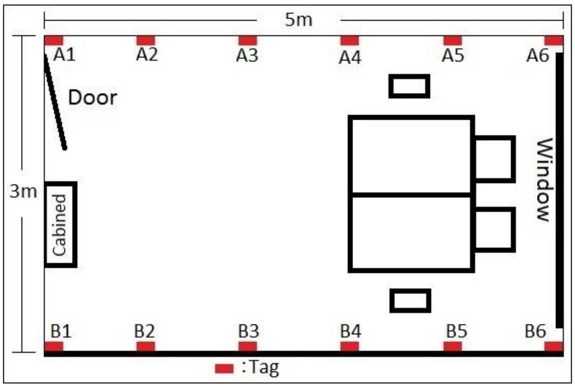


Fig. 7 Model 2

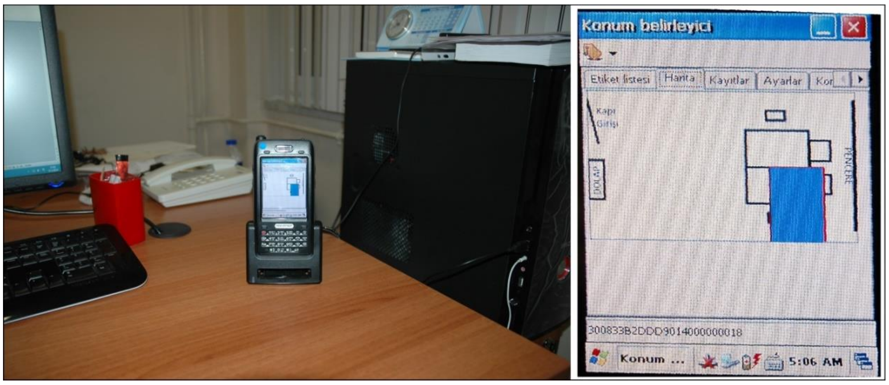


Fig. 8 The program was running on mobile device

Totally 106 position data have been obtained in the test with period set to be 1 second. In the other test with period set to be 2 seconds there have been obtained 110 position data. Error rates for these measurements are given in Table 6.

Table 6 Positioning test results for Model 2

Error Value (meters)	Explanation	1 Second Period		2 Seconds Period	
		Results	Percentage (%)	Results	Percentage (%)
±3	The distance between the estimated position and the actual position is 3 meters.	7	6,6	4	3,67
±2	The distance between the estimated position and the actual position is 2 meters.	8	7,55	0	0
±1	The distance between the es-	26	24,52	26	23,64

	timated position and the actual position is 1 meters.				
0	Estimated position is equal to actual position.	65	61,32	80	72,72
Total		106		110	

As seen in Table 6, while the rate of estimated positions which equal to actual positions (without error) has been about 61% in the test with 1 second period, it has been obtained as 73% for the test with 2 seconds period. In the case that position estimation error has been considered ± 1 meter in test with 1 second period, 91 measurements out of 106 which approximately correspond to 86% of total has provided the criteria. Beside, with the same position error consideration in test with 2 second period, 106 measurements out of 110 which approximately correspond to 96% of total has provided the criteria (Table 7).

Table 7 Position accuracy and percentage

Position Estimation Error (meters)	**1 Second Period**		**2 Seconds Period**	
	Estimation Count	**Percentage (%)**	**Estimation Count**	**Percentage (%)**
±1	91/106	85,85	106/110	96,36
±2	99/106	93,40	106/110	96,36

4.3 Model 3: Placement of Tags to the Center of 1 m Dimensional Quadratic Cells

In the third model, the tags have been attached to the ceiling of a room which has 4 m x 4 m dimension (Figure 9). They have been placed to the centers of 1 m dimensional quadratic cells. The height of the ceiling is 250 cm. In this model, room has been divided into 1 square meter cells. The estimated position is shown to the user as an area just like in Model 2. Size of each area is 1m x1m. Totally there have been used 16 passive tags and constructed 16 areas. The program has been tested only in 2 seconds period.

Room shown in Figure 9 has been used in the test with motion of this model. Tags have been labeled as shown in Figure 9. Totally 90 position estimation data have been obtained. Error rates for these measurements have been given in Table 8.

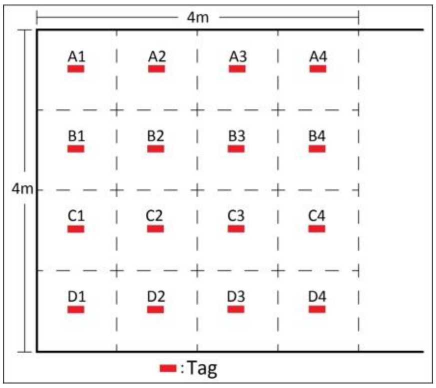


Fig. 9 Model 3

Table 8 Positioning test results for Model 3

Error Value (meters)	Explanation	2 Seconds Period	
		Results	**Percentage (%)**
±2	The distance between the estimated position and the actual position is 2 meters.	3	3,33
±1	The distance between the estimated position and the actual position is 1 meters.	19	21,11
0	Estimated position is equal to actual position.	68	75,56
Total		90	

In the test of Model 3 with motion, rate of estimated positions which equal to actual positions has been about 76%. In the case that position estimation error has been considered ± 1 meter, 87 measurements out of 90 which approximately correspond to 97% of total has provided the criteria. In the worst case, position estimation error has been obtained 2 meters. In 3 measurements out of 90 which approximately correspond to 3% of total, position estimation error has been observed 2 meters (Table 8).

In the second and third model, if the user stands in a fixed point and if exceptions are ignored, the position is determined correctly.

5 Conclusions

In this study, an RFID based indoor positioning system has been introduced. Usages and performances of various types of readers and tags have been investigated and the best combination has been selected. In general, in case of the tags were directly attached to the wall or ceiling, the interaction between tags and readers has been negatively affected and the performance of them has been decreased. To resolve this problem, the tags have been separated from the surfaces by using 20 cm long styrofoam material.

In the experiments of this study, geographical proximity approach has been used. As the results of tests performed on three different model proposed for indoor positioning, it has been shown that best rate for position estimations without error have been obtained from third model with the rate of approximately 76% and in the worst case, position estimation error has been obtained 2 meters.

In the study it has been observed that position estimation error is affected by several cases. One of them is the attachment frequency of tags. Increase of frequency reduces the error. Another factor is thought to be that direction of the antenna affects the quality of signal. The last factor that affects the signal is thought to be human body. A person who blocks the signal by standing between reader and tag makes the signal quality worse.

5 Acknowledgements

This study was supported by TUBITAK - The Scientific and Technological Research Council of Turkey (Project No: 112Y050) research grant. We are indebted for its financial support.

References

Lemieux, N., 2009, “Reading course on Indoor Location Systems” http://www.csd.uwo.ca/~nplemieu/readingCourse/ IndoorLocationReadingCourse.html (19 Feb. 2013)

Candy, J., 2008, “Indoor location techniques for mobile location-based indoor GIS applications” http://giswww1.bcit.ca/ georanger/find-indoor-loc.htm (1 Apr. 2013)

Khong, G., White, S., 2005, Moving right along: Using RFID for Collection Management at the Parliamentary Library, Information-Online 12th Exhibition & Conference, Sydney, 1-12.

Finkenzeller, K., 2003, RFID Handbook:Fundamentals and Applications in Contactless Smart Cards and Identification 2nd ed., Rachel Waddington, John Wiley & Sons, Ltd, West Sussex, 1-393.

Dziadak K., Kumar B., Sommerville J., 2009, Model for the 3D location of buried assets based on RFID technology, J.Comp. Civil Eng., pp. 148–159

Manish, B., Shahram, M., 2005, RFID Field Guide: Deploying Radio Frequency Identification Systems, Prentice Hall PTR, USA, 24-29.

L. Wang, Y. Lin, P.H. Lin, 2007, Dynamic mobile RFID-based supply chain control and management system in construction, Adv. Eng. Inform. 21, 377–390.

3D Indoor Navigation Prototype For Smartphones

Yasin Ortakci[1], Ismail Rakip Karas[2], Alias Abdul Rahman[3]

[1]Department of Computer Engineering, Karabuk University, Karabuk, Turkey
yasinortakci@karabuk.edu.tr

[2]Department of Computer Engineering, Karabuk University, Karabuk, Turkey
ismail.karas@karabuk.edu.tr

[3]Department of Geoinformatics, Universiti Teknologi Malaysia, Johor, Malaysia
alias.fksg@gmail.com

Abstract Nowadays, there are a lot of multi-storey, complex and huge buildings in the cities especially in metropolises. These buildings are almost like a small city with their tens of floors, hundreds of corridors and rooms and passages. Sometimes people lost their way in these huge buildings. Due to the size and complexity of these buildings, people need guidance to find their way to the destination in these buildings. In this study, a mobile application has been developed to visualize a pedestrian's indoor location as 3D in his/her smartphone. This mobile application has the characteristics of a prototype for indoor navigation systems. While the pedestrian is walking on his/her way, the smartphone will guide the pedestrian with the photos of indoor environment on the route, arrow marks and text information. As a future plan, an RFID (Radio-Frequency Identification) technology can be integrated to the system to detect the location of the pedestrian during his/her tour in the building. By this way, the system will navigate the users more accurately as a real-time navigation system.

Keywords: Indoor Navigation, Smartphone, Android, RFID

1 Introduction

GPS (Global Positioning System) has an extensive usage in outdoor areas for navigation. However, GPS cannot be used in indoor areas properly due to the weakness of its signals in the buildings. Nowadays, the dimensions of the buildings and the indoor environments are enlarged extremely. Therefore, it is difficult to find the way in these types of buildings due to the size and the complexity. For instance, when people visit multi-storey and complex buildings such as trade centers, shopping malls, skyscrapers, airports, hospitals and universities, they need a navigation system to find their way to the destination . Although there are a lot of studies about indoor navigation on 2D maps, pedestrians need more realistic navigation system that routes pedestrians in buildings as 3D (Musliman et al. 2009).

In this study, an indoor navigation system for smartphones is developed to visualize the pedestrians' locations in the building as 3D in their smartphones. This mobile application has the characteristics of a prototype for indoor navigation systems. It uses client server architecture, the client side is represented by smartphone and server side is represented by a web server which holds a spatial database. In the application, a user only selects the current location and destination to start the navigation. Then, the navigation system finds the shortest path and guides the user in the building. Within the scope of this study, some specific indoor areas are called nodes. The distance between nodes can differ. Indoor environments between all nodes are taken photo in four-meters intervals. A photo-frame library is established with these indoor photos. The spatial database in the server relates the each photo in the photo frame library to the nodes on the building networks. Thus, a link is established between a specific indoor environment and its photo. Client (smartphone) connects to the server on a wireless network connection and sends the data of client's current location and destination. When the location information arrives to the server, a web service is activated to find the shortest path from the-current location to the destination. This web service finds the photos on the path and their spatial descriptions. They are sent back to the application on the smartphone using the wireless network connection again. While the pedestrian is walking on his/her way on the route, the smartphone will guide the pedestrian by displaying the photos of the indoor environment on the route and giving some extra information.

The paper is organized as follows. Section 2 referred some other indoor navigation system studies in the literature. Android mobile operating system is described in Section 3. Section 4 details the design of our proposed indoor navigation system for smartphone. The conclusion and the future plan are mentioned in Section 5 and 6, respectively.

2 Background

Candy (2007) developed an indoor location based GIS application that works on a GIS web server for mobile phones. The application creates a web page displaying floor plan of the building and draws a route line that users must follow. The users can connect to this web page via an HTML-supported mobile phone. The mobile phone must have RFID technology to determine user's location. The loaded web page to the mobile phone for navigation is a graphical interactive map. One of the disadvantage of the system is that the map of the floor plan must be loaded as a whole since the map is displayed in the HTML browser. Besides, there are some other difficulties like using HTML browser on a small screen size phone.

Hammadi et al. (2012) introduced an Android based 2D indoor navigation system to guide the visitors inside the widespread buildings in their study. The system utilizes the NFC (Near Field Communication) to detect the location of users. According to their system, the location of user is detected with NFC tags. User must specify a destination point to start the navigation. The disadvantage of the system is lack of the interaction between the system and the user along the navigation process and the system has a poor visualization service. Another shortcoming of the system, the user must make the smartphone scan the NFC tag to detect his/her location since it does not detect the location of user automatically.

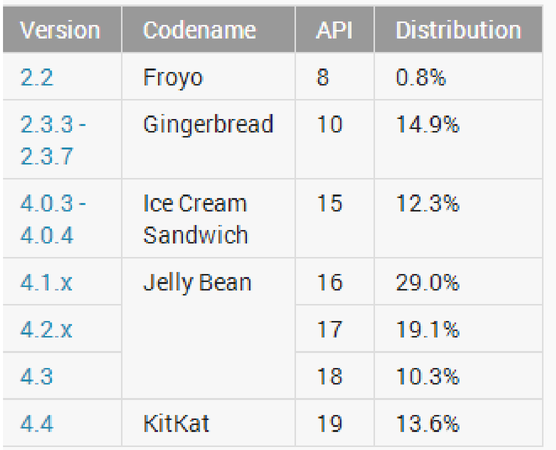
Cheung et al. (2006) described a Bluetooth-based indoor positioning system. They used stationary beacons and a Bluetooth-enabled mobile device. The system can detect the location at the scale of 2-3 meters areas, only capture movement trajectories of people in the building.

Pritt (2013) proposed an indoor location system for smartphones and tablets that make use of commonly available Wi-Fi networks. The proposed location system needs a calibration stage before the usage for the navigation. The user can track himself/herself in the floor plan displayed on the mobile device as 2D. The success rate of his location system is 97.5% in a suburban home and 100% in a shopping mall. However, the success rate can differ from device to device, since mobile devices have different sensitivities to Wi-Fi signals. Besides a real-time navigation module can be added to this system.

3 Android

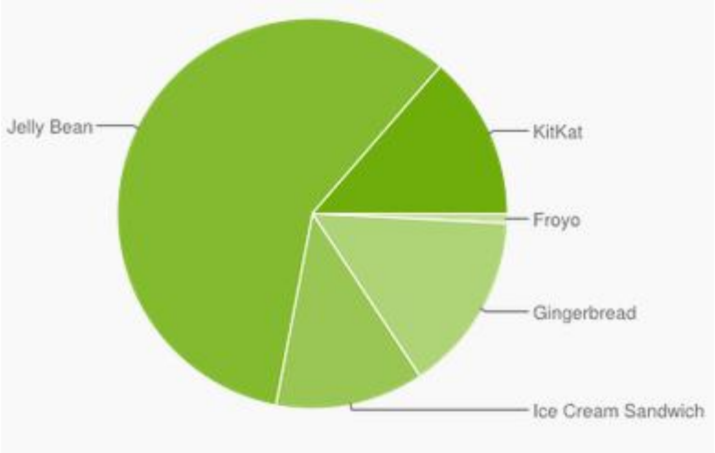
Android is an open source, mobile device and smartphone oriented mobile operating system developed by Google and Open Handset Alliance. Android mobile operating system is established upon Linux 2.6 kernel (Dimarzio 2008). Android has an extensive user group on the world. Android mobile devices are used by millions of people more than 190 countries all over the world. The number of Android applications, downloaded from Google Play, is over 1.5 billion. Google Play is the official Android application store managed by Google. Android is one of the most rapidly-developing mobile operating system with its huge user group and new specifications.

Android presents a lot of advantages to the developers as well as it presents to the users. Developers can publish their applications on Android devices to make the applications accessible by the millions of Android device users. Android is not only a mobile operating system, it is also a Software Development Kit (SDK) with its API (Application Program Interface) libraries, developer tools to build, test and debug applications. Developers can test their applications on the emulators supplied by Android SDK. The user interfaces in the applications can be designed using XML. Android provides a lot of facilities to the developers with its flexible structure. Android platforms and their usage rate is shown in Fig 1. Our proposed indoor navigation system can run on all of these Android platforms.



Version	Codename	API	Distribution
2.2	Froyo	8	0.8%
2.3.3 - 2.3.7	Gingerbread	10	14.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	12.3%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	19.1%
4.3		18	10.3%
4.4	KitKat	19	13.6%

(a)



(b)

Fig. 1 Usage rate of Android platforms (Android Developers, 4 June 2014)

Android offers opportunities of detecting the motion and the location with its location API. GPS (Global Positioning System) can be used to detect the location of Android devices. But generally, GPS is not sufficient for the indoor environment to detect the location.

4 System Design

Our navigation system has been carried out in three different steps. In the first step, the database of the navigation system was designed and this database was entered with a sample indoor environment data. Oracle Database 11g was preferred as a database management system due to its support of the spatial data (Atila et al 2013). The navigation system works according to client-server architecture. In the second step, smartphone application was developed as the client side application in the navigation system. The application was developed for smartphones and tablets that have Android operating system. Since the Android applications are generally developed based on Java programming language, Eclipse Interactive Development Environment is used as the development environment. In the third and last step, a script, which would work on server, was coded in PHP internet based programming language. This script takes arguments from Android application, finds the route for the users, transforms the route data (photos, location description) to JSON (JavaScript Object Notation) format and sends it back to the user via Android application. JSON was selected as a data transfer format since it is a lightweight data exchange format. Therefore, route data can be transferred from the server to the smartphone quickly.

The architecture of our navigation system is illustrated in Fig. 2 and these three steps of navigation system are detailed in the following parts.

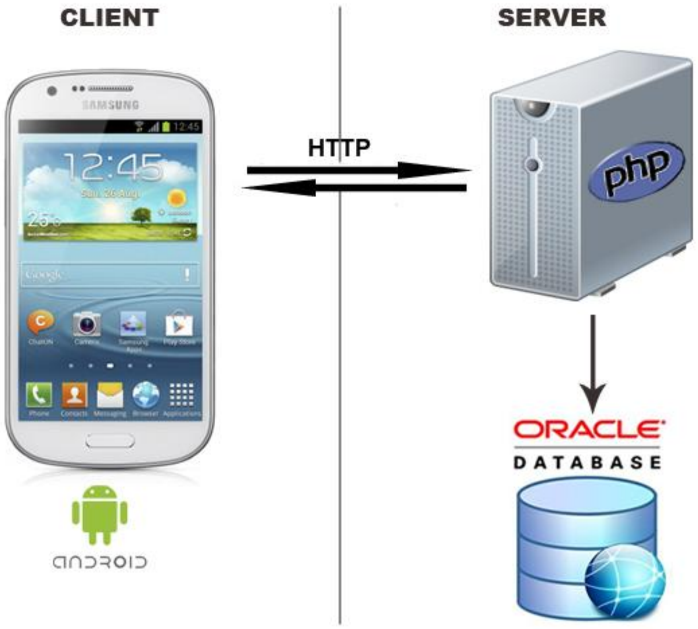
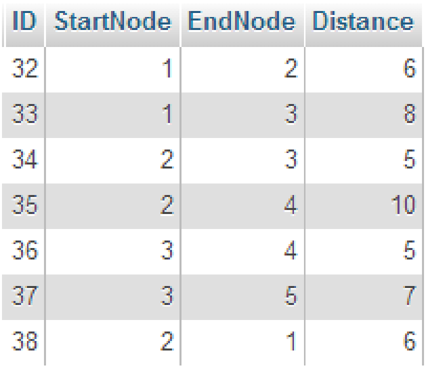


Fig. 2 Architecture of the navigation system

In the first step, a database called "tez" was created in Oracle Database 11g. The database has two tables related to one another. The first table, called "link", keeps the data of the links between the indoor nodes. The second table, called "sublink", is used for storing the indoor photos of each link. "link" table comprises of four fields called "ID", "StartNode", "EndNode" and "Distance", respectively. Each record in the "link" table keeps the link data between each node pairs and each record has a different ID number so "ID" field is the primary key of "link" table. In each record, "StartNode" field keeps the starting node of the link, "EndNode" keeps the end node of the link and "Distance" field keeps the distance of the links in meter. All of the fields in "link" table are integer. The following SQL code was used to create "link" table:

```
CREATE TABLE LINK(
    ID int not null auto increment,
    StartNode int not null,
    EndNode int not null,
    Distance float,
    PRIMARY KEY (ID)
)
```

In Fig 3, the record with 32 ID number keeps the information of a 6 meters link between node 1 and node 2. Besides, the inverse of this link is kept as a different record with 38 ID number.



ID	StartNode	EndNode	Distance
32	1	2	6
33	1	3	8
34	2	3	5
35	2	4	10
36	3	4	5
37	3	5	7
38	2	1	6

Fig 3. "Link" table

Another table, "sublink", keeps some information such as the spatial description, the indoor photos and the distance of each point, which has four-meters interval to one another, on the links. "sublink" table comprises of five fields called "ID", "LinkID", "Photo", "Distance" and "Description" respectively. "ID", "LinkID", "Distance" fields are integer and "Photo" and "Description" fields are varchar (a type of string definition). "ID" field is primary key and each indoor photo is kept in the separate records in the "sublink" table. "Photo " field keeps the address path of related picture in the server. "Description" field keeps the spatial description of the indoor environment for each photo. "Distance" field keeps the interval between related point and the starting point of link. "LinkID" field matches the points to the link in the "link" table. Therefore, "LinkID" field indicates which point belongs to which link. Besides, "LinkID" field is foreign key of the "sublink" table and supplies the conjunction between "link" and "sublink" table (Fig. 4). The following SQL code was used for creating "sublink" table:

```
CREATE TABLE SUBLINK(
    ID int not null auto_increment,
    LinkID int not null,
    photo varchar(255) not null,
    Distance int,
    Description varchar(255) not null,
    PRIMARY KEY (ID),
    CONSTRAINT fk_SubLink FOREIGN KEY (LinkID)REFERENCES Link(ID)
)
```

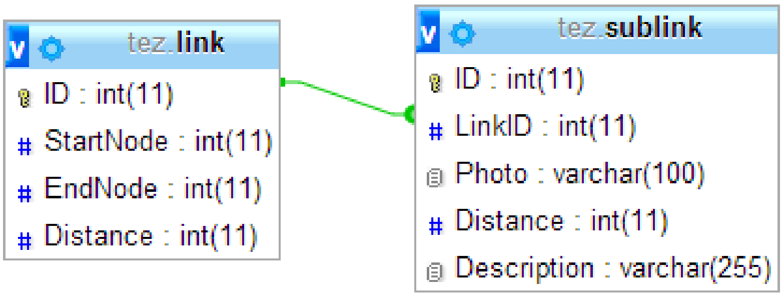
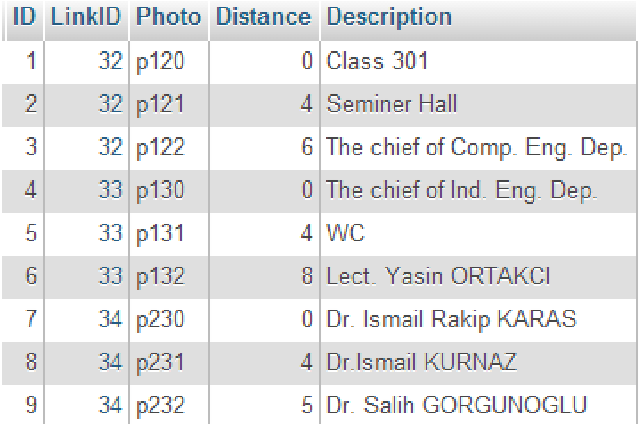


Fig 4. ER diagram of "tez" database

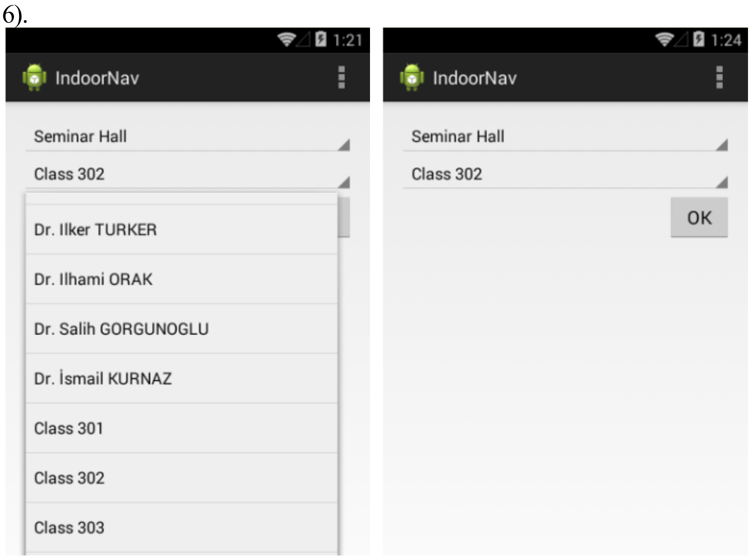
In this study, the environment photos among connected node pairs are taken in each four meters. These photos are stored in the "sublink" table in the separate records. In Fig. 5, 33 "ID" numbered record in "link" table connects node 1 and node 3 and the link has a distance of eight meters. The photos of the environment at 0th,4th and 8th meters of this link are taken respectively. Each photo is added in to the "sublink" table as a separate record with environmental description and distance to the starting node information of the point. The number of record in the "sublink" table belongs to a specific link in the "link" table differs according to the link length. For example, the length of the 33 ID-numbered link is 8 meters, so there are three records in "sublink" table for this link. These three records keep the information of the points at 0th meter, 4th meter and 8th meter of the link, respectively.



ID	LinkID	Photo	Distance	Description
1	32	p120	0	Class 301
2	32	p121	4	Seminer Hall
3	32	p122	6	The chief of Comp. Eng. Dep.
4	33	p130	0	The chief of Ind. Eng. Dep.
5	33	p131	4	WC
6	33	p132	8	Lect. Yasin ORTAKCI
7	34	p230	0	Dr. Ismail Rakip KARAS
8	34	p231	4	Dr.Ismail KURNAZ
9	34	p232	5	Dr. Salih GORGUNOGLU

Fig 5. "sublink" table

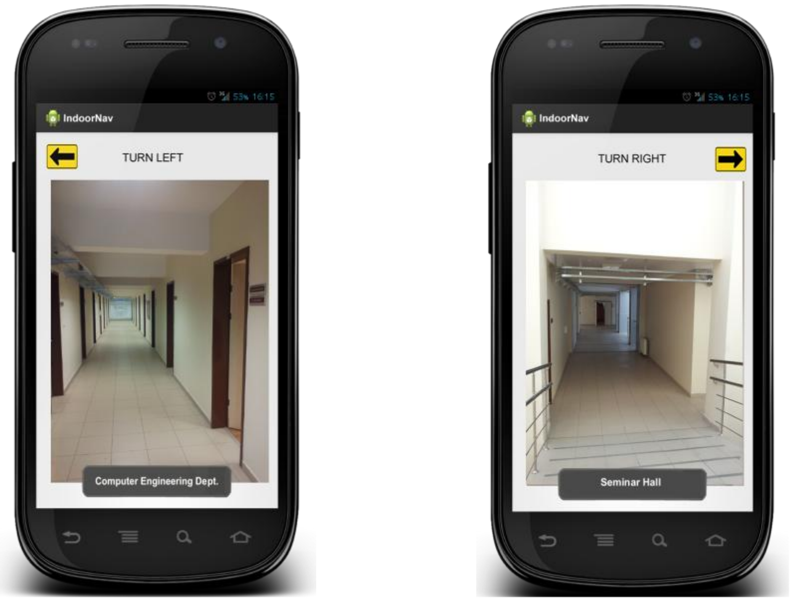
In the second step of the implementation, an Android application was developed for smartphones to supply interaction between the user and the navigation system. This mobile application has two different activity called Intro.java and Navigation.java, respectively. In Intro activity, user enters his/her current location and then selects the destination point. The graphical user interface of Intro activity has a dropdown list to select current location, a dropdown list to select the destination and an OK button to send this data to navigation module on the server (Fig. 6).



(a) (b)

Fig. 6 Intro activity screenshots

In the graphical user interface of Navigation activity (Fig. 7), there is a text view tool at the upper side of the screen to direct the users with text information. Besides, there are arrow marks on the screen to illustrate navigation instruction. There is an image view tool in the middle of the screen. This image view is used to display the indoor photos. The photos are provided from "Photo" field in the "sublink" table. A pop up message box, which describes the indoor environment, appears at the bottom of the screen while the indoor photo is being loaded. Content of pop up message is provided from "Description" field in "sublink" table.



(a) (b)

Fig. 7 Navigation activity screenshots

User's current location and the destination data are sent to the server script using HTTP protocols and IP numbers of the server. The script finds the shortest path between the starting point and the destination and sends the navigation data to Navigation activity on the smartphone. The navigation data includes a group of photos and their spatial descriptions. The navigation data may not be retrieved from server instantly because of insufficient internet connection speed in the wireless networks. Thus, AsyncTask class was used not to make wait users on the blank screen along retrieving the data from the server. AsyncTask class is an Android library class that welcomes user with a loading message while the navigation data is being loaded at the background.

In the third step of the implementation, a PHP script, working on a Apache web server, called "getData.php" was developed. First, the script connects to the "tez" database locally and finds the shortest path to the destination. The shortest path includes a group of nodes which will be visited in order. The links are found from "link" table by checking the start and the end node pairs on the path. Then, the photos and their spatial descriptions are retrieved from "sublink" table according to their LinkIDs and they are transformed to JSON format before being sent to Android application. An example JSON data is given below:

```
{"post":
   [
       {"description":"Class 301","photo":"p120"},
       {"description":"Seminer Hall","photo":"p121"},
       {"description":"Chief of Department","photo":"p122"},
       {"description":"Dr. Ismail Rakip KARAS","photo":"p230"},
       {"description":"Class 311","photo":"p231"},
       {"description":"Stairs","photo":"p232"},
       {"description":"WC","photo":"p310"},
       {"description":"Class 211","photo":"p311"},
       {"description":"Class 213","photo":"p312"}
   ],
   "success":1,
   "message":"Path is found successfully"
}
```

6 Conclusions

In the content of this study, The Outbuilding of Engineering Faculty in Karabuk University was selected as an application area. The photos of indoor environments of this building were taken each four meters. The size of photos was reduced in order to be processed efficiently. Then a photo-frame library is established on the server with these photos. The "photo" field of "sublink" table keeps the path of the photos in the library.

Our smartphone oriented navigation system worked successfully in the Outbuilding of Engineering Faculty. First, user connects to the internet on the Wi-Fi internet connection in the building and selects his/her current location and the destination in the application on the smartphone. Then navigation system finds the shortest path including the indoor nodes and points. Then it downloads the indoor photos on the path to the smartphone. Each photo is displayed on the screen in each four seconds. It is assumed for a pedestrian to walk four meters distance in a four-second time period. Using JSON format make data stream of photos quicker. Thus, user will not have to wait for the photos to be loaded to the screen of the smartphone while he/she was walking in his/her way.

The results show our prototype application can be a useful navigation tool for everyone who has a smartphone in the complex, multi-storey and huge building.

7 Future Plan

In this study, the user selects a starting node and a destination node and then runs the application just before starting to walk to the destination. Along the user's walk in the building, navigation system does not detect the location of the user automatically. Therefore it assumes that a pedestrian can walk four-meters distance in a four-seconds time period, and loads the photos of the points on the shortest path orderly in each four seconds. If the user does not obey the navigation instructions of the system and wander from the system-defined route, our application loses the control and cannot create a new route for user.

To overcome this disadvantage, an RFID based location determination technology can be integrated to the navigation system. The indoor environment of the building is equipped with the RFID tags in each point and the user handles an RFID reader. While the user is walking, the RFID reader will read the closest RFID tag and detect the location of the user as 3D (x, y, z) coordinates.

Another step in the location determination is to send 3D coordinate to the navigation system. This process can be handled in three different ways. First option, the RFID reader connects to the smart phone with Wi-Fi peer to peer connection. Second option, the RFID reader connects to the smart phone with Bluetooth connection. 3D user coordinate is sent to the server via smart phone in both ways. Third option, the RFID reader connects to the server directly and sends the coordinate to the system. All of these options can be feasible in our navigation system.

As a result, the location of users can be determined automatically adding the RFID technology to our navigation system. Thus, the navigation will work exactly real time. Even if the user does not obey navigation instructions and wander from the system-defined route, the RFID reader will detect the new location and produce a new route for the users. The users can easily find their way in the building without the need for any guidance. By this way, our navigation system can be applicable for all types of indoor navigation systems, even in emergencies such as earthquakes, fires etc. Furthermore, adding voice navigation to the system will increase usability.

Acknowledgements

This study was supported by TUBITAK - The Scientific and Technological Research Council of Turkey (Project No: 112Y050) research grant. We are indebted for its financial support.

References

Android Developers (2014) About. http://developer.android.com. Accessed 4 June 2014

Atila U, Karas IR, Abdul-Rahman A (2013) Integration of CityGML and Oracle Spatial for implementing 3D network analysis solutions and routing simulation within 3D-GIS environment. Geo-Spatial Information Science 16(4), 221-237

Candy J (2007) A Mobile Indoor Location-Based GIS Application. In: 5th International Symposium on Mobile Mapping Technologies (MMT07), 29-31 May 2007, Padua, Italy

Cheung K, Intille S, and Larson K (2006) An Inexpensive Bluetooth-Based Indoor Positioning Hack. In: Proceedings of UbiComp, 17-21 September 2006, CA, USA

DiMarzio JF (2008) Android A Programmer's Guide. Mc Graw Hill Hammadi O, Hebsi A, Zemerly M J et al. (2012) Indoor Localization and Guidance using Portable Smartphones. In: Web Intelligence and Intelligent Agent Technology, 4-7 December 2012, Macau, China, pp 337-341

Musliman IA, Alizadehashrafi B, Chen TK et al.(2009) Modeling Visibility through Visual Landmarks in 3D Navigation using Geo-DBMS. In: 3D Geoinfo Workshop, 4-5November 2009, Belgium

Pritt N (2013) Indoor location with Wi-Fi fingerprinting. In: Applied Imagery Pattern Recognition Workshop: Sensing for Control and Augmentation IEEE (AIPR), 23-25 October 2013, Washington, DC, pp 1-8

Modelling historic site as parametric model. Application to the Engelbourg castle - Thann, France

Mathieu Koehl*, Jimmy Fedczyszyn

Laboratoire ICube – UMR 7357, INSA de Strasbourg, France
(mathieu.koehl, jimmy.fedczyszyn)@insa-strasbourg.fr

Abstract. Within the framework of the valuation of the ruins of a castle destroyed in XVIIth Century, the project presented here shows diverse approaches of modelling and of 3D reconstructions of the castle shortly before its destruction. While surveys by topographic methods and by 3D TLS have been made for the modelling of the current state, several types of 3D reconstructions have been suggested: a first one based on purely geometrical aspects extracted from historic archive, a second one guided by the hypotheses of the archaeologists and the medieval context still existing in the surrounding region. This second scientific reconstruction was drawn using the *Trimble Sketchup* software with its inherent difficulties of modelling. Finally, the purpose of this paper is to show how the ESRI's *CityEngine* platform, initially planned for the construction of virtual large cities can be used on an only small site by using the shape grammar and the possibilities of parametric modelling to propose finally a new parametric, modifiable and adjustable reconstruction according to the hypotheses of the archaeologists.

Keywords: 3D modelling; Procedural modelling; Parametric model; Construction rules; Cultural heritage.

1 Introduction

In a project of valuation of the ruins of a castle destroyed in the XVIIth Century, several 3D modelling have been developed. Because of the ruined state of the current castle, this modelling can be considered as archaeological reconstruction. Several methods of reconstruction have been used. Then it has been a question of verifying the possibilities of a parametric modelling software, in this particular case ESRI's *CityEngine* platform within the background of an unconventional approach to this type of modelling. Indeed, *CityEngine* is rather intended for modelling of whole, small-scale virtual cities. This project allows to verify what could be its benefits in the context of an archaeological reconstruction, based on hypotheses which can be brought to vary strongly and often according to the improvement of the knowledge on the site and new hypotheses proposed by the archaeologists.

1.1 The studied area

The Castle of Engelbourg, object of this study, has been built at the beginning of the XIIIth Century, on the top of the Schlossberg. It is situated on the territory of the municipality of Thann (France), in the crossroads of Alsace and Lorraine, and dominates the outlet of the valley of Thur. Because of its strategic position this castle has been systematically destroyed during the XVIIth Century, and Louis XIV finished his fate by ordering his demolition in 1673, after the closure of Alsace with France. In this area there are nowadays only few vestiges, of which a section of tower, about 7m of diameter and 4m of wide put on its slice, unique characteristic in the regional castral landscape. It is visible from the valley and, baptized "the Eye of the witch", and became a key attraction of the municipality. The site, which extends over approximately one hectare, makes for several years the object of numerous archaeological studies and is in the center of a project of valuation of the vestiges of the Engelbourg castle today. This project joins more generally a program of cultural heritage, artistic and touristic development of the site, on the initiative of the city of Thann. The archaeological study is realized by the PAIR (Rhinish Interdepartmental Archaeological Pole). It is thus with the cooperation of all these actors that joins the team of the INSA of Strasbourg, the laboratory Icube / trio to participate in the elaboration of this program. It is indeed question, among the numerous planned works, to realize a 3D model of the site in its current state. In other words, a virtual model "such as seized", exploitable as well from a cultural and tourist point of view as a scientist who will serve as basis for the works of virtual reconstruction. The team of the INSA had had in charge the realization of this model, the acquisition of the data until the delivery of the virtual model, thanks to instruments like 3D TLS for data acquisition and CAD/modelling software tools. A student in Architecture, [1] realized a first reconstruction.

1.2 Base of 3D modelling

Acquisitions and modelling of the site in its current state has been realized between 2009 and 2013. They have been described in the various works of Master Degrees of [2], [3] or still [4]. Acquisitions have been made from 3D TLS surveys completed by the classic topography and the terrestrial photogrammetry. The modelling has been realized by using the *Trimble Sketchup* software. As described in [2], the various historic archaeological works have been also merged into this 3D model. Figure 1 shows a view of the surveyed site as in current state in the form of a global point cloud (acquisition 2009-2010) and Figure 2 the 3D model which has been reconstructed.

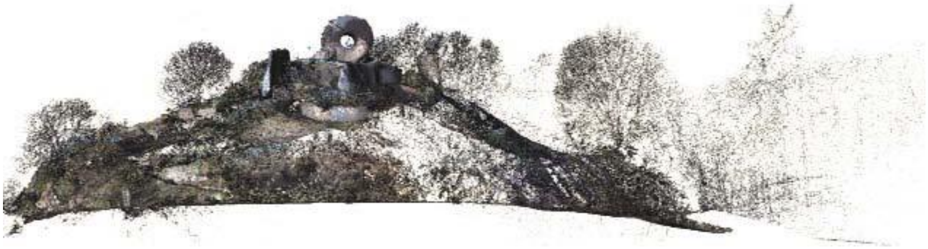


Fig. 1. Point cloud from TLS data acquisition (2009-2010)

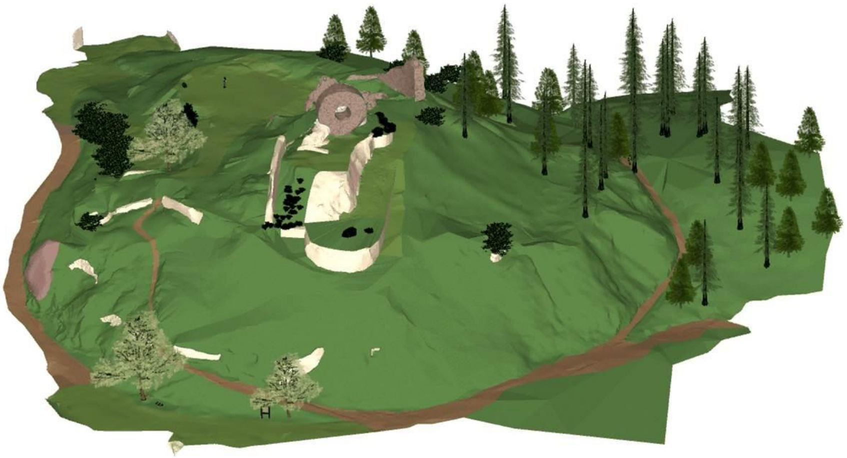


Fig. 2. Model of the site in *Trimble Sketchup*

These various works are explained, described and accessible on the web site [5].

2 First approach of archaeological reconstruction: geometrical approach in *Trimble Sketchup*

The reconstruction of the castle of Engelbourg was established from:
- the plan (1657) representing the castle in its last documented state (Fig.3),
- the detailed legend of the plan of 1657,
- the frontispiece representing the castle by Schenck (1628) (Fig. 4),

- Archaeological data recorded by Ehretsmann [6] [7] and Koch [8] [9],
- Comparisons and links with the corpus of castles and fortified cities.

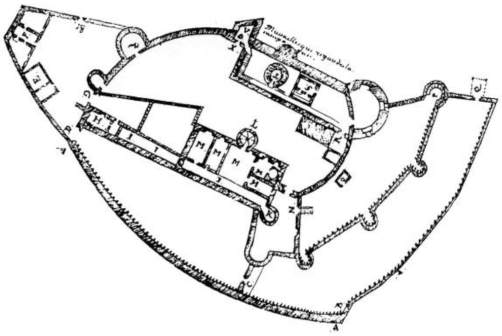


Fig. 3. Plan from 1657 – Last state known for the castle – Archives Départementales du Haut-Rhin

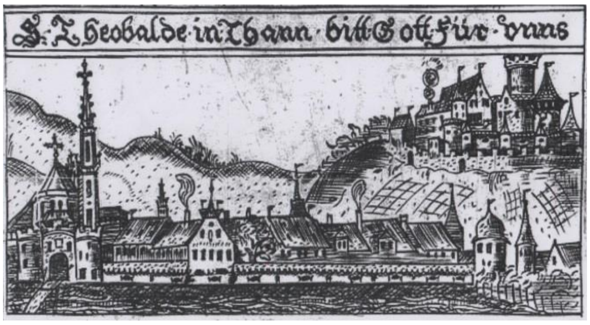


Fig. 4. Frontispiece by Schenck (1628) [10] – The castle is represented at upper right

The difficulties of use of the *Trimble Sketchup* software to carry this modelling to a successful conclusion were relatively important and did not allow to lead this modelling to a sufficient level of detail. The encountered difficulties were the following ones:

- Creation of a 3D model from a modelling of the current site not structured in the direction of the reconstruction,
- Very disrupted topography of the site,
- Too basic architectural knowledge concerning the castle,
- Strong presence of vegetation on the site and numerous undertaken restorations not allowing to find out the original ground levels,
- Difficulty of understanding of vestiges on the existing 3D model which proposed no real structuring of the various elements of the site.

Nevertheless the conclusions which have been carried out are the following ones:

The project allowed to provide a first overview of the castle, and so that a better understanding of the organization of the building. Reflections on the various construction phases and on the original height levels have been also possible to be started. (Fig. 5 visible on [5])

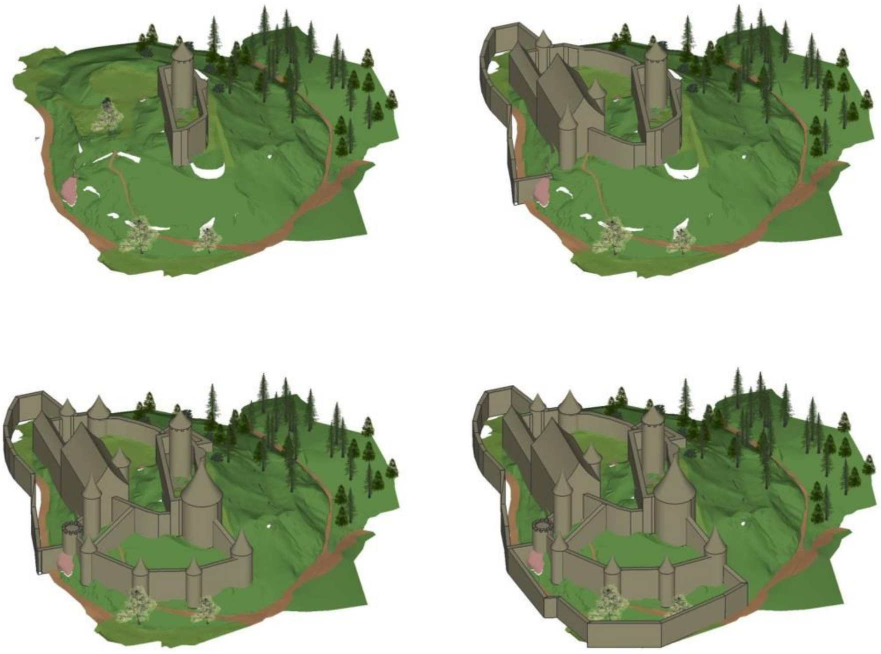


Fig. 5. Step by step modelling of the Engelbourg

To perfect the knowledge of the site, it seems at the moment necessary to study the former height levels and the layers of destructions as well as the former archives concerning the various works of restoration.

To improve the phase of reconstruction it has been necessary to realize a new 3D model with the former height levels as basis for the modelling of the castle. This new proposal is described later.

A new basis ground data as well as another type of software package have been used to explore this second approach.

3 Second approach of archaeological reconstruction: parametric modelling with *ESRI's CityEngine*

[11] invented a number of ground breaking techniques for the procedural modelling of 3D architectural contents which make up the foundation of *CityEngine* today. *CityEngine* has been presented for the first time outside of the research community in 2001 and has been released in 2008. Then it was acquired in 2011 by ESRI, one of the most powerful and well-known GIS Company. The main usefulness and highlight of *CityEngine* is to supply 3D contents from GIS data and also generate scenes and 3D models. Once the database loaded in *CityEngine* such as DTM, roads axes, footprints of buildings, vegetation, etc., the application of procedural rules can help to build the 3D entities. To better understand the process of generating 3D models, construction rules of 3D entities have been analysed and assigned in order to model a historic castle, the Engelbourg, Thann, France. [12] presented a very complete overview of the potential use of *CityEngine* in archaeological projects.

3.1 Timeline project creation

To start modelling with *CityEngine*, some initial real or virtual data have been required. Therefore a DTM, and in our case the map of the old demolished castle, textures and an idea of what the castle looked like (see 4, first approach) have been mandatory.

Project creation.

To create a scene, as a first step and to ensure a good data organization, it is highly recommended to create a folder to the desired location. The next step is to create a project. Then it is possible to create the scene associated to a coordinate system.

Digital Terrain Model (DTM).

A model of the ruins has already been constructed by TLS survey and CAD modelling as shown before. The terrain model and 3D model of the ruins are available in diverse 3D formats. The ruins of the castle were at first removed to keep only the terrain: the DTM (Fig. 6).

To use a DTM in the scene, it must be imported as a raster. After exporting the terrain in a 3D DXF-file, it was imported into ArcGIS/ArcMap where a raster DTM was generated by using a 3D Analysis Tool such as IDW interpolation. As result, we have obtained a greyscale elevation raster with values between 0 and 255.

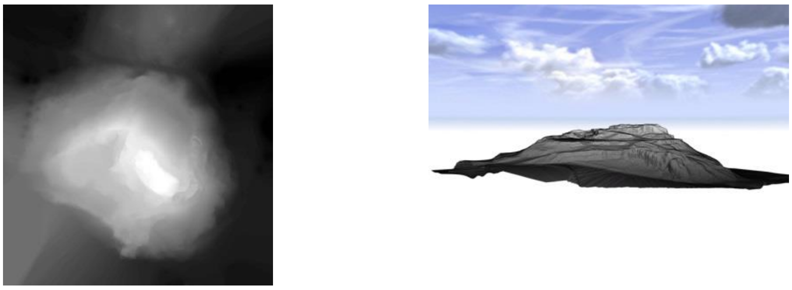


Fig. 6. DTM in raster format and integration in 3D scene

CityEngine allows to infer the frame like the next statement:

```
attr elevation = map_01 ( brightness , 411.33 , 443.25 ) + elevationDelta
```

This interpretation of this statement is that the black colour corresponds to a 411.33 meters elevation and the white colour to a 443.25 meters elevation. The elevation is then interpreted as Z values and as geometry in the 3D scene. While importing, it's possible to assign a texture to the DTM. But, in our case, without high resolution aerial photography, we have used here a generic green colour texture suggestive of grass.

Map of the Historic Site - Base.

The most important step on data baseline is setting up the base, the plan, the map. *CityEngine* is able to manage numerous well-used drawing formats such as DXF, DAE, OSM, KML, KMZ, SHP or OBJ.
Importing a DXF-file seems to be the most obvious solution (Fig.7).

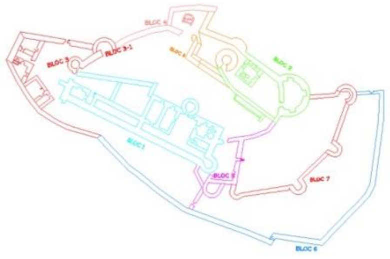


Fig. 7. Digitized Map in DXF-format obtained from Fig. 3

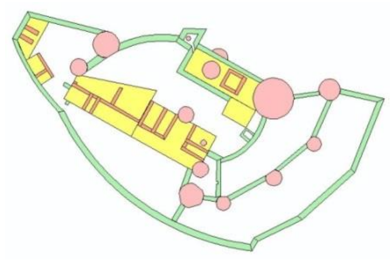


Fig. 8. Superimposition of different structured shapefiles

However, *CityEngine* interprets lines as entities which are designed to receive road rules and not building rules. The closed polylines have to be converted into polygons. So, in our study, these polygons have been classified in layers according to the composition of the building like ramparts, walls, buildings, towers, etc. depending on what the user wants to assign 3D generation rules. The layer contents are converted in different shapefiles (Fig. 8). To properly use polygons in the 3D scene, we had to create polygons representing in detail the building footprints.

Once the shapefiles have been prepared they were imported into the 3D scene. At this step, it is possible to create rules assigned to the footprints to model the historic site. However, it is better to project shapes on the terrain rather than leave them at the zero elevation. We have performed the alignment of the shapes to DTM (Fig.9). A projection method can be configured for this. But we recommend to use the *Minimum Translate* function, to translate shapes just below the terrain. So when shapes are extruded, no gap can persist between buildings and terrain. Once *Minimum Translate* function is selected, the next step is to choose the terrain on the heightmap field and add an offset if it is necessary.

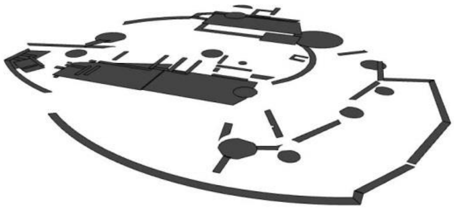


Fig. 9. Shape to DTM alignment

3.2 Rules

What is a rule?

In *CityEngine*, building modelling is performed through CGA-files, called rule-files. These files are text files using a single grammar, a programming language specifically designed to model the architectural content in three dimensions. The idea of grammar-based modelling is to define rules that iteratively refine a design by creating more and more details. *CityEngine*'s shape grammar is based on the so called "L-systems" developed in 1968 by the Hungarian biologist Lindenmayer to describe the growth of plants in a formalised language. The same principle is applied to architecture: the 3D geometry is created by iterative horizontal and vertical splits of an initial shape. The steps for shape creation are written in a so called "CGA rule" [13]. A CGA-rule-file defines how the geometry of the current building has to be created and is generally assigned to a shape (in our case), but it can also be assigned, to a linear entity like a street for example.

How to write a rule?

A rule-file is a collection of attributes and functions describing the geometry of the 3D extension of a shape. There are two ways available to edit a rule-file: either text mode, i.e. by typing code or a visual interactive mode.

```
/**
* Fichier : Exemple . cga
* Cree : 17 Jan 2014 10:47:50 GMT
* Auteur : Fedczyszyn
*/
attr Hauteur_min = 10
attr Hauteur_max = 30
attr Hauteur_etages = 3
attr Largeur_fenetres = 2
Lot --> extrude ( rand ( Hauteur_min , Hauteur_max )) Bloc
Bloc --> comp (f) { side : Facades }
Facades --> split (y) {~ Hauteur_etages : Etages }*
Etages --> split (x) {~ Largeur_fenetres : Tuiles }*
Tuiles --> setupProjection (0, scope .xy , 2, 3)
projectUV (0)
texture (" assets / facade / stone . jpg")
```

How to apply, assign a rule?

The assignment of a rule, or rather of a rule-file to a shape is a simple selection/association step. Once the file assigned, the user has to generate the model to display before proceeding to the next step.

How to use a rule?

Once the rule is created, assigned and generated, and once the parameters and attributes have been introduced in the rule, then the rule can be used by varying the different parameters and attributes. The settings can be modified by height minimum, maximum, floor height, width of the windows, etc.

By selecting the model in the main viewing window, sliders allow to modify the parameters through an Inspector window. It is possible to change the values visually by using sliders or by typing the exact values directly. The model fits itself in real time in the main viewing window.

3.3 Example of rule creation

For modelling the Engelbourg Castle, and to explain the concepts, five different rules have been established according to the layer classification of the castle components:

- Walls: extrusion rule and texture application.
- Ramparts: extrusion rule, battlements construction rules and texture application.
- Buildings: extrusion rule, wall thickness parameters, doors and windows positioning rules, roof creation rule and texture application.
- Towers: extrusion rule, wall thickness parameters, doors and windows positioning rules, roof creation rule and texture application.
- Embattled towers: extrusion rule, wall thickness parameters, doors and windows positioning rules, battlements creation, open inner cylinder creation and texture application.

With the adopted layer system, we can easily display one by one type of entities, such as towers, and assign them the corresponding rule-file in one operation to all of them.

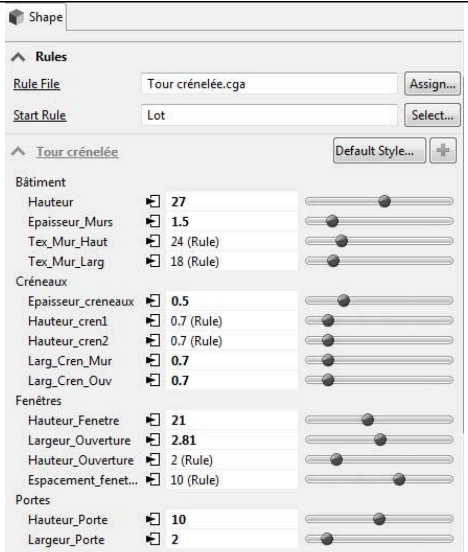
In this part are presented the code specifications for complex rule-files created for the embattled towers.

It is firstly described what is planned to model with these rules:

- The first step is to extrude the tower according to a height parameter.
- The second step is to divide this extrusion vertically and horizontally to position doors and windows.
- Starting from the initial shape, it is necessary to perform an offset inwards with the value of the wall thickness and then to extrude it.
- Then openings on the inner extrusion need to be created, exactly in front of doors and windows of the exterior wall to create an impression of thickness of the wall at the openings.
- At this stage, the result is a hollow cylinder with a thick wall. However at the openings it is necessary to texture the sides to avoid seeing the vacuum present between the two vertical extrusions.
- Then, starting from the initial shape, the disk above the tower is taken to create slots with vertical and horizontal divisions and texture for the battlements.

It was necessary to introduce 15 varying parameters (Fig. 10) to simulate models of tower and to obtain a model as expected. These parameters were classified in the rule-file as extrusion, outside facades, inside walls, crenellations, etc. (Fig. 11). The final model was based on a drawing of the reconstructed Castle made by a designer accordingly to archaeologist's hypothesizes. It is especially necessary to make the difference between realism and reliability [14].

We can then easily compare a manual drawing with the hypothesis of the first approach (Fig.12) with the parametric and textured model obtained by adapting the parameters to the best fitting of the same hypothesis in the parametric *CityEngine* model (Fig. 13). This approach allowed the archaeologists to propose some other hypothesis and to adapt their knowledge to the reality of the terrain.



Shape

Rules

Rule File: Tour crénelée.cga

Start Rule: Lot

Tour crénelée

Bâtiment	
Hauteur	27
Epaisseur_Murs	1.5
Tex_Mur_Haut	24 (Rule)
Tex_Mur_Larg	18 (Rule)
Créneaux	
Epaisseur_creneaux	0.5
Hauteur_cren1	0.7 (Rule)
Hauteur_cren2	0.7 (Rule)
Larg_Cren_Mur	0.7
Larg_Cren_Ouv	0.7
Fenêtres	
Hauteur_Fenetre	21
Largeur_Ouverture	2.81
Hauteur_Ouverture	2 (Rule)
Espacement_fenet...	10 (Rule)
Portes	
Hauteur_Porte	10
Largeur_Porte	2

Fig. 10. Modifiable parameters value for the crenellated tower rule file

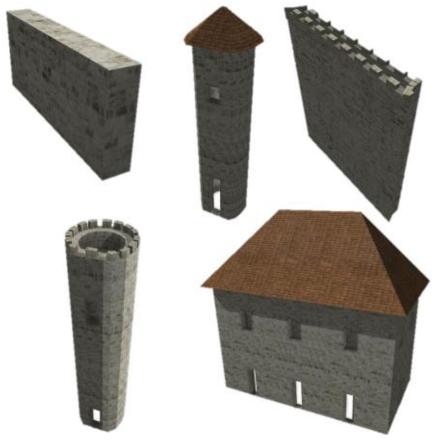


Fig. 11. Five rule based models constitutive of Engelbourg castle

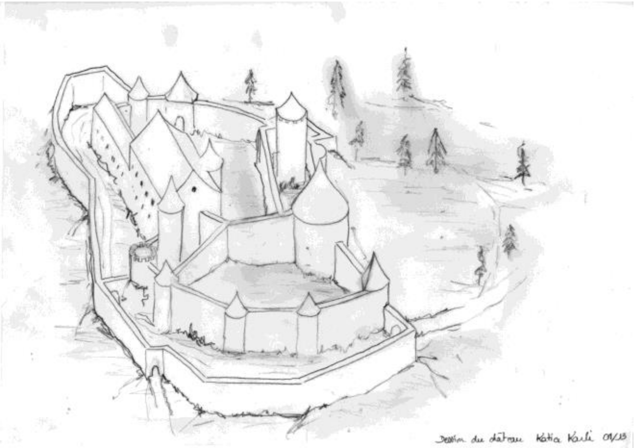


Fig. 12. Hand drawn reconstruction hypothesis by Karli [1]

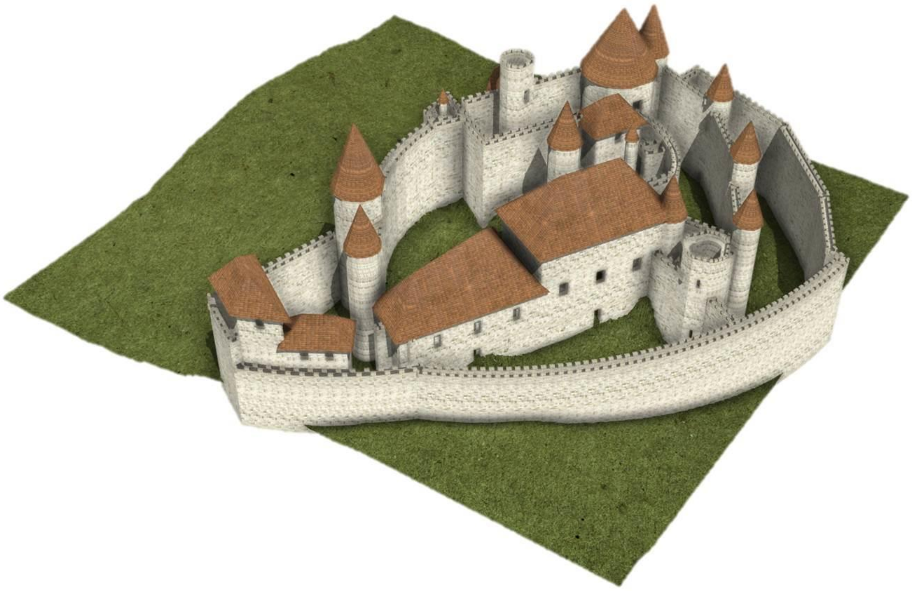


Fig. 13. Final reconstruction of Engelbourg castle in *CityEngine*

4 Prospects and Conclusion

4.1 Issues and prospects

Adopting the *CityEngine* software is quite time-consuming. It is not very intuitive and uses unusual operating mode, despite the many explanatory videos on the ESRI website. After modelling this historic site, the software is mastered but CGA-language remains still full of unexplored features and many other functions are available but not been used. For example, it is possible to insert an already 3D modelled object for a very detailed window, instead of simply texturing a Tile representing the windows.

One of the major problems encountered in this model is that the rules are only applicable to polygons. It has been difficult to model the openings on the circular towers. After import, the circular shapes are converted into polygons with a huge number of sides with a very small length. So it is impossible to have a window or door width of one meter. It has been therefore necessary to simplify the tower shapes with some straight sections for the openings.

At this stage, the model of Castle is quite perfectible and to improve it, a lot of overtime needs to be spent: create almost one rule-file for each entity, tower type or building part, etc.

However, if this is done, the project moves away from the initial software philosophy. So it would be more effective to work with another 3D modelling package. *CityEngine* is designed and so quite useful for modelling large areas such as cities with basic representations. The rules are supposed to be generic. One possible use after the final modelling is to export the model into a 3ws-file format, viewable with the *CityEngine* Web Viewer (Fig. 14) without any installation on a web browser. Thanks to this, the model can be presented to any public. The viewer is very intuitive.

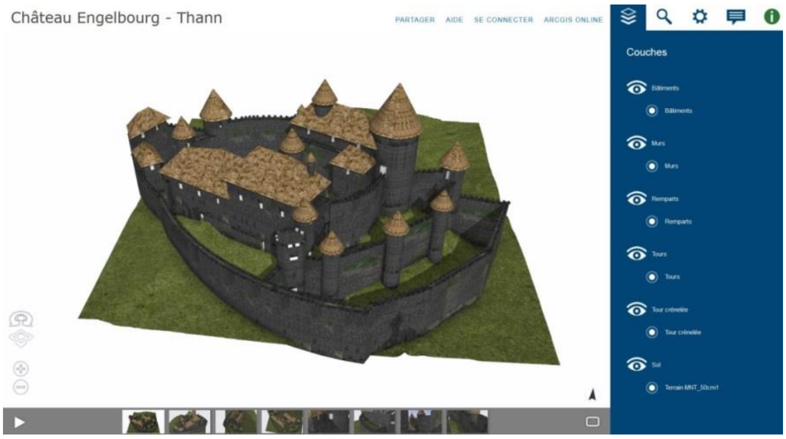


Fig. 14. *CityEngine* Web Viewer

4.2 Conclusion

CityEngine is a highly interesting and powerful software, totally different from any other software used for architectural modelling in three dimensions.

After spending many hours using *CityEngine*, the software is mastered by the operator but the CGA-language is not fully exploited.

Setting up the initial data is a crucial and mandatory step because it determines the whole modelling efficiency.

Community using *CityEngine* is currently a small group of experts and it is difficult to find information to develop and apply it in concrete programs. Further research is needed to perform deeper modelling and using coded rules in special cases like cultural heritage projects.

References

1. Karli, K.. Château de l'Engelbourg, Thann. Restitution 3D. Master Degree ENSAS, 50 pages (2012).

2. Berger, S., Etude de la modélisation d'une maquette 3D et de l'intégration de données. Application au projet archéologique et patrimonial de Thann (Alsace). Master Degree INSA de Strasbourg, 84 pages (2011).

3. Nobile, S., Intégration des outils numériques constitués sur le château de Thann (Haut-Rhin) dans un système d'information et une base de connaissances historiques, sous la forme d'un site internet. Master Degree INSA de Strasbourg, 64 pages, (2013)

4. Koehl, M. Brigand, N. Combination of virtual tours, 3D model and digital data in a 3D archaeological knowledge and information system. In: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXIX-B4, 439-444, (2012)

5. http://engelbourg-thann.fr/

6. Ehretsmann, M. Thann (Haut-Rhin) – Engelbourg, Rapport de fouilles programmées, campagnes de sondages 1992, Strasbourg : SRA d'Alsace, 150 pages (1992)

7. Ehretsmann, M. Analyse architecturale du château de l'Engelbourg à Thann (Haut-Rhin) Campagne de sondages archéologiques 1992. In : Cahiers alsaciens d'archéologie, d'art et d'histoire, pp. 179-194 (1995).

8. Koch, J. Thann, Haut-Rhin, Château de l'Engelbourg, Rapport de diagnostic, PAIR. (2010)

9. Koch, J. Thann – Engelsburg, Projet d'étude et de valorisation archéologique, Document-cadre, PAIR (2008)

10. Schenck, A. J. Sanctus Theobalus (1628)

11. Müller, P. PhD Thesis, ETH Computer Vision Lab. Zürich, Switzerland (2007)

12. Piccoli, Ch. CityEngine for Archaeology. In: 3D GIS for mapping the Via Appia. VU University Amsterdam, (2013)

13. Wonka *et al*. Instant Architecture. In A. P. Rockwood (ed.), Proceedings of ACM SIGGRAPH 2003 / ACM Transactions on Graphics 22(3), pp. 669-677. (2003)

14. Pescarin, S. Ancient landscape reconstruction. 1st Digital Cultural Heritage Workshop (2012)

15. Corrado, C. Automatic evolution of conceptual building architectures, Master of Science, Brock University, St. Catherines, Ontario, 109 pages (2011)

16. Cappellini, V., Saleri, R, Stefani, C, Nony, N., De Luca, L. A procedural solution to model roman mansory structures. In : IAPRS Volume XL-5/W2 (2013)

17. Edvardsson, K.N. 3d GIS modeling using ESRI's CityEngine – A case study from the University Jaume I in Castellon de la Plana Spain. Master of Science in Geospatial Technologies, 97 pages (2013)

18. Besuievsky, G. Patow, G. Procedural modeling historical buildings for serious games. VAR Vol.4 Nr 9, ISSN:1989-9947, pp. 160-166 (2013)

From Geological Exploration to 3D Geological and Numerical Models

Tatjana Kühnlenz [a*], Herbert Kunz [a], Detlef Schlüter [b], Jörg Hammer [a], Jürgen Hesser [a]

[a] Federal Institute for Geosciences and Natural Resources (BGR), Stilleweg 2, 30655 Hannover, Germany
Tatjana.kuehnlenz@bgr.de - herbert.kunz@bgr.de - joerg.hammer@bgr.de - juergen.hesser@bgr.de
[b] Ingenieurbüro bicad, Scheidestraße 4, 30625 Hannover, Germany

KEY WORDS: Modelling, Geology, Geometry, Triangulation, Software, CAD

ABSTRACT:

Three-dimensional geological and numerical modelling is an essential tool for planning of exploration works and evaluation of dynamic thermal, hydraulic and mechanical processes (THM). The different types of data, like maps, boreholes, seismic data and geological profiles are gathered and interlinked in the geological 3D models. Thus the 3D models serve as three-dimensional data base. Moreover, the geological 3D models are relevant as planning tools, presentation tools or as basis for the calculation of dynamic natural processes. The numerical modelling is crucial for the investigation of THM processes due to the exploitation of the underground (geothermal energy, mining, cavern sites, underground disposal and repositories) with regard to simulate the present status and to give a prediction for the usage in future. The complex mutually interacting processes are expressed in balanced equations and then calculated using computer-based numerical methods. This paper describes the process how to convert a geological model into a mesh for the numerical modelling. An essential part is the development of new methods for classification of model defects to ensure the mesh consistency, which is necessary for numerical calculations.

For 3D geological modelling BGR uses the software package openGEO, developed by bicad Hannover (Preuß 2002; Kühnlenz et al. 2011; Hammer et al. 2012). OpenGEO is a standalone program and uses the graphic core of AutoCAD with an activeX control and visual lisp commands.

The 3D models built with openGEO include all types of basic geological and geophysical data. Thereby, vector data (maps) as well as raster data (seismic data) can be implemented and processed in openGEO. These different data should be adjusted to each other, so that the inconsistencies can be removed. In order to realize this, all data must be georeferenced and evaluated in the 3D space in interaction with each other. The 3D models created in openGEO describe precisely even the most complicated geological structures like flow folds, overturned structures or multiple activated fracture zones.

To calculate various THM processes in different geological structures, the geological models have to be converted into valuable structures as basis for numerical simulation. Therefore, in BGR´s workflow some special tools were developed for the adaptation of geological 3D models to numerical calculation models.

The 3D modelling in openGEO based on a 3D line framework. Thus, the triangulation is oriented on the line framework, the backbone structure of the 3D model. This procedure allows the construction of highly complex structures. The openGEO 3D models describe the surfaces and the discontinuity planes with triangle meshes (Fig. 1). These triangles often have small interior angles. If these surfaces are used directly as basis for generating meshes for numerical calculations, these small angles can lead to severe problems. Therefore, a remeshing process must be performed to create triangles with an acceptable aspect ratio. In order to prevent changes in the shape of the 3D surfaces due to remeshing, the “feature lines” (ISO lines, geological outcrops) and the faces from the geological 3D model have to be transferred to the remeshing program. The “feature lines” remain unchanged in the remeshed model, so that the openGEO 3D model and the remeshed model have an identical shape even in complex geological structures.

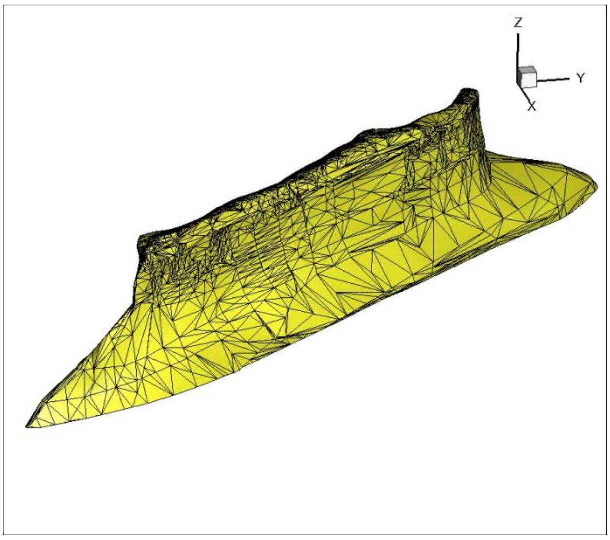


Fig. 1: openGEO 3D model of a salt dome (18 km length)

Since the last 10 years a pre- and post-processing program has been developed for numerical calculation programs used in the BGR. This Program (GINA) contain meshing features for structured 2D and 3D meshes. For unstructured 2D and 3D meshes an interface to freeware meshing programs was implemented. These programs are GMSH (http://geuz.org/gmsh; Geuzaine & Remacle 2009) and TetGen (http://wias-berlin.de/software/tetgen; Si). GINA in combination with the freeware program GMSH was used for the conversion of the openGEO surfaces into more suitable structures according to the requirements in regard to numerical calculation models.

Using GINA, the surfaces of the geological 3D model can be converted into a readable format for GMSH (own ASCII geometry file). Various parameter can be set to create a coarser or finer result mesh, or to obtain certain areas. With a special feature in GMSH (Remacle et al. 2010) all triangles of the geological surfaces can be merged into compound surfaces which can be easily remeshed (Fig. 2). The basis surface structure is preserved at the best.

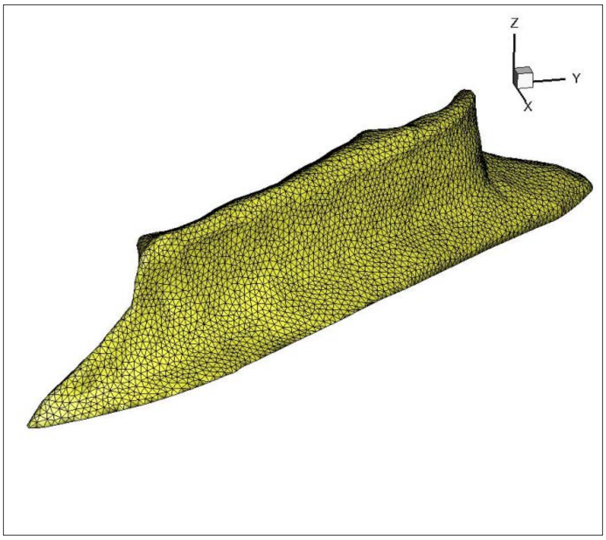


Fig. 2: the salt dome - remeshed with GMSH

However, 3D models built in openGEO which shall be remeshed should meet certain requirements: they have to be watertight, which means they are completely faced bounded. Mesh holes, intersections of faces and zero-area faces can lead to an abort of the remeshing process. Therefore, these types of mesh defects have to be removed within geological 3D models before their transformation to numerical calculation models. As different calculation programs have specific requirements concerning consistency, the defects will be checked within the particular program. Reasonably, check routines for model defects have to be implemented in the calculation programs, not in openGEO.

After loading the prepared 3D model into GINA three separate test routines are performed:

- Mesh holes will be detected if one edge of an element is not connected with another edge.

- Zero-area faces will be determined if the distance between one node and the opposite line of a triangle element is smaller than a limiting value.
- Intersecting elements will be detected with a feature of TetGen.

Afterwards three log files for the different types of mesh defects will be created by GINA. The error log files can be loaded and displayed in openGEO, so incorrect elements can be easily determined and reconditioned interactively by user. It has to be mentioned that some zero-area faces have already been eliminated using GINA. In the next step the adjusted model will be re-imported into GINA and remeshed with GMSH.

The remeshed 3D models can also be smoothed. This smoothing leads to little differences to the original geological model in some parts. It has to be assessed whether this deviation is in the range of tolerance or not. For this reason the original surfaces from openGEO and remeshed surfaces from GINA are loaded into the same 3D space in openGEO. Afterwards the distances between corresponding points are measured with a special function. This measurement allows the determination of model areas, where the distance exceed defined limits. To avoid such deviations, special points or lines can be fixed so that the structures are preserved exactly at these points. Also, the mesh density can be changed at these locations.

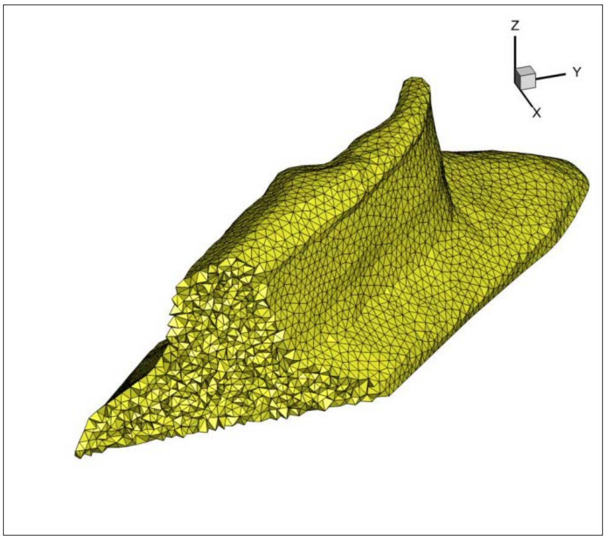


Fig. 3: tetrahedron elements of the salt dome created with TetGen

On the basis of these remeshed surfaces, tetrahedron elements can be generated for 3D calculation models (Fig. 3). For this BGR uses the freeware TetGen. This program has the advantage to detect self-acting closed volumes in 3D models. No volumes must be defined by the user. It is possible to define limits concerning radius and volume for the tetrahedron elements of the resulting mesh. Additional geometry elements (e. g. boreholes, galleries, etc.) can be implemented in the calculation model and meshed by TetGen (Fig. 4).

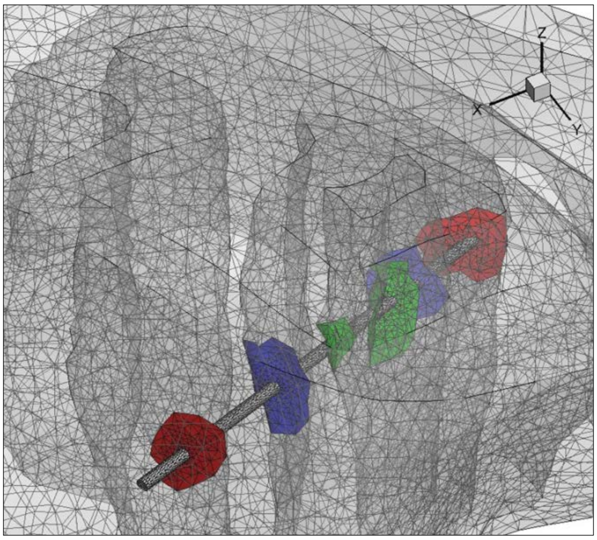


Fig. 4: gallery integrated in a geological 3D model

The prepared and converted openGEO geological 3D models are the basis for the numerical calculation programs which are used at BGR (JIFE and OpenGeoSys). Thus, BGR provides the complete process from geological exploration and geological 3D modelling up to numerical calculations for long-term safety assessment and for site evaluation in all types of host rock formations.

References:

Geuzaine, G. & Remacle, J.-F., 2009. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. In: *International Journal for Numerical Methods in Engineering* 79(11), pp. 1309-1331.

Hammer, J., Dresbach, C., Behlau, J., Mingerzahn, G., Fleig, S., Kühnlenz, T., Pusch, M., Heusermann, S., Fahland, S., Vogel, P. & Eickemeier, R. 2012. Geologische 3D-Modelle für UTD-Standorte - Generierung, Visualisierung, Nutzung. In: *Abschlussveranstaltung BMBF-Förderschwerpunkt „Entsorgung chemotoxischer Abfälle in tiefen geologischen Formationen", Februar 2012*. Materialienband, FZKA-PTE, Karlsruhe, pp. 221-273

Kühnlenz, T., Fleig, S., Mingerzahn, G., Behlau, J. & Hammer, J., 2011. The geological 3D-model of Gorleben salt dome - a tool for efficient exploration, data evaluation and data storing. In: *Quo Vadis Sal: XVI Miedzynarodowe Sympozjum Solne; Magazynowanie i Skladowanie w Wyrobiskach Solnych - Terazniejszosc i Przyszlosc Górnictwa Solnego,* Torun, Poland, pp. 39 - 41

Preuß, H., 2002. Progress in modelling the subsurface in Lower Saxony: experiences with OpenGeo. GIC-17: meeting of the Geoscience Information Consortium, 4 - 7 June 2002, Uppsala, Sweden, p. 9

Remacle, J.-F., Geuzaine, C., Compère, G. & Marchandise, E., 2010. High-quality surface remeshing using harmonic maps. In: *International Journal for Numerical Methods in Engineering* 83(4), pp. 403-420

Si, H. Tetgen: a quality tetrahedral mesh generator and three-dimensional Delaunay triangulator. http://wias-berlin.de/software/tetgen

Comparing Geomorphic Systems using Graph Algebra on Dual Graphs – First Results and further research Questions

Marc-O. Löwner, Thomas Becker

Abstract Natural landforms are of great importance for a variety of scientific and engineering disciplines. Their investigation can be improved by a comparison of features that have similar characteristics, structure and genesis. Current data frameworks do not support the description, search and comparison of landforms that are complex in geometry and material properties and are changing due to material transport processes. Here, the application and analysis of a topological multilayered graph approach is presented that represents every single state of a landform and its different layers at a given time state as a dual graph structure. Development from one state graph to another is represented by an interlayered graph, which maps nodes of one state to their corresponding representation of another state graph with weighted arcs. Since topology can be applied to represent sequences of geological layers and adjacency relationships of landforms, the exact geometric representation can be neglected. Graph algebra will be used to generate numbers that reflect similarities of neighborhood and genesis of two different landforms and helpgeoscientists in gaining insights into landform assemblages of different age and in different locations.

Marc-O. Löwner, Institute for Geodesy and Photogrammetry, Technische Universität Braunschweig, Germany, m-o.loewner@tu-bs.de

Thomas Becker, Institute for Geodesy and Geoinformation Science, Technische Universität Berlin, Germany, thomas.becker@tu-berlin.de

1 Introduction

The earth's surface is the boundary layer between the solid earth and the atmosphere and hydrosphere. Its future behavior and adoption on changing boundary conditions, therefor, is a key question when resilience of manmade constructions and natural resources are under investigation. This is particularly important if impact of a changing climate and human activities on agricultural land and built up areas is considered. Civil engineers and geoscientists need to assess earth's surface general characteristics, its decomposition, its 3D structure, its stability, as well as its genesis and future development. 3D semantical database modelling and topological analysis performed in spatial computer science may support these research areas when reliability and comparison of investigated natural objects are needed.

Natural landforms, their change, and processes that are causing these changes are in the competence of the science of Geomorphology (Hugget 2011). They may be defined as units of material, the sediment, which was accumulated under specific conditions and is reworked due to shape and material properties of the landform as well as external forces. Landforms may be disaggregated into layers of more or less homogenous material properties that have only partly crisp boundaries. Spatial arrangement of landforms and their change are determined by a triad of process, form and material. Since the interaction of these three factors is complex, comparing different landforms and their behavior seems to be more suitable to predict a landform's future development than physically based modelling approaches over a long time span. The latter lacks on unknown boundary conditions that, in addition, change in time.

Landforms are unique but comparable with each other in terms of their internal structure, their topological relationships to other landforms or parts of them and their temporal changes as well as their exposure to climate system. Of cause, landform development cannot be observed directly over a period of hundreds of years. Nevertheless, landforms embody the integral of all material transport processes that formed them over this time period. Thus, material layers of landforms represent an archive of the landforms own history. Therefore, analyzing outcrops, drillings or geophysical data is applied to reconstruct landform development, even if only their current status can be uncovered. However, application of these exploration methods allow for the reconstruction of discrete states of landform development.

On the basis of extrapolation geoscientists reconstruct history of states and processes that led to these states from the current situation until a hypothetical initial state, i.e. an undisturbed landform, is reconstructed. Amongst others (rf. Dikau 1999, Brunsden 1996) this investigation includes a space for time substitution. This is a method to describe and predict future behavior of a landform in a certain early state of maturity by comparing it with older ones (rf. Welch 1970, Brunsden & Kesel 1973). Investigation of the genesis of a mature landform may serve as an ex-post assessment of younger landform's future behavior and, as a precondition for that. Thus, geoscientists need to search already investigated landforms that have further developed than the object for which future behavior is to be predicted. Since landforms are unique, objects of comparison must have at least similar characteristics, structure or genesis compared to historical landforms that once encountered similar behavior in the past.

Spatial information science may support this work by providing a framework for the formal description of landforms, their parts, their spatial associations and the development of all these partitions. It has to enable first, to search for landforms that are comparable in their specific characteristic. Secondly, a GI framework should enable a documentation of landform development and, therefore, be able to describe different stages of a landform development as discrete states. Finally, the framework should be able to allow for an additional description of geometrical and semantically properties of landform layers.

Today, there is no comprehensive interoperable data model that fulfills these requirements. Due to their complex construction and characteristics and their interrelation with other landforms, defining a semantic data model is a reasonable challenge. A main problem is geometry. Landforms are divided by layers that have crisp boundaries but are hard to detect. On the other hand, some ‘boundaries’ reveal a smooth transition that lead to fiat boundaries (Smith 2001). However, semantic information, topological relationships, and change can be represented, even if data on geometry is not available.

Here we present the application and analysis of a topological multilayered graph approach (Löwner & Becker 2013) that represents every single state of a landform and its different (soil) layers at a given time state as a dual graph structure. Development from one state graph to another is represented by superposing an interlayered graph, which maps nodes of one state to their corresponding representation of another state graph with weighted arcs. Since topology can be applied to represent sequences of geological layers and adjacency relationships of landforms, the exact geometric representation can be neglected. Nevertheless, geometric representation is still possible. Graph algebra will be used to generate numbers that reflect similarities of neighborhood and genesis of two different landforms and help geoscientists in gaining insights into landform assemblages of different age and in different locations.

In the next section, directly related work is reviewed. Section 3 is devoted to two examples of slope developed under different boundary conditions as a use case for the analysis described in section 4. In section 5 we follow up with a discussion and outlook.

2 Related Work

2.1 Topological aspects of landforms

Some aspects of landforms and their development may be analysed using the concept of topology. This includes, first, the analysis of 3D topological relationships of landforms and landform layers. Their specific association builds up the georelief (Kugler 1974; Dikau 1996) and characterizes a specific geomorphic system. Topological investigation can help to identify the chronological order in which landforms were formed. Second, the interconnection of landforms through material transport processes can be formalized by directed graphs. Landforms do not exist in isolation but do interact with others.

(Zlatanova 2000) described topological relationships between eight multidimensional simple objects in $\mathbb{R}^3$. Based on (Egenhofer 1989, Egenhofer & Herring

1990) their boundary, interior and exterior are denoted by ∂a, a°, a^-, and ∂b, b° and b^-. A topological relationship $R(a, b)$ of two objects is then identified by composing the possible set of intersections resulting in a 9-bit binary value. If, for instance, two objects share a common boundary, the intersection between the boundaries is non-empty, i.e. $\partial a \cap \partial b = \neg\emptyset$. The binary value is then converted into a decimal number, e.g. R287 for '*a meets b*', where intersections $\partial a \cap \partial b$, $a^- \cap b^-$, $a^- \cap \partial b$, $a^- \cap b^\circ$, $\partial a \cap b^-$, $a^\circ \cap b^-$ are $= \neg\emptyset$ (Zlatanova et al. 2004).

Considering natural landforms, (Löwner 2013a) discussed four of these relationships that can be found investigating geomorphic systems. Analysing these topological relationship helps to sequence different material transport processes. Furthermore, geometric reconstruction of landform layers can be supported (Löwner 2013b). These are first, R476 = *coveredBy*, where body *a* lies on top of *b* or partly within *b* and is younger than *b*, second, R287 = *meet*, where *a* is adjacent to *b* but chronological order cannot be proved directly. Third, the trivial case R031 = *disjoint* has been distinguished from, fourth, R220 = *inside*, where *a* lies within the interior of *b* and, again, temporal order of the landform's genesis can be derived. In this case, genesis of *b* had started before the genesis of *a* but ended later. However, the most common case of successive bedded strata has not been considered.

Landforms are connected via material transport processes that are driven by climate and gravity. These transport processes erode material from one landform, transport it to and accumulate it on another one. In terms of systems theory in geomorphology (Chorley & Kennedy 1971) this can be described as a sediment cascade. The output of one subsystem deals as the input to another one. However, representation of geometry is not covered.

2.2 Application model for landform representation

Based on ISO 19107, i.e. Spatial Schema, (Herring 2001) and an inchoate model from Löwner et al. (2005), Löwner (2005a, 2010) proposed a GML application model (Lake et al. 2004) that represents the concept of a sediment cascade. It represents landforms with 3D geometry and contains a class *_Geoprocess* that represents nothing but the connectivity between these objects with the meaning of material exchange. An abstract class *_Geoobject* represents a solid landform feature and has one or more associations to an abstract class *_State*. For an associated *Timespan* the *_State* acts as an agent between the *_Geoobject* and its properties including geometry and an abstract class *_AttributeSet*. The class *_Geoprocess* connects two instances of a *_Geoobject* initializing new instances of corresponding *_States*. Thus, the concept of a *_Geoprocess* stands for the connecting edge between two landforms and is valid for a given time span. It directly enables a directed graph like representation of a sediment cascade where the sediment source may be interpreted as the "from-node" and the other, acting as a sink, as the "to-node".

An application of this directed graph representation is given by (Löwner & Otto 2008). Starting from a geomorphological map they analysed sediment cascades on the basis of the dual graph concept. An elaborated approach to save results of raster based analysis of sediment pathways from contributing sources is presented by (Heckmann & Schwanghart 2013). However, this predictive geomorphic mapping approach (Luoto & Hjord 2005) focuses on recent processes, only.

2.3 Mutilayered graph approach for the representation of landform development

The Multilayered Graph Model (MGM) was initially designed as a modular concept for route planning and localization in indoor environments (Becker et al. 2008). The MGM allows for the integration of conceptually separated indoor space models within a multi-layered representation. The layers are independent in such that they represent separate decompositions of space according to different semantic criteria, f.i. the building topography or sensor characteristics. The stack of different layers is then linked by joint-states which mutually constrain possible locations of objects in either space model. The modular framework allows the separate consideration of spaces and their geometric subspaces. The certain layers of the modular model describe the overlay of different space models. These space models specify the same real world object and are located in the same Euclidean space at the same time. However, this Euclidean space is subdivided through different semantics, which can be defined through the layers or objects itself but also by humans.

Since each layer provides a valid and consistent representation of space, the common framework itself can be seen as a valid multilayered space representation, which can be used as a whole to describe, for example, the indoor environment of buildings. For each layer, topological relationships such as connectivity and adjacency relations between 3D spatial objects are represented within topology space. In primal space, topology is induced by the corresponding 3D geometry in Euclidean space. A structure can be represented in topological and geometrical space according to ISO 19107 (Herring 2001). In primal space, topology is induced by the corresponding disjoint 3D geometry in Euclidean space. By applying a duality transformation based on Poincaré duality, the 3D cells in primal topology space are mapped to nodes (0D) in dual space. The topological adjacency relationships between 3D cells are then transformed to edges (1D) linking pairs of nodes in dual space. The resulting dual graph represents a Node-Relation-Structure as proposed by Lee (2004).

Löwner & Becker (2013) proposed a framework for the representation of landform development that simplifies the complex spatial relationships between 3D objects and the modelling of geological processes over time applying the Poincaré Duality (Munkres 1984). Single landform layers are represented as nodes (*DualStructures*) whereas the neighborhood of these layers is represented as edges

(*DualStructureRelations*). Finally, a *DualStructureState* represents a whole landform and its parts of stable conditions over a period of time. Change of a landform is represented as additional edges (*Abstract_GeoProcess*) between the nodes of layers of different states. The overall structure constitutes a multilayer graph, where all the nodes from all N layers are included but are separated into M partitions of time. In combination with the above described semantical data models (Sect. 2.2) all dual representations may be associated with geometric and semantic representations.

3 Use cases – two landforms and their development

Here, two examples of landforms and their evolution will be introduced as a use case. The first is a well investigated landform of a loess covered near Bonn, Germany (Löwner 2000, Preston 2001, Löwner et al. 2005B). The second one is an imaginary object that can be found in other publications or textbooks (Bork 1988, Morgan 2004).

Both landforms have in common, that they are composed of several soil layers with different material properties, which have been altered by material transport processes. This means that geometry and topology of landform layers have been changed. The general process domains on both slopes are redistribution of sediment by diffuse and linear erosion processes due to agricultural land use for several millennia (Bork et al. 1998). The anthropogenic clearing of vegetation cover represents the removal of a damping agent between climate and the soil system, forcing the landscape to readjust through redistribution of soil material. A smoothing effect on slope morphology reinforced by soil tillage can be observed on slope where transport capacity does not exceed a certain magnitude (e.g. Lindstrom et al. 1992, Blanco-Canqui & Lal, 2008). Alternatively, where ground cover is removed or strong rainfall events take place, linear erosional processes lead to more rugged and less smooth landscapes.

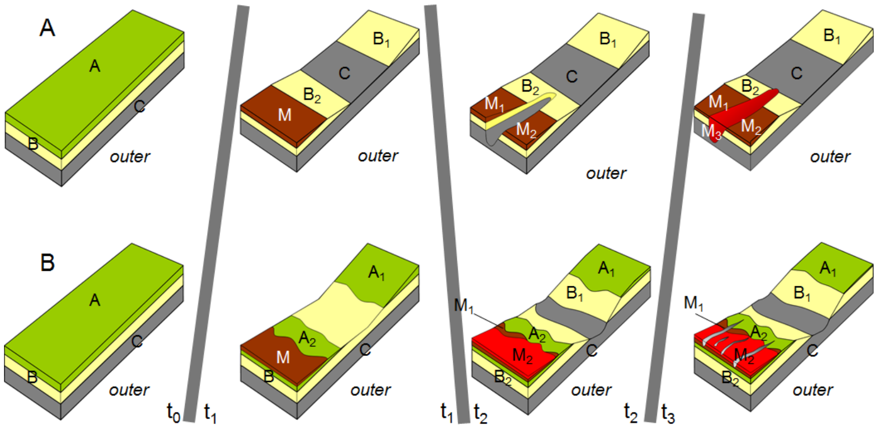


Fig. 1. Blog diagrams of two different landforms and their evolution over four discrete periods of time (see text for further explanation)

However, sundry boundary conditions led to a different evolution. Development of the first landform (Fig. 1 A) has been reconstructed as follows (see Löwner & Becker 2013 for a more detailed description):

- ($t_0 – t_1$) Erosion of the entire decalcified top soil, i.e. the A-horizon and parts of the lower bed B-horizon, followed by partly accumulation on the food slope.
- ($t_1 – t_2$) Cutting of a gully due to a strong rainfall event.
- ($t_2 – t_3$) Consecutively padding of the gully with material eroded at the upper slope and further sheet erosion at the top and mid slope followed by accumulation of material at the slope foot.

Development of the second landform (Fig. 1 B), however, is different:

- ($t_0 – t_1$) Erosion of the decalcified top soil, the A-horizon on the mid slope due to higher water supply and flow velocity and partly accumulation of this material at the foot slope
- ($t_1 – t_2$) Ongoing erosion on the mid slope and accumulation on the foot slope. Because the calcareous C-horizon is reached, a second colluvium (M_2) can be identified.
- ($t_2 – t_3$) Due to decreased slope, further erosion is limited to ploughing erosion on the top slope and small and pervasive linear erosion at the foot slope.

Although the two landforms have similar starting conditions, they develop quite different. This is due to different human impact, e.g. a more conservative agricultural activity, and extreme climate events.

4 Graph based comparison of two geomorphic systems

4.1 Graph based description of landforms

AbstractGeoProcesses that cause changes of geometry between discrete states of landform development, i.e. from t_n to t_{n+1}, are listed in table 1, for both, landform A and landform B. As specializations of *AbstractGeoProcesses*, erosion (e) and a complex geoprocess (Löwner 2010), erosion and accumulation (e/a) can be identified.

Table 1. *AbstractGeoProcesses* of landforms A and B (Fig. 1). Type of *AbstractGeoProcess* is presented in brackets whereby e stands for erosion and e/a for erosion and accumulation. O stands for the outer.

Period of time	AbstractGeoProcess of landform A	AbstractGeoProcess of landform B
$t_0 - t_1$	At0 → ot1 (e) Bt0 → B1t1 (e) Bt0 → B2t1 (e) Bt0 → Mt1 (e/a) Bt0 → ot1 (e)	At0 → A1t1 (e) At0 → A2t1 (e) At0 → Mt1 (e/a) At0 → ot1 (e) Bt0 → Bt1 (e) Bt0 → Mt1 (e/a) Bt0 → ot1 (e)
$t_1 - t_2$	B2t1 → B2t2 (e) B2t1 → ot2 (e) Mt1 → M1t2 (e) Mt1 → M2t2 (e) Mt1 → ot2 (e) Ct1 → ot2 (e)	A1t1 → A1t2 (e) A1t1 → M2t2 (e/a) A1t1 → ot2 (e) A2t1 → A2t2 (e) A2t1 → M2t2 (e/a) A2t1 → ot2 (e) Bt1 → B1t2 (e) Bt1 → B2t2 (e) Bt1 → M2t2 (e/a) Bt1 → ot2 (e)
$t_2 - t_3$	B1t2 → B1t3 (e) B1t2 → M3t3 (e/a) B1t2 → ot3 (e) B2t2 → B2t3 (e) B2t2 → M3t3 (e/a) B2t2 → ot3 (e) Ct2 → Ct3 (e) Ct2 → M3t3(e/a) Ct2 → ot3 (e) M1t2 → M1t3 (e/a) M1t2 → M3t3 (e) M1t2 → ot3 (e) M2t2 → M2t3 (e/a) M2t2 → M3t3 (e) M2t2 → ot3 (e)	A1t2 → A1t3 (e) A1t2 → o (e) A2t2 → A2t3 (e) A2t2 → o (e) B1t2 → B1t3 (e) B1t2 → ot3 (e) B2t2 → B1t3 (e) B2t2 → ot3 (e) M1t2 → M1t3 (e) M1t2 → ot3 (e) M2t2 → M1t3 (e) M2t2 → ot3 (e)

The overall multilayered graph structure represents first, the topological relationships of a single layer, i.e. the discrete state of a landform at a specific time sequence and, second, the evolution of landforms by directed edges between nodes of one single time state. A multilayered Graph for landform A is given in fig. 2.

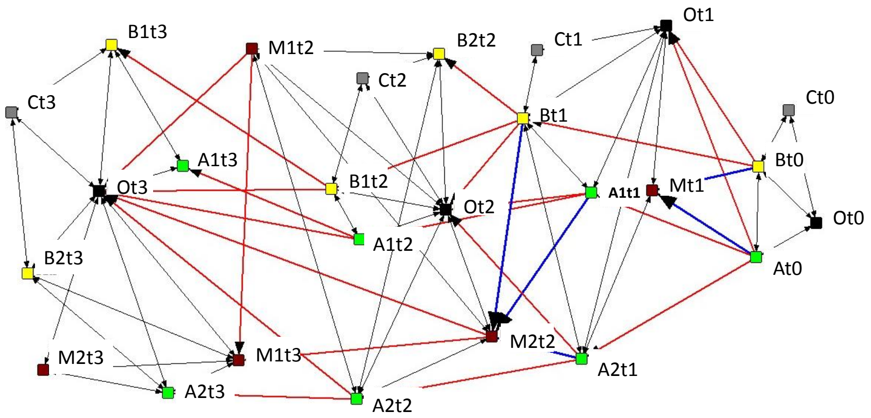


Fig. 2. Multilayered graph representation of the evolution of landform A (evolution from right to left), the node colors are similar to the layer colors depicted in fig. 1, where green depicts layer A, black the outer, yellow the layer B, grey the layer C, and brown the layer M. Grey arcs indicate the topological relationships, red edges erosion and blue edges erosion and accumulation.

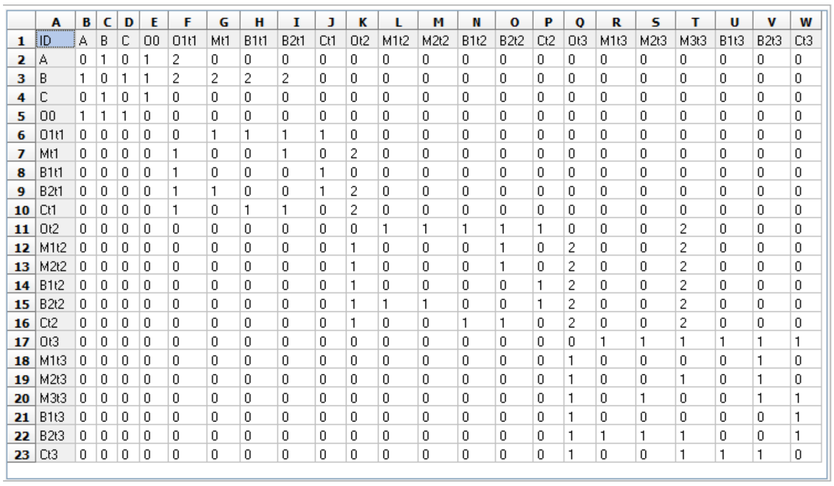
4.2 Graph based landform comparison

There are different ways to handle the landform comparison. The first works on node level and should answer questions such as “Are layers central in one network also central in another network?”. The second works on the dyadic level where answer should be given to the question whether a relation exist between the same layers or layer types. Finally, the third level (network analysis) should provide answers, how similar the graph structures or landform characteristics are. In the following, we apply different methods to both landforms and interpret the results obtained by graph analysis. The incidence matrices of both landforms are used as a base to perform our analysis, where the columns and the rows represent the different layers of the landform genesis.
According to Diestel (2005) a graph G is an ordered tuple of (V(G), E(G)) where V(G) is a nonempty set of vertices and $\mathrm{E(G)} \subseteq \{\{x, y\} \mid x, y \in V,\ x \neq y\}$ a set of edges. In terms of landform development V(G) equals to the set of layers of the respective landform to a certain point in time, where E(G) represents the set of Egenhofer Relations (adjacency) between the layers. A vertex x is incident to an edge e iff $x \in e$ and the both vertices that are incident to e are boundary of e and e links those vertices together. Two vertices $x,y \in G$; $x \neq y$ are adjacent in G and are called neighbors iff $xy \in E(G)$. Two edges $e, f \in E(G); e \neq f$ are adjacent, iff they share a common vertex. Thus, the degree d(v) of a vertex v is the number (degree) $|E(v)|$ of incident edges of v and thus the number of neighbors of v. The out-degree of a vertex is defined as the number of all row entries in ma-

trix A > 0 and the in-degree of vertex v is defined as the number of all column entries in matrix A > 0.
The adjacency matrix of G is a n × n matrix A with the entries

$$a_{i,j} = \text{number of edges with start node } v_i \text{ and end node } v_j$$

and for representing the different geoprocesses the interlayer relations are valued by 2 or 3, as depicted in fig.3.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	ID	A	B	C	O0	O1t1	Mt1	B1t1	B2t1	Ct1	Ot2	M1t2	M2t2	B1t2	B2t2	Ct2	Ot3	M1t3	M2t3	M3t3	B1t3	B2t3	Ct3
2	A	0	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	B	1	0	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	C	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	O0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	O1t1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Mt1	0	0	0	0	1	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0
8	B1t1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	B2t1	0	0	0	0	1	1	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0
10	Ct1	0	0	0	0	1	0	1	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0
11	Ot2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	2	0	0	0
12	M1t2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	2	0	0	2	0	0	0
13	M2t2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	2	0	0	2	0	0	0
14	B1t2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2	0	0	2	0	0	0
15	B2t2	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	2	0	0	2	0	0	0
16	Ct2	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	2	0	0	2	0	0	0
17	Ot3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
18	M1t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
19	M2t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
20	M3t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1
21	B1t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
22	B2t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
23	Ct3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0

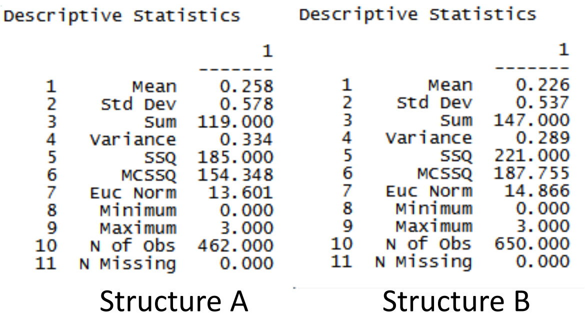
Fig. 3. Valued adjacency matrix of landform A where a 2 denotes the erosion geoprocess and a 3 denotes the erosion and accumulation geoprocess

Node level analysis

For doing network analysis at the node level we have the possibility to run a series of so-called ego level network statistics, (e.g., degree, betweenness centrality, constraint, etc.) potentially including the node attributes. Ego networks consist of a focal node ("ego") and the nodes to which ego are directly linked to.

Betweeness centrality is defined as the share of times that a node x needs a node y (whose centrality is being measured) in order to reach a node z via the shortest path.

Constraint is a summary measure that taps the extent to which ego's connections are to others who are connected to one another.
The nodes (layers) of our landforms can be ranked based on these measures and the ranking can be compared from one landform structure to the next.



```
Descriptive Statistics

                         1
                   -------
 1         Mean      0.258
 2      Std Dev      0.578
 3          Sum    119.000
 4     Variance      0.334
 5          SSQ    185.000
 6        MCSSQ    154.348
 7     Euc Norm     13.601
 8      Minimum      0.000
 9      Maximum      3.000
10     N of Obs    462.000
11    N Missing      0.000
```

Structure A

```
Descriptive Statistics

                         1
                   -------
 1         Mean      0.226
 2      Std Dev      0.537
 3          Sum    147.000
 4     Variance      0.289
 5          SSQ    221.000
 6        MCSSQ    187.755
 7     Euc Norm     14.866
 8      Minimum      0.000
 9      Maximum      3.000
10     N of Obs    650.000
11    N Missing      0.000
```

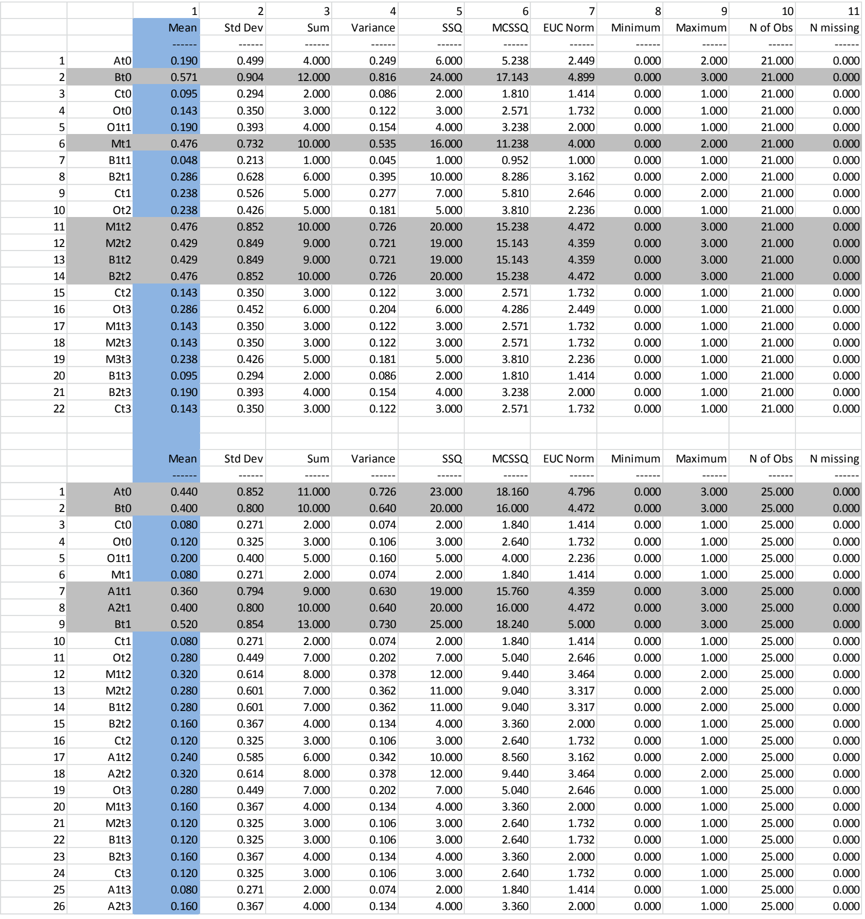
Structure B

Fig. 4. descriptive statistics for both structures

As one can see, structure B has more relations (650) than structure A (462), ranging from a minimum score of zero to a maximum score of 3. Thus, both structures are following the same inter- and intralayer semantic, where:

- a zero-relation means no adjacency relation,
- a one-relation means adjacency
- a two-relation means simple geoprocess relation – erosion
- a three-relation means complex geoprocess relation – erosion and accumulation.

The network density of both is nearly similar. Thus, the proportions of possible ties that are present in the network are 25.8 % for structure A and 22.6 % for structure B. Since we have asymmetric relations (directed in terms of geoprocesses) we are able to examine and compare the sending and receiving statistics for both structures. The sending statistics summarizes the rows of our incidence matrices and indicates the layers roles within the structure. As we can see there are differences within the statistics. We see that the layers Bt0, Mt1, M1t2, M2t2, B1t2, and B2T2 have a mean (or density) of tie sending of 42.9 % - 57.1 % in structure A and if they exist in structure B they have a mean of tie sending of 28 % - 52 %. That is, this actor sent four ties to the available nine other actors. In scanning down the column (in figure sending stats) or row (in figure receiving stats) of means, we note that there is quite a bit of variability across the layers – some send more and get more material (or are part of a geoprocess) than others.

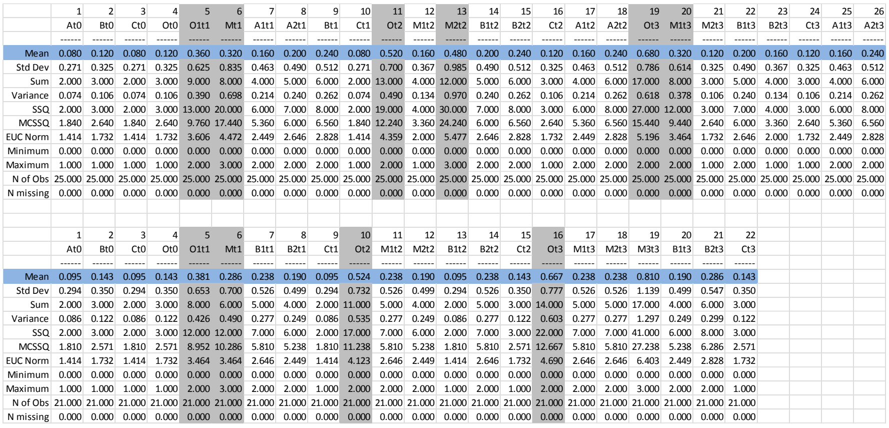


		1	2	3	4	5	6	7	8	9	10	11
		Mean	Std Dev	Sum	Variance	SSQ	MCSSQ	EUC Norm	Minimum	Maximum	N of Obs	N missing
		------	------	------	------	------	------	------	------	------	------	------
1	At0	0.190	0.499	4.000	0.249	6.000	5.238	2.449	0.000	2.000	21.000	0.000
2	Bt0	0.571	0.904	12.000	0.816	24.000	17.143	4.899	0.000	3.000	21.000	0.000
3	Ct0	0.095	0.294	2.000	0.086	2.000	1.810	1.414	0.000	1.000	21.000	0.000
4	Ot0	0.143	0.350	3.000	0.122	3.000	2.571	1.732	0.000	1.000	21.000	0.000
5	O1t1	0.190	0.393	4.000	0.154	4.000	3.238	2.000	0.000	1.000	21.000	0.000
6	Mt1	0.476	0.732	10.000	0.535	16.000	11.238	4.000	0.000	2.000	21.000	0.000
7	B1t1	0.048	0.213	1.000	0.045	1.000	0.952	1.000	0.000	1.000	21.000	0.000
8	B2t1	0.286	0.628	6.000	0.395	10.000	8.286	3.162	0.000	2.000	21.000	0.000
9	Ct1	0.238	0.526	5.000	0.277	7.000	5.810	2.646	0.000	2.000	21.000	0.000
10	Ot2	0.238	0.426	5.000	0.181	5.000	3.810	2.236	0.000	1.000	21.000	0.000
11	M1t2	0.476	0.852	10.000	0.726	20.000	15.238	4.472	0.000	3.000	21.000	0.000
12	M2t2	0.429	0.849	9.000	0.721	19.000	15.143	4.359	0.000	3.000	21.000	0.000
13	B1t2	0.429	0.849	9.000	0.721	19.000	15.143	4.359	0.000	3.000	21.000	0.000
14	B2t2	0.476	0.852	10.000	0.726	20.000	15.238	4.472	0.000	3.000	21.000	0.000
15	Ct2	0.143	0.350	3.000	0.122	3.000	2.571	1.732	0.000	1.000	21.000	0.000
16	Ot3	0.286	0.452	6.000	0.204	6.000	4.286	2.449	0.000	1.000	21.000	0.000
17	M1t3	0.143	0.350	3.000	0.122	3.000	2.571	1.732	0.000	1.000	21.000	0.000
18	M2t3	0.143	0.350	3.000	0.122	3.000	2.571	1.732	0.000	1.000	21.000	0.000
19	M3t3	0.238	0.426	5.000	0.181	5.000	3.810	2.236	0.000	1.000	21.000	0.000
20	B1t3	0.095	0.294	2.000	0.086	2.000	1.810	1.414	0.000	1.000	21.000	0.000
21	B2t3	0.190	0.393	4.000	0.154	4.000	3.238	2.000	0.000	1.000	21.000	0.000
22	Ct3	0.143	0.350	3.000	0.122	3.000	2.571	1.732	0.000	1.000	21.000	0.000

		Mean	Std Dev	Sum	Variance	SSQ	MCSSQ	EUC Norm	Minimum	Maximum	N of Obs	N missing
		------	------	------	------	------	------	------	------	------	------	------
1	At0	0.440	0.852	11.000	0.726	23.000	18.160	4.796	0.000	3.000	25.000	0.000
2	Bt0	0.400	0.800	10.000	0.640	20.000	16.000	4.472	0.000	3.000	25.000	0.000
3	Ct0	0.080	0.271	2.000	0.074	2.000	1.840	1.414	0.000	1.000	25.000	0.000
4	Ot0	0.120	0.325	3.000	0.106	3.000	2.640	1.732	0.000	1.000	25.000	0.000
5	O1t1	0.200	0.400	5.000	0.160	5.000	4.000	2.236	0.000	1.000	25.000	0.000
6	Mt1	0.080	0.271	2.000	0.074	2.000	1.840	1.414	0.000	1.000	25.000	0.000
7	A1t1	0.360	0.794	9.000	0.630	19.000	15.760	4.359	0.000	3.000	25.000	0.000
8	A2t1	0.400	0.800	10.000	0.640	20.000	16.000	4.472	0.000	3.000	25.000	0.000
9	Bt1	0.520	0.854	13.000	0.730	25.000	18.240	5.000	0.000	3.000	25.000	0.000
10	Ct1	0.080	0.271	2.000	0.074	2.000	1.840	1.414	0.000	1.000	25.000	0.000
11	Ot2	0.280	0.449	7.000	0.202	7.000	5.040	2.646	0.000	1.000	25.000	0.000
12	M1t2	0.320	0.614	8.000	0.378	12.000	9.440	3.464	0.000	2.000	25.000	0.000
13	M2t2	0.280	0.601	7.000	0.362	11.000	9.040	3.317	0.000	2.000	25.000	0.000
14	B1t2	0.280	0.601	7.000	0.362	11.000	9.040	3.317	0.000	2.000	25.000	0.000
15	B2t2	0.160	0.367	4.000	0.134	4.000	3.360	2.000	0.000	1.000	25.000	0.000
16	Ct2	0.120	0.325	3.000	0.106	3.000	2.640	1.732	0.000	1.000	25.000	0.000
17	A1t2	0.240	0.585	6.000	0.342	10.000	8.560	3.162	0.000	2.000	25.000	0.000
18	A2t2	0.320	0.614	8.000	0.378	12.000	9.440	3.464	0.000	2.000	25.000	0.000
19	Ot3	0.280	0.449	7.000	0.202	7.000	5.040	2.646	0.000	1.000	25.000	0.000
20	M1t3	0.160	0.367	4.000	0.134	4.000	3.360	2.000	0.000	1.000	25.000	0.000
21	M2t3	0.120	0.325	3.000	0.106	3.000	2.640	1.732	0.000	1.000	25.000	0.000
22	B1t3	0.120	0.325	3.000	0.106	3.000	2.640	1.732	0.000	1.000	25.000	0.000
23	B2t3	0.160	0.367	4.000	0.134	4.000	3.360	2.000	0.000	1.000	25.000	0.000
24	Ct3	0.120	0.325	3.000	0.106	3.000	2.640	1.732	0.000	1.000	25.000	0.000
25	A1t3	0.080	0.271	2.000	0.074	2.000	1.840	1.414	0.000	1.000	25.000	0.000
26	A2t3	0.160	0.367	4.000	0.134	4.000	3.360	2.000	0.000	1.000	25.000	0.000

Fig. 5. Sending statistics for structure A and structure B

The receiving statistics summarizes the columns of the incidence matrices and indicates the relation receiving of each layer. Interpreting the result of the receiving statistics (fig. 4) we note that especially in structure B more layers and different are involved in the genesis of that landform. Thus, the receiving statistics of specific layers, such as M2t2 are quite different, whereas others such as Mt1, M1t3, M1t2 are not. That indicates, part of the genesis of both structures might be very similar whereas at time t_1 something changed within the genesis of the structures.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	At0	Bt0	Ct0	Ot0	O1t1	Mt1	A1t1	A2t1	Bt1	Ct1	Ot2	M1t2	M2t2	B1t2	B2t2	Ct2	A1t2	A2t2	Ot3	M1t3	M2t3	B1t3	B2t3	Ct3	A1t3	A2t3
	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------
Mean	0.080	0.120	0.080	0.120	0.360	0.320	0.160	0.200	0.240	0.080	0.520	0.160	0.480	0.200	0.240	0.120	0.160	0.240	0.680	0.320	0.120	0.200	0.160	0.120	0.160	0.240
Std Dev	0.271	0.325	0.271	0.325	0.625	0.835	0.463	0.490	0.512	0.271	0.700	0.367	0.985	0.490	0.512	0.325	0.463	0.512	0.786	0.614	0.325	0.490	0.367	0.325	0.463	0.512
Sum	2.000	3.000	2.000	3.000	9.000	8.000	4.000	5.000	6.000	2.000	13.000	4.000	12.000	5.000	6.000	3.000	4.000	6.000	17.000	8.000	3.000	5.000	4.000	3.000	4.000	6.000
Variance	0.074	0.106	0.074	0.106	0.390	0.698	0.214	0.240	0.262	0.074	0.490	0.134	0.970	0.240	0.262	0.106	0.214	0.262	0.618	0.378	0.106	0.240	0.134	0.106	0.214	0.262
SSQ	2.000	3.000	2.000	3.000	13.000	20.000	6.000	7.000	8.000	2.000	19.000	4.000	30.000	7.000	8.000	3.000	6.000	8.000	27.000	12.000	3.000	7.000	4.000	3.000	6.000	8.000
MCSSQ	1.840	2.640	1.840	2.640	9.760	17.440	5.360	6.000	6.560	1.840	12.240	3.360	24.240	6.000	6.560	2.640	5.360	6.560	15.440	9.440	2.640	6.000	3.360	2.640	5.360	6.560
EUC Norm	1.414	1.732	1.414	1.732	3.606	4.472	2.449	2.646	2.828	1.414	4.359	2.000	5.477	2.646	2.828	1.732	2.449	2.828	5.196	3.464	1.732	2.646	2.000	1.732	2.449	2.828
Minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Maximum	1.000	1.000	1.000	1.000	2.000	3.000	2.000	2.000	2.000	1.000	2.000	1.000	3.000	2.000	2.000	1.000	2.000	2.000	2.000	2.000	1.000	2.000	1.000	1.000	2.000	2.000
N of Obs	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000
N missing	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	At0	Bt0	Ct0	Ot0	O1t1	Mt1	B1t1	B2t1	Ct1	Ot2	M1t2	M2t2	B1t2	B2t2	Ct2	Ot3	M1t3	M2t3	M3t3	B1t3	B2t3	Ct3
	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------
Mean	0.095	0.143	0.095	0.143	0.381	0.286	0.238	0.190	0.095	0.524	0.238	0.190	0.095	0.238	0.143	0.667	0.238	0.238	0.810	0.190	0.286	0.143
Std Dev	0.294	0.350	0.294	0.350	0.653	0.700	0.526	0.499	0.294	0.732	0.526	0.499	0.294	0.526	0.350	0.777	0.526	0.526	1.139	0.499	0.547	0.350
Sum	2.000	3.000	2.000	3.000	8.000	6.000	5.000	4.000	2.000	11.000	5.000	4.000	2.000	5.000	3.000	14.000	5.000	5.000	17.000	4.000	6.000	3.000
Variance	0.086	0.122	0.086	0.122	0.426	0.490	0.277	0.249	0.086	0.535	0.277	0.249	0.086	0.277	0.122	0.603	0.277	0.277	1.297	0.249	0.299	0.122
SSQ	2.000	3.000	2.000	3.000	12.000	12.000	7.000	6.000	2.000	17.000	7.000	6.000	2.000	7.000	3.000	22.000	7.000	7.000	41.000	6.000	8.000	3.000
MCSSQ	1.810	2.571	1.810	2.571	8.952	10.286	5.810	5.238	1.810	11.238	5.810	5.238	1.810	5.810	2.571	12.667	5.810	5.810	27.238	5.238	6.286	2.571
EUC Norm	1.414	1.732	1.414	1.732	3.464	3.464	2.646	2.449	1.414	4.123	2.646	2.449	1.414	2.646	1.732	4.690	2.646	2.646	6.403	2.449	2.828	1.732
Minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Maximum	1.000	1.000	1.000	1.000	2.000	3.000	2.000	2.000	1.000	2.000	2.000	2.000	1.000	2.000	1.000	2.000	2.000	2.000	3.000	2.000	2.000	1.000
N of Obs	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000	21.000
N missing	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Fig. 6. receiving statistics for structure A and structure B

Paired Samples

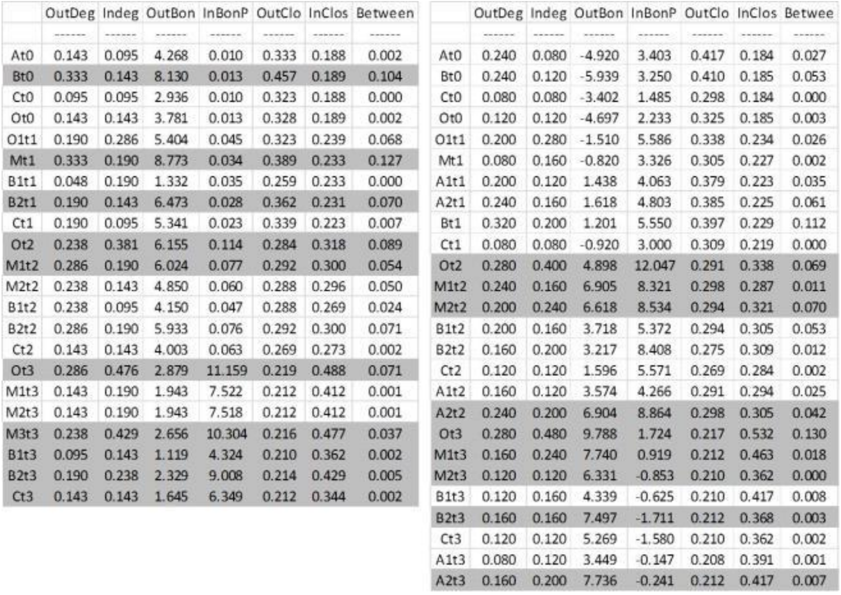
A paired samples test consists of the matched pairs of similar units, in our case the layers of both structures. By using a t-test the structure A is compared against structure B with the null hypothesis that there is no difference between both structures. The result gives the density of both matrices together with the difference, followed by a classical t-test. The estimated bootstrap standard errors are then reported together with the bootstrap standard error of the differences, the bootstrap 95% confidence intervals and the bootstrap t-statistic assuming independent samples. The bootstrap standard error, confidence interval, t-statistic and average value are then reported for the paired samples. Finally the proportion of differences (absolute, as large as and as small as) to the observed values are given:

Density of A.##h is: 0.1831
Density of B.##h is: 0.2262
Difference in density is: -0.0431
Number of bootstrap samples: 5000
Variance of ties for A.##h: 0.2515
Variance of ties for B.##h: 0.2893
Classical standard error of difference: 0.0288
Classical t-test (indep samples): -1.4934
Estimated bootstrap standard error for density of A.##h: 0.0395
Estimated bootstrap standard error for density of B.##h: 0.0379
Bootstrap standard error of the difference (indep samples): 0.0548
95% confidence interval for the difference (indep samples): [-0.1504, 0.0643]
bootstrap t-statistic (indep samples): -0.7866
Average bootstrap difference: -0.0408
Proportion of absolute differences as large as observed: 0.3219
Proportion of differences as large as observed: 0.8420
Proportion of differences as small as observed: 0.1582.

As we can see from the results, the both structures are nearly similar.

Centrality Measures

In order to determine the relative importance of a node or layer within our graph structure we use different centrality measures. The most popular ones are degree, betweenness, closeness and eigenvector. Those measures give us an idea about the importance of a layer for the genesis of that specific landform structure. Initially we used degree, Bonacich centrality, closeness and betweenness as a starting point for analysis (c.f. fig 5). Bonacich (1987) proposed a modification of the degree centrality, where centrality and power were a function of the connections of the node in node's neighborhood. The more connections the node in its neighborhood has, the more central it is. The fewer the connections of the nodes in the node's neighborhood, the more powerful is the node. The closeness centrality approach emphasizes the distance of a node to all others in the network by focusing on the distance from each node to all others.



	OutDeg	Indeg	OutBon	InBonP	OutClo	InClos	Between
	------	------	------	------	------	------	------
At0	0.143	0.095	4.268	0.010	0.333	0.188	0.002
Bt0	0.333	0.143	8.130	0.013	0.457	0.189	0.104
Ct0	0.095	0.095	2.936	0.010	0.323	0.188	0.000
Ot0	0.143	0.143	3.781	0.013	0.328	0.189	0.002
O1t1	0.190	0.286	5.404	0.045	0.323	0.239	0.068
Mt1	0.333	0.190	8.773	0.034	0.389	0.233	0.127
B1t1	0.048	0.190	1.332	0.035	0.259	0.233	0.000
B2t1	0.190	0.143	6.473	0.028	0.362	0.231	0.070
Ct1	0.190	0.095	5.341	0.023	0.339	0.223	0.007
Ot2	0.238	0.381	6.155	0.114	0.284	0.318	0.089
M1t2	0.286	0.190	6.024	0.077	0.292	0.300	0.054
M2t2	0.238	0.143	4.850	0.060	0.288	0.296	0.050
B1t2	0.238	0.095	4.150	0.047	0.288	0.269	0.024
B2t2	0.286	0.190	5.933	0.076	0.292	0.300	0.071
Ct2	0.143	0.143	4.003	0.063	0.269	0.273	0.002
Ot3	0.286	0.476	2.879	11.159	0.219	0.488	0.071
M1t3	0.143	0.190	1.943	7.522	0.212	0.412	0.001
M2t3	0.143	0.190	1.943	7.518	0.212	0.412	0.001
M3t3	0.238	0.429	2.656	10.304	0.216	0.477	0.037
B1t3	0.095	0.143	1.119	4.324	0.210	0.362	0.002
B2t3	0.190	0.238	2.329	9.008	0.214	0.429	0.005
Ct3	0.143	0.143	1.645	6.349	0.212	0.344	0.002

	OutDeg	Indeg	OutBon	InBonP	OutClo	InClos	Betwee
	------	------	------	------	------	------	------
At0	0.240	0.080	-4.920	3.403	0.417	0.184	0.027
Bt0	0.240	0.120	-5.939	3.250	0.410	0.185	0.053
Ct0	0.080	0.080	-3.402	1.485	0.298	0.184	0.000
Ot0	0.120	0.120	-4.697	2.233	0.325	0.185	0.003
O1t1	0.200	0.280	-1.510	5.586	0.338	0.234	0.026
Mt1	0.080	0.160	-0.820	3.326	0.305	0.227	0.002
A1t1	0.200	0.120	1.438	4.063	0.379	0.223	0.035
A2t1	0.240	0.160	1.618	4.803	0.385	0.225	0.061
Bt1	0.320	0.200	1.201	5.550	0.397	0.229	0.112
Ct1	0.080	0.080	-0.920	3.000	0.309	0.219	0.000
Ot2	0.280	0.400	4.898	12.047	0.291	0.338	0.069
M1t2	0.240	0.160	6.905	8.321	0.298	0.287	0.011
M2t2	0.200	0.240	6.618	8.534	0.294	0.321	0.070
B1t2	0.200	0.160	3.718	5.372	0.294	0.305	0.053
B2t2	0.160	0.200	3.217	8.408	0.275	0.309	0.012
Ct2	0.120	0.120	1.596	5.571	0.269	0.284	0.002
A1t2	0.160	0.120	3.574	4.266	0.291	0.294	0.025
A2t2	0.240	0.200	6.904	8.864	0.298	0.305	0.042
Ot3	0.280	0.480	9.788	1.724	0.217	0.532	0.130
M1t3	0.160	0.240	7.740	0.919	0.212	0.463	0.018
M2t3	0.120	0.120	6.331	-0.853	0.210	0.362	0.000
B1t3	0.120	0.160	4.339	-0.625	0.210	0.417	0.008
B2t3	0.160	0.160	7.497	-1.711	0.212	0.368	0.003
Ct3	0.120	0.120	5.269	-1.580	0.210	0.362	0.002
A1t3	0.080	0.120	3.449	-0.147	0.208	0.391	0.001
A2t3	0.160	0.200	7.736	-0.241	0.212	0.417	0.007

Fig. 7. Comparison of both centrality results of both structures

As shown in fig. 7 the outer nodes to each point in time belong to the group of the most important intermediary nodes. This is because all the change of material (all geoprocesses) are using the outer nodes as intermediate step during the genesis of the landform. Moreover the nodes which are transformed during the genesis of the landform (M1t1, M1t2, B2t3, M1t3, M3t3) are also considered as being central in both structures.

Dyadic level analysis

For dyadic level analyses, we can use statistical analyses, such as Exponential random graph models (ERGM) or quadratic assignment procedure correlation (QAP-correlation). Univariate ERGMs can be used to compare network of the same or of different size. QAP-Correlation calculates measures of nominal, ordinal, and interval association between the relations in two matrices. Since both methods are strongly related to social network analysis and the usefulness of applying those methods for our graph comparison, we decided to start with QAP Analysis for comparing our graph structures. This analysis is used to test the association between graphs. Here, one graph is a observed one while the other is a model or expected graph we would like to test against our similarity theory. In this case the Pearsson-Correlation is calculated which is a dimensionless measure for a linear relation between a minimum of two features. The correlation is between -1 and +1, while +1 means a complete positive linear relationship and -1 a complete negative relationship between the two features. If the correlation is 0 there is no linear relation. Since the dimension of our matrices (**dim A = 22** and **dim B = 26)** of the both structures are not equal, we have to render the matrix A by 4 rows and 4 columns. As a result, both matrices have the same dimension and can be compared with each other by applying the QAP-correlation analysis. As table 2 shows, the QAP analysis result in an overall correlation of 31.5 % of the structures.

Table 2. QAP Correlation of structure A and structure B

	A	B
A	1.000	0.315
B	0.315	1.000

5 Discussion

In this contribution we demonstrated the graph based analysis and comparison of two related but different landforms. In a first step the landform layers where transformed to several graphs applying the Poincaré Duality. Then change of landform layers from one distinct state of time to another is represented by adding directed edges. Multilayered graphs of both landforms have been analyzed and compared by different graph based techniques.

Node level analysis showed that network density of both multilayered graphs is nearly similar. The same accounts for the Paired Samples. However, centrality measures, i.e. Bonacich centrality of identic nodes in both graph structures revealed a much higher centrality of structure B. This could give a hint on different landform evolution. Since the A-horizon of landform is totally eroded form t0 to t1, this node cannot further contribute to centrality. A high centrality value therefore can be viewed as a dominant reworking but not erasing of landform layers.

Finally, dyadic level analysis results in a correlation of 31.5 % of both structures. Thus, similarity is far away from being equal.

Results of graph based analysis and comparison may help to search for and identify similar landforms in a data base structure. Geometrical analysis could hardly be efficient in such a way, since geometry often is not available.

Nevertheless, we are still far away from our target of automatically find, compare and predict landform behavior and their evolution. Open questions are:

- What influence does the render process of matrices have when performing valid dynamic level analysis?
- How can differences better be identified in both, dual topological relationships and in process driven change?
- Are centrality measures sufficient for a professional evaluation of a landform and its history, or
- are the applied methods only helpful in finding similar develloped landforms for further investigation?

Positive reply on the latter question would mean that the approach demonstrated here is suitable to search for similar mature landforms.

References

Becker T, Nagel C, Kolbe TH (2008) A Multilayered Space-Event Model for Navigation in Indoor Spaces. In: Lee J, Zlatanova S (eds) 3D Geoinformation Sciences. Lecture Notes in Geoinformation and Catrography, pp 61–77

Blanco-Canqui H, Lal R (2008) Tillage erosion. In: Blanco H, Lal R (eds) Principles of Soil Conservation and Management, p 109–135

Bonacich P (1987) Power and centrality: A family of measures. American journal of sociology 92(5):1170–1182

Bork H-R (1988) Bodenerosion und Umwelt. Verlauf, Ursachen und Folgen der mittelalterlichen und neuzeitlichen Bodenerosion. Landschaftsgenese und Landschaftsökologie, vol 13.

Bork H-R, Bork H, Dalchow C, Faust B, Piorr HP, Schatz T (1998) Landschaftsentwicklung in Mitteleuropa. Klett-Perthes, Stuttgart

Brunsden D (1996) Geomorphological events and landform change. Z Geomorph NF Suppl-Bd 40:273–288

Brunsden D, Kesel RH (1973) The evolution of a Mississippi river bluff in historic time. Journal of Geology 81:576–597

Chorley RJ, Kennedy BA (1971) Physical geography – a system approach. Prentice-Hall, London

Dikau R (1996) Geomorphologische Reliefklassifikation und -analyse. Heidelberger Geographische Arbeiten, vol 104, pp 15–23

Dikau R (1999) The need for field evidence in modelling landform evolution. In: Hergarten S, Neugebauer HJ (eds) Process modelling and landform evolution, Lecture Notes in Earth Sciences, Springer, Heidelberg, 3–12

Diestel, R (2005) Graph theory, Springer-Verlag Heidelberg 1996, 2005

Egenhofer MJ (1989) A formal definition of binary topological relationships, Lecture Notes in Computer Science, 457–472

Egenhofer MJ, Herring JR (1990) A mathematical framework for the definition of topological relations. In: Proceedings of the Fourth International Symposium on SDH, Zurich, Switzerland, pp 803-813

Heckmann T, Schwanghart W (2013) Geomorphic coupling and sediment connectivity in an alpine catchment exploring sediment cascades using graph theory. Geomorphology 182: 89–103

Herring J (2001) The OpenGIS abstract specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema), Version 5. OGC Document 01-101, p 168

Hugget RJ (2011) Fundamentals of Geomorphology. Routledge, London

Kugler H (1974) Das Georelief und seine kartographische Modellierung. Dissertation B, Martin-Luther-Universität Halle

Lake R, Burggraf DS, Trininic M, Rae L (2004) GML - Geography Mark-Up Language. John Wiley & Sons, Chichester, p 388

Lee J (2004) A spatial access oriented implementation of a 3-D GIS topological data model for urban entities. Journal Geoinformatica 8(3):237–264

Löwner M-O (2000) Geophysikalische und sedimentologische Untersuchungen zu Sedimentspeichern auf Gut Frankenforst (Investigations of sediment storages at property Frankenforst using geophysical and sedimentological methods). Diploma thesis, University of Bonn, Germany

Löwner, M-O, 2005: Semantische Modellierung und Repräsentation geomorphologischer Objekte in einem geographischen Informationssystem (GIS) unter besonderer Berücksichtigung von Wänden und steilen Hangbereichen. (Semantical modelling and representation of geomorphic objects in GIS with emphasis on walls and steep slopes). Dissertation University of Bonn, Germany

Löwner M-O, Dörschlag D, Plümer L (2005A) GeoSaMT – ein Konzept zur Kartierung und semantischen Modellierung von steilen Hangbereichen (GeoSaMT - A Concept for mapping and modeling steep slopes by the use of terrestrial photographs). In: Photogrammetrie Fernerkundung Geoinformation 4/2005:281–290

Löwner M-O, Preston, NJ, Dikau, R (2005B) Reconstruction of a colluvial body using geoelectrical resistivity. Z Geomorph, N. F. 49(2):225–238.

Löwner M-O, (2013A) 3D topological relationships of landforms and their Spatial Schema based representation. Geo-spatial Information Science 16 (4):238–246

Löwner M-O (2013B) On problems and benefits of 3D topology on under-specified geometries in geomorphology. In: Pouliot J, Daniel S, Hubert F, Zamyadi A (eds) Progress and New Trends in 3D Geoinformation Sciences. Lecture Notes in Geoinformation and Cartography, p 155-170

Löwner M-O (2010) New GML-based application schema for landforms, processes and their interaction. In: Otto, J-C, Dikau R (eds.) Landform – Structure, Evolution, Process Control. Lecture Notes in Earth Sciences 115,pp 21–36

Löwner M-O, Otto JC (2008) Towards an automatic identification of sediment cascades from geomorphological maps using graph theory. In: Proceedings of the International Symposium on Sediment dynamics in changing environments, Christchurch, New Zealand, 1-5 December 2008

Löwner M-O, Becker T (2013) A topological framework for the temporal aspects of landform development. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-2/W1, 237-246, doi:10.5194/isprsannals-II-2-W1-237-2013

Lindstrom M, Nelson W, Schumacher T (1992) Quantifying tillage erosion rates due to mouldboard ploughing. Soil and Tillage Research 24:243–255

Luoto M, Hjort J (2005) Evaluation of current statistical approaches for predictive geomorphological mapping. Geomorphology 67:299-315

Morgan RPC (2004) Soil Erosion and Conservation, Wiley-Blackwell

Munkres JR (1984) Elements of Algebraic Topology. Addison-Wesley, Menlo Park, CA

Preston, NJ, 2001: Geomorphic Response to Environmental Change: the Imprint of Deforestation and Agricultural Land Use on the Contemporary, Landscape of the Pleiser Hügelland, Bonn, Germany. Dissertation, University of Bonn, Germany

Smith B (2001) Fiat Objects, Topoi 20 (2):131–148

Welch DM (1970) Substitution of space for time in a study of slope development. Journ Geology 78:234–239

Zlatanova S (2000) On 3D topological relationships. In: Proceedings of the 11th International Workshop on Database and Expert System Applications (DEXA 2000), Greenwich, London, UK, 6-8 September 2000

Zlatanova S, Rahman AA, Shi W (2004) Topological models and frameworks for 3D spatial objects. Computers & Geosciences 30:419–428

The role, opportunities and challenges of 3D and geo-ICT in archaeology

B. De Roo [a, *], J. Bourgeois [b], P. De Maeyer [a]

a Dept. of Geography, Ghent University, Krijgslaan 281 (building S8), Ghent, Belgium - (berdien.deroo, philippe.demaeyer)@ugent.be

b Dept. of Archaeology, GhentUniversity, Sint-Pietersnieuwstraat 35, Ghent, Belgium – jean.bourgeois@ugent.be

Abstract Archaeology joins in the trend of three-dimensional (3D) data and geospatial information technology (geo-ICT). Currently, the spatial archaeological data acquired is 3D and mostly used to create realistic visualizations. Geographical information systems (GIS) are used for decades in archaeology. However, the integration of geo-ICT with 3D data still poses some problems. Therefore, this paper clarifies the current role of 3D, and the opportunities and challenges for 3D and geo-ICT in the domain of archaeology. The paper is concluded with a proposal to integrate both trends and tackle the outlined challenges. To provide a clear illustration of the current practices and the advantages and difficulties of 3D and geo-ICT in the specific case of archaeology, a limited case study is presented of two structures in the Altay Mountains.

Keywords: Archaeology, Data Standards, Data Exchange, 3D, GIS, GEO-ICT

1 Introduction

Three-dimensional (3D) information is having a rapid expansion in different areas. Employing these 3D data in existing practices offers new opportunities such as a more realistic overview of data and enhanced analyses. At the same time, however, this poses some challenges, like an increasing amount of data which has to be handled. Another trend is the increasing use of geospatial information and its integration with ICT. The latter is referred to as geo-ICT and includes among others geographical information systems (GIS). In this regard, one can also think of a growing amount of location-based applications for smartphones, resulting in the advantage that people become more familiar with spatial data.

In the archaeological domain, these two trends are also followed. Archaeological research projects are more and more using 3D techniques to reconstruct sites, and GIS have been part of archaeological research for decades (De Roo et al. 2013a; De Reu et al. 2013; Forte 2014).

[*] Corresponding author.

As archaeological excavations are destructive, many researchers aim for a complete 3D digital documentation of archaeological excavations (Katsianis et al. 2008; De Reu et al. 2013). Sharing this documentation will enable the transfer of knowledge between different parties occupied with archaeological data (e.g. policy makers, field archaeologists, tourists, etc.) (Shaw et al. 2009; McKeague et al. 2012). For the purpose of analysis and interpretation of 3D information, the application of a 3D GIS should be incorporated in this 3D documentation workflow. However, the creation of such a complete 3D digital workflow and the exchange of these data and documentation is challenging, e.g. due to the data structure and storage costs.

This paper intends to assess the current role of the third dimension in archaeology and to outline opportunities as well as challenges which need to be accepted when fully implementing 3D and geo-ICT in the archaeological process. Subsequently, a possible approach to the integration of 3D data and geo-ICT in archaeology is introduced. To illustrate clearly the role, opportunities and challenges of 3D and geo-ICT and to give a better insight in the proposed approach, a case study is included, which is shortly described in section 2.

2 The Yustyd Valley, Altay Mountains: Case study

2.1 Geographic location

Located on the border between South-Siberia and Central Asia, the Altay Mountains form part of the Eurasian Steppe, an area stretched over 5000 km. The climate of the Altay is more extreme than the typical continental climate in the Steppe. The Russian part of the Altay Mountains is situated in the Altay Republic, subject of the Russian Federation. The case study site lies in the Yustyd valley, in the east of the Kosh Agash district (Fig. 1).

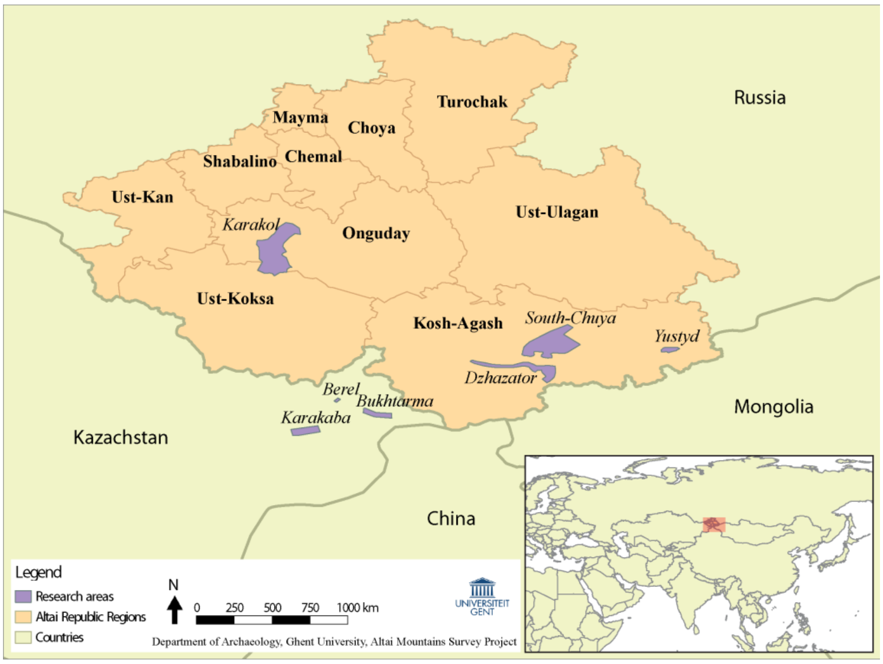


Fig. 1. Overview of the research areas in the Altai Republic

2.2 Archaeological setting

The Altai Mountain region and especially the Yustyd valley has an archaeological profusion (Plets et al. 2012). The lower and more flat part of the valley (2000-2200 m above sea level) is covered with a large concentration of archaeological structures (Gheyle 2009). Several campaigns have been organized to fully investigate the region.

For this paper only two of the more than 5100 found structures are considered, namely two so-called dwellings (Plets et al. 2012). These are geometric structures consisting of a single or double line of stones in rectangular or circular formation and a rectangular pavement in the center (Gheyle 2009; Plets et al. 2012). Based on Optically Stimulated Luminescence (OSL) technique, the two structures can be dated back to a period around 2000-1600 BC. Although the function remains unsure, they may be seen as virtual burials to remember a death and give this person a house for the next world (Jacobson-Tepfer 2008). The two considered dwellings, KA-771-016 and KA-771-006, are of the most simple typology (Fig. 2).

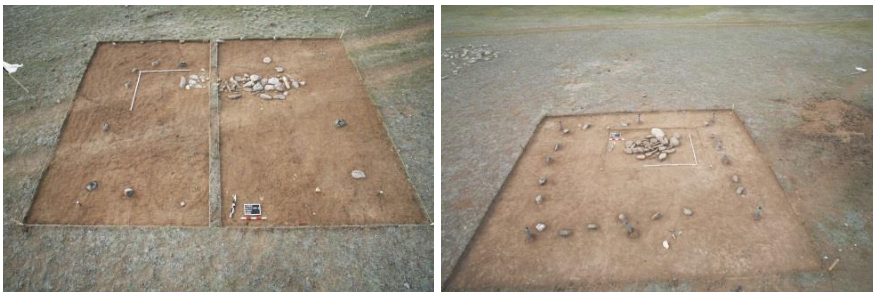


Fig. 2. Two dwellings found in the Yustyd valley: KA-771-006 (left) and KA-771-016 (right) (Department of Archaeology, Ghent University, Altay Mountains Survey Project)

These two dwellings were excavated in July 2011 to make a study of the chronology, typology and function possible (Plets et al. 2012). An accurate recording and documentation of the dwellings and the other structures in this area is highly important, since multiple factors threaten this archaeological remains. Seasonal river activity, vegetation and bioturbation are some of the natural threats, while increasing tourism, thefts, growing industry and car tracks are the most harmful effects caused by humans (Gheyle 2009). The two dwellings of this case study as well have been partly damaged by car tracks.

3 Current role of 3D in archaeology

The third dimension is naturally linked to archaeological data. On the one hand, archaeological objects have, like all real-world objects, a three-dimensional and mostly complex irregular shape. On the other hand, the excavated objects are found in the 3D space, which mostly implies the issue of depth. Both aspects of three-dimensionality play a substantial role in the analysis and interpretation process and therefore in the archaeological documentation. The shape of an object may for example give information on the functionality, whereas the depth or 3D spatial relation between finds reveal important temporal indications. The two dwellings consisting irregular shaped stones are found in the 3D space. Since the stones are found on a similar level and make up a regular shape, they are assumed to form one structure. In some cases, the shape of a stone may indicate a special function, e.g. an entrance, however, due to the simple type, this is not the case for our dwellings.

During the archaeological fieldwork, the location of finds and marks is recorded. This spatial data is nowadays acquired by means of land survey technologies, like total station and GPS. Born digital data is obtained which comprises not only the two horizontal dimensions but also the third, vertical dimension. Consequently, nearly all spatial archaeological data are 3D. The use of these digital sensors have also changed archaeological data recording in a quantitative way, since more data is gathered within a shorter period of time (De Reu et al. 2013; Stal et al. 2014). Formerly, GNSS measurements were used to record the internal corners of the dwellings (Plets et al. 2012). This resulted in 2D line drawings where the detailed configuration of the stones was not spatially acquired. Since the location of the individual stones is requisite for the study of the typology of the dwellings, more accurate registration was needed (Plets et al. 2012). Total station measurements were performed and during the excavation, site plans and sections are drawn on graph paper and scanned afterwards (Fig. 3).

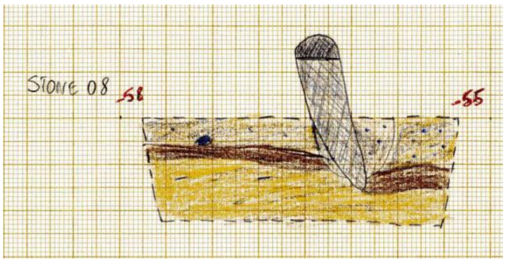


Fig. 3. Example of a scanned section drawing of stone 08 of KA-771-016 (Department of Archaeology, Ghent University, Altay Mountains Survey Project)

Although the spatial data is to a large extent available in 3D, two-dimensional representations still constitute the major deliverable. Either the vertical or one of the two horizontal dimensions is in this case ignored. When the combination of both horizontal and vertical dimensions is of importance a series of 2D maps are created (see Fig. 4), e.g. site plans for varying depths or a series of section drawings (Harris and Lock 1995). In the case of the dwelling, no major depth differences exist and thus, no series of horizontal maps is created. On the other hand, drawings or site plans are created during the different excavation phases, so showing different depths (Belien 2012). Several sections were drawn during the excavation, which result in a set of vertical maps, however, not in a linear relationship.

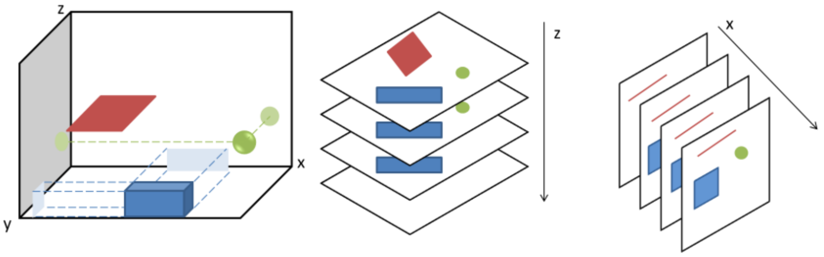


Fig. 4. 3D data as a series of 2D horizontal and 2D vertical maps

Nevertheless, by the use of 3D acquisition technologies as total station, GPS, laser scanning and digital photogrammetry an increase of 3D output products can be observed (Forte 2014; Stal et al. 2014). Digital elevation models, orthophotos and digital 3D models can be useful to support interpretations and geometric analysis, although the focus in the creation of these outcomes lays mainly on the visual representation (Forte 2014). In the summer of 2011, the two dwellings were photographed and using photo modelling and photogrammetry, orthophotos and 3D models were produced (Belien 2012; Plets et al. 2012). Based on the orthophotos digital site plans were reconstructed (Fig. 5).

The same 2D abstraction of the reality occurs when integrating the data in currently available GIS. Although in some cases the vertical dimension is stored as an attribute, an elaborate 3D analysis is impossible (Harris and Lock 1995; De Roo et al. 2013a). In conclusion, the final product of current excavations is generally a written report which includes some 2D plans of the site and list of the finds.

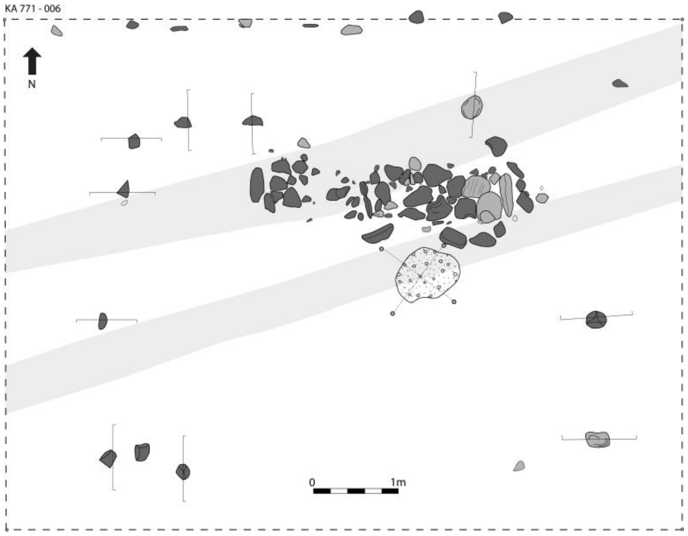


Fig. 5. Digital plan of dwelling KA-771-006, with location indication of the vertical sections. (Department of Archaeology, Ghent University, Altay Mountains Survey Project)

4 Opportunities of 3D and geo-ICT in archaeology

Considering the archaeological workflow in a very generalized way, it consists of a planning stage, the actual fieldwork, the analysis and interpretation and finally the reporting and communication. Due to the destructive nature of excavations, the precise documentation of the fieldwork is of indispensable importance. The resulting report in combination with the original acquired and processed data will be used as input for the planning of excavations in the future or form the basis for a detailed academic investigation. This way, a cyclic process exists. During the complete project cycle, the use of 3D information and geo-ICT can be profitable.

The planning stage includes among others the consultation of heritage inventories to assess the archaeological potentials of the project area. Such archaeological and cultural heritage inventories are usually maintained by (local) authorities as a tool for decision making in different fields. An inventory which gives a general overview of all archaeological and cultural heritage sites, comprises essential geospatial information and therefore, needs to be conceived as a geodatabase. If this database comprises the substantial metadata to understand the broad context of the site without giving detailed information on the precise findings, its use range can be strikingly increased and it can "support collaborative and even interdisciplinary research" (Labrador, 2012, p 241). Furthermore, a broader public can be reached by using web-based services, e.g. web maps. Since the case study is part of a research excavation, the planning stage mainly concentrates on the results of previous excavations by the institution itself and on the inventory maintained by the department of Archaeology of Ghent University. This database is called the Altai Archaeological Inventory (Altari) and maintained in Microsoft Access. A GIS project in ESRI ArcGIS is created, which links point layers for the sites and polygon layers for the individual structures to the Altari database. The database is therefore not a pure geodatabase. Web-based services would be advantageous for the project as well, since it is a collaboration between Ghent University and Gorno-Altaisk State University. Such a web-based services can facilitate the sharing and maintenance of an up-to-date geodatabase. Furthermore, a web-based service can be used in the promotion of the tourism in the region.

A geodatabase could provide advantages for excavation databases too. When the geodatabase is developed according to an archaeological data exchange standard, a common understanding and structure of the data set originates, which, for its part, would increase the data interoperability. This way, data can be "gathered once and used often" (McKeague et al., 2012, p. 49). Since spatial archaeological data are nearly always available in 3D, it is obvious to handle this third dimension in the geodatabase as well. The Altari database consists of data of both the site and the individual structures. Data on structures are similar on different sites because of a common table/form structure. However, the spatial data is not incorporated in the database.

The combined employment of 3D data and geo-ICT, mainly GIS, would offer favorable opportunities for the analysis. Although GIS are widely used in archaeology, the archaeological data complexity, i.e. 3D, temporal information and imperfection, makes an intensive application a difficult task. A GIS which is able to handle the three dimensions simultaneously and even incorporates the fourth, temporal, dimension would facilitate analysis and interpretation. One could think of performing geometric calculation, creating section cuttings, using colors to highlight or mask some elements, zooming and rotating to investigate the site form different viewpoints, executing 3D spatial analysis such as nearest neighbors, etc. Based on the GIS project in combination with the Altari database, 2D analyses are possible. Analyses and action which could be beneficial for the Altay Mountains Survey Project are 3D spatial analyses, for instance to investigate the orientation of the structures in combination with their depth. Since the dwellings are endangered by natural as well as human factors, their integration in a 3D GIS or in a web-map would enable revisiting and reinvestigated them.

5 Challenges of 3D and GEO-ICT in archaeology

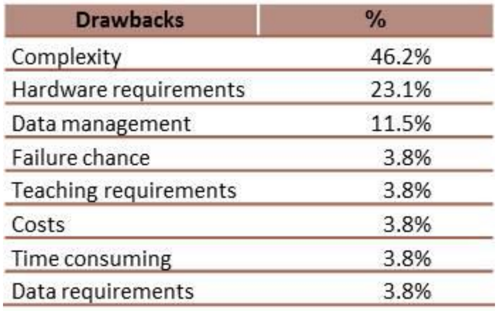
The exchange of archaeological data forms a major point of interest. The management of archaeological data is spread among various parties: academic researchers, national or local authorities, archaeological companies, heritage agencies, etc. (Wagtendonk et al. 2009; Labrador 2012; Huvila 2014). A similar list can be obtained for the parties who seek or use archaeological data. This results in the need for a common agreed-on data standard (Anichini et al. 2012; De Roo et al. 2013a). Data standards influence the consistency within databases, but assure as well a better data interoperability and exchange. A 3D GIS based on an accepted data model would further facilitate the data integration, spatio(-temporal) analyses and the understanding of the data. Developing such a standard demands considerable efforts. Due to the minor economic benefits (Green 2011), these struggles will not be taken by commercial software developers as a result of which the initiative needs to come from the scientific community. The data of the Altay Mountains Survey Project is used by different parties as well. On the one hand, Ghent University and Gorno-Altaisk State University acquire and maintain the data and mutually exchange them. On the other hand, local and national authorities, and other researchers may want to use the data for policy making or research as well. Considerable efforts have been made to integrate all the data form the Altay Mountains Survey Project by the creation of the Altari database and the GIS linkage (Gheyle 2009). Nevertheless, this database includes some elements which could hamper data consistency and querying functionalities, e.g. question marks are used to indicate unknown data. To facilitate data exchange, the input values for certain categories are written in both English and Russian. They are, however, stored in the same field, which may cause problems in query or other functionalities.

Second, a system suited to multiple purposes and audiences is required (McKeague et al. 2012; Huvila 2014). Due to the different objectives of the archaeological information providers and seekers, for all these groups their respective requirements need to be outlined. Together with the characterization of the use context, this forms an essential part of the human-centered design cycle, which is investigated by De Roo et al. (2013b) by means of a questionnaire. It is obvious that researchers make higher demands on the functionalities of a 3D geo-ICT system, then organizations do for touristic purposes. Furthermore, the planning of a pipeline will require accurate information on the location of the sites as well as on their respective importance and conservation state.

A third challenge concerns the collection and storage of 3D data. Even though acquiring 3D data during the fieldwork leads to a larger amount of data, topographic technologies record 3D data anyway. On the other hand, larger data volumes will result in higher storage requirements and thus higher costs. Therefore, the usefulness and challenges of incorporating 3D data and techniques need to be balanced against each other for each of the groups. To study the typology of the dwellings, accurate 3D documentation by means of photographs to produces 3D models fulfills the requirements (Plets et al. 2012). This detailed documentation has led to 2,85 GB and 1.85 GB of photos and scanned drawings for KA-771-006 and KA-771-016 respectively. Including the orthophotos of the different excavation stages resulted in 2,99 GB and 2.05 GB resp. A 3D reconstruction of the platform of dwelling KA-771-016 stored in a pdf-file required 8,71 MB of storage. Considering the area of the structures, approximately 21,5 m² and 17,4 m² respectively, and their simple typology, these numbers are quite high.

A fourth challenge is the necessary education and training on the use of new techniques and analyses. This risk can yet be reduced by keeping the system easy to use and comprehensible. By doing so, another issue is partly tackled, namely the risk to lapse into complexity. As shown in Table 1, this is identified by the users as major potential drawback of a 3D system (De Roo et al. 2013b).

Table 1. Drawbacks conceived of a 3D or 4D system (De Roo et al. 2013b)



Drawbacks	%
Complexity	46.2%
Hardware requirements	23.1%
Data management	11.5%
Failure chance	3.8%
Teaching requirements	3.8%
Costs	3.8%
Time consuming	3.8%
Data requirements	3.8%

6 Possible Approach to integrate 3D and geo-ICT

As described above, several challenges have to be accepted in order to reap the full benefits of 3D and geo-ICT in archaeology. The integration of 3D and geo-ICT during the complete archaeological project cycle would ideally result in a completely digital 3D workflow. Currently, 3D are mainly found in the first - acquisition - stage of the archaeological process and to a small extent also during the last – communication - phase, whereas Geo-ICT and especially GIS are employed in the middle - analysis and interpretation - stage. Extending the use of 3D to the middle stage and geo-ICT to the acquisition and communication phase will give rise to a complete digital 3D workflow. This division also exists in the Yustyd survey. The data are acquired in 3D using GNSS, total station and photo modelling, but then analyzed and interpreted using 2D GIS or site plans. 3D models were created from the two dwellings and the platforms. Although the database is used in the field as well, a linkage with GIS would even facilitate the acquisition stage.

Multiple parties dealing with archaeological data could be assisted by such a complete digital 3D workflow. For example, archaeological databases combined with a web-based GIS could blur the boundary between experts and leeks, and even involve a wider public (Labrador 2012). However, 3D is not always necessary or feasible for all parties dealing with archaeological data. As Huvila (2014) suggested an investigation of a supply-chain and customership of archaeological data and products is needed.

Since data exchange between various parties will be stimulated by a digital workflow, data interoperability is the major challenge. As explained in section 5, a data standard is desirable (De Roo et al. 2013a). Attention has to be given to existing international standards or data models from other research fields. One can think of CityGML, GeoSciML, etc. Implementing any links to those models will increase the usefulness for a broad range of parties interested in archaeological data.

We now turn to the potential approach for the integration of 3D and geo-ICT in a completely digital 3D archaeological workflow. The examination of an integrated object-oriented and data-driven approach for the basic structure of the database model is suggested (Fig. 6), since both approaches can directly be translated in an archaeology-specific relationship: object-space-time (Arroyo-Bishop and Lantada Zarzosa 1995) and place-people-event-stuff (Cripps 2012) respectively. The combination of objects, space and time is charactering for archaeological research, so it is obvious to handle those three elements simultaneously (Arroyo-Bishop and Lantada Zarzosa 1995). Besides, "archaeologists attempt to record and document the results of past events through a series of events or activities in the present" (Cripps et al. 2004, p.4). Such an approach, links places, people and objects by events and results in a place-people-event-stuff relationship as defined by (Cripps 2012).

First, we propose to use the event-oriented relationship as basis for the more general archaeological inventory (Fig. 6). Since this inventory will be used in the planning phase of an archaeological excavation or in policy decisions, for instance regarding land administration or spatial planning, no detailed analysis are requisite and the spatial information can be limited to 2D. This does not alter the possibility to incorporate, 3D reconstruction models for tourism purposes, conservation decisions, etc. As stated in section 4, we consider this inventory as a geodatabase, which includes essential metadata of the excavation or project. Detailed information on the excavation findings will be separated from the inventory, but kept traceable. The latter can be realized by, for example, including the contact details of the archaeological project manager or a link to the electronic deposit location.

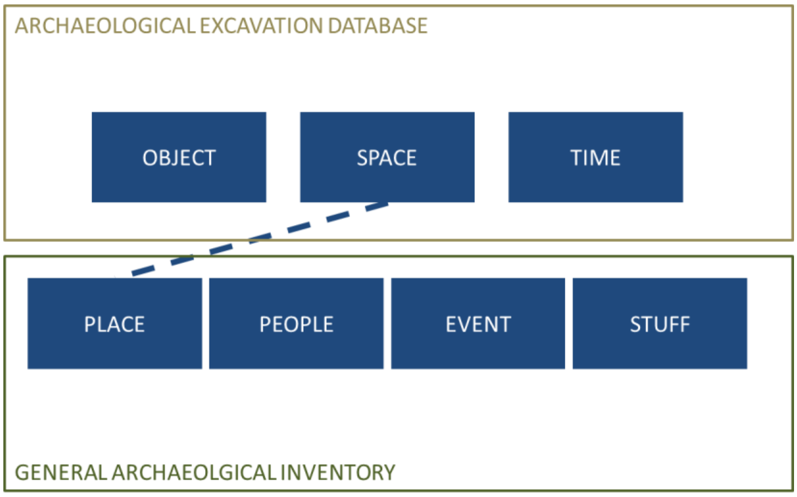

ARCHAEOLOGICAL EXCAVATION DATABASE
OBJECT
SPACE
TIME
PLACE
PEOPLE
EVENT
STUFF
GENERAL ARCHAEOLGICAL INVENTORY

Fig. 6. Proposed structure for an integrated archaeological database by combining object-based and event-based approach.

Second, the excavation database should be used during the field work as well as the analysis and interpretation. Detailed information on the objects, their location and temporal and other attributes should be recorded in it. As 3D spatial data is generally recorded, this database should be geospatial and incorporate 3D data where feasible. We propose to develop the database according to the 'object-space-time' relationship, where the connection with the general inventory can be made through the 'space-place' linkage (see Fig. 6). An integrated database can thus be developed.

With regard to the case study, the integrated database may look as follows (Fig. 7). For the general inventory the 'place' part will contain information on the administrative place, such as the country 'Russia', the province 'Kosh Agash', eventually cadastral parcels, etc. and the coordinates of the site. Preferably, this are the coordinates of the site boundaries, but at least it should be a point location '49°47'51"N, 89°09'10"E' for the Yustyd site. All persons and organizations involved in the archaeological project will be recorded in the general database, and linked to the events they have taken part in. These events can be both field (e.g. excavation, sampling, etc.) and desktop events (e.g. administration or report writing). Ghent University and Gorno-Altaisk State University will be identified as participating organizations, and the individual field archaeologists will be included in the data base and linked to what kind of activity they have participated in, e.g. excavation, OSL sampling, report writing, etc. The events are then linked to the stuff, which gives a general indication of what one can expected of the site. For the case study site, the same site type category as in the Altari database can be used, namely ritual/other. Additional information can be given: 'A line of square structures with platforms,…'. The detailed information of what is actually found on the site is stored in the 'object' part of the excavation database. All details such as the material ‘stone’, the type of structure ‘dwelling’, the findings ‘stones, …’, etc. can be stored. For all of them the shape is stored including the three dimensions or at least a 3D point is indicated. This will form the basis for linking the two databases. Another option is to link the two databases by the site ID ‘KA-771’. The latter will be useful if the site location is only known as a point. Since the temporal information is as important as the spatial information, this forms a separate axis in the database. Different temporal categories such as the excavation time ‘July 2011’ and the cultural period, in this case still unknown, can be stored (De Roo et al. 2014).

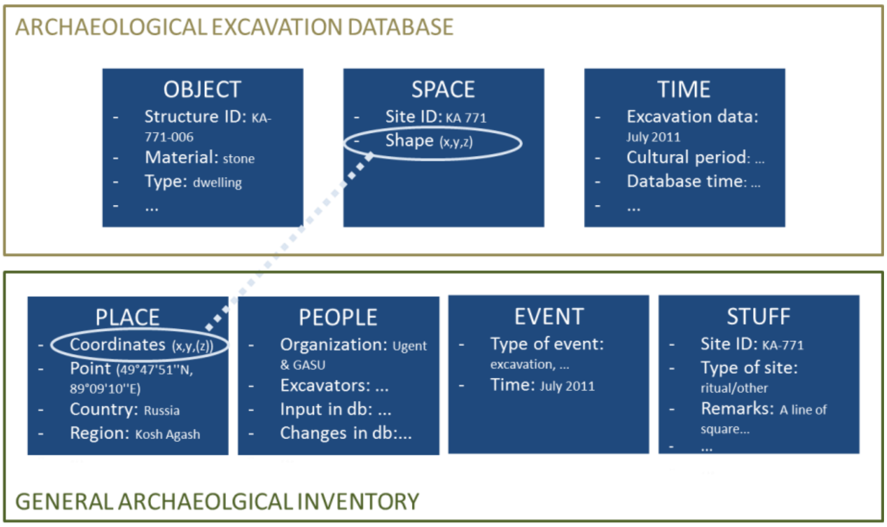


Fig. 7. Example interpretation of the integrated archaeological database

The integrated database, for its turn, can give rise to the development of information systems and analysis tools. According to the various actors who provide or use archaeological data, different objectives exists. This implies to define different levels of usage in line with these actors and their requirements (Anichini et al. 2012). For example, a tourist seeking for more information is not assumed to be able to make changes to the excavation database entries, while it is obvious a field archaeologist working in that project can do this. An archaeological information infrastructure will be created that way, and will allow archiving, accessing, integrating and mining disparate data sets (Kintigh 2006).

Finally, it should be noted that the proposed approach is not unassailable. Future work need to show the positive and negative implications arising from the implementation and testing in a real archaeological project cycle. Furthermore, elements such as implementation costs, storage space and costs, policy and control structures need to be thoroughly assessed as well. However, this research has shown that opportunities are granted by incorporating 3D and geo-ICT in the entire archaeological project cycle. An integrated archaeological database including a general inventory and a detailed excavation database, will facilitate interdisciplinary research and interoperable data for use within archaeological research and in other fields, such as land administration and other policy decisions.

References

Anichini F, Fabiani F, Gattiglia G, Gualandi ML (2012) A database for archaeological data recording and analysis. MapPapers 1 en II 21–38.

Arroyo-Bishop D, Lantada Zarzosa M (1995) To be or not to be: will an object-space-time GIS/AIS become a scientific reality or end up an archaeological entity? In: Lock G, Stancic Z (eds) Archaeol. Geogr. Inf. Syst. Taylor & Francis, London, pp 43–53

Belien E (2012) Methodologisch onderzoek naar de optimalisatie van documentatie van archaeologische opgraivngen door middel van fotogrammetrie en fotomodellering (Altaï-Rusland). Ghent University

Cripps (2012) Places, People, Events and Stuff: Building blocks for archaeological information systems. Proc. Comput. Appl. Archaeol. Conf. 2012.

Cripps P, Greenhalgh A, Fellows D, et al. (2004) Ontological Modelling of the work of the Centre for Archaeology.

Forte M (2014) 3D Archaeology: New perspectives and challenges-The example of Catalhöyük. J East Mediterr Archaeol Herit Stud 2:1–29.

Gheyle W (2009) Highlands and Steppes: An Analysis of the Changing Archaeological Landscape of the Altay Mountains from the Eneolithic to the Ethnographic Period. Ghent University

Green CT (2011) Winding Dali's clock :the construction of a fuzzy temporal-GIS for archaeology. 159.

Harris TM, Lock GR (1995) Multi-Dimensional GIS : Exploratory Approaches to Spatial and Temproal Relationships within Archaeological Stratigraphy. In: Kamermans H, Fennema K (eds) Interfacing Past. Comput. Appl. Quant. Methos Archaeol. CAA95. Vol. II (Analecta Paehistorica Leiden. 28). Institute of Prehisotry, University of Leiden, Leiden, pp 307–316

Huvila I (2014) Process and appropriation in the digitalisation of archaeological archives and archiving practices. Paris, France, pp 1–6

Jacobson-Tepfer E (2008) Culture and Landscape of the High Altai. Preserv. Frozen Tombs Altai Mt.

Katsianis M, Tsipidis S, Kotsakis K, Koussoulakou A (2008) A 3D digital workflow for archaeological intra-site research using GIS. J Archaeol Sci 35:655–667. Kintigh K (2006) The Promise and Challenge of Archaeological Data Integration. Am Antiq 71:567–578.

Labrador AM (2012) Ontologies of the Future and Interfaces for All: Archaeological Databases for the Twenty-First Century. Archaeologies 8:236–249. doi: 10.1007/s11759-012-9203-2

McKeague P, Corns A, Shaw R (2012) Developing a Spatial Data Infrastructure for Archaeological and Built Heritage. Int J Spat Data Infrastruct Res 7:38–65.

Plets G, Gheyle W, Verhoeven G, et al. (2012) Three-dimensional recording of archaeological remains in the Altai Mountains. Antiquity 86:884–897.

De Reu J, Plets G, Verhoeven G, et al. (2013) Towards a three-dimensional cost-effective registration of the archaeological heritage. J Archaeol Sci 40:1108–1121. De Roo B, Bourgeois J, De Maeyer P (2013a) On the way to a 4D archaeological GIS: state of the art, future directions and need for standardization. In: Addison AC, De Luca L, Guidi G, Pescarin S (eds) Proc. 2013 Digit. Herit. Int. Congr. Institute of Electrical and Electronics Engineers (IEEE), pp 617–620

De Roo B, Bourgeois J, De Maeyer P (2013b) A survey on the use of GIS and data standards in archaeology. Int J Herit Digit Era 2:491–507.

De Roo B, Van de Weghe N, Bourgeois J, De Maeyer P (2014) The temporal dimension in a 4D archaeological data model: applicability of the geoinformation standard. Innov 3D Geo-Information Sci Lect Notes Geoinf Cartogr. doi: 10.1007/978-3-319-00515-7

Shaw R, Corns A, McAuley J (2009) Archiving Archaeological Spatial Data: Standards and Metadata. Comput. Appl. to Archaeol. 2009

Stal C, Van Lieferinge K, De Reu J, et al. (2014) Integrating geomatics in archaeological research at the site of Thorikos (Greece). J Archaeol Sci 45:112–125.

Wagtendonk AJ, Verhagen P, Soetens S, et al. (2009) Past in Place: The Role of Geo-ICT in Present-day Archaeology. In: Scholten HJ, Velde R, Manen N (eds) Geospatial Technol. Role Locat. Sci. Springer Netherlands, Dordrecht, pp 59–86

Acknowledgments Financial support from the Special Research Fund (BOF) of Ghent University is gratefully acknowledged.

Prototyping Information Visualization in 3D City Models: a Model-based Approach

Claudine Métral[*], Gilles Falquet

Centre Universitaire d'informatique, University of Geneva,
7 route de Drize, 1227 Carouge, Switzerland
{claudine.metral gilles.falquet}@unige.ch

Abstract. When creating 3D city models, selecting relevant visualization techniques is a particularly difficult user interface design task. A first obstacle is that current geodata-oriented tools, e.g. ArcGIS, have limited 3D capabilities and limited sets of visualization techniques. Another important obstacle is the lack of unified description of information visualization techniques for 3D city models. If many techniques have been devised for different types of data or information (wind flows, air quality fields, historic or legal texts, etc.) they are generally described in articles, and not really formalized. In this paper we address the problem of visualizing information in (rich) 3D city models by presenting a model-based approach for the rapid prototyping of visualization techniques. We propose to represent visualization techniques as the composition of graph transformations. We show that these transformations can be specified with SPARQL construction operations over RDF graphs. These specifications can then be used in a prototype generator to produce 3D scenes that contain the 3D city model augmented with data represented using the desired technique.

Keywords. information visualization, 3D user interface design, 3D city models, model-based prototyping

1. Introduction

1.1 Motivation

3D city models are widely used all around the world. A current trend is adding semantics to the geometrical objects describing the city (buildings, transport elements, trees, city furniture, etc.) in order to enlarge the possible applications of such models. Adding semantics to 3D city models can be done according to various ways and for various additional information types. The added information can represent legal information, pollutant concentrations, noise levels, wind velocity, road names, historical facts, etc. A wide range of 3D visualization techniques is needed for visualizing such

[*] Corresponding author

information ranging from text (legal information, historical facts, road names) to scalar fields (pollutant concentrations, noise levels) or flows (wind velocity). When creating 3D city models, selecting relevant information visualization techniques is far from trivial. This task should be eased by providing a prototyping tool enabling the designer to visualize, for such or such technique, his set of data in his 3D city model, and to conduct usability tests with different techniques.

The efficiency of the prototyping approach relies on the possibility to develop prototypes at a fraction of the cost of the final interface. This is precisely why this approach is difficult to apply in 3D UI: There is a lack of tools and techniques to develop prototypes at a low cost. For instance, the well known low-fidelity paper and pencil prototyping techniques cannot be adapted to the third dimension.

The question we address in this work is therefore: Can we develop models, algorithms, and tools to efficiently create prototypes for the visualization of data in 3D virtual environment? More precisely, is there a way to specify prototypes (specify the 3D model, data, and visualization technique) and then have a generator produce the actual prototype, instead of building it manually?

1.2 Overview

This paper starts with a brief state of the art on information visualization and on 3D user interface prototyping.

The next sections describe the approach that we have developed. We propose a model-based approach for the generation of prototypes for the visualization of information in 3D city models. The idea is to provide the 3D city model interface designer with (1) a modeling language to specify the visualization technique to be tested and (2) an automated generation tool that takes as input the technique specification, the reference 3D city model and a data set, and produces a 3D test environment where the given data set is visualized according to the specified technique. The central part of the technique specification model is intended to describe the mapping of data set elements to visual objects. Adapting the data state model of Chi (2000), this mapping is decomposed into several stages: extracting features from data; mapping the features to abstract visual objects and relations; mapping these objects and relations to concrete visual objects (layout generation). A layout manager, which is a constraint solver, produces the concrete visual objects and places them in the 3D city model (output rendering). In many cases this output rendering implies complex algorithms in order to compute the geometrical properties, shape, position... of the concrete visual objects.

We then conclude with a work planned in a near future: the implementation and classification of such algorithms for a representative set of 3D visualization techniques.

2. Background and related work

2.1 Information Visualization in 3D Environments

Visualizing semantically enriched 3D models consists in displaying the geometric part of the model (the geometry of the modeled spatial objects) and displaying visual objects that represent the enrichment information attached to these objects. There are two cases to distinguish, depending on the spatial or non-spatial nature of the enrichment information. Non-spatial information is attached to a 3D object but it doesn't have any spatial component. The maximum capacity of a room or the name of a street are non-spatial, they can be displayed at different places in the scene, provided the link with the described object is made obvious for the user. The visualization problem for this type of information is closely related to information and data visualization and geovisualization (Chen et al., 2008, Bleish 2012). The main problem here, which is not addressed in traditional information visualization techniques, is to find an adequate location to display the information. These questions have been explored, for instance, for the associations of labels to roads (Vaaraniemi et al, 2013) or the association of historical information to buildings (Alamouri and Pecchioli, 2010).

In the case of spatialized information, the information elements already have a precise spatial location, for instance the noise level measured at a given point on a facade, or the wind flow (a vector field) in a street canyon. The visualization of this type of information is generally called scientific visualization. It may rely on simple visualization techniques that represent the information (such as air quality values or number of pedestrians) by objects (e.g. solids) put at specific places in the 3D city model (San José et al, 2012) (Marina et al, 2012). It may also involve sophisticated algorithms to compute isosurfaces, flow lines, vortexes, etc. (Laramee et al, 2004), (Post et al, 2003) (Amorim et al, 2012). Many techniques have been developed to visualize this type of information and the main question is to select the most efficient one for a given 3D model and a given user task.

2.2 3D user interface prototyping

Research work related to the design of user interfaces (Shneiderman, 1998) or more specifically related to 3D user interfaces (Bowman, 2004) exist for a long time. 3D user interfaces (3D UI) can be the result of different development approaches: the programmatic approach where the 3D UI is obtained by coding, the toolkit-based approach where the developer directly codes and implements the final interface by using predefined sets of elements and objects provided by the toolkit, and the methodological approach that relies on existing numerous methods for developing 3D UI (Gonzalez Calleros, 2010). But these approaches are not adapted to the prototyping of information visualization in 3D city models.

3. A Model-based Approach for Prototyping 3D Information Visualization

3.1 Knowledge Sources

We studied the scientific literature about applications based on 3D city models to obtain a global view of the domain. Although the techniques used in these applications are often not explicitly described, they provide enough information to draw initial classification axis. The studied models and applications are used for various tasks, such as:

- Evaluation of the wind comfort for pedestrians in a city street (Amorim et al, 2012) where 3D coloured polylines (colour representing wind velocity) are added to the geometrical model.
- Assessment of air quality in a street or neighbourhood by adding coloured solid objects to the 3D buildings (Lu et al, 2009), (San José et al, 2012).
- Analysis of pedestrian behaviour (Marina et al, 2012) where colored bars visualize spatial distribution of pedestrian movement.
- Analysis of human perception of space (Fisher-Gewirtzman, 2012) where colored lines represent visual exposure or visual openness in the 3D city model.
- Visualization of historically enriched 3D city models where information (text and images) has been added to the geometrical model (Alamouri and Pecchioli, 2010), (Hervy et al, 2012).

From this study we can define different cases:

- Each data element is individually represented by an object (point, solid…) situated at the (x,y,z) location of the data, with parameters (color, size, transparency…) representing the value of the data (Lu et al, 2009), (San José et al, 2012), (Marina et al, 2012).
- A set of data is represented as a whole by a complex object such as a polyline (Amorim et al, 2012).
- Each data (eg a text) is individually represented by an object (panel of text…) associated to one or several objects (building…) of the 3D city model (Alamouri and Pecchioli, 2010), (Hervy et al, 2012).
- The data describe relations between different objects of the 3D city model and can be represented by objects (lines…) between these objects (windows…) (Fisher-Gewirtzman, 2012).

These cases can be classified according to (1) the type of data and (2) the type of representation:

- The dataset elements can be associated to a point or a region of space, to an object of the 3D city model or to a relation between several objects of the 3D city model.
- The dataset elements can be individually represented by a visual object such as a sphere, a panel…, either at their exact location or with an offset. Alternatively they can be globally represented: in this case a dataset is represented by a set of visual objects, such as isosurfaces for a scalar field.

3.2 General Approach

We selected as starting point an ontology of 3D visualization techniques (Métral et al, 2014) that provided of a common and formalized representation of visualization techniques in 3D environments. This ontology describes input data, output rendering and usage of visualization techniques. But it does not describe how the techniques map their input data to the output rendering. These mapping descriptions are precisely what we need to automatically generate the output rendering of a specific visualization technique. Figure 1 illustrates the main elements of such a mapping.

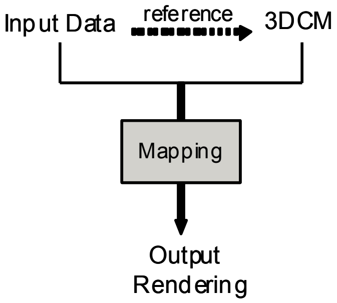


Figure 1. Visualisation technique as a mapping.

The question is thus, how to specify this kind of mapping?

Our approach is based on the notion of graph transformation and on the distinction between abstract and concrete visual objects. In the following we will use of the resource description framework RDF (RDF, 2014) and the SPARQL query language (SPARQL, 2013) to present the approach and its concrete implementation. A 3D city model, input data elements and an output rendering are described as RDF graphs. A visualization is a graph transformation defined by (1) an abstract representation that is specified as a graph transformation expressed with SPARQL constructions and (2) a concrete representation generated with a layout manager. The abstract visualization produces abstract visual objects and relations. The layout manager makes the representation concrete by computing object attributes (e.g. their coordinates) that were left undefined at the abstract level.

A last step is the execution of an output script to generate the final visualization, and hence the prototype, in the desired 3D modeling language (X3D, COLLADA, KML, etc.). Figure 2 shows the whole generation process.

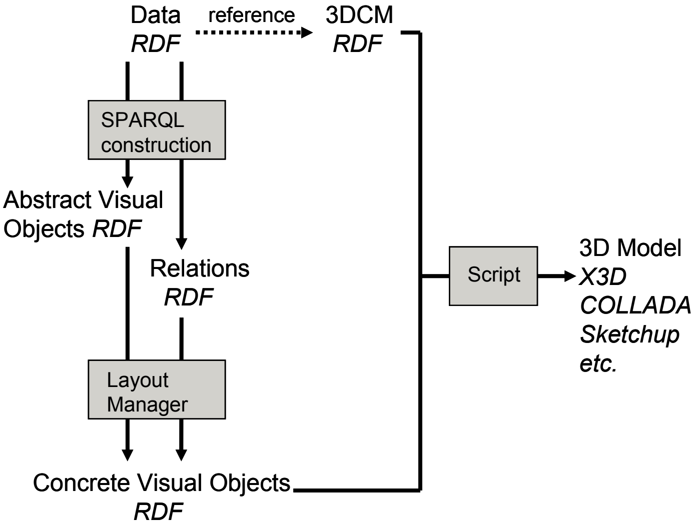


Figure 2. Mapping description in detail

4. RDF-based Specification of a Visualization Technique

We suppose that the 3D city models used for visualizing information are represented in CityGML (OGC 12-019, 2012). The CityGML XML Schema is almost compatible with RDF/XML, i.e. an XML file that conforms to the CityGML schema is almost an RDF/XML file. This comes from the fact that CityGML files obey to the alternating class-property structure: each element that represents a class entity (`bldg:Building`, `bldg:GroundSurface`, `gml:MultiSurface`, `gml:Polygon`, ...) contains only elements that represent properties (`bldg:boundedBy`, `bldg:lod2MultiSurface`, `gml:surfaceMember`, ...) and property elements don't have attributes. In addition, the `gml:id` attribute can serve as a resource identifier for the RDF (entities without `gml:id` will be mapped to RDF blank nodes). Therefore it is relatively straightforward to write a transformation script that generates an RDF version of a CityGML 3D city model.

4.1 Data Representation

As defined in (Métral et al, 2014) the information to visualize is composed of an information type (*PollutantConcentration*, *PedestrianCounting*, *IntervisibilityRate*...), a data type (*Integer, Vector, RichText*...) and an input location (*Point, Surface, CityObject*...). When these data represent measurement, their level of measurement (nominal, ordinal, quantitative) is in fact given by their datatype, more precisely, by the operations available (or not) for each datatype.

The input location – and thus the georeferencing of the data – is defined according to GML (OGC 07-036, 2007). The data elements are represented as RDF triples *subject property object*, depending of what these data represent. We distinguish three cases, depending on the spatial component of the considered data.

Data elements associated to a point or to a region (spatial data)
Each data element *x* is represented by a value *val* and by a location *loc*, which is a spatial object (not a city object) such as a point or a surface. The general RDF form of a spatial data element is:

```
x a DataElementClass.  x :value val.  x :location loc.
loc a SpatialObjectClass ; ... properties of the location
val ... properties of the value, for complex values
```

Such data can be a pedestrian counting with *val* being the counting value and *loc* the area used for the counting, or a pollutant concentration with *val* being the pollutant concentration level and *loc* the point or region associated:

```
:data1 a :PollutantConcentration.
:data1 :value 5.67.
:data1 :location :loc1.
:loc1 a :Point; :xcord 4.5 ; :ycoord 44 ; zcoord 1.5 .
```

Data elements related to an object of the 3D city model
Each data element *x* is represented by a value *val* and is related to the city object (*Building*, *Window*, *Vegetation*…) to which it refers. The general schema is

```
x a DataElementClass.  x :value val.  x :about y.
y a CityObjectClass ; ... properties of the city object
```

Such data can be the description of a building with *val* being the description text and *y* the related building:

```
:data2 a :RichText.  :data2 :value "....".
:data2 :about :co2.
:co2 a cgml:Building ; gml:geometry ...
```

In general the city object will be described in a separate RDF graph that represent the 3D city model.

Data elements representing a relation between objects of the 3D city model
Each data element *x* is represented by an optional value *val* and properties (`arg1`, `arg2`, ...) that point to the city objects (*Building*, *Window*, *Vegetation…*) it relates:

x `a` *RelationType*`.` *x* `:value val.` *x* `:arg1` y_1`.` *x* `:about` y_2`.` y_1 `a` *CityObjectClass*$_1$`.` y_2 `a` *CityObjectClass*$_2$`. ...`

Such data can represent the intervisibility between windows with *val* being the intervisibility level, and y_1 and y_2 the related windows:

```
:data3 a :IntervisibilityRelation.  :data3 :value 0.3 .
:data3 :arg1 :win1 ; :arg2 win2.
:win1  a cgml:Window ; ... . :win2 a cgml:Window ; ... .
```

4.2 Abstract Level

The abstract level mapping associates abstract visual objects and relations to input data (see Figure 2). The abstract vocabulary comprises simple visual objects (*Sphere*, *Line*, ...), complex objects (*IsoSurfaces, FlowLines, …*), and abstract spatial relations (*near*, *above*,*frontOf*, *inside*...). Each type of abstract objects has specific visual properties (color, shape, center, ...) that can be set a this level or left undefined. The abstract vocabulary may be arbitrarily extended with new elements, provided they are taken into account at the concrete level.

Since we represent data and the city model as RDF graphs, it is natural to use the SPARQL language to specify the graph transformation that corresponds to the abstract level mapping. The SPARQL query language provides several query forms (SPARQL, 2013). The *select* query form returns all (or a subset of) the variables bound in a query pattern match. The *construct* query form returns an RDF graph. The graph is built by substituting variables in a set of triple templates. A query consists in two parts: the *select* or *construct* clause and a *where* clause that provides the basic graph pattern to match against the data graph. In the SPARQL queries below all elements of the RDF graph that satisfy the *where* statement are selected. Then, for each of them, a new RDF graph is generated with the *construct* statement. We thus obtain a blank node of a certain abstract visual object type and with some visual properties. In fact, in the *select* statement we have the same RDF graph as the one used for the description of the input data, but with *loc*, *val*... as variables (this is why they are noted *?val*, *?loc*...).

Data elements associated to a point or to a region: individual representation
In this case the blank node (named `_:1`) is associated to the same location *loc* as the data element. Its visual properties are derived from the data value (f_1,f_2, ... represent computations on *?val* and its properties).

`construct {_:1 a` *AbstractVisualObjectType* `.`
  `_:1` *visualProperty*$_1$ f_1`(?val). _:1` *visualProperty*$_2$ f_2`(?val). ...`

```
  _:1 :location ?loc}
from DataGraph
where {?x a DataElementClass.  ?x :value ?val.  ?x :location
?loc}
```

Such data elements can be a pedestrian counting represented by a cone or a pollutant concentration (with the value *val* being the pollutant concentration level and location *loc* being the point associated) represented by a sphere located at *loc* and whose radius is computed from *val*:

```
construct {_:1 a :Sphere .
  _:1 :radius ?val/100.
  _:1 :location ?loc}
from DataGraph
where {?x a :PollutantConcentration.  ?x :value ?val.
  ?x :location ?loc.  ?loc a :Point}
```

Data elements associated to a point or to a region: global representation
In this case of a global representation a (complex) visual object represents the whole dataset:

```
construct {:vObj a AbstractComplexVisualObjectType .
:vObj :inputData _:1.  _:1 :value ?val.  _:1 :location ?loc}
from DataGraph
where {?x a DataElementClass.  ?x :value ?val.  ?x :location
?loc}
```

Each data element becomes an input data for the complex visual object `:vObj`, keeping its original location and value. Typical abstract objects of this type are isosurfaces, flowlines, stream ribbons, etc.

Data elements related to an object of the 3D city model
In the case of data elements related to a city object the location is in fact a blank node (named `_:2`) defined by a spatial relation (*near, above, inside…*) and by the related city object:

```
construct {_:1 a AbstractVisualObjectType .
  _:1 visualProperty1 f1(?val).   _:1 visualProperty2 f2(?val).  …
  _:1 :location _:2.
  _:2 a SpatialRelation; :arg1 _:1; :arg2 ?y }
from DataGraph
where {?x a DataElementClass.  ?x :value ?val.  ?x :about ?y.
?y a CityObjectClass}
```

For a text that has to be displayed on a panel located near the building it describes we can define:

```
construct {_:1 a :Panel .
   _:1 :content ?val.
   _:1 :location _:2.
   _:2 a :nearRealtion; :arg1 ?y; :arg2 _:1 . }
from DataGraph
```

```
where {?x a :RichText.  ?x :value ?val.  ?x :about y.  ?y a
cgml:Building}
```

Data elements representing a relation between objects of the 3D city model
In the case of data elements representing a relation between objects of the 3D city model, what we have to represent is a connection between the city objects (represented by a blank node `_:2` linking the different elements together):

```
construct {_:1 a AbstractVisualObjectType .
_:1 visualProperty1 f1(?val, ?y1, ?y2).
_:1 visualProperty2 f2(?val, ?y1, ?y2).
...
from DataGraph
where {?x a DataElement.  ?x :value ?val.
       ?x :arg1 ?y1.  ?x :arg2 ?y2.
       ?y1 a CityObject.  ?y2 a CityObject. ...}
```

An example is the representation of intervisibility levels between windows by colored lines drawn between the related windows:

```
construct {_:1 a :Line .
_:1 :color [a :Color; :red ?val; :green 0; blue 0].
_:1 :endpoint ?y1. _:1 :endpoint ?y2.
from DataGraph
where {?x a :IntervisibilityRelation.  ?x :value ?val.
       ?x :arg1 ?y1.  ?x :arg2 ?y2.
       ?y1 a cgml:Window.  ?y2 a cgml:Window}
```

Note that this representation is abstract because the line endpoints are objets (windows) not precise points in the 3D space.

4.3 Concrete level

The concrete level associates abstract visual objects to input data using the abstract visual objects and the relations generated at the abstract level (see Figure 2).

Data elements associated to a point or to a region: individual representation
In the case of a data element directly associated to a point it is possible to define a SPARQL statement similar to the one defined at the abstract level. In the case of a data element associated to a surface we have first to compute a point inside the surface and then to use this point for precisely positioning the concrete visual object.

```
construct {_:1 a ConcreteVisualObjectType .
_:1 visualProperty1 f1(?val).  _:1 visualProperty2 f2(?val).  ... }
_:1 preciseLocationf(?loc) .
from DataGraph
where {?x a DataElementClass.  ?x :value ?val.  ?x :location
?loc}
```

Data elements associated to a point or to a region: global representation
In the case of such a global representation we have to compute one or several visual objects that can have several parameters depending not only on the data elements but also on their abstract representation, and that will be placed at a precise location in the 3D scene. For example an abstract isosurface can be represented by several concrete multisurfaces whose number depends on the value intervals of the isosurface.

Data elements related to an object of the 3D city model
In the case of a data element associated to a city object we have to compute a point that will be used for precisely positioning the concrete visual object in the scene. This computation has to take in account the city object and the spatial relation between it and the abstract visual object. Some parameters have also to be computed. For the example of a text panel that has to be placed near a building the computed point can be used for positioning the panel. Parameters such as height or orientation of the panel have also to be computed.

Data elements representing a relation between objects of the 3D city model
In the case of a data element representing a semantic relation between several city objects we have to compute a point in each of these city objects. These points will be used for precisely positioning the concrete visual object that will rely them. Some parameters may also be computed. For the example of intervisibility lines between windows the computed points can be the center of the windows in order to draw a line between these points while parameters can define a color based on the intervisibility rate.

As we have seen with previous examples, using SPARQL queries for generating the concrete visual objects is only possible in very simple cases. For most cases we need a layout manager whose task is to compute precise locations as well as the needed parameters.

5. Layout manager

The layout manager is a constraint solver, that produces concrete visualization objects (in X3D, COLLADA, SketchUp...). The layout manager must conform to the abstraction principle, i.e. it must always be possible to reconstruct the abstract view from the concrete one. The layout managers tasks for the different cases are described below.

Data elements associated to a point or to a region: individual representation
The main task for such data elements is to transform data coordinates into output coordinates

Data elements associated to a point or to a region: global representation
The layout managers for these objects are sophisticated algorithms that compute surfaces for representing scalar fields, or lines for representing trajectories in vector fields, etc.

Data elements related to an object of the 3D city model
The layout manager must solve placement relations, i.e. find a location in the output space that clearly represents an abstract spatial relation. For example the *above* relation between a panel and a building can be represented by placing the bottom-centre of the panel 2 meters higher than the highest point of the building roof surfaces and at the barycentre of the building ground surface.

Data elements representing a relation between objects of the 3D city model
For semantic relations between city objects the layout manager must find locations in the city objects involved. For example, if the intervisibility between two windows is to be represented by a straight line segment, the layout manager must select appropriate points in each window surface as endpoints for the line.

6. Implementation

6.1 Generation process

The generation process proceeds as follows:

1. Transform the 3D city model into an RDF graph and upload it in a triple store able to store and manage RDF data.

2. Transform each dataset into an RDF graph and upload it in the triple store (as a separate graph).

3. For each dataset (RDF graph) apply the desired visualization transformation to produce a graph of abstract visual entities (objects and relations). To some extent this can be done dirctly with SPARQL queries. Indeed, in the current version of SPARQL, the *construct* query form doesn't support operations (such as *?val/100*) or functions (such as *concat*). To overcome this limitation some scripting or computing is needed (we can use the *select* query form that allows such operations with some scripting since the *select* query doesn't directly generate a RDF graph).

4. Apply a layout manager to compute the concrete representation (e.g. X3D or SketchUp objects) of each abstract visual entity.

Uploading datasets to the RDF store may require a “dictionary” to map the city object references present in a dataset to the corresponding nodes of the 3D city model graph (identified by their `gml:id`).

6.2 Specifying and Implementing the layout manager

For abstract objects located in a spatial region
If the region is a point the mapping is immediate, there is no real choice as to where the concrete object must be. If the region is a point set (line, surface, volume) the layout manager must choose a point within the region to place the concrete object. Thus the specification of such a layout manager amounts to selecting a function that maps a spatial region to a point (or more generaly to a region)

For abstract objects linked to a city object through a spatial relation
The layout manager is essentially a mapping from a city object and a relation to a point that represents and satisfies this relation. For instance, the *above* relation for a building should yield a point that is higher than the building and above its ground surface. Of course each spatial relation may be represented by several different mappings, so the specification of a layout manager consists in choosing one such mapping (e.g. in a library of mappings)

For abstract objects representing relations between city objects
The choice function will generally rely on geometric functions such as min, max, barycenter, etc. A layout manager is in charge of mapping an (set of) abstract visual object to a concrete user interface object.

The mapping computed by the layout manager aims to produce precise locations in the 3D city model. Alternatively the mapping could generate a concrete object with some parameters left undefined. For instance, a sphere whose radius is fixed but not its center point, a panel whose size is fixed but not its base point. The user thus could have the possibility to place the concrete object at the more suitable place according to his criteria.

7. Testing the Approach

To test the proposed model-based approach we selected a city model expressed in CityGML. This model represents a part of the city of Carouge in Switzerland and is composed of about one hundred buildings at LOD 2. Only buildings are represented in the model. We have used the Sesame triplestore for the storage and the retrieval of the RDF triples. Once loaded on the Sesame platform our model consists of about 80000 RDF triples.

A test case consists in using the approach to 1) specify a data set and a visualization technique (to check the expressiveness of the specification method), 2) to generate the test scene (to check if it correctly represents the visualization technique)

We present here a basic test case related to pedestrian counting. The data set contains the numbers of pedestrians counted at different locations. The visualization is made of cones whose location and height represent the counting location and value respectively. The output rendering is made of X3DOM elements corresponding to the cones and the city model geometry.

Data representation
The location of a data element is a 3D point with coordinates `xloc`, `yloc`, and `zloc`. The value is an integer number. The RDF representation of 42 pedestrians counted at $x = -13, y = 25, z = 0$ is

```
:pednum1  a :PedestrianCounting .
:pednum1 :value 42 .
:pednum1 :location :loc1 .
:loc1 :xloc -13.  :loc1 :yloc 25. :loc1 :zloc 0 .
```

Abstract level
The abstract representation is made of objects of type `Cone` with height and location directly drawn from the data. Here is the SPARQL query that generates these abstract objects (as two blank nodes named `_:1` and `_:2`):

```
construct {_:1 a :Cone.
_:1 :height ?val.
_:1 :location _:2. _:2 :xloc ?x. _:2 :yloc ?y. _:2 :zloc ?z}
where {?x a :PedestrianCounting.
       ?x :value ?val.
       ?x :location ?loc.
       ?loc :xloc ?x.  ?loc :yloc ?y. ?loc :zloc ?z}
```

Applying this query on the above data sample will generate

```
_:1 a :Cone.
_:1 :height 42.
_:1 :location _:2. _:2 :xloc -13.  _:2 :yloc 25. _:2 :zloc 0}
```

Concrete level
Even with the exact coodinates known we have to do some scripting since we have to transform the *x, y, z* numeric coordinates into the string "*x y z+val/2*" and the integer *height* value into the string "*height*" in order to generate correct X3DOM code. Here is the XML code with the values inset into the strings:

```
<transform rotation="1 0 0 1.5708" translation="-13 25 21">
   <shape>
      <appearance>
         <material diffusecolor="0 0 1"></material>
      </appearance
     <cone height="42"></cone>
   </shape>
</transform>
```

Figure 3 illustrates the previous data displayed as a cone in the 3D city model (and dispayed with another pedestrian counting data) :

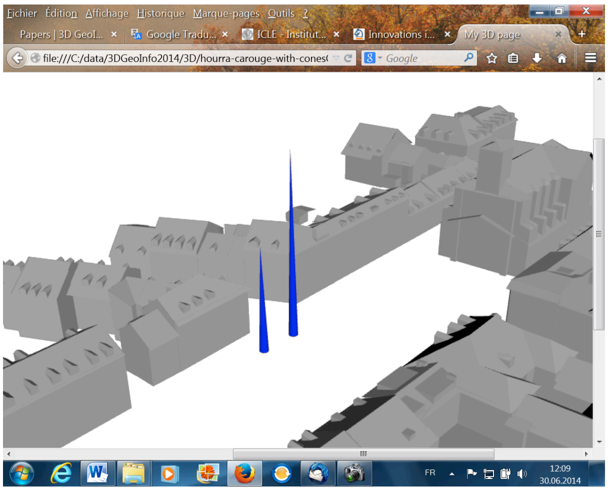

Fichier Édition Affichage Historique Marque-pages Outils ?
Papers | 3D GeoI...
Google Tradu...
ICLE - Institut...
Innovations i...
My 3D page
file:///C:/data/3DGeoInfo2014/3D/hourra-carouge-with-cones(
Google
FR
12:09
30.06.2014

Figure 3. Visualizing pedestrian counts as cones

Conclusion

In this paper we addressed the development of (rapid) prototyping tools for information visualization in 3D city models. A prototyping tool enables a designer to visualize in a specific 3D city model a specific set of data according to a specific visualization technique, and to conduct usability tests with different techniques.

We proposed a mapping of data set elements to visual objects in two stages: (1) a mapping of data features to abstract visual objects and relations, and (2) a mapping of these objects and relations to concrete visual objects (output rendering). A constraint solver named layout manager produces the concrete visual objects and places them in the 3D city model. In many cases the output rendering implies complex algorithms since the concrete visual objects are not simple 3D shapes but visual objects that embed functions to compute their geometrical properties, shape, position, etc. In a near future we plan to implement and classify such algorithms for a representative set of 3D visualization techniques.

References

Alamouri A, Pecchioli L (2010) Retrieving information through navigating in historical Baalbek. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/W15, pp 81-85

Aigner, W., Miksch, S., Muller, W., Schumann, H., & Tominski, C. (2008). Visual Methods for Analyzing Time-Oriented Data. *IEEE Transactions on Visualization and Computer Graphics*, *14*(1), 47–60. doi:10.1109/TVCG.2007.70415

Amorim JH, Valente J, Pimentel C, Miranda AI, Borrego C (2012) Detailed modelling of the wind comfort in a city avenue at the pedestrian level. In: Leduc T, Moreau G, Billen R (eds) Usage, Usability, and Utility of 3D City Models, 03008, pp 1-6

Aigner, W., Miksch, S., Muller, W., Schumann, H., & Tominski, C. (2008). Visual Methods for Analyzing Time-Oriented Data. *IEEE Transactions on Visualization and Computer Graphics*, *14*(1), 47–60. doi:10.1109/TVCG.2007.70415

aigner-tvcg08.pdf. (n.d.). Retrieved from http://www.cc.gatech.edu/~stasko/7450/Papers/aigner-tvcg08.pdf

Bleish, Susanne. (2012). 3D GEOVISUALIZATION – DEFINITION AND STRUCTURES FOR THE ASSESSMENT OF USEFULNESS (Vol. I–2, pp. 129–134). Presented at the XXII ISPRS Congress, Melbourne, Australia: ISPRS annals. Retrieved from http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/I-2/129/2012/isprsannals-I-2-129-2012.pdf

Bowman DA, Kruijff E, LaViola JJ, Poupyrev I (2004) 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA

Chen, C., Härdle, W., Unwin, A. (Eds.) (2008) Handbook of Data Visualization. Springer-Verlag, 2008.

Chi E H (2000) A Taxonomy of Visualization Techniques Using the Data State Reference Model. In: Proceedings of the IEEE Symposium on Information Vizualization 2000 (INFOVIS '00). IEEE Computer Society, Washington, DC, USA, pp 69-75

Fisher-Gewirtzman D (2012) 3D models as a platform for urban analysis and studies on human perception of space. In: Leduc T, Moreau G, Billen R (eds) Usage, Usability, and Utility of 3D City Models, 01001, pp 1-16

Gonzalez Calleros JM (2006) A Method for Developing 3D User Interfaces for Information Systems. DEA thesis dissertation, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Hervy B, Billen R, Laroche F, Carré C, Servières M, van Ruymbeke M, Tourre V, Delfosse V, Kerouanton JL (2012) A generalized approach for historical mock-up acquisition and data modelling: Towards historically enriched 3D city models. In: Leduc T, Moreau G, Billen R (eds) Usage, Usability, and Utility of 3D City Models, 02009, pp 1-10

Laramee R S, Hauser H, Doleisch H, Vrolijk B, Post F H, Weiskopf D (2004) The State of the Art in Flow Visualization - Dense and Texture-Based Techniques. In: Computer Graphics Forum 23 (2), pp 203-221

Lu A, Chen W, Ribarsky W, Ebert D (2009) Year-Long Time-Varying 3D Air Quality Data Visualization. In: Ras ZW, Ribarsky W (eds), Advances in Information and Intelligent Sys.,Springer-Verlag, Berlin Heidelberg, SCI 251, pp 289-306

Marina O, Leduc T, Rabino G, Karanakov B (2012) Integration of urban knowledge through visibilities studies and pedestrian behavior analysis in Old Bazaar in Skopje. In: Leduc T, Moreau G, Billen R (eds) Usage, Usability, and Utility of 3D City Models, 03006, pp 1-10

Métral C, Ghoula N, Silva V, Falquet G (2014) A Repository of Information Visualization Techniques to Support the Design of 3D Virtual City Models. In: Innovations in 3D Geo-Information Sciences, U. Isikdag (ed), Lecture Notes in Geoinformation and Cartography, Springer, 2014, XII, pp 175-194

OGC 07-036 (2007) OpenGIS Geography Markup Language (GML) Encoding Standard. Portele C (ed), Open Geospatial Consortium Inc, http://www.opengeospatial.org/standards/gml#downloads Accessed 27 June 2014

OGC 12-019 (2012) OGC City Geography Markup Language (CityGML) Encoding Standard. Gröger G, Kolbe TH, Nagel C, Häfele K-H (eds) Open Geospatial Consortium Inc, http://www.opengis.net/spec/citygml/2.0 Accessed 27 June 2014

Post F H, Vrolijk B, Hauser H, Laramee R S, Doleisch H (2003) The state of the art in flow visualisation: Feature extraction and tracking. In: Computer Graphics Forum, 22(4), pp 775-792

RDF Resource Description Framework (2014) http://www.w3.org/RDF/ Accessed 27 June 2014

San José R, Pérez J L, González R M (2012) Advances in 3D visualization of air quality data. In: Leduc T, Moreau G, Billen R (eds) Usage, Usability, and Utility of 3D City Models, 02002, pp 1-13

Shneiderman B (1998) Designing the user interface – Strategies for effective human-computer interaction. Addison-Wesley Longman Inc., Reading, Massachusetts, USA

SPARQL 1.1 query language – W3 recommendation (2013) Harris S, Seaborne A (eds), http://www.w3.org/TR/2013/REC-sparql11-query-20130321/ Accessed 27 June 2014

Vaaraniemi M, Freidank M, Westermann R (2013) Enhancing the visibility of labels in 3D navigation maps. In: Pouliot J, Daniel S, Hubert F, Zamyadi A (eds) Progress and New Trends in 3D Geoinformation Sciences. Lecture Notes in Geoinformation and Cartography (LNG&C) Series, Springer, Berlin, VI, pp 23-40

Acknowledgements

The model used for illustrating some visualization techniques relates to Carouge city (Switzerland) and has been provided by the Service de la mensuration officielle de l’Etat de Genève (SEMO).

INVESTIGATING INTEGRATION POSSIBILITIES BETWEEN 3D MODELING TECHNIQUES

Efi Dimopoulou*, Eva Tsiliakou[a], Vasso Kosti, George Floros, Tassos Labropoulos

School of Rural and Surveying Engineering, National Technical University of Athens,
9 Iroon Polytechneiou str, 15780 Zografou, Greece (efi@survey.ntua.gr;
eva.tsiliakos@gmail.com; vasso_kst@hotmail.com; flwrosg@gmail.com;
tassos.labropoulos@gmail.com)

Abstract In recent years, 3D city modeling gained increasing popularity for advanced applications, leading to the emergence of detailed 3D city models, entailing not only geometrical and topological, but also semantic information. This information is unavailable in most 3D modeling software, due to the lack of standardized 3D city objects' representation and the focusing of 3D modeling on visualization. So far, specific applications require additional data resulting in cumbersome data bases and technical processing difficulties. This problem is addressed by applying the CityGML data model, being accepted as an international standard for representing and exchanging spatial data of the three-dimensional real world objects at different Levels of Detail (LoD). Moreover, CityGML facilitates a higher level of data interoperability between various applications. Given the great number of currently available 3D modeling techniques, as well as the range of users, data integration and interoperability is a great challenge towards the advancement of 3D city modeling. Therefore, the aim of this paper is to investigate integration and interoperability between procedural modeling techniques and BIM-ready software within the CityGML framework from a semantic viewpoint. ESRI CityEngine environment and Trimble SketchUp Pro software were used to create 3D building models and evaluate modeling techniques. Integration processes with CityGML, and level of detail of the final 3D model were also investigated.

Keywords: 3D modeling, Procedural modeling, CityEngine, Sketchup Pro, CityGML

1 Introduction

The need of the modern world to better comprehend and enhance the perception of the real entities and phenomena has led to the description of our environment in a higher dimensionality. This is evident especially in 3D city modeling applications, in areas such as smart city planning or environmental simulation. 3D city models are characterized by complexity, while a semantic basis is required to complement their geometrical and topological aspects. In recent years, the integration of semantics into 3D city models has been widely accepted [Zhu et al., 2011]. On the contrary, 3D city modeling, focusing on visualization or datasets, resulted in cumbersome data bases and technical difficulties in processing. Furthermore difficulties and questions arised, concerning the effectiveness of the integration of the semantic aspects in 3D modeling. This semantic enrichment is crucial, due to the structural complexity and the multiplicity of space within the multidimensional urban environment, in which a range of different RRRs (Rights, Restrictions and Responsibilities) intersect with the corresponding land parcels. This range of land rights, restrictions and responsibilities requires proper 3D registrations complying with each legal structure [Dimopoulou and Elia, 2012]. Additionally, the semantic modeling of cities requires the appropriate qualification of 3D data [Gröger & Plümer, 2012]. Current trends focus on the semantic enrichment of distinctive city objects or 3D geometries which can be decomposed into their structural elements including attributes and their correlations. The semantic modeling approach along with the appliance of 3D geometry and topology of real-world objects is realized by the CityGML open data model [Kolbe, 2009]. However, questions arise about the most effective way to complement the geometries' semantics or how to efficiently extract semantics from pure geometric models [Zhu et al., 2011], issues which can be tackled by the concept of interoperability. The emergence of novel 3D modeling methodologies and techniques in computer graphics as well as the development of a range of 3D file formats has certainly assisted in this direction. Nevertheless, data integration and interoperability is a great challenge towards the advancement of 3D city modeling. In this paper interoperability options are investigated within 3D city/ building modeling techniques from a semantic viewpoint. Referring to interoperability, it was initialy introduced for information technology or systems engineering services, to allow for information exchange [IEEE, 1990]. A definition subjected to further modifications in the time. In our paper, the term is accepted as the ability of two or more computer systems to communicate and exchange information, based on specific data formats and protocols or referring to a common information exchange reference model.

However, achieving interoperability often presents a demanding procedure, since exchanging information between systems or models requires the set up of correspondence of the concepts from one system to the other [Métral et al, 2010].

Dealing with heterogeneous data, interoperability may be achieved when they follow the same modeling standard [Becker et al., 2012]. Interoperable models and options have emerged for urban application, such as CityGML. Kolbe [2009] suggests that the CityGML semantic model, tackles semantic heterogeneity or data amalgamation issues within the context of geo-web services and spatial data infrastructures. Becker et al., [2012] further discuss that while CityGML 'ensures interoperability on a semantic and syntactic level, the explicit treatment of geometric and topological heterogeneity is neglected'. The conflicting arguments from the literature suggest the need for the creation of synergies between different technologies applied to urban information systems.

This paper investigates the need discussed above, and examines integration possibilities of different datasets derived from BIM ready tools or procedural modeling processes, within the context of CityGML. It is structured as follows: first the procedural modeling is presented, along with the building modeling via BIM ready tools, such as Sketchup Pro, focusing on the semantic enrichment. An overview of the CityGML standard follows, including recommendations related to the alignment between different formats. In the next chapter, two case studies are described in detail. They employ different methodologies, and their conversion possibilities into CityGML is investigated. In the last section, the preliminary results and conclusions are presented. They refer to the entire modeling process and efficiency for both visualization and semantics, as well as the final models interoperability.

2. 3D Modeling Techniques

2.1. Procedural modeling

The emergence and use of a variety of 3D modeling methods and techniques have considerably increased during the last decade in various fields employing processes for the generation of urban structures' 3D geometry, via a mixture of datasets and modeling techniques. Amongst the latter, procedural modeling applies to a 3D modeling notion executed via algorithmic generation, based on the conception that all real world buildings are defined by rules, since repetitive patterns and hierar- chical components describe their geometry. For this reason, 'it is really important to know the hierarchical and relative positions of the elements' [Jesus et al, 2012] in order to achieve the concept of bottom-up creation, since the 3D model is split into sides at first and later into floors, which are split into tiles. Additional tile modification concerns the design of windows, doors as well as other façade elements which complete the building's actual view. Within this context, detailed 3D models and composite facade textures derive in different LODs (both for the exterior or the interior) from sets of rules, which are called grammars, considering an initial 2D geometry.' Procedural modeling mainly uses the CGA (Computer Generated Architecture) shape grammar to programmatically produce building models with high visual quality and geometric detail' [Müller et al., 2006; Zhang et al., 2014]. Shape grammars, originally defined by George Stiny and James Gips in 1971 are a formalism for the manifestation of procedural context, containing a set of production rules (such as shape rules).

Shape grammar was reexamined by Peter Wonka who introduced Set Grammar in 2003. Set grammars followed the introduction of Split grammar, with which models can be manipulated by "cutting" them along x and y axis into various forms.

Finally, Pascal Müller extended the Split Grammar in 2006, and defined a context-sensitive Set grammar, CGA which is practically the basis for 3D modeling of architectural buildings via CGA grammars. According to Müller et al. [2006], 'CGA is a novel shape grammar, with which shape rules allow the user to specify interactions between the entities of the hierarchical shape descriptions'. Procedural modeling is a very applicable and effective method in architecture modeling cases where the generation of the 3D model using mainstream 3D modeling techniques is uneconomical both in terms of human labor, resources and time and can be further used for the 3D modeling of the buildings' interior. 'However, procedurally generated models are typically restricted to mere geometric representation' [Tutenel et al. 2008, 2010], lacking in terms of semantics, which seem to be a challenge since they are limited to shape geometry's semantics, or the semantic hierarchy of the 3D model's elements. According to Jesus et al. (2012) GIS data can be used in procedural modeling tools, but these do not provide an easy and uniform way to incorporate semantic information from different data sources'. Another issue that needs to be addressed is the interoperability capacity of procedurally generated models. These models can be usually exported to CAD or formats used purely for geometric representation especially for the gaming or movies industry and hardly ever for integration within a semantic standard such as CityGML.

2.2. Building modeling by Sketchup Pro

Sketchup Pro is a user friendly software for the design of 3D models that provides full ability to design entire buildings, both the exterior and the interior building parts. Within Sketchup several design methods may be used, either through solids or surfaces and floors, depending mostly on the final model's requirements or specifications. Concerning its interoperability options with other CAD software, such as AutoCAD, the connection with Google Earth and its ability to generate accurate 3D models by using architectural plans, make Sketchup particularly useful. The connection with the 3D Warehouse of Google and the use of its components optimize the models appearance considerably in terms of realism. The program provides tools for grouping objects, thus facilitating manipulation and management of specific groups of homogeneous features and supports adding layers according to the project's requirements. Furthermore, Sketchup Pro comprises features for classifying objects and exporting files according to some of the industry's most common Building Information Modeling (BIM) standards. BIM as a digital version of all the substantial and functional features of a building through its entire life cycle [Isikdag et al., 2007; Isikdag et al., 2013], refers to a sophisticated geometric and semantic representation of the building parts. BIMs such as IFC enclose advanced semantics and geometric decompositions, while their sophisticated attributes are not available in other building or city models. Based on the above remarks, the fact that Sketchup supports file export according to BIM standards is a step forward in terms of semantic enrichment within Sketchup models. Models created in SketchUp Pro can also be exported to AutoCAD and other CAD formats, 3Dstudio format (.3ds), Collada (.dae), KML (Keyhole Markup Language)/KMZ, VRML (.wrl), while SketchUp can also export 2D images (JPG, TIFF, etc.) (Panchal et al, 2011). For the purpose of this pa per, SketchUp Pro 8 was used and proved to be a user friendly tool that is easy to learn but at the same time it concerns a highly detailed and time-saving modeling process.

2.3. Semantic modeling - CityGML

Real world objects such as buildings may be decomposed into parts (rooms, wall surfaces, doors, windows etc) according to their usage or building parts' type. This information can be ontologically modeled in entities, attributes and relations to be used by different users in a number of applications. In this direction, CityGML developed as an official standard of the OGC. According to the www.citygml.org definition "CityGML is a common information model and XML-based encoding for the representation, storage, and exchange of virtual 3D city and landscape models" which "is implemented as an application schema for the Geography Markup Language version 3.1.1 (GML3), an official extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211" (www.citygmlwiki.org). CityGML "defines in different, well-defined Levels-of-Detail, the three- dimensional geometry, topology, semantics and appearance of the most relevant topographic objects in urban or regional contexts" [Gröger & Plümer, 2012]. CityGML as a multipurpose information model [Kolbe et al. 2005] enables the selection of features for thematic visualizations and focuses on the representation of the semantic properties of 3D city models and their structures. The semantics is expressed by classes, which aim to cover a large group of commonly used real- world objects [Zlatanova et al, 2012]; because of its XML base, CityGML presents good interchange format, though not easily processable in large applications unless specific compatibility rules are implemented [Falkowski et al., 2009; Kolbe, 2009]. For example in CityGML, all boundary coordinates (outer and interior boundaries) must be located on the same plane, straight curves are only allowed and features are geometrically represented by Boundary Representation, while buildings are represented by solids defined by their (non-overlapping and without gaps) bounding surfaces. Other specifications' constraints facilitating compatibility options with other databases comprise the coordinate reference system denoted explicitly (facilitating the integration of data) and the ability for implicit geometries for features having complex geometry and occurring multiple times. Moreover, the theory of cell complexes, where geometrical objects can be shared by two solids, also apply to the specifications' constraints [Gröger & Plümer, 2012]. The semantic and constraint features of a CityGML building model include an AbstractBuilding, with each building being able to be composed of Rooms and IntBuildingInstallations, while all required obstacles can only be partially represented by Building- Furniture, IntBuildingInstallation and indirectly from BoundarySurfaces [Brown et al., 2013]. Regardless, the use of the CityGML format enables a high degree of semantic and syntactic interoperability between applications [Herrlich et al, 2010]. Furthermore, according to the LoD concept of CityGML, data from different sources and application areas may be more easily integrated when features are represented in the same level of detail [Gröger & Plümer, 2012]. Because size and complexity of these models continuously grow, a LoD concept effectively supporting the partitioning of a complete model into alternative models of different complexity and providing metadata, addressing informational content, complexity and quality of each alternative model is indispensable [Benner et al., 2013]. CityGML supports 5 different levels of details covering also semantic aspects, starting from LoD0 to LoD4 (indoor), amongst which the most prominent are the LoD of buildings [Zlatanova et al, 2012]. Nonetheless, Biljecki et al. [2013] argue that despite the popularity of the LoD concept, "albeit it is one of the most successful contributions of CityGML that has been the most cited concept of CityGML so far and has been anticipated and used in scientific research", it is criticised and revisions are often suggested. Criticism mostly refers to the association of geometry to semantics and their incorporation and represention as a whole, especially when referring to interior building structures. Consequently, the revision of the LoD concept has been suggested [Benner et al., 2013] as well as the disconnection of semantic and geometrical LoDs, due to problems in definition of CityGML by specific LoDs [Biljecki et al., 2013]. Benner et al. [2013] conduct a short synopsis on various LoD concepts, and indentify the shortcomings of the current CityGML LoD concept for 3D city modeling. They develop an alternative concept distinguishing between a Geometric Level of Detail (GLoD) and a Semantic Level of Detail (SLoD), implementing the new concept via a UML model.

A CityGML model is organized as a “profile” containing several thematic modules (relief, land use, city furniture, building and many more) and their feature types described by feature classes, attributes, semantic definition, relationships and 3D spatial representation.

Applications based on CityGML and particularly its semantics refer to the wider thematic areas of urban development, energy management, property taxation, navigation (indoor as well), natural disasters simulation, environment preservation, cultural heritage registration and military operations.

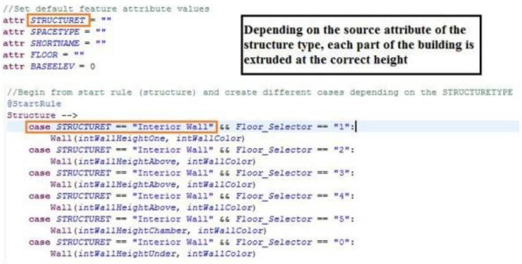
3. Case Study

In this paper, two cases are presented in order to evaluate available conversion tools, integration possibilities and constraints within the CityGML framework. The first focuses on the procedural generation of a 3D building model using ESRI CityEngine whereas the second, examines the possibilities of the Trimble SketchUp Pro software application.

The 3D building model was created after collecting the essential input data provided by the Urban Planning authorities and the Cadastre (National Cadastre & Mapping Agency S.A.). More specifically, the architectural and coverage floor plan, as well as a cross section of the building were provided by the competent planning authority. In addition, a georeferenced orthophoto of the scene’s terrain was used, and ground-based images were captured, in order to obtain the building's different textures. Utilizing these data, a 3D building model was generated, and the transformation possibilities within the CityGML framework were evaluated.

3.1. Building with CityEngine

Procedural modeling was used in order to explore the method’s capabilities related to precise 3D building modeling for both interior and exterior space. A basic condition for achieving this, especially for the interior level, is to collect the necessary information (raster and vector), both two-dimensional and three-dimensional if available, and to organize them in a functional way. Specifically, the information extracted by the collected data was used in order to decompose the building into its building parts and organize them in thematic layers. The building parts were further characterized by their individual attributes (e.g. the type of structure, the floor, etc.), thus securing a high level of functionality while using and identifying them. The height information for the entirety of the building parts derived from the floor plans' attributes. Each building part was assigned the corresponding height value respectively using ESRI ArcScene software and finally the preprocessed data was imported into CityEngine. The most significant part of the whole process was the creation of the rule files which were used for the generation of the building's 3D model. The building parts (e.g. interior and exterior walls) were extruded by using the rules, while specific characteristics were assigned to all parts. The latter was achieved via linking the rule with the characteristics already assigned to the building parts (Fig. 1).



```
//Set default feature attribute values
attr STRUCTURET = ""
attr SPACETYPE = ""
attr SHORTNAME = ""
attr FLOOR = ""
attr BASEELEV = 0

//Begin from start rule (structure) and create different cases depending on the STRUCTURETYPE
@StartRule
Structure -->
    case STRUCTURET == "Interior Wall" && Floor_Selector == "1":
        Wall(intWallHeightOne, intWallColor)
    case STRUCTURET == "Interior Wall" && Floor_Selector == "2":
        Wall(intWallHeightAbove, intWallColor)
    case STRUCTURET == "Interior Wall" && Floor_Selector == "3":
        Wall(intWallHeightAbove, intWallColor)
    case STRUCTURET == "Interior Wall" && Floor_Selector == "4":
        Wall(intWallHeightAbove, intWallColor)
    case STRUCTURET == "Interior Wall" && Floor_Selector == "5":
        Wall(intWallHeightChamber, intWallColor)
    case STRUCTURET == "Interior Wall" && Floor_Selector == "0":
        Wall(intWallHeightUnder, intWallColor)
```

Depending on the source attribute of the structure type, each part of the building is extruded at the correct height

Fig. 1 Part of the rule file indicating the link with the initial attributes

The rule containing the description of the façade's geometry can be integrated in the same rule file so as to produce the outer of the building simultaneously. By using this 3D modeling method, a detailed model of the building's interior space and of the outer shell has been created, combining the building’s geometrical representation as well as the semantics and the representation of the thematic characteristics (Fig. 2 & 3).

What is more related to this paper’s scope though is the investigation of integration possibilities and interoperability of the procedural modeling technique within the CityGML framework. CityEngine offers the capability to export the generated 3D model into limited file formats(OBJ Wavefront, FBX Autodesk, DAE COLLADA, KML, RIB, VOB, SHP ESRI and GDB ESRI FileGDB) and it does not offer a direct format compatible with CityGML. Most of the export options (e.g. OBJ, KML) have the disadvantage of missing the semantic information incorporated in the initial 3D building model. After research, it seems that the most appropriate method for reaching a compensatory level of integration with CityGML is by exporting the 3D building model as a ESRI FileGDB and then implementing further transformations steps via the FME software (Safe Software). The transformation took place for the whole building, while the CityGML model focused on the ground floor.

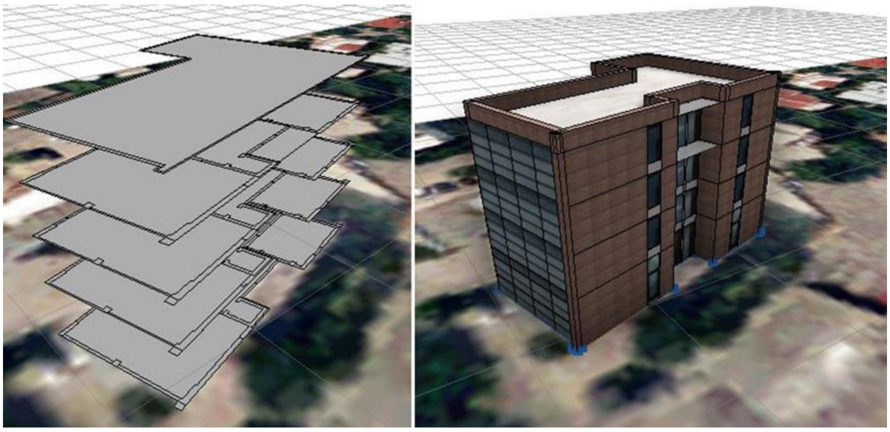


Fig. 2 2.5D Data Representation (Left) & Façade Geometry (Right) in CityEngine

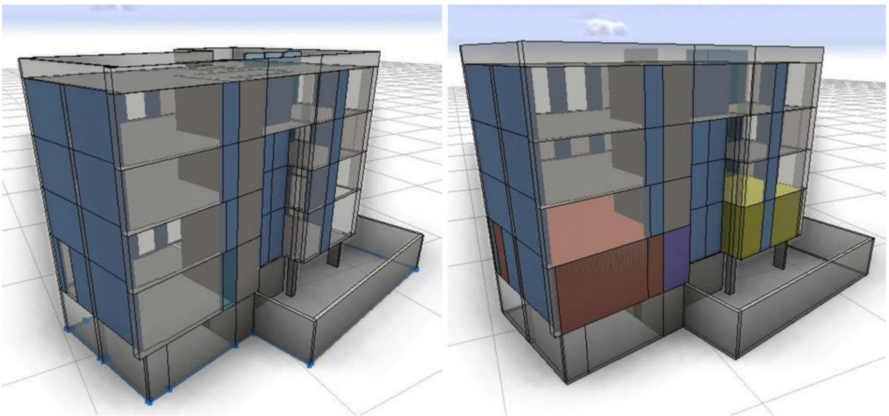


Fig. 3 3D Building Model with interior 3D structures (Left) and with interior 3D volume spaces (Right)

FME (Feature Manipulation Engine) is a powerful transformation engine produced by Safe Software Inc., which helps users convert data (both geometry and attributes) into several formats. FME supports the creation of CityGML format, since CityGML is an international standard for modeling and exchanging 3D city models and there is also a growing need for semantic models. More specifically, it reads and writes CityGML up to version 2.0, it supports any arbitrary ADEs and all thematic modules and levels of detail. In addition, FME includes key transformers for CityGML such as Attribute Creator and Geometry Property Setter. The procedure followed in this case study is described below (Fig. 4).

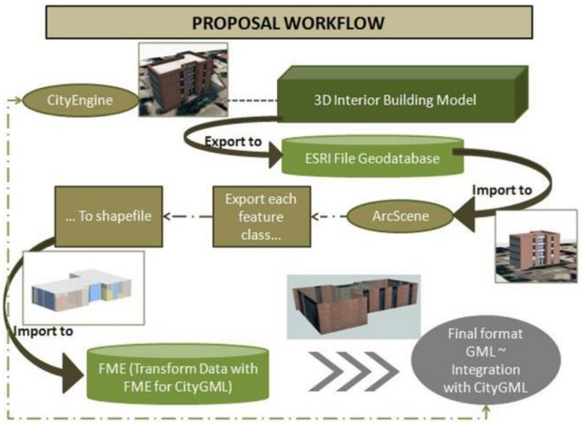
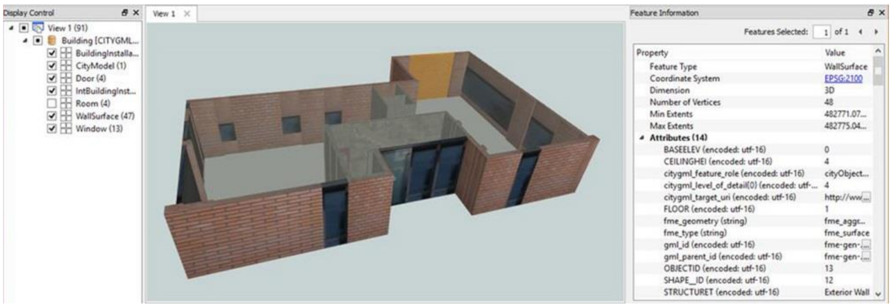

PROPOSAL WORKFLOW
CityEngine
3D Interior Building Model
Export to
ESRI File Geodatabase
Import to
ArcScene
Export each feature class...
... To shapefile
Import to
FME (Transform Data with FME for CityGML)
Final format GML ~ Integration with CityGML

Fig. 4 3D Building model transformation workflow from CityEngine to CityGML

As mentioned before, the 3D building model exported as an ESRI File Geodatabase, was imported into ArcScene, for exploring further representation and analysis options. Since FME interface does not support File Geodatabase feature classes (version 2.0), it was necessary to export each feature class as a separate Shapefile Feature Class (Fig.4). For example, one layer was exported containing all the interior building parts whereas another layer contained the components of the building's exterior shell. It is worth mentioning that the geometry type of the exported building model from CityEngine is MultiPatch, which caused multiple problems at an early stage of this research, related to the conversion into a CityGML compatible format. The first step after importing the data into FME Workbench interface was the distinction of the building parts according to CityGML specifications. For instance, the exterior walls constituted a feature type (WallSurface) and the interior walls a distinct feature type. The main steps for writing CityGML from our shapefiles were the addition of CityGML specific attributes (for instance, gml_id, gml_name and gml_parent_id) as well as geometry properties. It is crucial to use exact names for the attributes (for example, gml_id and not GMLid). As far as the geometry properties are concerned, FME provides the necessary transformation tools (e.g. GeometryPropertySetter, GityGMLPropertySetter etc.), which assign the appropriate feature role to the corresponding geometry type. Thus, the geometry type is defined as MultiSurface and the geometry properties vary depending on the feature type (e.g. for an exterior wall, the geometry property "CityGML_feature_role" should be identified as "boundedBy" according to

CityGML specifications whereas the same attribute for the stairs is identified as "interiorBuildingInstallation", for rooms is "interiorRoom", and finally for doors and windows is "opening"). The data was further organized depending on its specific use within the building and on the CityGML specifications' documentation (e.g. Wall Surface, Door, Window, etc.). Overall, the feature types generated are WallSurface, Window, IntBuildingInstallation, Door, Window, Room and BuildingInstallation. After writing to the preferable format using FME Workbench, the output model can be optimized and represented via FME Data Inspector (Fig. 5).

A problem that encountered when using FME as the transformation software concerns the texture mapping on the building parts. Specifically, all data must be structured in a very detailed way, since difficulties arised in the process of defining areas where different textures (e.g. windows) should apply. This inefficiency could be resolved through editing the initial data's characteristics, by enriching them with attributes such as the material of the structure. The next step could be the utilization of the Geometry XQuery, an option of FME for isolating a structure's specific parts. Nonetheless, FME is an efficient tool for generating CityGML format considering the data is well-structured, as in the example mentioned before.



- ■ View 1 (91)
  - ■ Building [CITYGML...
    - ☑ BuildingInstalla...
    - ☑ CityModel (1)
    - ☑ Door (4)
    - ☑ IntBuildingInst...
    - □ Room (4)
    - ☑ WallSurface (47)
    - ☑ Window (13)

Feature Information

Features Selected: 1 of 1

Property	Value
Feature Type	WallSurface
Coordinate System	EPSG:2100
Dimension	3D
Number of Vertices	48
Min Extents	482771.07...
Max Extents	482775.04...
Attributes (14)	
BASEELEV (encoded: utf-16)	0
CEILINGHEI (encoded: utf-16)	4
citygml_feature_role (encoded: utf-16)	cityObject...
citygml_level_of_detail(0) (encoded: utf-...	4
citygml_target_uri (encoded: utf-16)	http://ww...
FLOOR (encoded: utf-16)	1
fme_geometry (string)	fme_aggr...
fme_type (string)	fme_surface
gml_id (encoded: utf-16)	fme-gen-...
gml_parent_id (encoded: utf-16)	fme-gen-...
OBJECTID (encoded: utf-16)	13
SHAPE__ID (encoded: utf-16)	12
STRUCTURET (encoded: utf-16)	Exterior Wall

Fig. 5 CityGML Building Model in FME Data Inspector

3.1.1 Results

The process of transforming 3D building models proceduraly generated into CityGML results in the following remarks:

- Interoperability issues can be addressed, although in a rather complicated manner concerning the software used, but still is a feasible process.
- It is advisable to organize the structure of the initial data according to CityGML specifications, so as to facilitate the transformation procedure.

- FME provides the necessary tools in order to harmonize the initial 3D model with CityGML standard.
- The time required for the generation of the entire building (comprising the interior parts and the outer shell) by procedural modeling is more laborious than modeling building blocks sharing the same architectural type by using a single rule file. Programming skills may reduce the time required for the whole process. The same applies for the transformation by the FME tool; familiarity with the process minimizes the time needed.
- Since CityGML does not support legal and administrative objects, the geometric and semantic modeling of property rights and restrictions of the buildings examined can only be implemented by extending the CityGML with an Application Domain Extension (ADE) [Çağdaş, 2012].
- The procedurally generated 3D building model has to be enriched with semantic information for the creation of the final CityGML model. The approach may be characterized as semi-automatic, since the building had been decomposed to its building parts at an early stage of the whole procedure. Also, the buildings parts were organized into specific thematic layers and were attributed according to their use (e.g. structure type). Despite the fact that this stage was quite auxiliary to the conversion process, additional transformations had to be completed within the FME interface in order to create a CityGML semantic building model. This was achieved by adding feature and geometry properties manually into FME Workbench.
- The notion of level of detail (LoD) was examined according to CityGML specifications. More specifically, LoD 4 was theoretically achieved since the building's interior space was modeled. Nevertheless, interior elements such as furniture were not included as in the CityGML definition of LoD, while the façade of the building is procedurally modeled in LoD 2 without creating a detailed architectural building model (LoD 3).

3.2. SketchUp Pro

Google SketchUp Pro 8 was used for the design of the 3D building model. The boundaries of the entire modeling area were digitized on the corresponding true orthophoto, in AutoCAD 2013. The digitized boundaries of the surrounding buildings (in .dwg file format) were imported and further processed within SketchUp Pro 8. The buildings' heights were assigned according to the number of floors with an average height of 3 meters per floor, in LoD 1.

The 3D model of the relevant building, which will be extruded in a higher level of Detail (LoD 4 for the interior space and LoD 2 for the exterior shell) was completed by using the SketchUp powerful drawing tools based on the building's architectural plans. SketchUp provides additional tools to add textures and colors to the model's outer and interior parts, as shown in Fig. 6.

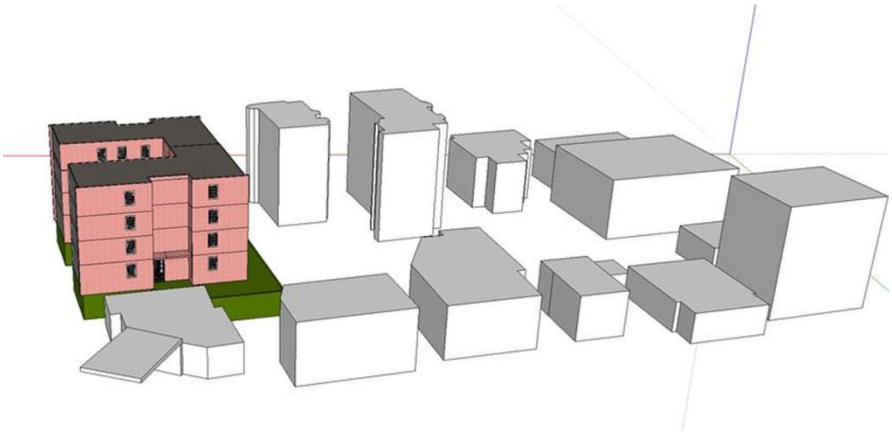


Fig. 6 General view of the 3D modeling area

The 3D building model's interior was gradually constructed focusing on the distinct creation of each floor level, resulting in the creation of different .skp files for each floor. The .skp files were additionally processed by assigning specific layer names to their building parts. The CityGML entities used were WallSurface, GroundSurface, RoofSurface, Door and Window and the layers created were labeled accordingly as: Building-1_WallSurface etc. Once all CityGML-based layers, corresponding to building parts had been created, the model was geolocated via connecting to Google Earth using EPSG 900913 Geodetic Parameter Dataset. The final model (Fig. 7), was created at LoD 4 for the interior and LoD 2 for the exterior.

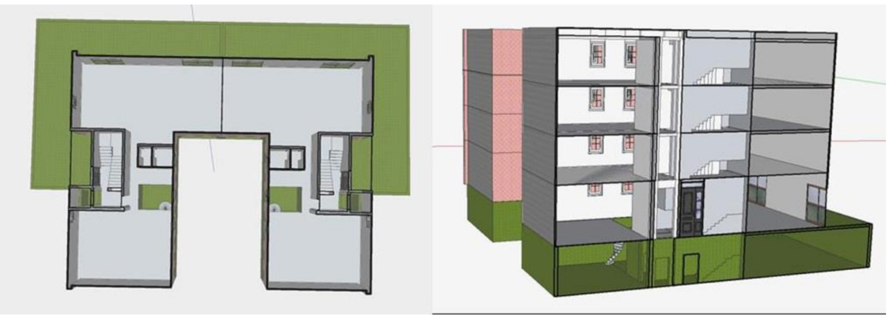


Fig. 7 View of the building's interior

The next step is the conversion of the Sketchup 3D model into GML format, which may be accomplished by utilizing various approaches:

- using commercially available plug-ins for immediate conversion at different LoDs (GEORES),
- using free external plugins, "CityGML Toolbar for Sketchup" (by GEORES) for the creation of CityGML layers and "CityGML editor" (by Westfälische Hochschule) for reading, editing, and exporting CityGML files in LoD2
- through FME Workbench module.

Finally, the latter was used for the current transformation. As mentioned above each building part was saved as a distinct .skp file, five in total, imported and processed within FME Workbench (Fig. 8), producing a single GML file by key transformers. The final GML file can be viewed with FME Data Inspector module (Fig. 9).

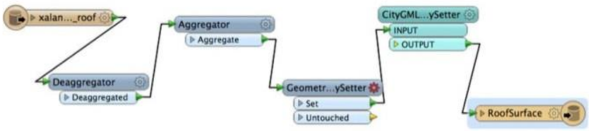
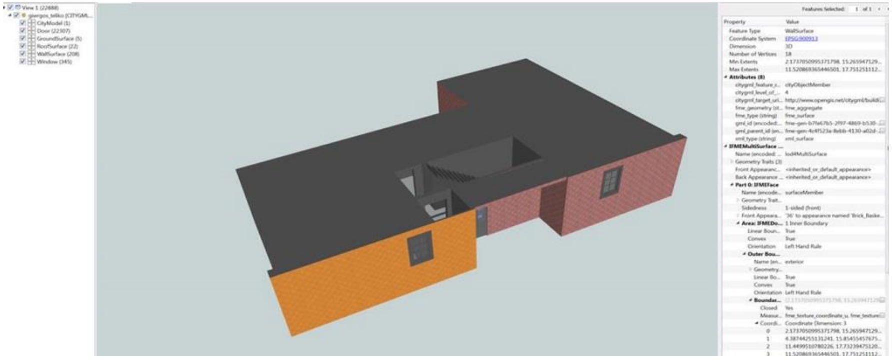


Fig. 8 Example of the FME Workbench



- ☑ View 1 (22888)
  - ☑ giwrgos_teliko [CITYGML...
    - ☑ CityModel (1)
    - ☑ Door (22307)
    - ☑ GroundSurface (5)
    - ☑ RoofSurface (22)
    - ☑ WallSurface (208)
    - ☑ Window (345)

Features Selected: 1 of 1

Property	Value
Feature Type	WallSurface
Coordinate System	EPSG:900913
Dimension	3D
Number of Vertices	18
Min Extents	2.1737050995371798, 15.265947129...
Max Extents	11.520869365446501, 17.751251112...
Attributes (8)	
citygml_feature_r...	cityObjectMember
citygml_level_of_...	4
citygml_target_uri...	http://www.opengis.net/citygml/buildi...
fme_geometry (st...	fme_aggregate
fme_type (string)	fme_surface
gml_id (encoded:...	fme-gen-b7fe67b5-2f97-4869-b530-...
gml_parent_id (en...	fme-gen-4c4f523a-8ebb-4130-a02d-...
xml_type (string)	xml_surface
IFMEMultiSurface ...	
Name (encoded: ...	lod4MultiSurface
Geometry Traits (3)	
Front Appearanc...	<inherited_or_default_appearance>
Back Appearance ...	<inherited_or_default_appearance>
Part 0: IFMEFace	
Name (encode...	surfaceMember
Geometry Trait...	
Sidedness	1-sided (front)
Front Appeara...	'36' to appearance named 'Brick_Baske...
Area: IFMEDo...	1 Inner Boundary
Linear Boun...	True
Convex	True
Orientation	Left Hand Rule
Outer Bou...	
Name (en...	exterior
Geometry...	
Linear Bo...	True
Convex	True
Orientation	Left Hand Rule
Boundar...	(2.1737050995371798, 15.265947129...
Closed	Yes
Measur...	fme_texture_coordinate_u, fme_texture...
Coordi...	Coordinate Dimension: 3
0	2.1737050995371798, 15.265947129...
1	4.38744255131241, 15.85455457675...
2	11.4499510780226, 17.73239475120...
3	11.520869365446501, 17.751251112...

Fig. 9 View of the GML file in FME Inspector

The final GML file can be inserted into 3D CityDatabase, either via Oracle or PostGis database. The latter was selected and a .gml importer/exporter for PostGIS was downloaded from www.3dcitydb.org and installed. The workflow of the whole process is displayed in fig. 10.

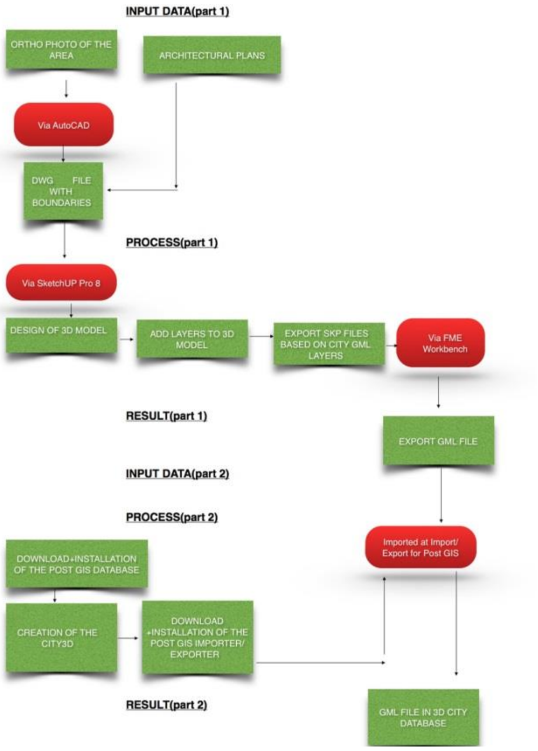


Fig. 10 Workflow of the whole process from data input to the final 3D model

3.2.1 Results

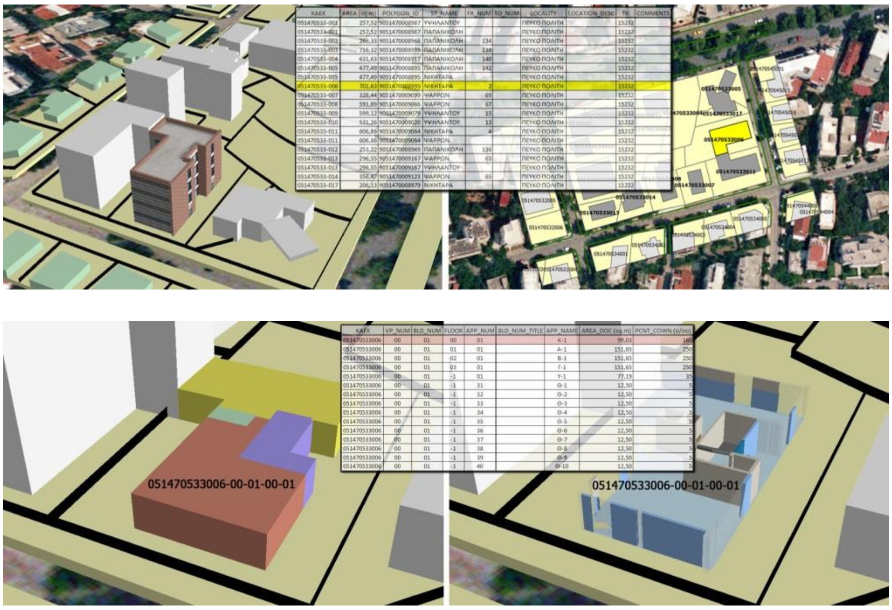
- SketchUp Pro supports geo-location only through Google Earth. As a result, the building footprints were digitized with a relative accuracy, since Google Earth imagery is not true orthophoto rectified.
- SketchUp Pro does not recognize multi-surface building parts. For example both sides (inner & outer) of a wall have to be attributed to the "wall" layer.
- Connecting to the 3D SketchUp Warehouse and exploiting component libraries offers a more realistic and professional visual to the model.
- An "inside to outside" drawing approach was selected, from the inner parts of the building to the outer shell, being a more flexible process in terms of design.
- SketchUp Pro 8 provides powerful tools and procedures, such as object snapping, grouping of features with similar attributes and interactive viewing (Section Plane) for cost & time efficiency, flexibility of use, accuracy in geometry and productivity.

3.3. Integrating additional information to the 3D model

The 3D model created in CityGML can be further exploited for different applications, such as Cadastre, land administration or facility management. As already stated by Döllner et al. [2006], "although typical cadastral databases do not contain 3D data, they provide essential input for 3D building models and a kind of official foundation for virtual 3D city models". Furthermore according to Gröger & Plümer [2012], it is common practice to have "an external reference pointing to the identifier of the object in another information system" in CityGML systems, to "facilitate updates of CityGML features and support the retrieval of additional information".

Accordingly, a field was added to the model's database including the Cadastral ID number of each property unit, thus providing a possible link to the relevant legal rights' visualization, documentation and management. The 3D model may be joined to the Hellenic Cadastral IT system's database, according to the different LoDs selected, thus connecting to the relevant part of the Cadastral ID number.

Fig. 11 & 12 present examples of this linkage from land parcels' level (LoD0) to a more detailed level of single residential or professional properties within buildings (LoD4).



KAEK	AREA (sq.m)	POLYGON_ID	ST_NAME	FR_NUM	TO_NUM	LOCALITY	LOCATION_DESC	TK	COMMENTS
051470533-001	257,52	9051470008987	ΥΨΗΛΑΝΤΟΥ			ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-001	257,52	9051470008987	ΠΑΠΑΝΙΚΟΛΗ			ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-002	266,33	9051470008968	ΠΑΠΑΝΙΚΟΛΗ	134		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-003	716,32	9051470008935	ΠΑΠΑΝΙΚΟΛΗ	138		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-004	631,63	9051470008917	ΠΑΠΑΝΙΚΟΛΗ	140		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-005	477,49	9051470008895	ΠΑΠΑΝΙΚΟΛΗ	142		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-005	477,49	9051470008895	ΝΙΚΗΤΑΡΑ			ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-006	701,63	9051470008995	ΝΙΚΗΤΑΡΑ	2		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-007	328,44	9051470009099	ΨΑΡΡΩΝ	69		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-008	591,89	9051470009066	ΨΑΡΡΩΝ	67		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-009	599,12	9051470009079	ΥΨΗΛΑΝΤΟΥ	15		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-010	531,26	9051470009020	ΥΨΗΛΑΝΤΟΥ	13		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-011	606,86	9051470009064	ΝΙΚΗΤΑΡΑ	4		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-011	606,86	9051470009064	ΨΑΡΡΩΝ			ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-012	253,22	9051470008949	ΠΑΠΑΝΙΚΟΛΗ	136		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-013	296,55	9051470009167	ΨΑΡΡΩΝ	63		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-013	296,55	9051470009167	ΥΨΗΛΑΝΤΟΥ			ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-014	356,47	9051470009123	ΨΑΡΡΩΝ	65		ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	
051470533-017	206,13	9051470008979	ΝΙΚΗΤΑΡΑ			ΠΕΥΚΟ ΠΟΛΙΤΗ		15232	

KAEK	VP_NUM	BLD_NUM	FLOOR	APP_NUM	BLD_NUM_TITLE	APP_NAME	AREA_DOC (sq.m)	PCNT_COWN (o/oo)
051470533006	00	01	00	01		Κ-1	99,01	165
051470533006	00	01	01	01		Α-1	151,65	250
051470533006	00	01	02	01		Β-1	151,65	250
051470533006	00	01	03	01		Γ-1	151,65	250
051470533006	00	01	-1	01		Υ-1	77,19	35
051470533006	00	01	-1	31		Θ-1	12,50	5
051470533006	00	01	-1	32		Θ-2	12,50	5
051470533006	00	01	-1	33		Θ-3	12,50	5
051470533006	00	01	-1	34		Θ-4	12,50	5
051470533006	00	01	-1	35		Θ-5	12,50	5
051470533006	00	01	-1	36		Θ-6	12,50	5
051470533006	00	01	-1	37		Θ-7	12,50	5
051470533006	00	01	-1	38		Θ-8	12,50	5
051470533006	00	01	-1	39		Θ-9	12,50	5
051470533006	00	01	-1	40		Θ-10	12,50	5

Fig. 11, 12 Possible links of 3D model to relevant cadastral records at different LoDs

4. Conclusions & Discussion

3D city models are characterized by complexity, while a semantic basis is required to complement the geometrical and topological aspects. In recent years, the integration of semantics into 3D city models is widely accepted [Zhu et al., 2011]. However, concerns have been expressed due to the lack of standardized 3D city objects representation. Several questions have arised concerning the most effective way to complement the geometries' semantics or the efficiency of semantics extraction from pure geometric models [Zhu et al., 2011]. The emergence of novel 3D modeling methodologies and techniques in computer graphics as well as the development of a range of 3D file formats that sustain interoperability and promote the distribution of information between different CAD and GIS packages has certainly assisted in this direction. Nonetheless, there are a few concerns related to 3D modeling and semantics integration. They mainly refer to the optimal representation of semantics within a holistic city or building model as well as to what kind of semantic information is required for the sufficient depiction of the various concepts of these models.

Due to the great number of existing 3D modeling techniques, data integration and interoperability options present a challenge to researchers. This paper investigated transformation alternatives for 3D building models created by commercially available software systems, into CityGML. The 3D modeling techniques investigated in this research work included compatibility options between procedural modeling employed through ESRI CityEngine software environment and BIM-ready tools through Trimble Sketchup Pro within CityGML's requirements. CityGML (e.g. FME) or GIS/ CAD tools with CityGML import/export were taken into account. The conclusions of this study may be summarized as follows:

- CityEngine, although may export the generated 3D model into several file formats, does not offer a direct format compatible with CityGML. Most of the export options (e.g. OBJ, KML) have the disadvantage of missing semantic information incorporated in the initial 3D building model. On the contrary, there is more flexibility when converting the Sketchup model to CityGML format, as more conversion tools are available.
- The most appropriate way to reach a level of integration with CityGML is by using the FME software for the final transformation steps. Although FME provides the necessary tools in order to harmonize the initial 3D model with CityGML standard, it doesn't support the textured model that was procedurally generated in CityEngine. Thus the appropriate transformer tool was used for this purpose. In contrast, the textures from the Sketchup model were directly identified and represented via FME inspector.
- Finally, it was shown that Sketchup provides the option to export models directly in different LoDs, while CityEngine does not equally support an interactive export in various LoDs.

Despite the use of CityGML in a variety of applications, it does not actually support legal and administrative objects as well as the geometric and semantic modeling of their property rights and restrictions. In this direction, modeling and integration strategies are required by the administration, the cadastral and the planning authorities, in the form of data harmonization and exchange, based on international standards, in order to facilitate integrated management of land related information. This is an ongoing research focusing on investigating integration possibilities with several BIM models, such as Industry Foundation Classes-IFC, whose sophisticated attributes are not yet widely available.

5. References

Becker, S., Walter, V., & Fritsch, D. (2012). Integrated Management of Heterogeneous Geodata with a Hybrid 3d Geoinformation System. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,. Melbourne, Australia. I-2.*

Benner, J., Geiger, A., Gröger, G., Häfele, K. H., & Löwner, M. O. (2013). Enhanced LoD concepts for virtual 3D city models. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2, W1.

Biljeckia, F., Zhaoa, J., Stotera, J., & Ledouxa, H. (2013, September). Revisiting the concept of level of detail in 3D city modelling. In ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. Proceedings of the ISPRS 8th 3D GeoInfo conference & WG II/2 workshop (pp. 63-74).

Brown, G., Nagel, C., Zlatanova, S., & Kolbe, T. H. (2013). Modelling 3D topographic space against indoor navigation requirements. In Progress and New Trends in 3D Geoinformation Sciences (pp. 1-22). Springer Berlin Heidelberg.

Çağdaş, V. (2013). An Application Domain Extension to CityGML for immovable property taxation: A Turkish case study. International Journal of Applied Earth Observation and Geoinformation, 21, 545-555.

Dimopoulou, E., & Elia, E. (2012). Legal aspects of 3D property rights, restrictions and responsibilities in Greece and Cyprus. In Proceedings of the 3rd International Workshop on 3D Cadastres, Developments and Practices pp. 25-26.

Döllner, J., Baumann, K., Buchholz, H., 2006. Virtual 3D City Models as Foundation of Complex Urban Information Spaces. In: Schrenk, M. (Ed.), Proc. 1st international conference on Urban Planning and Spatial Development in the Information Society (REAL CORP). CORP - Competence Center of Urban and Regional, Planning, pp. 107-112.

Falkowski , K., Ebert , J., Decker , P., Wirtz , S., Paulus, D., 2009. Semiautomatic generation of full CityGML models from images. In: Geoinformatik 2009, ifgiPrints, Vol. 35, Institut für Geoinformatik Westfälische Wilhelms-Universität

Gröger, G., Kolbe, T., & Czerwinski, A. (2007). Candidate OpenGIS® CityGML Implementation Specification (City Geography Markup Language). Open Geospatial Consortium Inc, OGC.

Gröger, G., Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. ISPRS Journal of Photogrammetry and Remote Sensing 71 (2012) 12–33

Herrlich, M., Holle, H., & Malaka, R. (2010). Integration of cityGML and collada for high-quality geographic data visualization on the PC and Xbox 360. InEntertainment Computing-ICEC 2010 (pp. 270-277). Springer Berlin Heidelberg.

Institute of Electrical and Electronics Engineers (1990) IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.

Isikdag, U., Aouad, G., Underwood, J., & Wu, S. (2007). Building information models: a review on storage and exchange mechanisms. Bringing ITC Knowledge to Work.

Isikdag, U., Zlatanova, S., & Underwood, J. (2013). A BIM-Oriented Model for supporting indoor navigation requirements. Computers, Environment and Urban Systems, 41, 112-123.

Jesus, D., Coelho, A., Rebelo, C., & Cardoso, A. (2012, May). Modeling Urban Environments from Geospatial Data: A Pipeline for Procedural Modeling. InProceedings of the third workshop on Procedural Content Generation in Games p. 5. ACM.

Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. In 3D geo-information sciences (pp. 15-31). Springer Berlin Heidelberg.

Kolbe, T., H., 2009. Representing and Exchanging 3D City Models with CityGML. Proceedings of the 3rd International Workshop on 3D Geo-Information, Seoul, Korea. Lectures in Geoinformation and Cartography, Springer Verlag.

Kolbe, T.H., 2008. Representing and Exchanging 3D City Models with CityGML. In: Lee, J., Zlatanova, S. (Eds.), 3D Geo-information Sciences. Springer, Berlin, pp.15–31.

Métral, C., Billen, R., Cutting-Decelle, A. F., & Van Ruymbeke, M. (2010). Ontology-based approaches for improving the interoperability between 3D urban models. Journal of Information Technology in Construction, 15.

Muller, P., Wonka, P., Haegler, S., Ulmer, A., & Gool, L.V., (2006). Procedural Modeling of Buildings. ACM Transactions on Graphics, 25(3), pp. 614-23.

Panchal, H., Khan, R., Sengupta, S., (2011). GIS-based Smart Campus System using 3D Modeling. Geospatial World Forum, 18-21 January 2011, Hyderabad, India,

Stiny, G., Gips, J., 1972. Shape Grammars and the Generative Specification of Painting and Sculpture. Information Processing 71, 1460-1465.

Tutenel, T., Smelik, R. M., Bidarra, R., & de Kraker, K. J. (2010). A Semantic Scene Description Language for Procedural Layout Solving Problems. AIIDE,10, 1-6.

Wonka, P., Wimmer, M., Sillion, F., & Ribarsky, W., 2003. Instant architecture. ACM Transactions on Graphics, 22(3), pp. 669-677.

Zhang, H., Li, Y., Liu, B., & Liu, C. (2014). The application of GIS 3D modeling and analysis technology in real estate mass appraisal taking landscape and sunlight factors as the example. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1, pp. 363-367.

Zhu, Q., Zhao, J., Du, Z., Zhang, Y., Xu, W., Xie, X., & Wang, T. (2011). Towards semantic 3D city modeling and visual explorations. In Advances in 3D Geo-Information Sciences (pp. 275-294). Springer Berlin Heidelberg.

Zlatanova, S., Stoter, J., & Isikdag, U. (2012, June). Standards for exchange and storage of 3D information: Challenges and opportunities for emergency response. In Proceedings of the Fourth International Conference on Cartography and GIS, Albena, Bulgaria (pp. 17-28).

http://www.citygml.org/index.php?id=1523, What is CityGML?

http://www.citygmlwiki.org/index.php/Main_Page, CityGML - City Geography Markup Language

http://www.esri.com/software/cityengine

http://www.safe.com/fme/

http://www.slideshare.net/SafeSoftware/how-to-easily-read-and-write-citygml-data-using-fme

AUTOMATIC CONTROL OF BUILDINGS' ZONING REGULATION COMPLIANCE BASED ON LIDAR-DERIVED MODELS

U.Isikdag [a,] *, I. Buyuksalih [b],

[a] Beykent University, Istanbul, Turkey - uisikdag@gmail.com
[b] BIMTAS, Istanbul,Turkey- ibuyuksalih@yahoo.com

Commission II, WG II/2

KEY WORDS: LiDAR, Zoning, Building, Turkey, Control

ABSTRACT:

Airborne LiDAR (Light Detection And Ranging) is now becoming more widely used as a technique for deriving 3D geometric information with accuracy .The use of LiDAR increased the productivity in production of accurate Digital Elevation Models (DEMs) and Digital Surface Models (DSMs). The LiDAR technique facilitates the rapid production of models which contain ground and above surface objects in 3D (Jaafar et al., 1999). Today automatic / semi-automatic generation of 3D digital building models and other city furniture from LiDAR data is a very active area of scientific research. Elaksher and Bethel (2002) indicates that manual surface reconstruction is very costly and time consuming, and the development of automated algorithms is of great importance. Airborne LIDAR emerged as a relatively new technology for obtaining Digital Surface Models (DSMs) of the earth's surface and now utilized as a fast method for sampling the earth's surface with a high density and high point accuracy. Airborne LiDAR is known as an active remote sensing technology. It actively transmits pulses of light toward an object of interest, and receives the light that is scattered and reflected by the objects. An airborne LiDAR system is typically composed of three main components: a laser scanner unit, a Global Positioning System (GPS) receiver, and an Inertial Measurement Unit (Liu, 2008). Airborne LIDAR systems derive irregularly spaced 3D point measurements of objects including ground, buildings, trees and cars scanned by the laser beneath the aircraft. As mentioned by Zhang et al.(2006) building footprints are one of the fundamental GIS data components that can be used to estimate energy demand, quality of life, urban population and property taxes in an urban environment. Accurate building footprint data is considered as essential for construction of urban landscape models, assessment of urban heat island effect and estimation of building base elevation for flood insurance. Furthermore, footprint data in combination with height values can be used to generate 2,5D building models for visualization. As explained by Cheng et.al (2008) in order to obtain the segmented building points from raw LiDAR data, the first process is usually to separate the ground points from non-ground points, and then identify the building points from non-ground points. Numerous algorithms have been developed to separate ground points from non-ground points.

Recent research illustrated that building density indicators such as Building Coverage Ratio (BCR), Floor Area Ratio (FAR), and other ones can be numerically and automatically derived from Building Representations in high-resolution airborne LiDAR data. Bailang et al (2010) explained that the recognition of building objects in LiDAR data and their association with land lots and urban districts allow for the calculation of various building density attributes. The building density attributes derived in their analysis include three types: (1) geometric and volumetric attributes for individual buildings; (2) density attributes at the land lot scale; and (3) geometric, volumetric and density attributes at the urban district scale. These attributes are used to characterize and analyse the physical form and structure of the urban landscape for environmental studies and land use management at different scales. In zoning studies, Building Coverage Ratio (BCR) and Floor Area Ratio (FAR) are the most commonly used metrics in zoning regulations in different countries. These metrics are developed for controlling the building density in certain areas of the city.

In Turkey zoning regulations that are enforced are indicated in "Implementation Development Plan"- a municipal plan controlling development and construction within an urban zone and includes all implementation details marked on a map scaled 1/1000.The Implementation Development Maps (scaled 1/1000) are prepared by municipalities, based on Implementation Development Plans of that city, and all implementation details of the zoning regulations are shown as marked in these 2D Maps. Similar to many countries, Turkish zoning regulations enforced in Implementation Development Plans focus on 3 metrics, Building Coverage Ratio (BCR) and Floor Area Ratio (FAR) and HMax. The BCR can be defined as the ratio of the building coverage area (i.e. the area of building footprint) to the size of land lot. Since the footprint represents the planimetric shape of a building, the BCR measures the building density in two-dimension (2D) space (Bailang et al., 2010). The Floor Area Ratio (FAR) can be defined as the ratio of gross building floor area to the size of land lot. The value of FAR is determined not only by the planimetric shape of the building, but also by the vertical distribution of the floors in different height, thus it depicts the three-dimensional (3D) building density. The HMax can be defined as the maximum allowable height of the building. In Turkey all the building projects are checked prior to the start of the construction process if they conform the allowed BCR, FAR and HMax. For instance, for a given land lot of 1000 sqm. if BCR=0.5 and FAR=2 and Hmax=20, the proposed building can have 4 floors with total gross area of 2000 sqm where the area of building footprint is 500 sqm, or can have 5 floors again with total gross area of 2000 sqm, but in this second case the area

* Corresponding author.

of building footprint needs to be 400 sqm. In fact, a building with a footprint area of 500 sqm cannot have 5 floors (which would exceed FAR=2).In addition the height of the building cannot exceed the gived HMax for that lot. Considering 300 cm as the average floor height for a regular building in Turkey, HMax for the given case do not allow buildings with 7 floors. Although the projects are checked prior to the start of the construction by municipalities, in metropolitan cities such as Istanbul, Ankara, Izmir, it observed in practice that final form of the constructed buildings do not conform / comply with the zoning regulations. Furthermore in mega-cities such as Istanbul, there have been a great number of buildings that had been built without taking permission from municipal authorities and these are termed as “Illegal Construction or Illegal Buildings”. Since a long time, these buildings (i.e. the building that do not conform with zoning regulations) are considered as one of the biggest problems in urban development in Turkey. Last year the Greater Municipality of Istanbul have made investments for obtaining LiDAR data for several parts of the city. Today, with this LiDAR data in place, it is possible to derive models and information (e.g. the heights of the buildings).This paper focuses on and explains our efforts on automatically checking Buildings’ Zoning Regulation Compliance by integrating geometric information derived from 3D LiDAR data and semantic information acquired from 2D Implementation Development Maps. Our initial test results revealed that the approach holds a great potential for Buildings’ Zoning Regulation Compliance.

References

J. Jaafar, G. Priestnall and P.M. Mather 1999. The effects of LIDAR DSM grid resolution on categorising residential and industrial buildings. Proceedings of the ISPRS Workshop, 9-11 NOVEMBER 1999, La Jolla, USA.

Zhang, K. Yan, J.; Chen, S-C. 2006 Automatic Construction of Building Footprints From Airborne LIDAR Data, Geoscience and Remote Sensing, IEEE Transactions on , Vol.44, No.9, pp.2523,2533, Sept. 2006 DOI: 10.1109/TGRS.2006.874137

Elasksher, A. Bethel, J. 2002. Building Extraction Using Lidar Data, ASORS-ACSM Annual Conference and FIG XXII Congress, Apr. 22-26, 2002.

Liu, X. 2008. Airborne LiDAR for DEM generation: some critical issues Progress in Physical Geography, Vol. 32, No. 1. pp. 31-49.

Cheng, L. Gong, J. Chen, X. Han, P. 2008. Building boundary extraction from high resolution imagery and LiDAR data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B3b.

Bailang,Y. Hongxing,L Jianping,W. Yingjie,H. Li,Z. 2010, Automated derivation of urban building density information using airborne LiDAR data and object-based method. Landscape and Urban Planning. 98(3-4), 210-219. DOI:10.1016/j.landurbplan.2010.08.004

ENABLING GEO-DESIGN: EVALUATING THE CAPACITY OF 3D CITY MODEL TO SUPPORT THERMAL DESIGN IN BUILDING

I. H. Hijazi[a,] *, M. HAJ HUSSEIN [a], R. Koenig[c]

[a] Dept. of Urban Planning Engineering, An-Najah University Nablus, Palestine - eehab@najah.edu
[b] Dept. of Building Engineering, An-Najah University, Nablus, Palestine - haj_mj@najah.edu
c Chair of Information Architecture, Dept. of Architecture, ETH Zurich, Switzerland -
reinhard.koenig@arch.ethz.ch

Commission II, WG II/2

KEY WORDS: BIM, CityGML, gbXML, CAD, Thermal design

ABSTRACT:

This paper reports a study that has been undertaken as part of an on-going project to examine the capacity of 3D city models to support thermal design of building. The two standard used for this is CityGML and gbXML. The first is an OGC standard for the exchange of 3D city information, and the second is a CAD standard to exchange of information between engineering and environmental analysis software. In particular, our premise is that effective thermal design relies on the ability to exchange urban environment information such as surrounding building, green areas, streets, trees. CityGML is a 3D information model that provides a detailed information about the topography and manmade objects - surrounding environment data in a robust way using an open information standard GML. The focus of this present work has been on the processes involved in design with the urban context and how it could affect the design workflow for advising on thermal performance issues. We begin by looking at the kinds of data that needs to be exchanged to support such design in context. We then test the exchange using a pilot project model and finally review the capacity of the CityGML standard to support such processes.

1. INTRODUCTION

The development of semantic 3D city models has allowed for new approaches to town planning and urban management (Benner et al. 2005) such as emergency and catastrophe planning, checking building developments, and environmental design.

Building sector in the world presents a great amount of energy consumption in comparison to other sectors such as industry, transportation. This energy is mostly used to improve the thermal performance of living spaces (NREL, 2006; Karasu, 2010; Geetha & Velraj, 2012).

Building thermal performance and energy efficiency is greatly influenced by urban planning legislation and surroundings environments (Geetha & Velraj, 2012)). Architects took account of the relationship between newly designed building and the surrounding environment. 3D city models can play an essential role for energy engineer by facilitating design simulation within the surrounding environment and urban context in early design stages. Currently, thermal simulation tools do not easily integrate the urban context (surroundings) of the new design; this is related to the interoperability issues between CAD and GIS.

This paper report the result of investigation undertaken to examine the capacity of 3D City models specifically CityGML - OGC standard; to support design process related to thermal performance issues. The paper use a case study approach to define the data required for heat demand calculation and thermal data processing including geometrical data processing. Then the paper investigates the possibility to extract the required data from the 3D city models in order to be exchanged using Green Building XML schema (gbXML) and to be used by thermal simulation software. The exchange then tested using a pilot project model and finally reviews the capacity of the 3D City model standard to supports such process.

2. BACKGROUND IN CITYGML/GBXML

gbXML is an extensible markup language that was developed to facilitate the transfer of building information between different CAD standard and engineering environmental analysis software. It aims to enable interoperability between design models and a wide variety of engineering software. Nowadays, gbXML is considered a de facto industry standard schema. The standard has industry support and wide adoption by the leading CAD vendors, Autodesk, Graphisoft, and Bentley (gbXML, 2014).
CityGML is an OGC standard, which provides a specification for the representation of 3D urban objects (CityGML, 2012). It is the only 3D information model for the exchange of 3D city models. One of the reasons for creating such a model was to enrich 3D city models with thematic and semantic information. The information model of CityGML is an XML-based format implemented as an application schema of Geography Markup Language (GML3). Today, CityGML seems to provide the best framework for semantic-geometric relations of 3D objects above the earth surface (Emgaard and Zlatanova, 2008;

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

Groneman and Zlatanova, 2009). It maintains a good taxonomy and aggregations of Digital Terrain Models (DTM), sites (including buildings), vegetation, water bodies, transportation facilities, and city furniture. The underlying model differentiates between consecutive Levels of Detail (LOD), where objects become more detailed with increasing LOD regarding both geometry and thematic differentiation. In LODs 2-4 of CityGML the building facade is defined in the form of boundary surfaces, i.e. wall surface, roof surface, ground surface or closing surface. The LOD4 allows the representation of interior building elements, e.g. rooms, furniture, interior wall surfaces. Nevertheless, the current version of CityGML integrates the subsurface features, such as underground constructions (e.g. tunnels).

3. ONGOING WORK AND OUTLOOK

An important step in this study is the development of the use case. A conceptual model is built for a typical quarter (Figure 1a), a building (in green) need to be extended horizontally and vertically. The urban model is consisting of 25 plots of 36X36 m2 for each (Figure 1b). Seven typologies are proposed for this urban scale model (Figure 1a). The plots number 8 and 22 are green areas on the site, while plot number 13 is the location for our apartment block. A vertical extension of two floors (in red) is asked for this case block, which is already composed of three floors (Figure 1c).

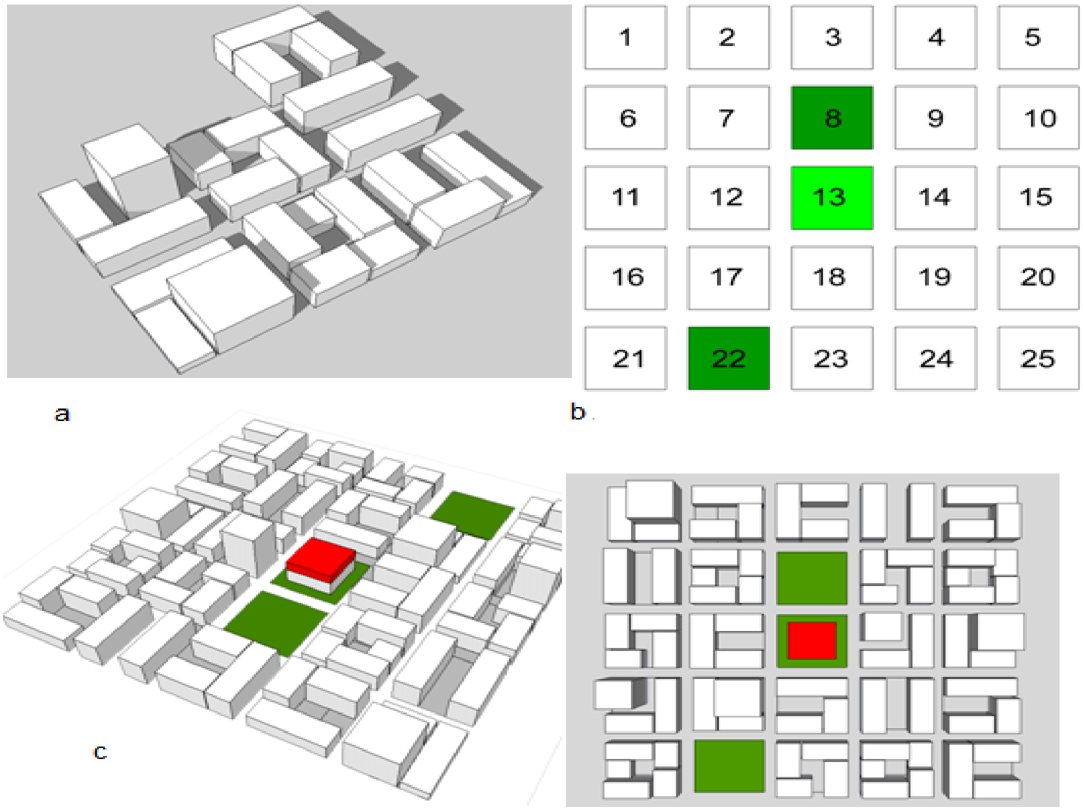


Figure 1: The quarter model, a) the seven typologies, b) plots uses, c) the proposed model in 3D and 2D

The purpose is to get to know the different variables affecting thermal performance. Until today, the factors that intervene is categorized under four main categorizes (Figure 2).

- Space uses and users activities
- Geographical and climatic factors
- Design variables
- Surrounding features

4. EXPECTED RESULTS

The work will illustrate the information requirement for thermal analysis and simulation in geo-context. Also, it will provide initial understanding for building a formal framework for the geometric and semantic transformation of 3D city objects (related to thermal design) between the two data models, gbXML and CityGML. To demonstrate the applicability of the developed framework, a number of trial conversions will be carried out between gbXML and the CityGML models, the conversion will be performed using the tool that will be developed.

Moreover, the work will demonstrate a new possible application for thermal analysis that includes indoor rooms – outdoor space. The prototype capabilities will allow us to answer questions such as: What are the effects of change in the outdoor on the thermal performance of inside of the buildings? Or where the places outside that cause a big change in the temperature of the indoor spaces?

Acknowledgment: This paper has been prepared as a part of research project Modelling City System (MCS) in the chair of Information Architecture (IA), department of Architecture, at ETH Zurich, supported by the Grants Climate-KIC.

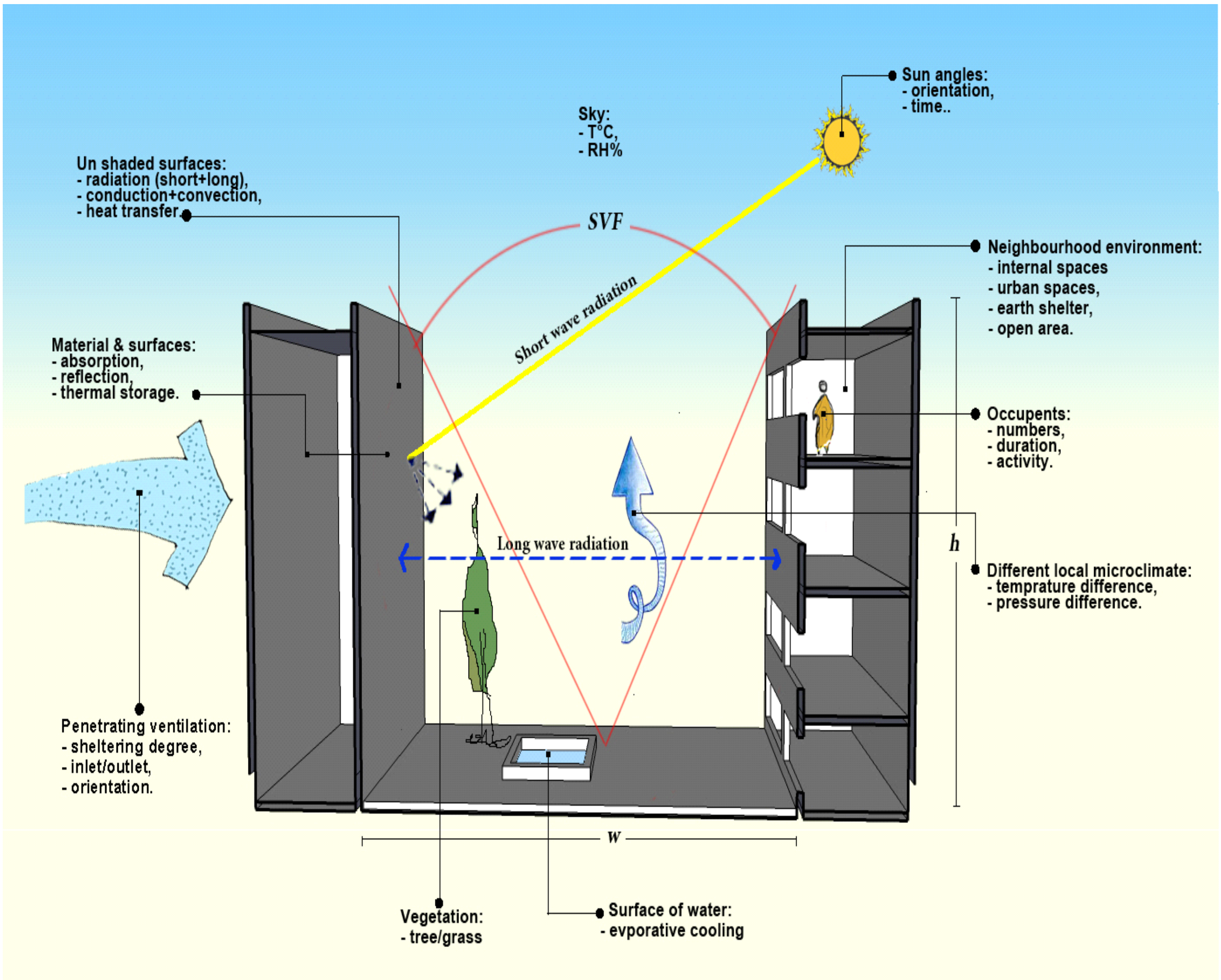


Figure 2: illustrates the different variables intervened in the question of thermal-energy performance of one room located in the new built floor of our apartment block.

5. REFERENCES

A. Groneman, S. Zlatanova, "TOPOSCOPY: a modelling tool for CITYGML". In Onsrud H, R van de V (Eds.), GSDI Association, pp. 1-13, 2009.

Benner, J., A. Geiger and K. Leinemann (2005). "Flexible generation of Semantic 3D building models" In: Proc of the 1st Intern. Workshop on Next Generation 3D City Models, Gröger/Kolbe (Eds.), Bonn, pp. 17-22.

CityGML, 2012, CityGML Encoding Standard document version 2.0.0, Retrieved Mai 2, 2012 from the World Wide Web: http://www.citygml.org/1522/

Emgaard, K. and Zlatanova, S. (2008). Design of an integrated 3D information model, In: Coors V. and Fendel E. and Zlatanova S. (Eds.). Urban and regional data management: UDMS annual 2007, Taylor & Francis Group, London, UK.

Geetha N. B. & Velraj R. (2012) Passive cooling methods for energy efficient buildings with and without thermal energy storage – A review, Energy Education Science and Technology Part A: Energy Science and Research, 29 (2): 913-946

gbXML, 2014, Green Building XML schema, Retrieved Mai 2,2014 from the World Wide Web: http://www.gbxml.org/

Karasu, A. (2010). Concepts for Energy Savings in the Housing Sector of Bodrum, Turkey: Computer based analysis and development of future settlements using renewable energy, PhD Thesis, Technique University Berlin, Germany.

National Renewable Energy Laboratory 'NREL' (2006). Zero Energy Buildings: A Critical Look at the Definition, ACEEE Summer Study, Pacific Grove, California.

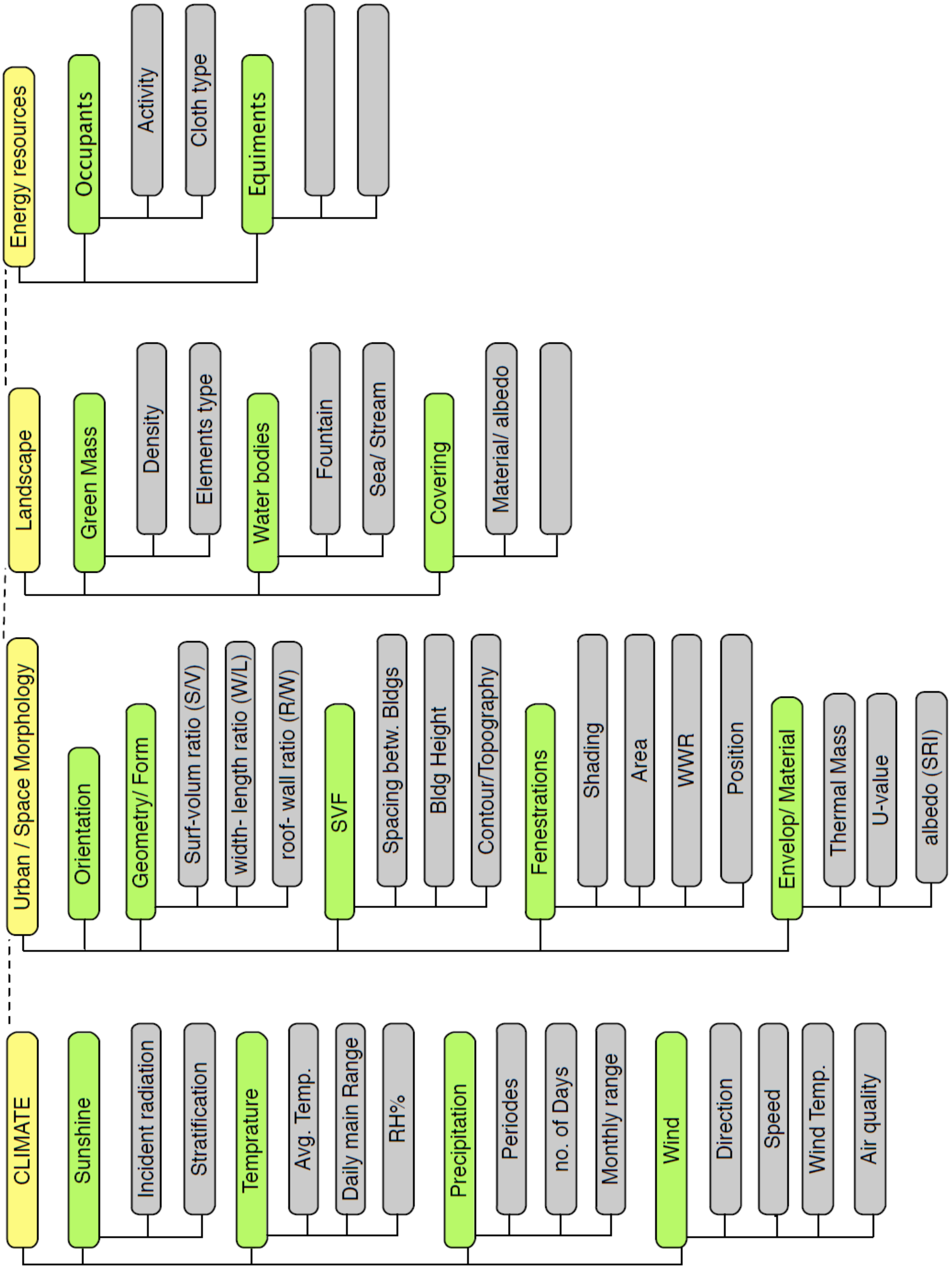


Figure 3: Information requirement for indoor/outdoor thermal analysis