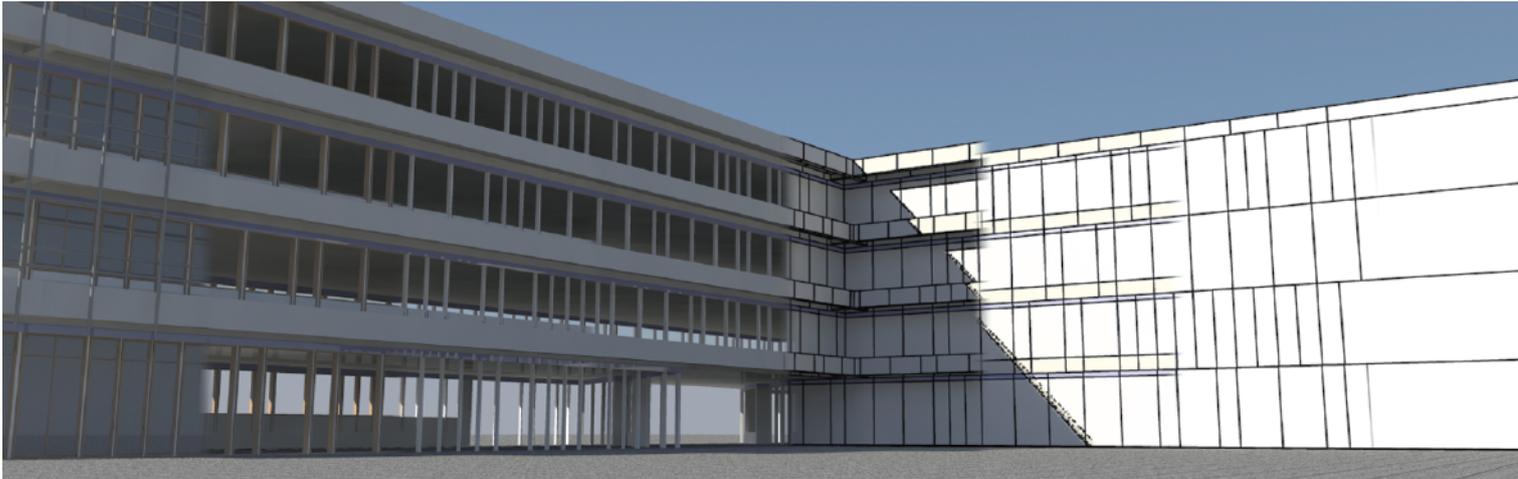


Dieter Finkenzeller



Modellierung komplexer Gebäudefassaden in der Computergraphik



universitätsverlag karlsruhe

Dieter Finkenzeller

**Modellierung komplexer Gebäudefassaden in der
Computergraphik**

Modellierung komplexer Gebäudefassaden in der Computergraphik

von
Dieter Finkenzeller



universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH), Fakultät für Informatik
Tag der mündlichen Prüfung: 4. Februar 2008
Erster Gutachter: Prof. Dr. rer. nat. A. Schmitt
Zweiter Gutachter: Prof. Dr. rer. nat. O. Deussen

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziiert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2008
Print on Demand

ISBN: 978-3-86644-231-3

Danksagung

Die Erforschung des Themas führte ich während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Betriebs- und Dialogsysteme der Universität Karlsruhe durch.

An erster Stelle möchte ich Herrn Prof. Dr. Alfred Schmitt für die Betreuung meiner Arbeit und dafür, dass er mir die Durchführung meines Forschungsthemas ermöglicht hat, herzlich danken. Seine Hinweise und Ratschläge zu meiner Arbeit waren eine wertvolle Hilfe zur erfolgreichen Gestaltung der Abhandlung. Herrn Prof. Dr. Oliver Deussen danke ich für die Übernahme des Korreferats. Außerdem danke ich ihm dafür, dass er mich – trotz der Entfernung – bei meiner Forschung unterstützt hat. Die fachlichen Diskussionen mit ihm haben mir bei meiner Arbeit sehr geholfen. Herrn Prof. Dr. Hartmut Prautzsch gilt mein Dank für die fachliche Unterstützung bei meinen geometrischen Problemstellungen.

Bei den Implementierungen im Rahmen von Diplom- und Studienarbeiten geht mein besonderer Dank an Kevin Fucik für die Gesimse und Mauerwerkstexturen sowie an Thomas Ring für die Anregungen bei den Arbeiten zu den Zyklopenmauerwerken. Matthias Baas danke ich besonders für seinen Rat bei der Implementierung des Prototyps und den vielen ergebnisreichen Diskussionen über *Python* und *RenderMan* im Verlauf meiner Arbeit. Günter Härdle und Sven Thüring möchte ich für die fachliche Beratung im Bereich der Architektur danken. Günter Härdles Literaturhinweise haben einen entscheidenden Beitrag zum Gelingen meiner Arbeit geleistet. Sven Thüring stand mir vor allem bei der Gestaltung der Gebäudebeispiele mit praktischen Ratschlägen zur Seite. Meinen Kollegen Dr. Jan Bender und Raphael Straub danke ich besonders für die Hilfestellung bei fachlichen Fragestellungen und für die Durchsicht und Korrekturvorschläge meiner Ausarbeitung. Für die Korrekturen und Verbesserungsvorschläge bedanke ich mich recht herzlich bei Kerstin und Dr. Michael Fautz. Ebenso möchte ich hier Bertrand Klimmek meinen Dank für das Korrekturlesen meiner Veröffentlichungen aussprechen. Benjamin Bertram danke ich vor allem für die wertvolle und erfolgreiche Prüfungsvorbereitung.

Den werten Kollegen am Institut möchte ich für die angenehme und entspannte Atmosphäre und die gute Zusammenarbeit danken. Vor allem brachten die stets lustigen Institutsfilmabende eine willkommene Abwechslung. Schließlich bedanke ich mich bei meinen Eltern für ihre fortwährende Unterstützung.

Inhaltsverzeichnis

1	Einführung	1
1.1	Zielsetzung	2
1.2	Aufbau der Arbeit	3
2	Grundlagen	5
2.1	Prozedurale Techniken	5
2.1.1	Grammatiken	5
2.1.2	Lindenmayer-Systeme	6
2.1.3	Shape grammars	6
2.2	Repräsentation von 3D-Modellen	7
3	Verwandte Arbeiten	9
3.1	CAD-Programme	9
3.2	Texturierung architektonischer Elemente	10
3.3	Erzeugung von Dachstrukturen	14
3.4	Geometrisches Modellieren architektonischer Elemente	15
3.5	Erzeugung und Beschreibung von Grundrissen	16
3.6	Systeme für die prozedurale Modellierung von Gebäuden und Städten	18
3.6.1	Wissenschaftliche Systeme	18
3.6.2	Kommerzielle Produkte	28
3.7	Vergleich der Techniken	30
3.8	Fazit	38
4	Grundrissbeschreibung	41
4.1	Allgemeines	41

4.2	Grundriss eines Stockwerks	42
4.2.1	Grundrissmodul	42
4.2.2	Punkt-Punkt-Verbindung	44
4.2.3	Kante-Kante-Verbindungen	46
4.2.4	Grundriss	48
4.2.5	Merkmale	50
4.2.6	Einschränkungen und Sonderfälle	57
4.3	Erweiterung auf Stockwerke	58
4.3.1	Unterteilung	58
4.3.2	Merkmale	62
4.4	Dachrepräsentation	64
4.4.1	Dachformen	65
4.4.2	Kombination von Dächern	70
5	Fassadenbeschreibung	73
5.1	Gebäudegrundstruktur	73
5.2	Basiswände und Hausecken	74
5.2.1	Wände und Zwischenwandelemente	75
5.2.2	Wand-, Zwischenwand- und Eckpartitionen	75
5.3	Mauerwerke	76
5.3.1	Mauerziegelverbände	78
5.3.2	Zyklopenmauerwerke	88
5.4	Dekorationen	93
5.4.1	Gesimse, Konsolen und Friese	93
5.4.2	Ornamente	105
5.5	Fenster und Türen	114
5.5.1	Verfeinerung der Fenster- und Türöffnungen	114
5.5.2	Fenster- und Türstöcke	117
5.5.3	Fenster- und Türrahmen	120
6	Semantik von Gebäudefassaden	125
6.1	Anforderungen	126
6.2	Typisierte Graphen	127

6.2.1	Gerichtete Graphen	127
6.2.2	Typen	128
6.2.3	Attribute	128
6.3	Wissensbasen	128
6.4	Datenbanken	129
6.5	Vergleich zwischen TGraphen, Wissensbasen und Datenbanken . . .	130
6.6	Fassadensemantik mit TGraphen	132
6.6.1	Konzepte und Relationen allgemein	133
6.6.2	Konzepte und Relationen der Gebäudestruktur	134
6.6.3	Konzepte und Relationen des Fassadenstils	137
6.6.4	Anwendung des Stils auf die Gebäudestruktur	141
6.6.5	Vom Weltmodell zum Zwischenmodell	142
6.6.6	Vorteile der Technik	143
7	Geometrie-Fabrik	145
7.1	Ordnungsrelationen	145
7.2	Geometrieerzeugung	146
7.3	Prototyp	148
7.4	Ergebnisse	152
8	Zusammenfassung und Ausblick	159
8.1	Zusammenfassung	159
8.2	Ausblick	162
	Anhang	165
A	Definition des semantischen Modells	165
A.1	Konzepte und Relationen der Gebäudestruktur	165
A.2	Konzepte und Relationen des Fassadenstils	168
	Literaturverzeichnis	175

Abbildungsverzeichnis

3.1	Ein pflanzenähnliches Ornament.	11
3.2	Eine Steinwand.	12
3.3	Ein Torbogen mit verschiedenen Steintypen.	13
3.4	Ein Mauerwerk mit natürlich wirkenden Ziegelsteinen.	13
3.5	Beispiel für ein komplexeres Fenster.	15
3.6	Ein einfach unterteiltes Fenster.	16
3.7	Eine ganze Stadt, basierend auf L-Systemen.	20
3.8	Gebäude, erstellt mit der <i>split grammar</i>	21
3.9	Pompeji, erstellt mit L-Systemen und der <i>shape grammar</i>	23
3.10	Aus Teilen eines realen Gebäudes zusammengesetztes Gebäude.	25
3.11	Eine virtuelle Stadt in Echtzeit generiert.	26
3.12	Einfaches Gebäude, modelliert mit einem <i>3D Studio Max</i> Plugin.	27
3.13	Grobe Gebäudestruktur, modelliert mit SketchUp.	29
3.14	Gebäude, erzeugt mit ArCon.	30
4.1	Grundrissmodul.	43
4.2	Grundriss der Petronas Towers.	44
4.3	Zwei PP-Module; links mit einer, rechts mit zwei PP-Verbindungen.	46
4.4	Grundriss, bestehend aus zwei Modulen.	49
4.5	Merkmale eines Grundrisses.	50
4.6	Merkmal ist gleich dem Umriss des Moduls.	51
4.7	Merkmal mit zwei dominanten Verbindungen.	52
4.8	Merkmal mit zwei Verbindungen, wobei die linke dominant ist.	52
4.9	Merkmal mit zwei Verbindungen, wobei die rechte dominant ist.	53
4.10	Merkmal mit zwei Verbindungen, wobei keine dominant ist.	54

4.11	Merkmal zwischen zwei Verbindungen.	54
4.12	Merkmal an einer PP-Verbindung mit angedocktem Modul.	55
4.13	PP-Verbindung als Merkmal.	55
4.14	Merkmal zwischen zwei PP-Verbindungen.	56
4.15	1. Sonderfall bei PP-Verbindungen.	58
4.16	2. Sonderfall bei PP-Verbindungen.	58
4.17	Unterteiltes Grundrissmodul mit Verbindungsmöglichkeiten.	60
4.18	Gebäudestruktur, die aus verschiedenen Stockwerken aufgebaut ist.	61
4.19	Drei Merkmale bei Stockwerken.	62
4.20	Merkmal: externe Verbindung.	63
4.21	Merkmale bei Stockwerken.	64
4.22	Flachdach.	66
4.23	Schematische Darstellung der allgemeinen Dachform.	66
4.24	Walmdach.	68
4.25	Satteldach.	68
4.26	Pulldach.	69
4.27	Zeltdach.	69
4.28	Zwei Grundrissmodule mit Dächern.	71
4.29	Zwei Grundrissmodule mit angepassten Dächern.	71
5.1	Nahtstellen der groben Gebäudestrukturen.	74
5.2	Bildschirmfoto aus dem Online-Spiel <i>World of Warcraft</i>	77
5.3	Beispiele für Läuferverbände.	78
5.4	Beispiele für Kreuz- und Blockverbände.	78
5.5	Beispiele für armierte Mauerwerke.	79
5.6	Beispiele für Quaderstein- und unregelmäßige Schichtmauerwerke.	79
5.7	Beispiel für einen Mauerziegelverband.	82
5.8	Beispiel für ein Grauwertbild.	83
5.9	Verschiedene einfache Mauerziegelverbände.	84
5.10	Verschiedene überlappende Mauerziegelverbände.	85
5.11	Konkaver Quaderstein.	86
5.12	Beispiele für Quaderstein- und unregelmäßige Schichtmauerwerke.	87
5.13	Beispiel für ein Zyklopenmauerwerk.	88

5.14	Beispiel für ein Voronoi-Diagramm.	89
5.15	Beispiel einer problematischen Ecke.	91
5.16	Beispiele für Zyklopenmauerwerke.	92
5.17	Gesimse und Konsolen.	94
5.18	Illustrationen zur Profildefinition.	96
5.19	Transformation der Profilbasen.	97
5.20	Beispiel für ein Gesims mit Beschreibung.	97
5.21	Beispiel für ein Gurtgesims.	98
5.22	Platzierung der Konsolen im Gesims.	99
5.23	Beispiele für Friese.	100
5.24	Beispiel für eine Fries-Textur mit Blattwerk-Ornament.	101
5.25	Probleme der Texturierung bei Gesimsen.	102
5.26	Fehlerhafte und korrigierte Texturierungen.	103
5.27	Beispiel für die Lösung des Texturierungsproblems mit Anfangs-, Mittel- und Endtexturen.	103
5.28	Lösung für das Texturierungsproblem mit Anfangs-, Mittel- und Endtexturen.	104
5.29	Ornamente, klassifiziert nach ihren Motiven.	106
5.30	Ornamente, klassifiziert nach ihrer Form.	107
5.31	Mittelachsentransformation.	109
5.32	Ein-, Ausgaben, Eigenschaften und Funktionsweisen des Ornament- elements.	112
5.33	Beispiel für ein Ornament.	113
5.34	Definition einer Fenster- oder Türöffnung.	115
5.35	Verfeinerung der Kante durch einen Kantenzug.	116
5.36	Beispiel für die Verfeinerung einer Fensteröffnung.	117
5.37	Zusammenspiel der Fenster-/Türstöcke mit dem Ornamentrechteck.	118
5.38	Beispiel für die Verfeinerung einer Fensteröffnung mit Fensterstock und Ornamentrechteck.	119
5.39	Verschiedene Fensterformen als Vektormodell.	122
5.40	Ornamentierte Fensterstöcke mit Quellornamenten.	123
6.1	Beispiel für ein einfaches semantisches Netz.	129
6.2	Semantisches Modell der groben Gebäudestruktur.	136

6.3	Semantisches Modell des Fassadenstils.	141
6.4	Vom Konzeptmodell zum 3D-Modell.	144
7.1	Abhängigkeiten der Fassadenelemente und deren Ordnungsrelationen.	147
7.2	Konfiguration der Ausgabeformate.	149
7.3	Konzept- und Weltmodell.	150
7.4	Das Zwischenmodell.	151
7.5	Freistehendes Haus mit Gesimse und Ornamente.	152
7.6	Nahaufnahme des Hauses.	152
7.7	Gesamtansicht des Modells des Informatikgebäudes.	153
7.8	Sicht aus dem zweiten Stock.	153
7.9	Das Informatikgebäude im klassischen Stil.	154
7.10	Dieselbe Sicht wie in Abbildung 7.8, wobei das Gebäude in einem klassischen Stil gehalten ist.	154
7.11	Die Grundstruktur des Fakultätsgebäudes.	155
7.12	Großes Gebäude mit Haupt und Seitenflügel.	155
7.13	Eine Gebäudezeile, aus einfachen Häusern aufgebaut.	156
7.14	Die Gebäudezeile aus voriger Abbildung mit Klinker als Mauer- werkstextur.	156
7.15	Die Gebäudezeile mit einem Kreuzverband.	157
7.16	Modell des Karlsruher Schlosses (Rückansicht).	157
7.17	Foto des Karlsruher Schlosses.	158
7.18	Modell des Karlsruher Schlosses.	158

Tabellenverzeichnis

3.1	Gesamtsysteme zur prozeduralen Modellierung von Gebäuden. . . .	33
3.2	Texturierungstechniken und ihre Anwendbarkeit auf architektonische Elemente.	34
3.3	Techniken zur Erzeugung von Dächern.	35
3.4	Verfahren zur geometrischen Modellierung architektonischer Elemente.	36
3.5	Methoden zur Beschreibung von Grundrissen.	37
4.1	Daten eines Grundrissmoduls.	43
6.1	Vergleich von TGraphen, Wissensbasen und relationalen Datenbanken hinsichtlich der gestellten Anforderungen.	132

Kapitel 1

Einführung

Der Entwurf komplexer Strukturen ist ein Kerngebiet der Computergraphik. Speziell im Bereich der Unterhaltung – hierunter fallen vornehmlich Computerspiele und Animationsfilme – hat leistungsstarke Hardware komplexere und hochwertigere Szenen ermöglicht. Um so wichtiger ist es, den Entwurf dieser Szenen durch leistungsfähige Software zu unterstützen. Die manuelle Modellierung virtueller Umgebungen, wie Landschaften und Städte, ist sehr mühsam. Jedes Objekt muss aus einfachen Strukturen aufgebaut werden. Ebenso ist es aufwendig Texturen für jedes Objekt zu erstellen. Änderungen zu einem späteren Zeitpunkt sind nur noch schwer durchführbar. Abhilfe erwartet man sich deshalb vom computergestützten Modellieren auf Basis formaler Spezifikationen und Prozeduren. Damit verlagert sich zum einen die Modellierung auf eine abstraktere Ebene. Der Prozess der Modellierung wird dadurch beschleunigt, womit eine größere Vielfalt an Modellen einhergeht. Zum anderen werden dadurch Zeit und damit verbunden auch Kosten eingespart.

Das prozedurale Modellieren von Pflanzen und Landschaften ist bereits eingehend untersucht und hat sehr gute Ergebnisse hervorgebracht. Ebenso sind bei der automatischen Generierung von Städten ansehnliche Erfolge zu verzeichnen. Eine besondere Herausforderung stellt aber die computergestützte Modellierung detaillierter Gebäudefassaden auf symbolischer Ebene dar. Darunter versteht man die Abbildung der Fassadenelemente auf ein semantisches Modell, in dem die Eigenschaften der Fassade bearbeitet werden. Anschließend wird auf Grundlage des Modells die vollständige Geometrie der Fassade erzeugt.

1.1 Zielsetzung

Die computergestützte Modellierung detaillierter Gebäudefassaden muss grundlegenden Anforderungen genügen, von denen der Erfolg des Vorgehens maßgeblich abhängig ist. Im Folgenden werden diese näher erläutert:

- **Interaktion**

Die Modellierung einer Gebäudefassade ist als interaktiver Prozess zu realisieren. Der Benutzer muss jederzeit eingreifen können, um Änderungen vorzunehmen und das Modell seinen Wünschen entsprechend anzupassen. Die Benutzung soll auf einem abstrakten, intuitiven Niveau erfolgen.

- **Trennung von Gebäude und Stil**

Die Beschreibung des groben Gebäudemodells muss unabhängig vom Stil erfolgen. Dual dazu können Stile für sich selbst existieren.

- **Semantische Modelle**

- **Konzeptmodell**

In einem abstrakten Modell ist eine Spezifikation des groben Gebäudes und des Stils zu erstellen. In ihm spiegeln sich die Elemente der Gebäudestruktur und des Stils wider. Abhängigkeiten zwischen Elementen werden als binäre Relationen nachgebildet.

- **Weltmodell**

Die konkrete Ausprägung eines Gebäudes, also seine grobe Struktur und sein Stil, sind Instanzen des abstrakten Modells. Dieses Modell enthält auch Vorschriften für Verfeinerungen der Gebäudestruktur.

- **Zwischenmodell**

Durch Anwendung der Unterteilungsvorschriften des Weltmodells entsteht als letztes semantisches Modell das Zwischenmodell. Hier, wie auch in den anderen Modellen, können Änderungen auf semantischer Ebene vorgenommen werden. Dieses Modell beinhaltet minimale geometrische Informationen, anhand derer automatisch Geometrie und Texturen der gesamten Gebäudefassade berechnet werden.

Die Modelle sind auf eine konkrete Datenstruktur abzubilden, die ihre syntaktische Struktur widerspiegelt.

- **Räumliche Abhängigkeiten**

Anhand der semantischen Modelle müssen Nachbarschaftsbeziehungen innerhalb eines Stockwerks und stockwerkübergreifend einfach ableitbar sein.

Damit wird einerseits bei der Erzeugung der Gebäudefassade sichergestellt, dass aneinander grenzende Elemente automatisch angepasst werden können. Andererseits werden dadurch Texturen ermöglicht, die von der darunter liegenden Geometrie abhängig sind.

- **Prozedurale Texturen**

Texturen für Mauerwerke sollen über eine einfache Beschreibungssprache definiert werden können. Sie lassen sich so bei Bedarf in beliebiger Auflösung erzeugen. Damit wird ein Einsatz in Echtzeitanwendungen denkbar, da die Auflösung der Texturen abhängig vom Betrachter gewählt werden kann. Außerdem kann damit die Qualität der Textur in Abhängigkeit von der verfügbaren Graphikhardware erstellt werden. Ebenso lässt sich damit die Qualität der Texturen an die verfügbare Graphikhardware anpassen.

- **Flexible Datenhaltung**

Die in den vorherigen Punkten geforderten Eigenschaften verlangen nach einer einheitlichen, flexiblen Datenhaltung. Zum einen müssen Änderungen am semantischen Modell schnell vollziehbar sein. Zum anderen muss die Datenstruktur auch die Beschreibung der Texturen aufnehmen können.

- **Kompression durch Konstruktionsbeschreibung**

Dadurch, dass aus der semantischen Beschreibung der Fassade mittels Zwischenmodell Geometrie und Texturen generiert werden können, benötigt ein vollständiges Gebäude nur wenig Speicherplatz.

Bisher existiert kein System, das allen Anforderungen gerecht wird. Die Aufgabe besteht darin, ein System zur Modellierung komplexer Gebäudefassaden zu entwickeln. In dieser Arbeit werden Techniken vorgestellt, die die Anforderungen erfüllen und im Zusammenspiel ein prototypisches Gesamtsystem ergeben. Dabei werden bekannte Techniken aufgegriffen und weiterentwickelt. Ebenso werden neue Techniken vorgestellt.

1.2 Aufbau der Arbeit

Kapitel 2 enthält einen Überblick über notwendige Grundlagen der Arbeit. Darunter fallen formale Grammatiken und Repräsentationen von 3D-Modellen.

In Kapitel 3 wird eine Übersicht der verwandten Arbeiten gegeben. Dabei wird auf einzelne Techniken – gruppiert in Teilbereiche, z. B. Grundriss- und Dachbeschreibungen – wie auch auf Gesamtsysteme zur computergestützten Erzeugung architektonischer Elemente, Gebäude und Städte eingegangen. Am Ende des Kapitels werden die bekannten Verfahren anhand verschiedener Kriterien tabellarisch miteinander verglichen.

Kapitel 4 beschäftigt sich mit dem Aufbau grober Gebäudemodelle. Dabei wird das Gebäude aus groben Strukturen zusammengesetzt. Zu diesen zählen z. B. Haupttrakt, Seitenflügel, Mauervorsprünge, Dächer usw. Dieses Verfahren wird erst für ein einzelnes Stockwerk eingeführt und anschließend auf mehrere Stockwerke erweitert. Damit können beliebige Grundrisse pro Stockwerk beschrieben werden, wobei räumliche Abhängigkeiten leicht ableitbar sind.

Elemente der Fassade, wie Fenster, Türen, deren Verfeinerungen, Gesimse, Ornamente, Wände, Mauerwerke usw., werden in Kapitel 5 eingeführt. Hier liegt das Augenmerk vor allem auf der Entwicklung einfacher, aber mächtiger Beschreibungsformen. Dadurch lässt sich eine hohe Flexibilität in der Formgebung der Elemente erreichen. Speziell die Beschreibung der Ornamente und Mauerwerke erlaubt die Erzeugung auflösungsunabhängiger Texturen.

Die aus den Kapiteln 4 und 5 gewonnenen Erkenntnisse fließen in Kapitel 6 in die semantischen Modelle ein. Sie erlauben eine dreistufige symbolische Beschreibung der Gebäudefassade. Im *Konzeptmodell* wird eine abstrakte semantische Repräsentation grober Gebäudestrukturen und Stile definiert, die für Gebäudefassaden einheitlich ist. Das *Weltmodell* enthält konkrete Gebäudefassaden, die sowohl Gebäudestrukturen als auch Stile umfassen. Instanzen des Weltmodells sind konkrete Ausprägungen der Objekte des Konzeptmodells. Ebenso enthält das Weltmodell Vorschriften, die die grobe Gebäudestruktur feiner unterteilen. Durch Anwendung dieser Vorschriften entsteht das *Zwischenmodell*, das auch symbolische Informationen beinhaltet. Auf Grundlage dieses Modells werden detaillierte Geometrie und Texturen einer Gebäudefassade erstellt. In allen Modellen kann der Benutzer eingreifen und Anpassungen vornehmen.

Kapitel 7 widmet sich zu Beginn der Geometrie-Fabrik. Es wird ein Verfahren vorgestellt, mit dem die Geometrie und die Texturen einer Gebäudefassade aus dem Zwischenmodell erzeugt werden. Weiterhin werden Ordnungsrelationen zwischen Fassadenelementen definiert, um diese automatisch einander anpassen zu können. Dafür werden räumliche Informationen benötigt, die sich aber dank der semantischen Modelle leicht ableiten lassen. Ebenso werden für die Geometrien exakt passende Texturen erzeugt, so dass sich alle Fassadenelemente nahtlos aneinander fügen lassen. Anschließend wird kurz auf die Implementierung und Bedienung des erstellten Prototyps eingegangen. Das Kapitel schließt mit Ergebnissen, die mit dem Prototyp und den vorgestellten Techniken erzielt wurden.

Das letzte Kapitel fasst die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf mögliche Erweiterungen und Anwendungsgebiete.

Kapitel 2

Grundlagen

Einführend werden in diesem Kapitel relevante Grundlagen der prozeduralen Modellierung besprochen. Als wichtigste Repräsentanten werden Lindenmayer-Systeme, Chomsky-Grammatiken und sog. *shape grammars* vorgestellt. Der nächste Abschnitt behandelt Datenmodelle für die Beschreibung geometrischer Formen.

2.1 Prozedurale Techniken

In diesem Abschnitt werden Grammatiken, Lindenmayer-Systeme und *shape grammars* behandelt. Diese Techniken sind im Allgemeinen im prozeduralen Modellieren recht beliebt. Die vielfältigen Anwendungsmöglichkeiten zeigen uns später die Gesamtsysteme in Abschnitt 3.6. Eine weitere Möglichkeit mit dem Computer Strukturen generativ zu erzeugen, bieten allgemein die Programmiersprachen. Sie stellen die größtmögliche Flexibilität dar, sind aber dadurch – verglichen mit Grammatiken – deutlich fehleranfälliger. Im Folgenden wird aber nur auf die wichtigsten Merkmale der Grammatiken-Techniken eingegangen.

2.1.1 Grammatiken

Formale Sprachen, Grammatiken und die Chomsky-Hierarchie sind ein gut erforschtes Thema der theoretischen Informatik. Eine umfassende Einführung in diesen Themenbereich wird von Salomaa [Sal73] und Schöning [Sch92] gegeben.

Eine Grammatik, die eine Sprache (Typ 0 bis 3) erzeugt, ist definiert als ein Quadrupel $G = (V, \Sigma, P, S)$ mit folgenden Eigenschaften: V ist die endliche Menge der *Variablen* und Σ die endliche Menge der *Terminalsymbole*. Weiterhin gilt: $V \cap \Sigma = \emptyset$. P ist die endliche Menge an Produktionsregeln, $P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$. $S \in V$ ist die *Startvariable*.

Allgemein werden die Variablen und Terminalsymbole als Zeichen kodiert. Bei der Anwendung einer Regel wird dann die entsprechende Variable (linke Seite der Regel) durch die Zeichen (Variablen und/oder Terminalzeichen) auf der rechten Seite der Regel ersetzt. Aufgrund des Mechanismus, wie ein Wort der Sprache erzeugt wird, werden Grammatiken auch als Ersetzungssysteme (*rewriting systems* [Sal73]) bezeichnet.

In unserem Fall kommt auf der linken Seite der Produktionsregeln genau eine Variable vor, d. h. es handelt sich um sog. kontextfreie Grammatiken bzw. eine Sprache vom Typ 2.

2.1.2 Lindenmayer-Systeme

Eine weitere Form von Ersetzungssystemen stellen die Lindenmayer-Systeme oder kurz L-Systeme dar. L-Systeme wurden 1968 vom ungarischen Biologen Aristid Lindenmayer vorgeschlagen, um das Wachstum von Pflanzen mittels einer mathematischen Theorie zu beschreiben. Einen umfangreichen Einblick bietet die Arbeit von Prusinkiewicz und Lindenmayer [PL96]. Die zugrunde liegende Idee ist den Grammatiken sehr ähnlich: Ausgehend von einem Anfangswort, das aus einer oder mehreren Variablen besteht, werden Regeln angewendet und die Variablen entsprechend der Regelvorschrift ersetzt. Im Gegensatz zu den Grammatiken gibt es hier keine Terminalsymbole, da diese ja, bildlich gesprochen, absterbenden Zweigen gleich kämen. Des Weiteren werden bei L-Systemen in einem Schritt immer alle Variablen ersetzt. Wie auch bei den Grammatiken beschäftigen wir uns auch hier nur mit den kontextfreien L-Systemen (*OL-Systeme*). Die formale Definition ist ähnlich zu den Grammatiken, nur dass hier die Terminalsymbole fehlen.

Ein *OL-System* ist ein Tripel $OL = (V, \omega, P)$. V ist wieder das Alphabet der Variablen, $\omega \in V^+$ ist ein Anfangswort und P sind die Produktionsregeln, $P \subseteq (V \times V^*)$.

L-Systeme sind in zahlreichen Anwendungen zu finden. Die wohl bekanntesten Arbeiten mit L-Systemen beschäftigen sich mit der Simulation bzw. der Erzeugung von Pflanzen. Die wichtigsten Vertreter sind in diesem Zusammenhang die Arbeiten von Prusinkiewicz et al. [PL96] und Deussen et al. [DHL+98]. Wie wir in Abschnitt 3.6 sehen werden, sind L-Systeme auch für die Erzeugung von Städten bzw. Straßennetzen sehr gut geeignet. Sie lassen sich aber für die Generierung von Gebäuden nur beschränkt einsetzen.

2.1.3 Shape grammars

Als letzte relevante Technik wird nun kurz auf die Arbeit von Stiny et al. [SG71] zu *shape grammars* eingegangen. Sie dient als Grundlage für verschiedene weiterführende Arbeiten, wie z. B. Wonka et al. [WWSR03]. Informell spezifizieren sie

eine Zeichnung als die Definition einer Sprache für zweidimensionale Formen, die Auswahl einer Form, die gezeichnet werden soll (die sog. *selection rule*, deren Definition weiter unten gegeben wird), die Spezifikation eines Schemas, wie Flächen in einer Form darzustellen sind und die Regeln, wo und in welcher Größe eine Form auf einer Zeichenfläche bestimmter Größe zu zeichnen ist.

Eine Klasse von Zeichnungen definieren sie formal wie folgt: (S, M) , wobei S ein Klasse von Formen spezifiziert, bestehend aus einer *shape grammar* mit Regeln. M spezifiziert die Materialien für die durch S erzeugten Formen mittels einer endlichen Liste von Zeichenregeln und der Zeichenfläche. *shape grammars* werden ähnlich wie Chomsky-Grammatiken beschrieben. Im Unterschied zu diesen besteht bei den *shape grammars* das endliche Alphabet aus Formen. Bei Chomsky werden eindimensionale Zeichenketten erzeugt, wohingegen bei *shape grammars* n -dimensionale Formen generiert werden. Eine *shape grammar* SG ist das Quadrupel (V_T, V_M, R, I) , mit V_T als Menge der Terminalsymbole, V_M als Menge der Variablen, R als Menge der Regeln und I die Startform, bestehend aus Elementen aus V_T^* und einem Element aus V_M . Elemente aus V_T^* bestehen aus einer endlichen Aneinanderreihung von Elementen aus V_T , deren Ausrichtung und Größe beliebig verändert sein darf. Gestartet wird mit der Anfangsform I . Anschließend werden Regeln aus R angewendet, bis keine Regel mehr angewendet werden kann. Die von SG erzeugte Sprache $L(G)$ kann eine endliche oder unendliche Menge endlicher Formen darstellen.

Jeder Anwendungsschritt einer Regel definiert eine Stufe (bezeichnet als *level*) der Form. Durch die Beschränkung der Anwendungsschritte lassen sich Minimum- und Maximumlevel für eine *shape grammar* definieren. Weiterhin werden die durch die Level beschriebenen Areale eingefärbt. Die Einfärbung wird mithilfe eines Venn-Diagramms vorgenommen, womit sich überlagernde Areale verschiedener Level berücksichtigen lassen.

Als konsequente Fortsetzung erweitern Stiny et al. ihre *shape grammars* auf dreidimensionale Objekte. Hier wird fast völlig analog zu den 2D-Grammatiken verfahren. Ein Unterschied besteht darin, dass beim Zeichnen überlappende Areale (hier Räume) nicht unterschiedlich eingefärbt, sondern geschnitten werden.

2.2 Repräsentation von 3D-Modellen

Das Datenmodell eines 3D-Objekts in der Computergraphik beinhaltet verschiedene Informationen, wie z. B. Form, Textur bzw. Materialeigenschaften oder physikalische Daten. In unserem Fall ist primär die Formbeschreibung von Interesse. Nachfolgend wird ein kurzer Überblick über die unterschiedlichen Repräsentationen *Punktmodell*, *Volumenmodell* und *Oberflächenmodell* gegeben. Da wir uns für das Oberflächenmodell entscheiden, wird hierauf detaillierter eingegangen.

Beim *Punktmodell*, das auch die einfachste Form der Repräsentation darstellt, wird die Oberfläche des Objekts allein durch eine Punktwolke, bestehend aus 3D-Punkten, beschrieben. Über den Zusammenhang der Punkte zur Definition der Oberfläche wird keine Aussage gemacht. Dieser wird erst durch die Auswertung ihrer Nachbarschaftsbeziehungen ermittelt. Dadurch wird schließlich eine Transformation in ein Dreiecksnetz, z. B. zur Oberflächendarstellung, möglich.

3D-Objekte werden beim *Volumenmodell* über räumlich disjunkte Volumenprimitive repräsentiert. Hierbei werden die drei Typen *Voxelmodell*, *Octree-Modell* und *Volumenstab-Modell* unterschieden. Eine kurze prägnante Beschreibung wird in der Arbeit von Fautz [Fau02] gegeben.

Oberflächenmodelle (engl. *boundary representations*; kurz *B-rep*) bieten im Gegensatz zu Punktmodellen zusätzlich zu den Koordinaten der Punkte noch Informationen über die Struktur der Oberfläche. So definiert z. B. ein Punktepaar eine Kante. Eine Fläche wird durch eine Liste von Kanten definiert, die koplanar, geschlossen und überkreuzungsfrei sind. Nachbarschaftsbeziehungen zwischen Flächen werden anhand ihrer gemeinsamen Kanten festgelegt. Wird hierüber eine geschlossene Oberfläche definiert, so können damit volumetrische 3D-Modelle repräsentiert werden. Genauer gesagt, gliedert sich die Beschreibung eines geschlossenen Körpers bzw. Festkörpers in zwei Teile: die *topologische* Beschreibung der Verbindung und Orientierung der Punkte, Kanten und Flächen und die *geometrische* Beschreibung, um die Oberfläche im dreidimensionalen Raum darzustellen. Das Thema der Oberflächenmodelle wird ausführlich in den Arbeiten von Baumgart [Bau72], Mäntylä [Män88] und Hoffmann [Hof89] beschrieben. Mäntylä stellt vor allem die verschiedenen topologischen Repräsentationen bis hin zur *winged-edge representation* ausführlich dar. Wie das Oberflächenmodell in dieser Arbeit eingesetzt wird und welche topologischen Informationen modelliert werden, ist Thema von Kapitel 7.

Eine weitere Technik, 3D-Objekte zu beschreiben, ist die CSG-Darstellung (*constructive solid geometry*). Aus graphischen Objekten – für die einfache mathematische Formeln¹ existieren – werden durch die booleschen Mengenoperationen *Vereinigung*, *Differenz* und *Schnitt* neue, kompliziertere Objekte erzeugt (siehe Hoffmann [Hof89]).

¹Hier sind implizite Funktionen der Form $f(x, y, z) = 0$ gemeint. Ein Zylinder entlang der y -Achse hat z. B. folgende Darstellung: $x^2 + z^2 - r^2 = 0, h_1 \leq y \leq h_2, r \neq 0$.

Kapitel 3

Verwandte Arbeiten

Das Thema dieser Arbeit fußt in verschiedenen Disziplinen der Informatik, auf deren Stand der Forschung in diesem Kapitel eingegangen wird. Es ist wie folgt untergliedert. Zu Beginn wird kurz auf die klassische Vorgehensweise zur Gestaltung von Gebäuden eingegangen. Darunter fallen die üblichen CAD-Programme, die in der Architektur Verwendung finden. Im Anschluss daran werden für die Arbeit wichtige Techniken der prozeduralen Texturerzeugung behandelt. Die Erzeugung von Dächern stellt einen eigenständigen Problembereich dar, dem in einem weiteren Abschnitt Rechnung getragen wird. Der darauf folgende Abschnitt widmet sich spezialisierten Techniken zur Beschreibung architektonischer Elemente. Anschließend werden Gesamtsysteme zur prozeduralen Modellierung von Gebäudefassaden und Gebäuden vorgestellt, die einige der zuvor beschriebenen Techniken nutzen. Hierbei wird zwischen wissenschaftlichen Systemen, deren Techniken bekannt sind, und kommerziellen Systemen unterschieden.

Abschließend werden alle relevante Techniken anhand eines vorgestellten Kriterienkataloges miteinander verglichen und auf ihre Eignung zur Lösung der Aufgabenstellung überprüft.

3.1 CAD-Programme

CAD-Programme dienen primär der Erstellung technischer Zeichnungen. Sie werden vor allem in Fachbereichen eingesetzt, die sich mit der Erstellung von Konstruktionsbeschreibungen befassen. In der Architektur unterscheidet man zwischen *geometrieorientierter* und *produktorientierter* CAD-Software. Zu den geometrieorientierten Systemen zählt vor allem das Produkt *AutoCAD* der Firma *Autodesk*. Die Stärke der Systeme liegt in der reinen Verwaltung der Geometrie. Damit sind sie zwar sehr flexibel und vielseitig einsetzbar, was aber dadurch erkauft wird, dass z. B. die Geometrie einer Wand, kein Objektwissen besitzen. Der Kreativität sind

praktisch keine Grenzen gesetzt, Änderungen hingegen sind extrem aufwendig. Die produktorientierten Systeme folgen dem Ansatz des Baukastenprinzips. Diese Systeme modellieren ein Gebäude auf Objektbasis, d. h. sie verwalten Bauteile und Baugruppen wie auch Räume und vor allem deren Beziehung untereinander. Eine Weiterentwicklung dieser Systeme für die Architektur verkörpern die *CAAD*-Systeme¹. Die Vielfalt der möglichen Gebäude wird zwar dadurch eingeschränkt, Änderungen können aber sehr einfach durchgeführt werden. Beide Typen sind primär daraufhin ausgelegt Materialien und Kosten für den Bau des Gebäudes abzuschätzen. Feine Details, wie das Relief eines Gesimses oder prozedurale Strukturen wie Ornamente oder Mauerwerke – im Sinne der Computergraphik – sind nicht vorgesehen. Ebenso sind diese Systeme nicht für prozedurales Modellieren ausgelegt. Im Abschnitt 3.6.2 wird für die CAD-Systeme exemplarisch das System *ArCon* herausgegriffen und dessen Funktionsweise untersucht.

3.2 Texturierung architektonischer Elemente

Texturen sind in der Computergraphik ein beliebtes Mittel, um den visuellen Eindruck von 3D-Objekten zu verbessern. Ebenso bieten sie den Vorteil, dass mittels in Hardware implementierter Techniken Geometrie eingespart werden kann. Das ist z. B. an Ornamenten, die in Steinplatten eingelassen sind, gut ersichtlich. Das Relief des Ornaments muss nicht geometrisch nachgebildet werden, sondern kann direkt als Textur auf die Geometrie aufgebracht werden. Soll das Relief zusätzlich noch korrekt beleuchtet werden, so wird zusätzlich zur Textur noch das Höhenprofil des Ornaments als eine Art Textur (engl. *bump map*) aufgebracht.

Als Texturen können Fotos von realen Objekten, z. B. Steinwände oder Asphalt, oder prozedural erzeugte Bilder verwendet werden. In diesem Abschnitt werden wir uns nur den Techniken der prozeduralen Texturierung widmen. Ebert et al. [EMP⁺03] bieten eine sehr umfangreiche Übersicht zu verschiedenen Techniken der prozeduralen Texturierung. Darunter fallen z. B. Fraktale, Perlin-Noise, atmosphärische Effekte, usw. Weiterführende Texturierungstechniken werden in den Arbeiten von Cohen et al. und Nicoll et al. [CSHD03, NMMK05] vorgestellt. Cohen et al. erstellen unendlich große Texturen aus einem endlichen Satz Texturen. Dabei besitzen die endlichen Texturkacheln Kanten, an die wiederum andere Kacheln nahtlos angeschlossen werden können. So werden visuelle Artefakte, wie sie bei der üblichen Kachelung von nur einer Texturkachel durch Wiederholung entstehen, vermieden.

Im Zusammenhang mit der Erzeugung von Gebäudefassaden fällt das Interesse hauptsächlich auf die Erstellung von Ornamenten und auf Methoden zur Beschreibung von Mauerwerken.

¹CAAD: Computer Aided Architectural Design

Ornamente sind in erster Linie Verzierungen oder Ausschmückungen, die auf vielerlei Gegenstände anwendbar sind. Eine Übersicht über die gestalterische Vielfalt wird in Meyers „Handbuch der Ornamentik“ [Mey27] gegeben. In unserem Fall wollen wir uns auf die Verzierung von Wänden und Fensterrahmen mit pflanzenähnlichen Motiven beschränken. Der von Wong et al. [WZS98] entwickelte Ansatz bietet eine hervorragende Basis zur computergestützten Erzeugung pflanzenähnlicher Ornamente (siehe Abbildung 3.1). In Abschnitt 5.4.2 wird auf den Ansatz von Wong et al. genauer eingegangen und erläutert, wie er in dieser Arbeit eingesetzt wird.



Abbildung 3.1: Ein pflanzenähnliches Ornament aus der Arbeit von Wong et al. [WZS98].

Unter einem Mauerwerk versteht man eine Wand oder ein Teil einer Fassade, das aus natürlichen oder künstlichen Steinen gefertigt ist. Verschiedene Typen von Mauerwerk aus natürlichem Stein sind in Neuferts „Bauentwurfslehre“ [NN96] zusammengefasst. In dieser Arbeit werden Zyklopenmauerwerke, Quadersteinmauerwerke und Mauerziegelverbände behandelt. Mauerziegelverbände stellen eine interessante Form von Mauerwerk dar. Dort werden aus gleich großen oder verschieden großen Ziegeln sich wiederholende Muster generiert. Eine Übersicht der Vielfalt von Mauerziegelverbänden ist ebenso in Neuferts „Bauentwurfslehre“ zu finden.

Miyata [Miy90] stellt in seiner Arbeit eine Methode zur Generierung von Steinmauermustern vor. Über einen sechsstufigen Prozess werden die Texturen für eine Steinmauer erzeugt. Die Ausmaße der Steine berechnen sich über eine Standardgröße zuzüglich einer zufälligen Abweichung in Höhe und Breite. Eine Wand, bestehend aus Steinen, wird über einen Graphen beschrieben. Dabei repräsentieren die Stoßfugen der Steine die Knoten des Graphen. Ein Stein wird schließlich durch die Kanten, die seine Eckpunkte verknüpfen, beschrieben. Aneinander grenzende Steine teilen sich gemeinsame Eckpunkte und definieren darüber die gesamte Wand. Um eine unregelmäßige Struktur der Steine zu erreichen, werden zum einen die

Eckpunkte der Steine leicht verschoben und zum anderen werden die Verbindungen der Eckpunkte (also die Begrenzungskanten der Steine) über Polygonzüge realisiert. Den letzten Schliff, eine unregelmäßige Oberfläche, erhalten die Steine über ein Höhenprofil. Ein Beispiel zu Miyatas Verfahren ist in [Abbildung 3.2](#) gezeigt. Mit Miyatas Methode werden nur rechteckige Polygone texturiert. Das Verfahren lässt sich aber leicht auf beliebige Polygone erweitern. Das in [Abschnitt 5.3](#) vorgestellte Verfahren zur Texturierung verwendet eine andere Methode, um Steine zu platzieren, die an die folgende Technik angelehnt ist.



Abbildung 3.2: Eine Steinwand (aus [\[Miy90\]](#)).

Legakis et al. [\[LDG01\]](#) beschreiben ein Verfahren, mit dem Quaderstein- und Ziegelmauerwerke für einfachere 3D-Objekte erzeugt werden können. Hierbei wird zwischen Ecken, Kanten und Flächen unterschieden. Steine an Ecken beeinflussen die Steine ihrer anliegenden Kanten und Steine an Kanten wirken sich auf die Steine ihrer anliegenden Flächen aus. Damit erreichen sie, dass die Mauerformation auch über Ecken und Kanten hinweg korrekt angebracht wird, wie in [Abbildung 3.3](#) zu sehen ist. Verschiedene Strukturen innerhalb einer Fassade, z. B. Erker, Mauervorsprünge usw., werden nicht berücksichtigt. Weiterhin ist die Form der Steine auf Rechtecke beschränkt und es gibt keine Möglichkeit, Muster in Mauerziegelverbänden automatisch erstellen zu lassen. Die generierten Steine und Mauerwerke werden außerdem direkt als Geometrie erzeugt. Schon bei einem kleineren Wohnhaus würde hier die Anzahl der Dreiecke drastisch in die Höhe schnellen. Die in [Abschnitt 5.3](#) beschriebenen Techniken, basierend auf diesem Verfahren, produzieren für einfache Polygone die exakt passende Textur eines Mauerwerks. Unterstützt werden Zyklopenmauerwerke, Quadersteinmauerwerke und Mauerziegelverbände anhand einfacher Regeln.

In ihrer Arbeit beschreiben Nicoll et al. [\[NMMK05\]](#) eine Technik zur automatischen Erzeugung fast regulärer Texturen. Diese findet man z. B. in Textilien und



Abbildung 3.3: Ein Torbogen mit verschiedenen Steintypen (aus [LDG01]).

Mauerwerken. Das zugrunde liegende Muster, z. B. die Form des Ziegelverbands, ist gleichbleibend, aber die Ziegel variieren leicht. Mit der vorgeschlagenen Technik ist es möglich, aus Fotos der Materialien das zugrunde liegende Muster zu extrahieren. Aus dem Muster wird schließlich eine *fractional fourier texture mask* berechnet, die für die Erzeugung der unregelmäßigen Details im Muster zuständig ist. Als Beispiele werden neben Textilien auch Ziegelmauerwerke (siehe [Abbildung 3.4](#)) gezeigt. Damit wäre das Verfahren geeignet, um z. B. die Textur einzelner Ziegelsteine zu generieren. Diese Technik ist aber nur bedingt für geometrieabhängige Texturierung geeignet, da das Muster an den Löchern und Begrenzungen der Geometrie aufgebrochen und an die Geometrie angepasst werden müsste.

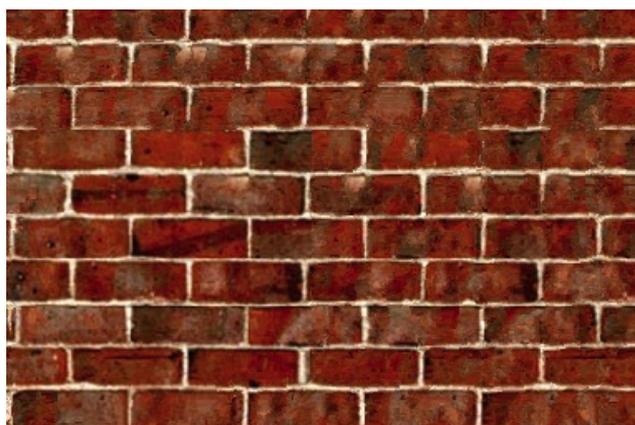


Abbildung 3.4: Ein Mauerwerk mit natürlich wirkenden Ziegelsteinen (aus [NMMK05]).

3.3 Erzeugung von Dachstrukturen

Ein weiteres Problem, das bei der automatischen Erzeugung von Gebäuden eine wichtige Rolle spielt, ist die Generierung von Dächern. Deren automatische Erzeugung ist ein bekanntes Thema in der Computergraphik. In diesem Abschnitt werden verschiedene Techniken zur automatischen Erzeugung von Dächern vorgestellt und abschließend kurz miteinander verglichen.

Felkel et al. [FO98] stellen einen robusten Algorithmus zur automatischen Erzeugung von Dachgeometrien vor. Ausgangspunkt ist hier ein beliebiges Umrisspolygon. Die vorgestellte Methode lässt das Umrisspolygon schrumpfen, bis schließlich nur noch ein Skelett übrig bleibt, das dann als Dachfirst dient. Aus First und Umrisspolygon wird schließlich das Dach generiert. Verschiedene Dachtypen und die automatische Anpassung aneinander grenzender Dachtypen sind nicht vorgesehen. Bei Felkel et al. werden zugrunde liegende Strukturen des Umrisspolygons nicht berücksichtigt. Es ist z. B. nicht möglich, zwischen Hauptgebäude und Anbauten zu unterscheiden.

Ein Verfeinerung stellt die Arbeit von Laycock et al. [LD03] dar. Hier wird das Umrisspolygon in rechteckige Blöcke unterteilt, die sich auch überschneiden dürfen. Jeder dieser Blöcke hat seine eigene Dachform und sein First verläuft entlang einer der beiden Hauptachsen. Wird nun das gesamte Dach generiert, so wird der Verlauf der Firste aneinander grenzender Dächer untersucht. Verlaufen sie parallel zueinander, so werden die beiden sich überschneidenden Dachhälften geeignet vereinigt. Verläuft dagegen ein First senkrecht zum anderen First, so wird der First vom kleineren Dach verlängert, so dass er sich exakt an das größere Dach anschmiegt. Einen sehr ähnlichen Ansatz verfolgen Müller und Wonka in ihrer Arbeit [MWH⁺06].

Beide Techniken lassen die Möglichkeit außer Acht, dass sich die Dächer eines Gebäudes auch auf verschiedenen hohen oder verschiedenen Stockwerkenden befinden können. Felkels Technik ist sehr gut geeignet, um Dachgeometrien für Umrisspolygone zu erstellen, für die keine weiteren Strukturinformation bekannt sind. Ein Anwendungsgebiet ist z. B. die Arbeit zur Hochwassersimulation von Bender et al. [BFO04]. In einer 3D-Darstellung des exponierten Gebiets kann der Benutzer interaktiv durch die Szene navigieren. Um den Realitätsgrad zu erhöhen, werden aus den Grundrissinformationen² des Landesvermessungsamtes automatisch Häuser mit Giebeldächern erzeugt. Laycocks Arbeit ist dann gut geeignet, wenn es sich bei den Umrisspolygonen um Polygone mit nur rechten Winkeln handelt.

Da aber in der hier vorliegenden Arbeit ein neuer Ansatz zur Erzeugung von Umrisspolygonen anhand der Strukturinformationen eines Gebäudes vorgestellt wird, sind die beiden Techniken nur bedingt geeignet. Zwar könnten mit der Tech-

²Es handelt sich genau genommen nur um den Umriss des Grundrisses.

nik von Felkel auch hier die Dächer erzeugt werden. Dies wäre aber nicht sinnvoll, da so die gegebene Strukturinformation nicht berücksichtigt würde. Die Techniken von Laycock bzw. Müller können schon deshalb nicht eingesetzt werden, da sie auf Polygone mit rechten Winkeln angewiesen sind. Außerdem würde auch hier die Strukturinformation ungenutzt bleiben.

3.4 Geometrisches Modellieren architektonischer Elemente

Die schnelle Generierung detaillierter Fassadenmodelle ist Ziel dieser Arbeit. Die Fassade lässt sich in viele verschiedene architektonische Elemente unterteilen. In diesem Abschnitt werden Arbeiten zur Erzeugung von 3D-Modellen dieser architektonischen Elemente vorgestellt.

Die Arbeit von Havemann et al. [HF01] beschäftigt sich mit der Repräsentation derartiger 3D-Modelle.³ Hier wird vor allem auf die interaktive Darstellung gesamtter Städte Wert gelegt. Komplexe Strukturen sollen dynamisch detailliert bzw. grob dargestellt werden können. Havemann et al. erreichen ihr Ziel über Unterteilungsflächen mit gewichteten Kanten. Es wird z. B. nur die grobe Form eines Gesimses modelliert. Anschließend werden Rundungen approximativ über Unterteilungsflächen in Abhängigkeit zur Entfernung des Betrachters erzeugt. Die Modelle werden individuell angefertigt und sind damit nicht parametrisiert. Eine einfache Anpas-

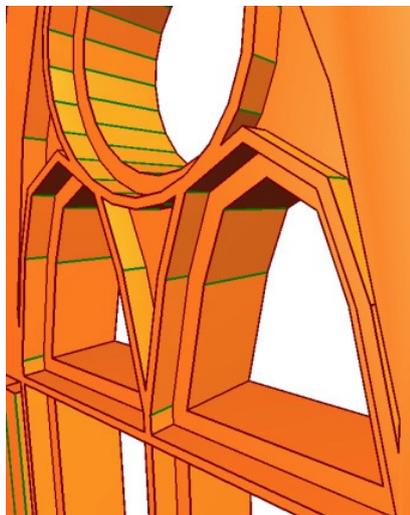


Abbildung 3.5: Beispiel für ein komplexeres Fenster (aus [HF01]).

³Die von Havemann et al. vorgestellte Technik ist in verallgemeinerter Form fester Bestandteil von Maya [Aut08].

sung ist zu einem späteren Zeitpunkt nicht mehr möglich. In Abbildung 3.5 ist ein Beispiel aus Havemanns Arbeit zu sehen.

Birch et al. [BBJ⁺01] stellen Techniken zur schnellen Modellierung von Fenstern, Erkern, Torbögen und Säulen vor. Die verschiedenen Techniken sind in einem Modellierungswerkzeug vereint und sollen den Benutzer in der Modellierung unterstützen. Fenster werden über ein Rechteck beschrieben, das nun horizontal oder vertikal in kleinere Fenster unterteilt werden kann (siehe Abbildung 3.6). Jedem (Teil-)Fenster wird eine eigene feste Geometrie für den Rahmen zugeordnet. Der Verlauf des Unterteilungsprozesses wird in einem Modellierungsprotokoll (*construction history*) festgehalten, so dass zu einem späteren Zeitpunkt Modifikationen vorgenommen werden können. Für Erker stehen verschiedene Grundmodelle zur Auswahl. Die Rahmen der Fenster des Erkers und die Fenster selber werden texturiert, um einen gefälligeren Eindruck zu erzielen. Die Form der Torbögen, ihr Profil und das Profil der Säulen werden mit Hilfe von Splines modelliert.



Abbildung 3.6: Ein einfach unterteiltes Fenster (aus [BBJ⁺01]).

3.5 Erzeugung und Beschreibung von Grundrissen

Die Gebäudegrundrisse spielen eine entscheidende Rolle in dieser Arbeit. Über eine geschickte Beschreibung der Grundrisse können bereits in einem sehr frühen Stadium Strukturen des Gebäudes modelliert werden. Diese Informationen wirken sich später auf fast alle architektonischen Elemente aus, wie z. B. die Form der Fenster, der Gesimse und sogar der Dachformen. Aus diesem Grund werden in diesem Abschnitt die wichtigsten Techniken zur Erzeugung von Grundrissen behandelt.

Wie bereits in Abschnitt 3.3 beschrieben, unterteilen Laycock et al. [LD03] Gebäudegrundrisse in sich überlappende Rechtecke. Sie erreichen damit, dass mar-

kante Gebäudeabschnitte, wie z. B. Haupttrakt, Nebentrakt und Anbauten bereits auf der Ebene der Grundrissbeschreibung modelliert und somit auch unterschieden werden. Anschließend nutzen sie diese Informationen zur Erzeugung des Daches, das die Gebäudestrukturen widerspiegelt. Dadurch, dass sich aber die Polygone des Grundrisses überschneiden, ist keine eindeutige Zuordnung der Schnittfläche möglich. Wie weitere Stockwerke behandelt werden, ist nicht angegeben.

Die beiden im Folgenden vorgestellten Verfahren sind jeweils Teil eines Gesamtsystems. Sie werden in Abschnitt 3.6.1 vollständig beschrieben.

Parish et al. [PM01] erzeugen konvexe Gebäudegrundrisse aus größeren sog. Blöcken, die durch das L-System, das die Straßenzüge erzeugt, vorgegeben werden. Die Grundrisse beinhalten keinerlei Strukturinformationen über das Gebäude.

In der Arbeit von Müller et al. [MWH⁺06] werden die Gebäude durch Kombination einzelner geometrischer Körper, wie z. B. Quader und Zylinder, erzeugt, die sich auch überlappen dürfen. Damit werden aber zum einen nur sehr grobe Strukturen des Gebäudes beschrieben und zum anderen sind damit auch alle Stockwerke für eine Gebäudestruktur festgelegt, d. h. Stockwerke in einer Struktur werden nicht unterschieden. Jeder dieser geometrischen Körper kann zusätzlich mit einer Dachform versehen werden.

Harada et al. [HWB95] entwickeln einen recht interessanten allgemeinen Ansatz zur Lösung von 2D-Layout-Problemen. Anhand eines Beispiels zeigen sie, wie damit auch Grundrisspläne erzeugt werden können. Bei den Objekten, die angeordnet werden, handelt es sich um achsenparallele Rechtecke. Zwischen Objekten können Beziehungen und Bedingungen modelliert werden. Eine Bedingung könnte z. B. lauten: die Küche soll stets neben dem Esszimmer angeordnet werden. Der Benutzer kann Objekte hinzufügen, löschen oder interaktiv ihr Layout, solange alle Bedingungen erfüllt sind. Die von der Umstrukturierung des Layouts betroffenen Objekte werden vom System, basierend auf einem physikalischen Ansatz, automatisch angeordnet. Wird aber darüber hinaus vom Benutzer eine Änderung erzwungen, die eine Verletzung der Bedingungen zur Folge hätte, so versucht das System, die Objekte völlig neu zu platzieren, um wieder allen Bedingungen zu genügen. Die Objekte bewegen sich dann nicht mehr kontinuierlich, sondern „springen“ an ihren neuen Platz.

Garanin et al. [GG71] behandeln in ihrer Arbeit das Problem der Raumplanung für Industriekomplexe. Die Herangehensweise zur Lösung des Problems formulieren sie wie folgt: Zuerst definieren sie das Ergebnis der Raumplanung als Variante ω eines allgemeinen Layouts des Industriekomplexes, das einer Liste von Kriterien (wie z. B. vernünftiges Layout von Leitungen und Straßen; Gruppierung von Gebäuden ähnlicher Funktionalität usw.) genügen muss. Als Nächstes formulieren sie die notwendigen Informationen bzw. Bedingungen α , die ein Architekt üblicherweise zur Bewältigung der Aufgabe benötigt. Darunter fallen u. A. die geographischen Gegebenheiten und die verschiedenen anzusiedelnden Industriezweige. Gesucht ist

eine Methode A , die die Überführung von α nach ω bewerkstelligt: $\alpha \xrightarrow{A} \omega$. Dazu bestimmen sie eine Baumstruktur, die die verallgemeinerten Formen gängiger Aufteilungen von Industriekomplexen beinhaltet. Eine Variante ω ist eine sequenzielle Aufteilung des Territoriums unter Berücksichtigung der Baumstruktur und der Bedingungen α , die schließlich eine Lösung für die Raumplanung darstellt.

In den beiden Arbeiten von Greuter et al. [GPSL03a, GPSL03b] werden Grundrisse durch Verschmelzen zufällig ausgewählter Polygone erzeugt. Dabei kann aber von Stockwerk zu Stockwerk nur höchstens ein Polygon zum aktuellen Grundriss hinzugefügt werden. Diese Vorgehensweise erlaubt ihnen, je Stockwerk einen unterschiedlichen Grundriss zu erzeugen. Als Dachtyp ist nur das Flachdach vorgesehen.

Ein Hauptproblem der beschriebenen Arbeiten, ausgenommen die von Müller [MWH+06] und Greuter [GPSL03a, GPSL03b], ist die Limitierung auf achsenparallele Rechtecke zur Grundrissbeschreibung. Ein Nachteil der Arbeit von Müller ist, dass Überschneidungstests durchgeführt werden müssen, um räumliche Information über aneinander grenzende Strukturen zu erhalten. Wenn auch eingeschränkt, können Greuter et al. pro Stockwerk verschiedene Grundrisse erzeugen, aber es werden keine Strukturinformationen modelliert. Um aneinander grenzende Strukturen zu bestimmen, müssten auch hier Überschneidungstests durchgeführt werden.

3.6 Systeme für die prozedurale Modellierung von Gebäuden und Städten

Die vorgestellten Systeme werden in wissenschaftliche und kommerzielle unterschieden. Die Unterscheidung wird getroffen, da die Methoden der wissenschaftlichen Systeme, im Gegensatz zu den kommerziellen, zugänglich sind.

3.6.1 Wissenschaftliche Systeme

Hier werden die wichtigsten wissenschaftlichen Systeme zur prozeduralen Generierung von Gebäuden und Städten aufgeführt. Sie bieten hinsichtlich der verwendeten Techniken einen guten Einblick. Damit wird auch sofort ihre Mächtigkeit ersichtlich.

3.6.1.1 Procedural Modeling of Cities [PM01]

Parish et al. erzeugen hier ganze Städte auf Basis von L-Systemen. Die Erzeugung der Städte verläuft in einem vierstufigen Prozess.

In Bildern, sog. Belegungskarten, werden durch farbige Areale verschiedene

Regionen gekennzeichnet. Dabei wird zwischen Wasser und Land unterschieden. Die Landareale werden weiter unterschieden in bebaute und nicht bebaute Flächen. Bebaute Flächen sind in soziologischer Hinsicht aufgeteilt. Dabei werden regional unterschiedliche Bevölkerungsdichten miteinbezogen. Ebenso wird in die Belegungskarten die regional durchschnittliche Gebäudehöhe kodiert. Weiterhin werden auch geographische Aspekte berücksichtigt. Damit kann eine Vielzahl von Parametern bereits in Belegungskarten graphisch kodiert werden, was wiederum eine große Variation der Städte erlaubt.

Basierend auf diesen Daten werden mit einem erweiterten L-System die Straßenzüge erzeugt. Zum Setzen und Modifizieren der Parameter des L-Systems werden zwei Funktionen angeboten. Mit der einen werden die Parameter zum Erreichen der globalen Ziele (Verbinden zweier Regionen hoher Bevölkerungsdichte oder Fortsetzen der Straßen usw.) gesetzt. Über die andere werden die Parameter hinsichtlich lokaler, einschränkender Gegebenheiten (z. B. Straße führt zu einem Wasserareal oder in einen Park) überprüft und angepasst. Sollte eine Anpassung nicht möglich sein, so wird das L-System-Modul entfernt. Für die Straßenerzeugung werden vier verschiedene Straßennuster unterstützt.

- *Rechtwinkliges Raster* – wie es in Teilen New Yorks üblich ist.
- *Radial bzw. konzentrisch* – wie in Paris oder Karlsruhe.
- *Verzweigend* – hier wird keine konkretes Muster verfolgt. Die Straßenzüge verlaufen zufällig.
- *Steigungs- bzw. gefälleorientiert* – die Straßen werden geographischen Gegebenheiten angepasst, wie z. B. in San Francisco.

Die für die Gebäudeerzeugung relevanten Gebiete werden durch die Straßenzüge in Blöcke unterteilt. Anschließend werden die Blöcke bis zum Erreichen einer benutzerdefinierten Mindestgröße weiter unterteilt. Dabei werden ausschließlich konvexe Flächen erzeugt, konkave sind nicht erlaubt. Zu kleine Flächen oder Flächen, die nicht an eine Straße grenzen, werden entfernt. Die maximale Gebäudehöhe wird der Belegungskarte entnommen.

Über ein weiteres L-System werden die Gebäudegeometrien erzeugt. Sie werden als Zeichenketten binärer Operationen repräsentiert und arbeiten auf einfachen geometrischen Formen. Jede der erzeugten Flächen repräsentiert genau ein Gebäude. Insgesamt können die drei Gebäudetypen Hochhaus, Geschäftsgebäude und Wohnhaus generiert werden. Die Gebäudetypen werden wiederum anhand der Belegungskarten festgelegt. Das eingesetzte L-System verfügt über folgende Regeln zur Geometrieerzeugung: Transformation, Extrusion, Verzweigung und Terminierung. Objekte, wie Dächer und Antennen, liegen als vorgefertigte Geometrie vor. Die Fassaden der Gebäude werden mittels prozeduraler Texturen erstellt. Dabei

werden die verschiedenen Merkmale einer Fassade auf jeweils eine Schicht projiziert. Eine Schicht repräsentiert z. B. die horizontale und vertikale Verteilung der Fenster. Eine weitere Schicht enthält spezialisierte Informationen zu dedizierten Spalten und Zeilen einer Fassade. Hierüber werden spezielle Ausprägungen für Erdgeschosse oder Mittelkonsolen definiert. Die endgültige Fassade entsteht durch Überlagerung dieser Schichten. Das Beispiel einer Stadt ist in [Abbildung 3.7](#) zu sehen.



Abbildung 3.7: Eine ganze Stadt, deren Erzeugung auf L-Systemen basiert (aus [\[PM01\]](#)).

3.6.1.2 Instant Architecture [\[WWSR03\]](#)

Wonka et al. präsentieren in eine sog. *split grammar* zur Erzeugung von Gebäudefassaden. Ihre Arbeit beinhaltet die Erstellung einer umfangreichen Datenbank an Grammatikregeln, wobei aber aus einem Satz von Regeln eine ganze Reihe verschiedener Designs entstehen, statt für jedes Modell spezialisierte Grammatiken zu erzeugen. Aufgrund der Komplexität der Grammatik stehen in jedem Ableitungsschritt eine Vielzahl von Regeln zur Auswahl. Die Entscheidung, welche Regel verwendet wird, bestimmt sich durch ein zu Beginn gewähltes Designziel. Um ungewollte Designs zu vermeiden, werden wichtige Desginmerkmale herausgearbeitet. Diese werden anschließend über einen separaten Mechanismus räumlich passend angeordnet.

Für ihre *split grammar* gehen sie von einer einfachen geometrischen Form aus, basierend auf der Arbeit von Stiny [\[SG71\]](#). Diese wird dann durch die Grammatik weiter in kleinere Formen zerlegt. Beim Prozess der Unterteilung werden Eigenschaften (z. B. Materialeigenschaften) den neu entstandenen Formen zugewiesen. Weiterhin können diese Eigenschaften nachfolgenden Formen vererbt werden. Eine weitere Grammatik, die *control grammar*, entscheidet wie und welche Eigenschaften auf die neuen Formen übertragen und welche neuen Formen erzeugt werden.

Architektonische Prinzipien werden über diese Grammatik eingehalten bzw. durchgesetzt. Über den Mechanismus *rule selection* werden aus einer Menge möglicher Regeln nur diese herausgegriffen, die entweder ein plausibles Resultat liefern oder eine ausreichend große Vielfalt in der Gebäudestruktur erzeugen oder die vom Benutzer vorgegebenen Designziele erfüllen. Mit dieser Vorgehensweise lassen sich viele verschiedene Gebäude erzeugen. Ebenso werden auch hier detaillierte Strukturen über vorgefertigte geometrische Modelle und Texturen bereitgestellt. Eine Parametrisierung ist nicht vorgesehen. In Abbildung 3.8 ist ein Gebäude aus Wonkas Arbeit dargestellt.

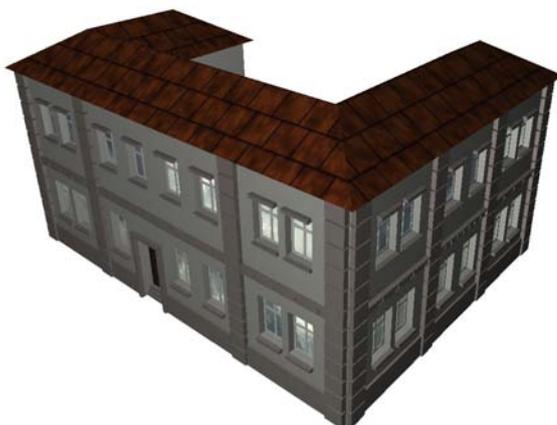


Abbildung 3.8: Gebäude, erstellt mit der *split grammar* (aus [WWSR03]).

3.6.1.3 Procedural Modeling of Buildings [MWH⁺06]

Diese Arbeit von Müller et al. basiert auf den beiden eben vorgestellten Arbeiten von Parish [PM01] und Wonka [WWSR03].

Für die prozedurale Generierung von Gebäudefassaden wird hier eine kontext-sensitive *shape grammar*, eine Erweiterung der *split grammar*, vorgestellt. Die Motivation, eine neue Grammatik zu erstellen, ist dadurch begründet, dass die *split grammar* überlappende architektonische Strukturen nicht entdecken und auflösen kann.

Mit dieser *shape grammar* für Computer-Graphik-Architektur (kurz: *CGA shape*) wird dieses Problem behoben. Sie basiert auf geometrischen Formen, ähnlich wie es bereits von Stiny [SG71] praktiziert wird. Jede Form besitzt, wie bei Wonka et. al [WWSR03], zusätzliche Attribute: die Position im Raum, ein lokales Koordinatensystem und Vektoren für Größe (orientierte Hüllkörper, bezeichnet als *scope*). Wie bei Chomsky-Grammatiken gibt es auch hier Alphabete über Terminal- (Σ) und Non-Terminal-Symbole (V). Die Produktionsregeln sind allgemein wie folgt

definiert: $id : predecessor : cond \rightarrow successor : prob$. Mit id wird eine Regel eindeutig bezeichnet. $predecessor \in V$ bezeichnet eine Form, die zu ersetzen ist. Die Ersetzung findet aber nur statt, wenn die in $cond$ geforderte Bedingung erfüllt ist. Ersetzt wird durch $successor$ mit der Wahrscheinlichkeit $prob$.

Aus der allgemeinen Regeldefinition wurden folgende spezialisierte Klassen von Regeln entworfen: Über sog. *scopes rules* werden Regeln zur Transformation für Formen definiert, wie sie auch üblicherweise in L-Systemen benutzt werden. Diese beinhalten Translation, Rotation und Skalierung. Mit den *basic split rules* wird die aktuelle Form mit ihren Attributen, *scope*, entlang einer Achse oder mehrerer Achsen unterteilt. Die Anzahl der Unterteilungen und die Stärke jeder Unterteilung kann entweder absolut oder relativ angegeben werden. Des Weiteren kann auch wiederholt unterteilt werden, so dass ein Element bestimmter Breite so oft wie möglich in ein vorhandenes Element eingefügt wird. Ebenso können Flächen und Kanten aus 3D-Körpern entnommen und bearbeitet werden. Die dabei entstandenen Flächen können z. B. wieder zu 3D-Körpern extrudiert werden.

Die grobe äußere Hülle eines Gebäudes wird durch Kombination einzelner einfacher Formen, wie z. B. Quader und Zylinder, erstellt. Dabei werden die einzelnen Formen durch die Grammatik transformiert. Für die Erzeugung von Dächern wird ein Satz von Standarddächern mitgeliefert. Um Überlappungen einzelner Formen zu entdecken, werden Verdeckungstests mittels Octrees durchgeführt. Die Überlappungstests können durch Ausnutzen des Ableitungsbaums beschleunigt werden. Damit werden z. B. Formen aus früheren Ableitungsschritten nicht in die Überprüfung miteinbezogen. Dadurch, dass sich Formen überlappen können, müssen ggf. Unterteilungen, die z. B. Fenster setzen, neu angeordnet werden, um ungewollte Unterteilungen zu vermeiden. Es kann z. B. eine einheitliche Unterteilung so vorgenommen werden, dass nur die am Schnitt der Überlagerung angrenzenden Elemente angepasst werden, nicht aber das gesamte sichtbare Element. Mit dieser Technik werden auch räumliche Anordnungen von Fassaden berücksichtigt. Dabei sind aber räumliche Informationen nicht direkt aus dem verwendeten Modell ableitbar, sondern müssen erst über Verdeckungstests bestimmt werden. Für architektonische Elemente, wie z. B. Gesimse und Fensterstöcke, werden vorgefertigte Modelle verwendet, entweder als Geometrie oder Textur. Eine Parametrisierung dieser Elemente, wie z. B. die Profile der Gesimse und Fensterformen, ist nicht vorgesehen. Somit können derart feine Strukturen am fertigen Modell nur noch sehr schwer verändert werden. Über die Form der Texturierung (z. B. Mauerwerke für Wände) der Gebäude wird keine Aussage gemacht. Diese Technik lässt sich vielseitig einsetzen. So wurde damit Pompeji rekonstruiert (siehe Abbildung 3.9).



Abbildung 3.9: Pompeji, erstellt mit L-Systemen und der *shape grammar* für Computer-Graphik-Architektur (aus [MWH⁺06]).

3.6.1.4 Build-by-Number: Rearranging the Real World to Visualize Novel Architectural Spaces [BA05]

Bekins et al. präsentieren eine halbautomatische Technik zur Erstellung neuer virtueller Gebäude anhand von Aufnahmen realer Gebäude. Der Stil der neuen Gebäude ist bei diesem Verfahren direkt vom Stil des aufgenommenen Gebäudes abhängig. Die Vorgehensweise zur Erzeugung der Gebäude erinnert stark an das Prinzip „Malen nach Zahlen“ (engl. *paint-by-number*).

Die Technik lässt sich grob in drei verschiedenen Bereiche aufteilen. Zunächst wird vom Benutzer das fotografierte Gebäude aufbereitet, um anschließend automatisch zugrunde liegende Regeln extrahieren zu können. Danach können mit den gefundenen Regeln neue Gebäude gleichen Stils modelliert werden. Die drei Bereiche werden im Folgenden genauer betrachtet.

Zu Beginn erstellt der Benutzer von Hand ein grobes 3D-Modell zum Foto eines Gebäudes seiner Wahl. In diesem Modell werden die groben Strukturen des Gebäudes, wie Stockwerke, Fenster, Türen usw. mit farbigen Rechtecken gekennzeichnet. Dabei werden gleiche Strukturen gruppiert und verdeckte Flächen markiert. Daraus wird die nächst feinere Strukturierung des Modells wie folgt vorgenommen. Die oben genannten groben Strukturen des Gebäudes werden jetzt durch dreidimensionale Blöcke aufgebaut. In einem Szenengraph werden die räumlichen Beziehungen zwischen den Blöcken modelliert, wobei die Knoten die Blöcke und die Kanten deren räumliche Beziehung darstellen. Zu den Blöcken werden weitere Eigenschaften, wie Größe, Seitenverhältnis usw. gespeichert. Um noch feinere Strukturen modellieren zu können, werden zwei Operationen eingeführt. Über die eine wird

die relative Position zweier Blöcke zueinander beschrieben, d. h. eine Fläche des einen Blocks ist koplanar zur einer Fläche des anderen Blocks. Mit der anderen werden entweder Blöcke oder Flächen unterteilt.

Die korrekte Unterteilung des Gebäudes obliegt dem Benutzer. Sie ist die Grundlage für die automatische Erkennung von Strukturmustern im aufgenommenen Gebäude. Wiederkehrende Muster, wie z. B. Anordnungen verschiedener Fenstertypen innerhalb einer Wand, werden extrahiert. Daraus werden schließlich Regeln generiert.

Anhand der gefundenen Regeln können neue Gebäude im Stil des Originalgebäudes erzeugt werden. Hierzu werden folgende drei Schemata angewendet. Beim *face schema* wird eine neue Fläche beliebiger Größe anhand der Regeln mit verschiedenen Texturen, die aus dem Originalbild gewonnen wurden, texturiert. Das *floor schema* ist ein ungerichteter zyklischer Graph der den Grundriss bzw. den Umriss eines jeden Stockwerks beschreibt. Die Knoten des Graphen beinhalten die zuvor erstellten Flächen. Über die Kanten sind die Nachbarschaftsbeziehungen zwischen den Flächen repräsentiert. Zusätzlich werden in den Kanten die Winkel, die die Flächen zueinander einnehmen, gespeichert. Dadurch kann beim Anwenden der Schemata Kontextinformation genutzt und ausgewertet werden. Im *model schema* wird schließlich über einen Graphen das gesamte Gebäude abgebildet. Seine Knoten entsprechen den Stockwerken und die Kanten verknüpfen die Stockwerke miteinander. Bei den Stockwerken wird zwischen den Typen Erdgeschoss, Dachgeschoss und Zwischengeschoss unterschieden. Erd- und Dachgeschosse kommen in der Grammatik nur am linken und rechten Ende einer Regel vor. Zwischengeschosse hingegen können beliebig oft wiederholt und kombiniert werden. Schließlich kann aber ein Dachgeschoss noch Kanten zu weiteren Dachaufbauten, wie z. B. Türme, besitzen. Um eine visuell ansprechende Texturierung zu erhalten, werden die Schattierungen aus den Originaltexturen entfernt. Damit werden zu starke Kontraste beim Aneinanderfügen der Texturen vermieden und die Gebäudefassade erhält ein angenehmes homogenes Aussehen. Ebenso kann damit auch das Originalgebäude modelliert und die nicht sichtbaren Flächen nachträglich texturiert werden. Ein Beispiel ihrer Arbeit ist in [Abbildung 3.10](#) zu sehen.

Belkin et al. [[BA05](#)] haben erkannt, dass die räumliche Information über aneinander grenzende Strukturen relevant ist. Sie nutzen diese, um die Wände in einem Stockwerk abhängig von ihren Kontexten zu texturieren. Feine Strukturen wie Gesimsprofile, beliebige Fenster- und Dachformen, wie auch beliebige Grundrisse auf verschiedenen Stockwerken werden aber nicht unterstützt.

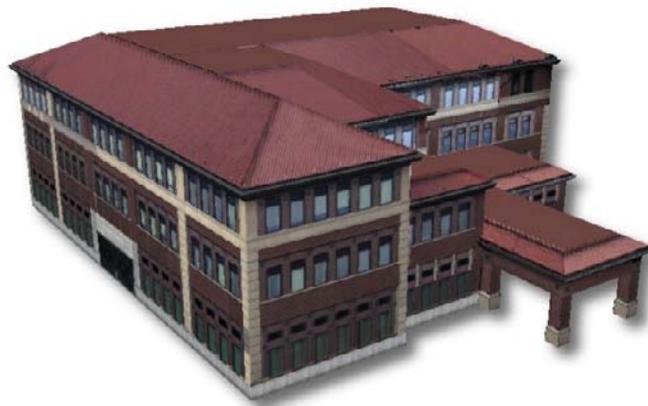


Abbildung 3.10: Aus Teilen eines realen Gebäudes zusammengesetztes Gebäude (aus [BA05]).

3.6.1.5 Real-time Procedural Generation of 'Pseudo Infinite' Cities [GPSL03a, GPSL03b]

In den beiden Arbeiten von Greuter et al. liegt das Augenmerk auf der prozeduralen Erzeugung ganzer Städte in Echtzeit. Abhängig vom Blickpunkt des Benutzers werden nur die im Sichtfeld liegenden Gebäude dargestellt. Der Benutzer kann sich völlig frei in der Stadt bewegen und es werden zur Laufzeit ständig neue Gebäude generiert. Gebäudedaten, wie Breite, Höhe, Anzahl der Stockwerke, Gebäudetyp usw., werden über einen Pseudo-Zufallszahlengenerator festgelegt. Dabei wird jedes Gebäude eindeutig über seine Position in einer Stadtkarte beschrieben. Ein Hash über die 2D-Koordinate, also die Position des Gebäudes, dient der Initialisierung der Zufallszahlen. Damit ist sichergestellt, dass die Gebäude völlig zufällig, aber fest, beim Durchschreiten der Stadt erstellt werden. Fallen Gebäude aus dem Sichtfeld des Benutzers, so kann die entsprechende Datenstruktur gelöscht werden. Über die Gebäude-Koordinate und dem daraus resultierenden Hash-Wert kann zu einem späteren Zeitpunkt das Gebäude exakt rekonstruiert werden. Folglich ist der Speicherbedarf der darzustellenden Gebäude relativ konstant.

Stockwerke werden über zweidimensionale Polygone beschrieben. Jedes einzelne Stockwerk wird über zufällig ausgewählte Polygone und Rechtecke, die miteinander verschmolzen werden, erzeugt. Dabei werden die Gebäude, Stockwerk für Stockwerk, in einem iterativen Verfahren von oben nach unten generiert. Im ersten Schritt wird ein zufälliges Polygon erzeugt und extrudiert. Nachfolgende Schritte nehmen jeweils ein neues zufälliges Polygon zum bereits vorhandenen hinzu und verschmelzen es zu einem neuen Polygon. Anschließend wird das Polygon extrudiert und es entsteht ein neues Stockwerk. Dachstockwerke und Erdgeschosse entsprechen der Form ihrer jeweils direkt angrenzenden Stockwerke. Damit sie sich besser abheben, werden sie geschrumpft. Die Texturierung erfolgt über eine definierte Anzahl vorgefertigter Fenster- und Wandtexturen, wobei die Texturen

gleiche Größe besitzen und kachelbar sind. Entsprechend der Höhe der Textur wird jedes Stockwerk unterteilt und texturiert. Die Gebäude können also in Höhe, Anzahl der Stockwerke und Textur variiert werden. Ein Bildschirmfoto einer in Echtzeit erzeugten Stadt ist in [Abbildung 3.11](#) dargestellt.

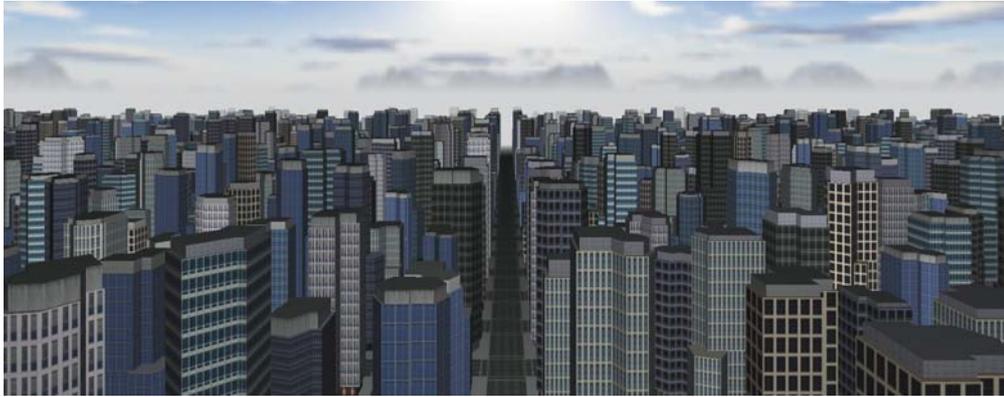


Abbildung 3.11: Eine virtuelle Stadt, die vollständig in Echtzeit generiert wird (aus [\[GPSL03a\]](#)).

Ausgezeichnete architektonische Elemente (Balkon, Erker, Vorsprünge, Gesimse) von Gebäuden können damit nicht erzeugt werden. Dadurch, dass die Positionen der Gebäude über ein regelmäßiges 2D-Raster gegeben sind, können – im Gegensatz zur Arbeit von Müller [\[MWH+06\]](#) – nur rechtwinklige, völlig regelmäßige Straßenzüge erzeugt werden. In einer neueren Arbeit von Greuter et al. [\[GSL07\]](#) wird diese Einschränkung ein wenig aufgeweicht, indem das Verfahren von Miyata [\[Miy90\]](#) zur Berechnung von Mauerwerken herangezogen wird. Aber auch bei dieser Methode basieren die Straßenzüge auf einem regelmäßigem Raster, bei dem lediglich die 2D-Koordinaten der Kreuzungspunkte leicht verschoben werden. Ebenso weisen die erzeugten Gebäude weniger Vielfalt auf, da sie nur durch Extrusion der zwischen Straßen eingeschlossenen Flächen entstehen. Verschiedene Stadtbezirke, wie in der Arbeit von Müller [\[MWH+06\]](#), können nicht beschrieben werden.

3.6.1.6 Procedural Modelling of Vernacular Housing for Virtual Heritage Environments [\[BJDA01, BBJ+01\]](#)

Birch et al. präsentieren ein Plugin für *3D Studio Max* zum interaktiven Modellieren von Gebäuden. Ihr Augenmerk liegt vor allem auf interaktiver Modellierung, wobei die vom Benutzer ausgeführten Aktionen in einer Art Konstruktionshistorie gespeichert werden. Des Weiteren werden in einer Bibliothek die Modelle der verschiedenen Gebäudetypen gespeichert.

Für die Modellierung eines Gebäudes dient ein Quader als Grundlage, dessen

Ausmaße entlang der Raumachsen verändert werden können. Um komplexere Gebäude zu erstellen, können weitere Quader an den Seiten des Ausgangsquaders angebracht werden. Zusätzlich können vom Benutzer Beschränkungen (z. B. Verschiebung nur entlang einer Achse) für jeden Quader festgelegt werden. Weitere, bereits vorgefertigte Modelle, wie beispielsweise Türen und Fenster, können an den Quadern angebracht werden. Ebenso wird eine spezielle Technik für die Modellierung von Fenstern vorgeschlagen. Ausgehend von einem Rechteck, kann ein Fenster in genau zwei Unterfenster unterteilt werden, welche wiederum in genau zwei Unterfenster unterteilt werden können usw. Die Fensterrahmen werden dabei automatisch erzeugt und die Fensterscheiben können mit verschiedenen Texturen versehen werden. Unterstützt werden auch Erker als eine Art spezialisierter Fensterform. Für jeweils einen Quader, der einen Raum verkörpert, wird zusätzlich eine Dachform spezifiziert. Zur Verbesserung des optischen Eindrucks werden die Gebäude zusätzlich mit verschiedenen Texturen versehen. Weiterhin werden noch splinebasierte Techniken zur interaktiven Modellierung vorgestellt. Damit lassen sich beliebige Formen, wie z. B. Säulen erzeugen. Diese Formen können wiederum zu komplexen Formen zusammengesetzt werden. Das Gebäude in Abbildung 3.12 wurde mit dem entwickelten Plugin für 3D Studio Max modelliert.

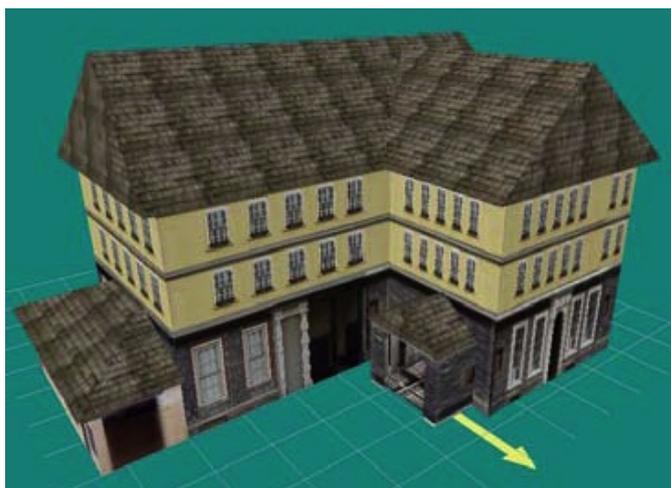


Abbildung 3.12: Einfaches Gebäude, modelliert mit dem von Birch et al. [BJDA01] entwickelten Plugin für *3D Studio Max*.

In künftigen Arbeiten sollen damit z. B. Verstrebungen an Kirchendecken generiert werden. Sämtliche Aktionen die beim Prozess der Modellierung durch den Benutzer vorgenommen werden, werden in einer Konstruktionshistorie gespeichert, wie es z. B. auch in Maya üblich ist. Die vom Benutzer erzeugten Modelle werden nach Zeitperiode und Stil zusammengefasst und in einer Modellbibliothek abgelegt. Abhängigkeiten bzgl. Seitenverhältnissen, Maßen usw. können eingebracht werden, um falsche Eingaben des Benutzers zu verhindern. Über spezialisierte Benutzungsschnittstellen werden alle wichtigen Daten, z. B. für eine Kirche, eingegeben.

Eine Einschränkung ist, dass Gebäude nur aus rechteckigen Grundformen aufgebaut werden können. Außerdem werden Details, wie Gesimse, Formen für Fensterrahmen usw., nur unzureichend unterstützt. Im Unterschied hierzu wird in dieser Arbeit eine symbolische Beschreibung der Gebäudefassade auf verschiedenen Detailstufen vorgenommen, die auch die räumlichen Beziehungen der architektonischen Strukturen untereinander widerspiegelt.

3.6.2 Kommerzielle Produkte

In diesem Abschnitt werden kommerzielle Produkte zur Erzeugung von Gebäuden und Gebäudefassaden vorgestellt. Die Einsicht in die darunter liegende Technik ist meist verborgen. Die Mächtigkeit und die Grenzen der Systeme werden dargelegt.

3.6.2.1 SketchUp [[Goo08](#)]

SketchUp ist ein einfach zu bedienendes 3D-Modellierungsprogramm, das von Google vertrieben wird. Es ist in zwei verschiedenen Varianten erhältlich, einer Kaufversion und eine freie Version mit etwas eingeschränkter Funktionalität. Die eigentliche Intention ist die Erstellung von Gebäuden für *Google-Earth*. Mit dieser Software lassen sich nicht nur Häuser, sondern beliebige Objekte modellieren. Gebäude können sehr einfach und schnell erzeugt werden. Durch Unterteilen eines Rechtecks entstehen Polygone, die den Grundriss des Erdgeschosses definieren. Diese Polygone können extrudiert und damit zum 3D-Körper werden. Auf den polygonalen Flächen lassen sich weitere Polygone erzeugen, entweder unter Einbezug der Kanten oder völlig frei. Sehr feine Strukturen, wie Gesimse oder geometrieabhängige Texturen, müssen von Hand erzeugt werden. Ist das Gebäude einmal erstellt, so können zwar noch Änderungen vorgenommen werden, aber die darunter liegende Gebäudestruktur, wie z. B. Mauervorsprünge, Erker oder der Gebäudehaupttrakt, werden nicht berücksichtigt. Ebenso müssten auch die geometrieabhängigen Texturen neu erstellt werden. Die Modellieretechniken von SketchUp eignen sich hervorragend, um die grobe Struktur eines Gebäudes zu erstellen. Leider lassen sich beim Konstruktionsverlauf die Strukturen des Gebäudes nicht mitprotokollieren, um zu einem späteren Zeitpunkt Änderungen vornehmen zu können.

Die in dieser Arbeit entwickelten Techniken eignen sich hervorragend, um in SketchUp integriert zu werden. Die grobe Struktur eines Beispielgebäudes dieser Arbeit wurde mit SketchUp nachmodelliert (siehe Abbildung [3.13](#)).

3.6.2.2 ArCon [[Arc00](#)]

ArCon – Visuelle Architektur bietet professionelle Software zur Planung und zum Bau von Häusern. Es sind vier verschiedene Varianten erhältlich, die sich in Hin-

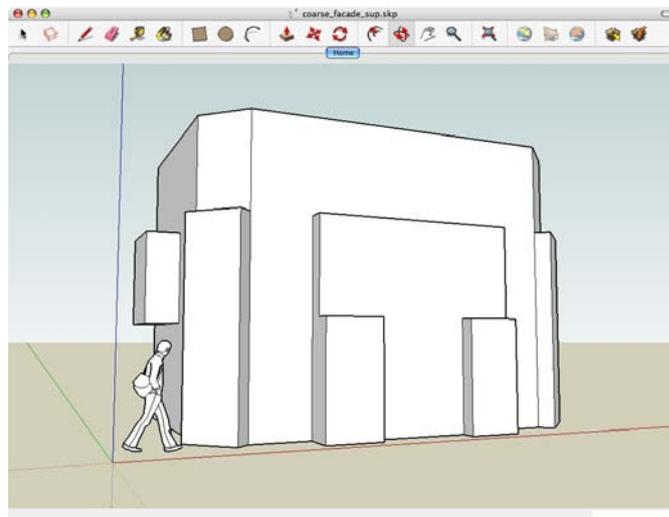


Abbildung 3.13: Grobe Gebäudestruktur die mit SketchUp [Goo08] modelliert wurde.

sicht des Leistungsumfangs unterscheiden, wobei die Demoversion ArCon+ 5.0 getestet wurde. Der Schwerpunkt der Software liegt in der Planung und dem Bau von Gebäuden. Es wird gezielt die einfache Erstellung von Grundrissen unterstützt. In einer vollständig interaktiven Umgebung kann der Bauherr oder Architekt das Gebäude visuell entwickeln. Hierfür stehen verschiedene Bibliotheken zur Verfügung, die vorgefertigte Modelle enthalten, die im Folgenden vorgestellt werden. Aus einer ca. 4000 Teile umfassenden Bibliothek können verschiedene Elemente für Fenster, Dachformen, Dachgauben usw. ausgewählt werden. Ebenso steht eine weitere Bibliothek für die Inneneinrichtung zur Verfügung. Eine zusätzliche Bibliothek beinhaltet vielfältige Texturen für Dächer, Wände, Böden usw. Ein Haus kann komplett interaktiv aus den bereitgestellten Elementen zusammengesetzt werden oder man greift auf vorgefertigte Modelle zurück und passt sie den eigenen Vorstellungen entsprechend an. Die Treppen, die die Stockwerke miteinander verbinden, werden dabei automatisch erzeugt. Um das visuelle Erscheinungsbild des Gebäudes zu verbessern, kann es noch in ein frei definierbares Geländemodell eingebettet werden. Die gesamte Szene kann photorealistisch berechnet werden. In der interaktiven 3D-Darstellung können die Räume des Gebäudes durchwandert werden, um die Wirkung des Gebäudes so realistisch wie möglich zu vermitteln. Damit können Unstimmigkeiten zu einem frühen Zeitpunkt erkannt und eliminiert werden. Für das fertige Gebäudemodell kann abschließend eine Kostenabschätzung für den Bau automatisch berechnet werden. In Abbildung 3.14 ist ein Gebäude zu sehen, das mit ArCon modelliert wurde.

Es ist offensichtlich, dass diese Software nicht für die Computergraphik und -animation gedacht, sondern ausschließlich für die Planung und den Bau von Häu-



Abbildung 3.14: Gebäude, erzeugt mit ArCon [Arc00].

sern. Ebenso sind, abgesehen von der automatischen Generierung der Treppen, keine prozeduralen Elemente enthalten.

3.7 Vergleich der Techniken

Die in den vorherigen Abschnitten vorgestellten Techniken werden wie folgt miteinander verglichen: verschiedene Kriterien werden thematisch in Blöcke zusammengefasst. Anschließend werden die Techniken anhand dieser Kriterien tabellarisch miteinander verglichen. Zusätzlich wird am Ende jeder Tabelle das in dieser Arbeit entwickelte System und seine Techniken aufgeführt. Ob die Techniken die Kriterien erfüllen, wird mit folgenden Symbolen gekennzeichnet:

- +: das Kriterium wird erfüllt.
- -: das Kriterium wird nicht erfüllt.
- o: das Kriterium wird eingeschränkt berücksichtigt.

In Tabelle 3.1 werden die wissenschaftlichen und kommerziellen Gesamtsysteme miteinander unter folgenden Gesichtspunkten verglichen:

- **prozedurale Technik:** Unter diesem Punkt sind in der Tabelle die verwendeten prozeduralen Techniken aufgelistet. Dazu zählen die L-Systeme, die Grammatiken, die Formengrammatiken (*shape grammars*, siehe Abschnitt 2.1.3) wie auch Programmiersprachen. Sie werden mit L, G, S und P abgekürzt.
- **beliebige Stockwerke:** Dieses Feld gibt darüber Auskunft, ob das System für jedes Stockwerk einen beliebigen Grundriss zulässt.

- **Gebäudestrukturinformation:** Über diesen Eintrag ist zu ersehen, ob Informationen über die grobe Struktur, wie z. B. Erker, Mauervorsprünge usw., beim Gebäudemodell vorgesehen sind.
- **räumliche Information:** Für das Erzeugen der Geometrie einer Fassade bzw. eines Gebäudes ist es außerordentlich wichtig, dass man Anfragen über räumliche Zusammenhänge aneinander grenzender architektonischer Elemente beantworten kann. Das kann zum einen über die Gebäudestrukturinformation geschehen oder in anderer Form zur Laufzeit berechnet werden.
- **architektonische Details:** Hier wird die Frage beantwortet, inwieweit das System parametrisierte Details, wie z. B. Fensterformen, Gesimse usw., unterstützt.

In Tabelle 3.2 werden die Texturierungstechniken anhand nachfolgender Kriterien betrachtet:

- **geometrieabhängig:** Kann die Textur abhängig von der darunter liegenden Geometrie erzeugt werden?
- **Eignung für architektonische Elemente:** Inwiefern eignet sich die jeweilige Technik für die Texturierung architektonischer Elemente? Hier wird auch berücksichtigt, ob sich die Verfahren zur Erzeugung der Texturen derart verändern lassen, so dass sich die Texturen optimal an die Geometrien anpassen.

Die vorgestellten Methoden der Dachgenerierung sind in Tabelle 3.3 zusammengefasst. Dabei werden die Methoden auf folgende Punkte überprüft:

- **verschiedene Dachtypen:** Werden von der jeweiligen Methode verschiedene Dachtypen (Satteldach, Pultdach, Walmdach usw.) unterstützt?
- **Berücksichtigung der Gebäudestruktur:** Wird die darunter liegende Gebäudestruktur bei der Erzeugung der Dächer berücksichtigt?
- **automatische Anpassung:** Werden aneinander grenzende Dächer automatisch aneinander angepasst?

Die Mächtigkeit der Verfahren zur geometrischen Modellierung architektonischer Elemente ist unter den nachfolgenden Gesichtspunkten in Tabelle 3.4 veranschaulicht.

- **Parametrisierbarkeit:** Inwieweit lassen die jeweiligen Verfahren eine Parametrisierung der Elemente zu?

- **Komplexität:** Wie komplex bzw. umfangreich können die mit dem Verfahren erzeugten Elemente werden?
- **Vielfältigkeit:** Wie unterschiedlich sind die generierte Elemente, wie groß ist die Vielfalt?

In Tabelle 3.5 sind die Techniken zur Erzeugung der Grundrisse veranschaulicht. Dabei sind die folgenden vier Fragen von Interesse:

- **Grundrisstrukturinformation:** Kann bereits auf Grundrissebene Information über den strukturellen Aufbau des Gebäudes modelliert werden?
- **beliebige Stockwerke:** Kann der Grundriss derart erweitert werden, dass verschiedene Grundrisse für jedes Stockwerk möglich sind?
- **räumliche Information:** Lassen sich für das gesamte Gebäude räumliche Anfragen noch leicht beantworten?
- **Dach:** Werden die Strukturinformationen auch für die Konstruktion der Dächer genutzt?

	<i>prozedurale Techniken</i>	<i>beliebige Stockwerke</i>	<i>Gebäudestrukturinfo.</i>	<i>räumliche Information</i>	<i>arch. Details</i>
Wissenschaftliche Systeme					
Procedural Modeling of Cities [PM01]	L,G	+	–	–	–
Instant Architecture [WWSR03]	G,S	o	+	+ ¹	–
Procedural Modeling of Buildings [MWH+06]	L,G,S	+	+	o ²	–
Build-by-number [BA05]	G	o	o	+	–
Real-time procedural generation of ‘pseudo infinite’ cities [GPSL03a]	– ³	o	–	o	–
Procedural Modelling of Vernacular Housing for Virtual Heritage Environments [BJDA01]	–	o	o	o	–
Kommerzielle Systeme					
SketchUp [Goo08]	–	+	–	–	–
ArCon [Arc00]	–	+	o	o	– ⁴
System und Techniken dieser Arbeit					
ProcMod	P	+	+	+	+

Tabelle 3.1: Gesamtsysteme zur prozeduralen Modellierung von Gebäuden.

¹Diese Informationen stehen nur im Rahmen der vorgenommenen Unterteilung zur Verfügung.

²Die Strukturinformationen werden über Schnittberechnungen mittels Octree bestimmt.

³Städte und Gebäude werden zufällig generiert.

⁴Es wird eine umfassende Bibliothek von Objekten mitgeliefert.

	<i>geometrieabhängig</i> <i>Eignung für arch. Elem.</i>	
Texturierung		
Wang Tiles for image and texture generation [CSHD03]	–	○
Fractional Fourier Texture Masks [NMMK05]	–	–
Computer-generated floral ornament [WZS98]	+	+
A method of generating stone wall patterns [Miy90]	–	+
Feature-based cellular texturing [LDG01]	+	+
Wissenschaftliche Systeme		
Procedural Modeling of Cities [PM01]	–	–
Instant Architecture [WWSR03]	–	–
Procedural Modeling of Buildings [MWH+06]	–	○
Build-by-number [BA05]	+	○
Real-time procedural generation of ‘pseudo infinite’ cities [GPSL03a]	–	○
Procedural Modelling of Vernacular Housing for Virtual Heritage Environments [BJDA01]	–	○
Kommerzielle Systeme		
SketchUp [Goo08]	–	–
ArCon [Arc00]	–	○
System und Techniken dieser Arbeit		
ProcMod	+	+

Tabelle 3.2: Texturierungstechniken und ihre Anwendbarkeit auf architektonische Elemente.

	<i>verschiedene Dachtypen</i> <i>berücks. Gebäudestruktur</i> <i>autom. Anpassung</i>		
Dachtechniken			
Straight Skeleton Implementation [FO98]	–	–	–
Automatically Generating Roof Models from Building Footprints [LD03]	+	o ¹	+
Wissenschaftliche Systeme			
Procedural Modeling of Cities [PM01]	–	–	–
Instant Architecture [WWSR03]	–	–	–
Procedural Modeling of Buildings [MWH ⁺ 06]	+	o ¹	+
Build-by-number [BA05]	–	–	–
Real-time procedural generation of ‘pseudo infinite’ cities [GPSL03a]	–	–	–
Procedural Modelling of Vernacular Housing for Virtual Heritage Environments [BJDA01]	+	+	–
Kommerzielle Systeme			
SketchUp [Goo08]	–	–	–
ArCon [Arc00]	+	+	o
System und Techniken dieser Arbeit			
ProcMod	o ²	+	+

Tabelle 3.3: Techniken zur Erzeugung von Dächern.

¹Gebäudestrukturen sind nicht eindeutig zuzuordnen, da sich ihre Grundflächen überlappen dürfen und außerdem rechtwinklig sein müssen.

²Die klassischen Dachstrukturen wie Flach-, Walm-, Sattel-, Pult- und Zeltdach werden unterstützt. Mansardendächer werden derzeit nicht angeboten.

	<i>Parametrisierbarkeit</i>	<i>Komplexität</i>	<i>Vielfältigkeit</i>
Modellierung architektonischer Elemente			
A versatile 3D model representation for cultural reconstruction [HF01]	–	○ ¹	○ ¹
Rapid procedural-modelling of architectural structures [BBJ ⁺ 01]	+	○	○
Wissenschaftliche Systeme			
Procedural Modeling of Cities [PM01]	–	– ²	– ²
Instant Architecture [WWSR03]	–	○ ³	○ ³
Procedural Modeling of Buildings [MWH ⁺ 06]	–	○ ³	○ ³
Build-by-number [BA05]	–	–	–
Real-time procedural generation of ‘pseudo infinite’ cities [GPSL03a]	–	–	–
Procedural Modelling of Vernacular Housing for Virtual Heritage Environments [BJDA01]	○	○	○
Kommerzielle Systeme			
SketchUp [Goo08]	–	–	–
ArCon [Arc00]	–	– ⁴	– ⁴
System und Techniken dieser Arbeit			
ProcMod	+	+	+

Tabelle 3.4: Verfahren zur geometrischen Modellierung architektonischer Elemente.

¹Die Modelle werden von Hand erzeugt.

²Es werden vorgefertigte Texturen statt 3D-Modelle verwendet.

³Es werden entweder vorgefertigte 3D-Modelle oder Texturen verwendet.

⁴ArCon beinhaltet eine sehr umfangreiche Datenbank von 3D-Modellen.

	Gebäudestruktur	beliebige Stockwerke	räumliche Information	Dach
Grundrisstechniken				
Automatically Generating Roof Models from Building Footprints [LD03]	+	-	+ ¹	+
Interactive physically-based manipulation of discrete/continuous models [HWB95]	+	-	+ ¹	-
A Heuristic Method of Forming up the Schemes of Layouts of Industrial Complexes [GG71]	+	-	+ ¹	-
Wissenschaftliche Systeme				
Procedural Modeling of Cities [PM01]	-	+	-	-
Instant Architecture [WWSR03]	+	o ²	o ²	-
Procedural Modeling of Buildings [MWH ⁺ 06]	+	+	+ ³	+
Build-by-number [BA05]	+	o	+	-
Real-time procedural generation of 'pseudo infinite' cities [GPSL03a]	-	+	-	-
Procedural Modelling of Vernacular Housing for Virtual Heritage Environments [BJDA01]	+	o	o	o
Kommerzielle Systeme				
SketchUp [Goo08]	-	+	-	-
ArCon [Arc00]	-	+	-	-
System und Techniken dieser Arbeit				
ProcMod	+	+	+	+

Tabelle 3.5: Methoden zur Beschreibung von Grundrissen.

¹Es ist nur die Modellierung des Umrisses der bebauten Fläche vorgesehen.

²Diese Informationen stehen nur im Rahmen der vorgenommenen Unterteilung zur Verfügung.

³Die Strukturinformationen werden über Schnittberechnungen mittels Octrees bestimmt.

3.8 Fazit

Im Folgenden werden die Daten der Tabellen ausgewertet und Limitierungen der Techniken kurz diskutiert. Am Ende des Abschnitts wird ein kleiner Vorgeschmack auf die Lösungen und Erweiterungen der angesprochenen Einschränkungen gegeben.

Aus Tabelle 3.1 geht hervor, dass bisher kein System parametrisierte Fassadendetails, wie z. B. Fenster und Gesimse, unterstützt. Ebenso berücksichtigen diese Systeme die Gebäudestruktur (Seitenflügel, Hauptflügel, Mauervorsprünge usw.) gar nicht oder nur unzureichend. Um räumliche Abhängigkeiten zwischen aneinander grenzenden Elementen zu ermitteln, müssen Schnittoperationen durchgeführt werden.

Wie in Tabelle 3.2 zu sehen ist, existieren zwar Techniken zur geometrieabhängigen Texturierung, wobei diese aber bisher nicht in ein Gesamtsystem zur Modellierung von Gebäuden eingebracht wurden.

Tabelle 3.3 zeigt auf, dass die Techniken, um Dächer zu erzeugen, nicht vollständig zufriedenstellend gelöst sind. Zum einen wird zwar die zugrunde liegende Gebäudestruktur berücksichtigt, wobei sich die Strukturen aber durchdringen und auf rechteckige Grundflächen beschränkt sind. Zum anderen können auch für einfache Polygone automatisch Dächer generiert werden, wobei aber weder die Gebäudestruktur noch der Benutzer hinreichend Einfluss auf die Dachform nehmen können.

Tabelle 3.4 greift Techniken zur Modellierung architektonischer Elemente auf. Hier fällt auf, dass weder die verschiedenen Systeme noch die Techniken eine hinreichende Parametrisierung, Komplexität und Vielfalt besitzen.

In Tabelle 3.5 werden schließlich die Techniken zur Beschreibung und Erzeugung von Grundrissen hinsichtlich der beiden folgenden Fähigkeiten miteinander verglichen. Zum einen wird die Frage beantwortet, ob räumliche Abhängigkeiten leicht abzuleiten sind. Zum anderen wird überprüft, ob die Grundrissbeschreibung Einfluss auf die Erzeugung der Dächer nimmt. Mit Ausnahme der Arbeit von Müller et al. [MWH⁺06] wird von keiner weiteren Technik das Zusammenspiel von Gebäudestruktur und Dach wie auch die räumlichen Abhängigkeiten berücksichtigt. Wobei die Gebäudeteile bei Müller et al. auf rechteckige und runde Grundformen beschränkt sind. Ebenso besitzen sie keine Informationen über ihre Funktionalität im Gebäude. Da sich die Gebäudeteile durchdringen dürfen, müssen erst Schnittoperationen durchgeführt werden, um räumliche Abhängigkeiten aufzulösen. Anschließend werden erst die Stockwerke eingepasst.

In dieser Arbeit wird eine Technik zur Beschreibung von Grundrissen bzw. Umrissen vorgestellt, die die Gebäudestruktur berücksichtigt und die einfache Ableitung räumlicher Abhängigkeiten erlaubt. Ebenso fließen die Informationen des

Grundrisses in die Erzeugung der gesamten Dachform ein. Jedes Gebäudeteil besitzt seine eigene Dachform, wobei diese schließlich zu einer Dachstruktur miteinander verschmolzen werden. Die Gebäudeteile besitzen eigenes Objektwissen, d. h. sie kennen ihren Verwendungszweck im Gebäude. Es werden verschiedene architektonische Elemente in parametrisierter Form bereitgestellt. Dazu zählen horizontale Strukturen (z. B. Gesimse und Sockel), beliebige Fenster- und Türöffnungen sowie deren Rahmen. Schließlich werden die räumlichen Abhängigkeiten genutzt, um aneinander grenzende Strukturen anzupassen. Damit ist die Voraussetzung für geometrieabhängiges Texturieren erfüllt. Hierfür werden pflanzenähnliche Ornamente, Zyklopenmauerwerke wie auch beliebige Muster für Mauerziegelverbände angeboten.

Kapitel 4

Grundrissbeschreibung

In diesem Abschnitt werden im ersten Teil Methoden und Formalismen für die Beschreibung des Grundrisses¹ eines Stockwerks definiert. Die Beschreibung wird derart gewählt, dass markante Abschnitte wie Mauervorsprünge, Erker, Hausecken, Wände usw. und ebenso deren Nahtstellen modelliert werden. Ein Grundriss wird dabei aus einzelnen konvexen Polygonen, die miteinander verbunden sind, aufgebaut. Anschließend wird gezeigt, wie sich darüber die markanten Abschnitte herausarbeiten lassen.

Methoden, die die Erweiterung eines Grundrisses zu Stockwerken hin zu ganzen Gebäuden erlauben, werden im zweiten Teil beschrieben. Dabei liegt das Interesse auf den Nahtstellen zweier aufeinander folgender Stockwerke.

Im letzten Abschnitt werden Techniken zur Repräsentation verschiedener Dachformen vorgestellt. Auch steht hier wieder die Herausarbeitung der Nahtstellen zu angrenzenden Elementen, wie das darunter liegende Stockwerk oder angrenzende Wände, im Vordergrund.

4.1 Allgemeines

Die Techniken zur Beschreibung von Grundrissen im Bereich der generativen Modellierung in der Computergraphik sind bisher in ihrer Ausdrucksstärke nur unzureichend. Aktuelle Arbeiten, wie in Abschnitt 3.5 beschrieben, repräsentieren die Gebäudestruktur in den Grundrissen nur in sehr geringen Umfang bzw. gar nicht. Daraus ergeben sich verschiedene Probleme. Zum einen müssen Schnittoperationen zur Bestimmung von Gebäudeteilen durchgeführt werden, um räumliche In-

¹Es sei erwähnt, dass der Begriff Grundriss hier nicht im strengen Sinne der Architektur verwendet wird, sondern mit Grundriss bzw. Grundrisspolygon ist in dieser Arbeit die Umrisslinie eines Grundrisses gemeint.

formationen zu gewinnen. Da diese Gebäudeteile aber keine Information über ihre Funktion im Gebäude besitzen, müssen diese Informationen nachträglich erzeugt werden. Dabei sind z. B. die Anzahl an Stockwerken, die Fensterverteilung usw. der einzelnen Teile aneinander anzupassen. Um diesen Vorgang zu automatisieren, müssen viele Vereinfachungen und Annahmen gemacht werden, die wiederum Fehlerquellen beinhalten können.

Die im Folgenden vorgestellten Beschreibungsformen umgehen derartige Probleme, bieten dabei aber eine große Vielfalt an verschiedenen Grundrissen, sogar über Stockwerke hinweg. Informationen über die Gebäudestruktur, wie Mauervorsprünge oder Erker werden damit berücksichtigt. Dadurch lassen sich räumliche Informationen über die Elemente des Gebäudes bzw. der Fassade sehr leicht ableiten. Genutzt werden diese Informationen, um aneinander grenzenden Strukturen zu entdecken und um sie aneinander anzupassen. Damit werden Schnittoperationen und die damit einhergehenden Probleme vermieden.

4.2 Grundriss eines Stockwerks

In diesem Abschnitt werden Methoden für die Beschreibung des Grundrisses eines einzelnen Stockwerks definiert. Ein Grundriss wird hierbei aus sog. Grundrissmodulen aufgebaut. Dafür werden die Module über zwei Typen von Verbindungen miteinander verknüpft und beschreiben den Grundriss eines Stockwerks.

4.2.1 Grundrissmodul

Definition 4.1. Grundrissmodul

Ein Grundrissmodul (floorPlanModule oder fpm) ist ein konvexes Polygon, definiert als Tupel $fpm = (V, E)$ mit folgenden Bedingungen: V ist eine endliche Menge von Punkten im \mathbb{R}^2 . E ist eine endliche Menge gerichteter und über $[0, 1]$ parametrisierter Kanten mit $E \rightarrow V \times V$.

Jedes Modul ist von einem bestimmten Typ (Mauervorsprung oder Risalit, Balkon usw.) der später in der Beschreibung der Fassade zum Tragen kommt. Weiterhin erhalten die Module zusätzliche Daten wie Wanddicke, Höhe, Material, die für die geometrische und graphische Darstellung von Bedeutung sind. In Tabelle 4.1 sind alle Daten eines Moduls aufgelistet.

Das Polygon eines Grundrissmoduls ist wie folgt definiert:

$$(\mathbf{p}_i, \mathbf{p}_{i+1}) \in E \quad \text{mit} \quad i = 1, \dots, n-1 \quad \text{und} \quad (\mathbf{p}_n, \mathbf{p}_1) \in E .$$

Mit dem letzten Zusatz $(\mathbf{p}_n, \mathbf{p}_1) \in E$ ist sicher gestellt, dass das Polygon geschlossen ist. Außerdem wird $\mathbf{p}_i \neq \mathbf{p}_j$ für $i \neq j$ vorausgesetzt. Um den vollständigen geschlossenen Kantenzug eines Moduls kompakt schreiben zu können, sei $\mathbf{p}_{n+1} = \mathbf{p}_1$.

<i>Daten</i>	<i>Bedeutung</i>
Polygon	Daten des konvexen Polygons.
Typ	Architektonischer Typ des Moduls, z. B. Risalit, Balkon, usw.
Dicke	Dicke der Umrandung (i. Allg. Dicke der Wand).
Höhe	Höhe, i. Allg. handelt es sich um die Höhe der Wand. Sie wird üblicherweise über den Grundriss, welcher das Modul enthält, vorgegeben. Bei einem Balkon repräsentiert die Höhe die Höhe der Balustrade.
Material	Hier werden Farbe, Textur, Struktur/Typ des Mauerwerks usw. gesetzt.

Tabelle 4.1: Daten eines Grundrissmoduls.

Damit ist die Rückkante von \mathbf{p}_n nach \mathbf{p}_1 auch in $(\mathbf{p}_i, \mathbf{p}_{i+1})$ mit $i = 1, \dots, n$ enthalten. Für die Kanten $e_i, i = 1, \dots, n$ wird analog verfahren.

Beispiel für ein Grundrissmodul (fpm_1):

$$\begin{aligned} \text{fpm}_1 = (V_1 &= \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5\}, \\ E_1 &= \{e_1 = (\mathbf{p}_1, \mathbf{p}_2), e_2 = (\mathbf{p}_2, \mathbf{p}_3), e_3 = (\mathbf{p}_3, \mathbf{p}_4), \\ &e_4 = (\mathbf{p}_4, \mathbf{p}_5), e_5 = (\mathbf{p}_5, \mathbf{p}_1)\}) . \end{aligned}$$

In Abbildung 4.1 ist obiges Beispiel graphisch dargestellt.

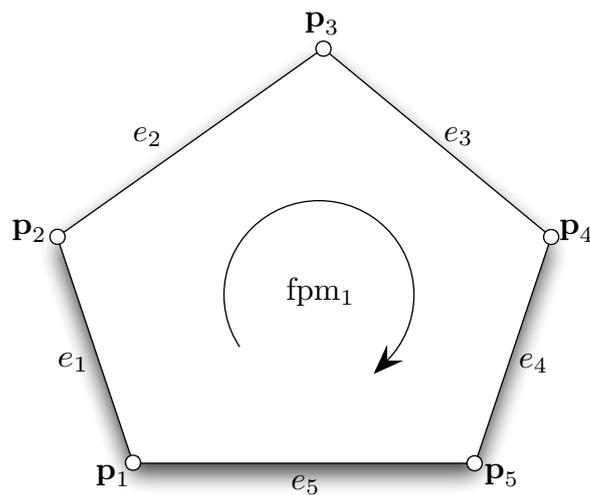


Abbildung 4.1: Grundrissmodul.

Ein Grundriss besteht aus mehreren Grundrissmodulen (*fpm*s). Diese Module sind über zwei Typen von Verbindungen miteinander verknüpft. In den folgenden Abschnitten werden die Verbindungen und der Grundriss definiert.

4.2.2 Punkt-Punkt-Verbindung

Mit einer Punkt-Punkt-Verbindungen (kurz PP-Verbindung) werden zwei Module über eine Kante miteinander verbunden. Grundrisse, wie z. B. der des Petronas Towers (siehe Abb. 4.2), können darüber beschrieben werden. Voraussetzung für diese Verbindung ist, dass die neu eingeführte Kante keines der beteiligten Module und keine andere PP-Verbindung schneidet.

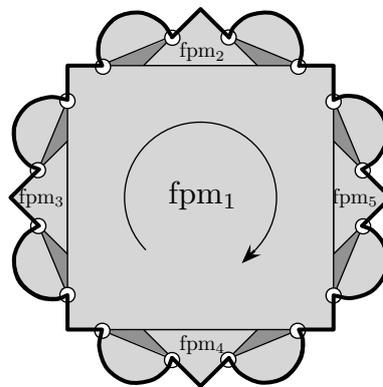


Abbildung 4.2: Grundriss der Petronas Towers: Er besteht aus den fünf Modulen fpm_1, \dots, fpm_5 die über KK-Verbindungen miteinander verknüpft sind und den Modulen in Form eines Halbkreises (polygonal angenähert), die über PP-Verbindungen an den ersten fünf Modulen angebracht sind. Der resultierende Umriss ist fett gezeichnet.

Definition 4.2. Punkt-Punkt-Verbindung

Eine Punkt-Punkt-Verbindung ist ein 6-Tupel

$$pp_connection = (fpm_1, fpm_2, c, d, a, b) .$$

Wobei fpm_1 und fpm_2 Grundrissmodule, c bzw. d Kanten und a bzw. b Skalare sind.

Die sechs Parameter sind wie folgt definiert:

- $fpm_1 = (V_1, E_1) ;$
- $fpm_2 = (V_2, E_2) ;$
- $c = (\mathbf{p}_i, \mathbf{p}_{i+1}) \in E_1 ;$

- $d = (\mathbf{q}_j, \mathbf{q}_{j+1}) \in E_2$;
- $a, b \in [0, 1]$ beschreibt die Punkte $\mathbf{r}_a = (1 - a) \cdot \mathbf{p}_i + a \cdot \mathbf{p}_{i+1}$ auf der Strecke c und $\mathbf{r}_b = (1 - b) \cdot \mathbf{q}_j + b \cdot \mathbf{q}_{j+1}$ auf der Strecke d ;
- $k = (\mathbf{r}_a, \mathbf{r}_b)$ ist die erzeugte Kante zwischen den beiden Modulen fpm_1 und fpm_2 .

Eine Kante eines Moduls kann natürlich mehrere Verbindungen zu weiteren Modulen aufweisen. Die erzeugten Kanten zwischen den Modulen sollten sich nicht schneiden, da in einem späteren Schritt die Kanten als Grundlage zur Erzeugung der Wände herangezogen werden und diese sich dann durchdringen würden.

Gegeben sind die Menge der Verbindungen

$$C_{\text{pp}} = \{\text{pp_connection}_1, \dots, \text{pp_connection}_m\}$$

und die Menge der Module

$$M = \{\text{fpm}_1, \dots, \text{fpm}_n\} ,$$

die miteinander verbunden werden:

$$\begin{aligned} \text{pp_connection}_i &= (\text{fpm}_j, \text{fpm}_k, c_j, d_k, a_i, b_i) , \\ i &= 1, \dots, m \quad j, k = 1, \dots, n \quad j \neq k . \end{aligned}$$

Für die Verbindungen muss folgende Bedingung erfüllt sein:

$$(\mathbf{r}_{a_i}, \mathbf{r}_{b_i}) \cap (\mathbf{r}_{a_j}, \mathbf{r}_{b_j}) = \emptyset \quad i, j = 1, \dots, m \quad i \neq j .$$

4.2.2.1 PP-Module

Über eine PP-Verbindung wird eine weitere Kante zum Grundriss bzw. Umriss hinzugefügt. Jetzt ist es möglich, dass damit, wie in Abbildung 4.3 gezeigt, ein neues Modul entsteht (bezeichnet als PP-Modul, hinsichtlich seiner Entstehung über eine PP-Verbindung), das diese Kante bzw. Kanten beinhaltet. Wie wir später sehen werden, ist das Modul notwendig, damit es zum einen auf das nächste Stockwerk erweitert werden kann. Zum anderen wird das Modul zur Erzeugung der Dachstruktur (Abschnitt 4.4) herangezogen.

Die Berechnung der Eckpunkte des Moduls wird wie folgt vorgenommen: Ausgehend vom Startpunkt \mathbf{r}_a wird im Uhrzeigersinn der Kantenzug abgelaufen. In diesem Fall wird nicht entlang der Außenlinie gelaufen, sondern entlang des inneren Kantenzugs. Das bedeutet, dass im Punkt \mathbf{r}_b entgegen der Richtung des anschließenden Polygons gelaufen wird. Das Verfahren endet, wenn der Ausgangspunkt \mathbf{r}_a wieder erreicht wird.

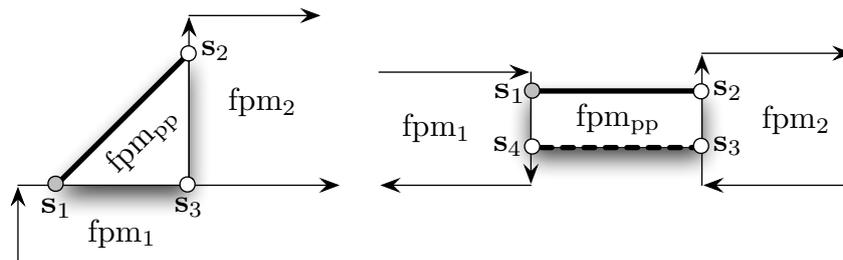


Abbildung 4.3: Zwei PP-Module; links mit einer, rechts mit zwei PP-Verbindungen.

Abbildung 4.3 zeigt zwei Beispiele. Im ersten (links in der Skizze) besteht das gefundene PP-Modul aus (s_1, s_2, s_3) , mit dem Startpunkt s_1 . Das zweite Beispiel zeigt, dass auch weitere PP-Verbindungen Teil des Kantenzugs des PP-Moduls sein können. In diesem Fall ergibt sich als Kantenzug (s_1, s_2, s_3, s_4) . Sollte die zweite PP-Verbindung (in der zweiten Skizze gestrichelt dargestellt) fehlen, so bricht das Verfahren an dem Punkt ab, der einen Innenwinkel größer als 180° aufweist. Damit wird gewährleistet, dass nur konvexe PP-Module erzeugt werden. Zum anderen werden dadurch auch reine Mauern entdeckt, die nicht Bestandteil einer Wand eines Gebäudes sind. Die Kanten der PP-Module, die nicht der Kante einer PP-Verbindung entsprechen (also Teilkanten der angrenzenden Module), definieren Verbindungen zu den anliegenden Modulen, wovon z. B. bei der Dacherzeugung Gebrauch gemacht wird. Im ersten Beispiel sind das die Kanten (s_2, s_3) und (s_3, s_1) und im zweiten (s_2, s_3) und (s_4, s_1) .

4.2.3 Kante-Kante-Verbindungen

Eine Kante-Kante-Verbindung (kurz KK-Verbindung) beschreibt die Verbindung zwischen zwei beliebigen Grundrissmodulen oder der Kante einer PP-Verbindung und einem Grundrissmodul. Dabei wird für eine Kante eines Moduls bzw. der PP-Verbindung ein Bereich im normierten Intervall $[0, 1]$ angegeben, in dem die vollständige Kante des zweiten Moduls angebunden wird. Zu diesem Zweck muss das zweite Modul transformiert (rotiert, skaliert und verschoben) werden, um in das Intervall auf der Kante des ersten Moduls bzw. der PP-Verbindung zu passen.

Definition 4.3. Kante-Kante-Verbindung: Modul–Modul

Hier werden zwei Module über eine gemeinsame Kante verbunden. Ein Kante-Kante-Verbindung für Grundrissmodule ist ein 6-Tupel

$$ee_connection = (fpm_1, fpm_2, c, d, a, b) .$$

Wobei fpm_1 und fpm_2 Grundrissmodule, c bzw. d Kanten und a bzw. b Skalare sind.

Die sechs Parameter sind wie folgt definiert:

- $\text{fpm}_1 = (V_1, E_1)$, an diesem Modul wird das zweite Module fpm_2 angedockt; im Weiteren wird fpm_1 als *dominantes Grundrissmodul* bezeichnet;
- $\text{fpm}_2 = (V_2, E_2)$;
- $c = (\mathbf{p}_i, \mathbf{p}_{i+1}) \in E_1$;
- $d = (\mathbf{q}_j, \mathbf{q}_{j+1}) \in E_2$;
- $a, b \in [0, 1]$ mit $a < b$, beschreibt die Punkte $\mathbf{r}_a = (1 - a) \cdot \mathbf{p}_i + a \cdot \mathbf{p}_{i+1}$ und $\mathbf{r}_b = (1 - b) \cdot \mathbf{p}_i + b \cdot \mathbf{p}_{i+1}$ auf der Strecke c ;
- es gilt: $d = (\mathbf{r}_a, \mathbf{r}_b)$.

Eine Kante eines dominanten Moduls darf auch mehrere Verbindungen zu weiteren Modulen aufweisen. Dabei ist zu beachten, dass die Intervalle, an welchen die Kanten der anderen Module angebunden werden, sich nicht überlappen.

Gegeben sind die Menge der Verbindungen

$$C_{ee} = \{\text{ee_connection}_1, \dots, \text{ee_connection}_m\} ,$$

das dominante Modul fpm_i und die Menge der angedockten Module

$$M = \{\text{fpm}_1, \dots, \text{fpm}_m\}$$

mit den Verbindungen

$$\text{connection}_i = (\text{fpm}_i, \text{fpm}_j, c_i, d_j, a_j, b_j) , \quad j = 1, \dots, m, \quad i \neq j .$$

Für die Verbindungen muss folgende Bedingung erfüllt sein:

$$[a_i, b_i] \cap [a_j, b_j] = \emptyset \quad i, j = 1, \dots, m \quad i \neq j .$$

Somit ist eine Ordnung auf den Verbindungen jeder Kante definiert, wovon bei der Berechnung der Umrisslinie Gebrauch gemacht wird.

Definition 4.4. Kante-Kante-Verbindung: Modul-PP-Verbindung

Hier wird die Kante einer PP-Verbindung mit der Kante eines Moduls verbunden. Eine Kante-Kante-Verbindung für ein Grundrissmodul und eine PP-Verbindung ist ein 5-Tupel

$$\text{ee_connection} = (k, \text{fpm}, c, a, b) .$$

Wobei k die Kante der PP-Verbindung und fpm das Grundrissmodul, d ein Kante und a bzw. b Skalare sind.

Die sechs Parameter sind wie folgt definiert:

- $k = (\mathbf{k}_1, \mathbf{k}_2)$ ist die Kante einer PP-Verbindung;
- $\text{fpm} = (V, E)$;
- $c = (\mathbf{p}_i, \mathbf{p}_{i+1}) \in E$;
- $a, b \in [0, 1]$ mit $a < b$, beschreibt die Punkte $\mathbf{r}_a = (1 - a) \cdot \mathbf{k}_1 + a \cdot \mathbf{k}_2$ und $\mathbf{r}_b = (1 - b) \cdot \mathbf{k}_1 + b \cdot \mathbf{k}_2$ auf der Kante k ;
- es gilt: $c = (\mathbf{r}_a, \mathbf{r}_b)$.

Eine Kante einer PP-Verbindung kann weitere Verbindungen zu anderen Modulen besitzen. Dabei ist zu beachten, dass sich die Intervalle, an denen die Kanten der anderen Module angebinden werden, nicht überlappen.

Gegeben sind die Menge der Verbindungen

$$C_{\text{pp}} = \{\text{pp_connection}_1, \dots, \text{pp_connection}_m\} ,$$

die PP-Verbindungskante k und die Menge der angedockten Module

$$M = \{\text{fpm}_1, \dots, \text{fpm}_k\}$$

mit den Verbindungen

$$\text{pp_connection}_i = (k, \text{fpm}_i, c_i, a_i, b_i) , \quad i = 1, \dots, m .$$

Für die Verbindungen muss nun folgende Bedingung erfüllt sein:

$$[a_i, b_i] \cap [a_j, b_j] = \emptyset \quad i, j = 1, \dots, m \quad i \neq j .$$

Wie bei den KK-Verbindungen, ist auch hier eine Ordnung definiert, die bei der Berechnung der Umrisslinie zum Tragen kommt.

4.2.4 Grundriss

Ein Grundriss enthält verschiedene Grundrissmodule (fpms) die über KK- und PP-Verbindungen verknüpft sind. Im Weiteren wird der Begriff Verbindungen vereinfachend für KK- und PP-Verbindungen verwendet.

Definition 4.5. Grundriss

Ein Grundriss ist ein Tupel $\text{floorPlan} = (M, C)$ und wie folgt definiert: M ist eine endliche Menge von Grundrissmodulen und C eine endliche Menge von Verbindungen mit $C = C_{\text{ee}} \cup C_{\text{pp}}$.

Zu bemerken ist, dass für M und $C_{ee} \subset C$ (also den KK-Verbindungen) folgende Eigenschaft gilt: $|M| = |C_{ee}| + 1$.

Ein Beispiel für einen Grundriss, bestehend aus zwei Modulen, ist in Abbildung 4.4 zu sehen. Formal wird das Beispiel wie folgt beschrieben, wobei das Grundrissmodul fpm_1 bereits in Abschnitt 4.2.1 definiert wurde. fpm_2 und die Verbindung sind wie folgt definiert:

$$\begin{aligned} \text{fpm}_2 &= (V_2 = \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}, \\ &E_2 = \{k_1 = (\mathbf{q}_1, \mathbf{q}_2), k_2 = (\mathbf{q}_2, \mathbf{q}_3), \\ &\quad k_3 = (\mathbf{q}_3, \mathbf{q}_4), k_4 = (\mathbf{q}_4, \mathbf{q}_1)\}) \\ \text{connection} &= (\text{fpm}_1, \text{fpm}_2, e_5, k_2, a, b). \end{aligned}$$

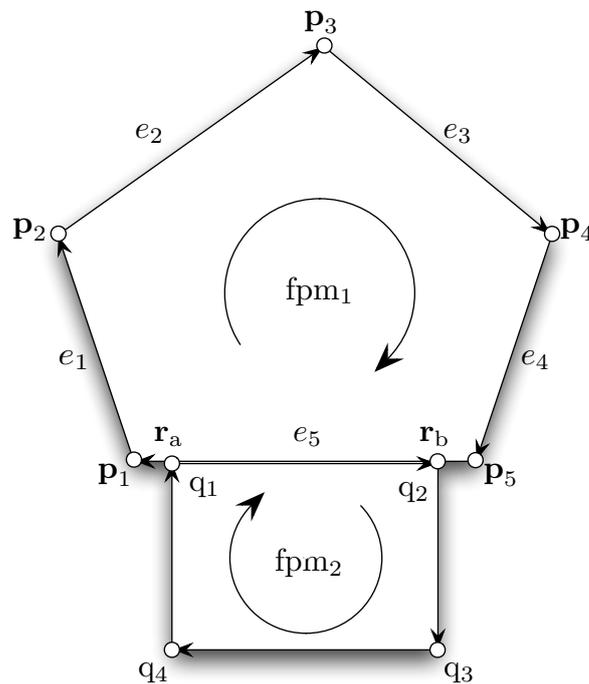


Abbildung 4.4: Grundriss, bestehend aus zwei Modulen.

4.2.5 Merkmale

Die Umrisslinie eines Grundrisses wird entsprechend der Grundrissmodule und deren Verbindungen in Kantenzüge unterteilt. Diese Kantenzüge, im Weiteren als *Merkmale* bzw. *Features* bezeichnet, verlaufen ausschließlich entlang der Außenlinie eines Grundrisses. In Abbildung 4.5 ist ein Grundriss zu sehen, bestehend aus drei Grundrissmodulen und dessen Merkmalen. Dabei ist der rote, gepunktete Kantenzug ein Merkmal des obersten Polygons. Die beiden blauen, durchgezogenen Kantenzüge sind dem mittleren Polygon zugeordnet und der grüne, gestrichelte Kantenzug dem letzten Polygon.

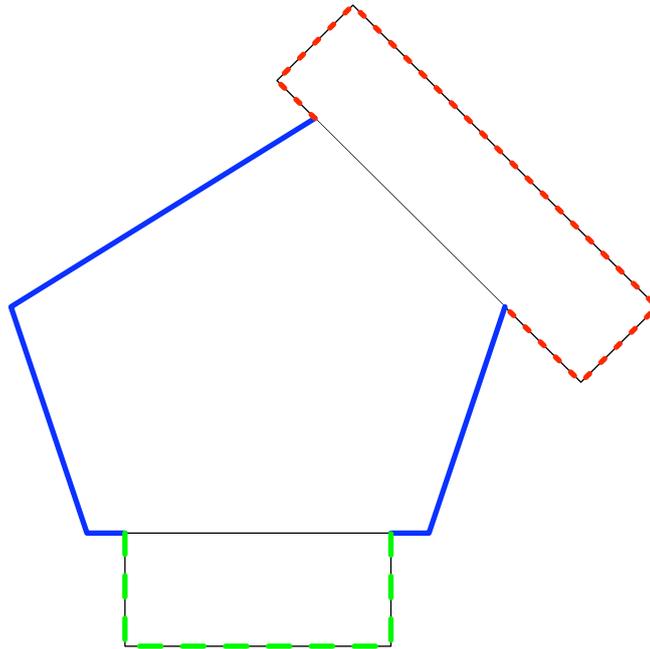


Abbildung 4.5: Merkmale eines Grundrisses.

Definition 4.6. Grundrissmerkmale

Ein Grundrissmerkmal ist ein Kantenzug $F = f_1, \dots, f_n$ eines Grundrissmoduls, der entlang der Umrisslinie eines Grundrisses, i. Allg. zwischen zwei Verbindungen, verläuft.

Im Folgenden werden die Merkmale formal definiert. Es wird eine Methode angegeben, mit der die Merkmale eines Grundrisses bestimmt werden. Gegeben seien ein Grundriss

$$G = (M, C),$$

mit den Modulen

$$M = \{fpm_1, \dots, fpm_l\}$$

und den Verbindungen

$$\begin{aligned} C &= C_{ee} \cup C_{pp} \\ C_{ee} &= \{ee_connection_1, \dots, ee_connection_m\} \\ C_{pp} &= \{pp_connection_1, \dots, pp_connection_n\}, m < l, \end{aligned}$$

wobei die Grundrissmodule fpm_1, \dots, fpm_l alle miteinander über die Verbindungen in C verbunden sind.

Merkmale können in verschiedenen Formen auftreten. Nachfolgend sind alle Möglichkeiten, klassifiziert nach der Art der Verbindung, aufgelistet. Im Weiteren werden die möglichen Merkmale erst für KK-Verbindungen und anschließend für PP-Verbindungen aufgezählt.

1. Der Grundriss besteht aus genau einem Modul $fpm \in F$ (Abbildung 4.6).

Dann ist das Merkmal $F = f_1, \dots, f_m$ mit $f_1, \dots, f_m \in E$.

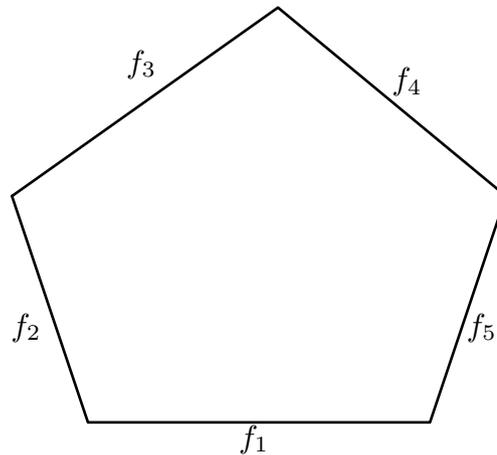


Abbildung 4.6: Merkmal ist gleich dem Umriss des Moduls.

2. Das Merkmal ist an beiden Enden über dominante Verbindungen zu weiteren Modulen abgeschlossen (Abbildung 4.7).

Merkmal $F = f_1, f_2, \dots, f_{m-1}, f_m$ mit

$$\begin{aligned} fpm_k, fpm_i, fpm_j &\in F, \quad k \neq i, j \\ (fpm_k, fpm_i, c_1, d_1, a_1, b_1) &\in C_{ee} \\ (fpm_k, fpm_j, c_2, d_2, a_2, b_2) &\in C_{ee} \\ f_2, \dots, f_{m-1} &\in E_k \\ f_1 &= c_1[b_1, 1] \\ f_m &= c_2[0, a_2]. \end{aligned}$$

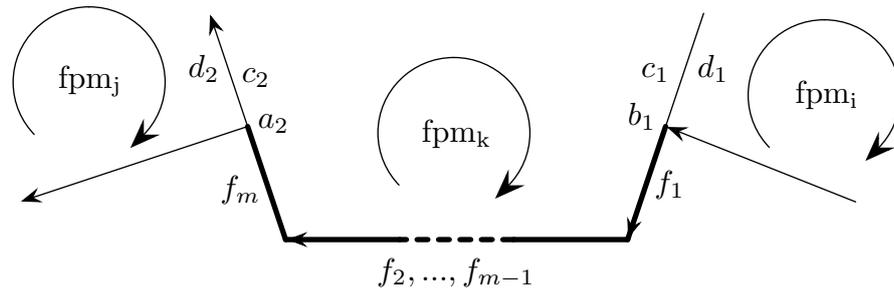


Abbildung 4.7: Merkmal mit zwei dominanten Verbindungen.

3. Das Merkmal ist an beiden Enden über Verbindungen zu weiteren Modulen abgeschlossen, wobei die linke (im Uhrzeigersinn gesehen) dominant ist (Abbildung 4.8).

Merkmal $F = f_1, \dots, f_m$ mit

$$\begin{aligned} fpm_k, fpm_i, fpm_j &\in F, \quad k \neq i, j \\ (fpm_i, fpm_k, d_1, c_1, a_1, b_1) &\in C_{ee} \\ (fpm_k, fpm_j, c_2, d_2, a_2, b_2) &\in C_{ee} \\ f_1, \dots, f_{m-1} &\in E_k \\ f_m &= c_2[0, a_2] . \end{aligned}$$

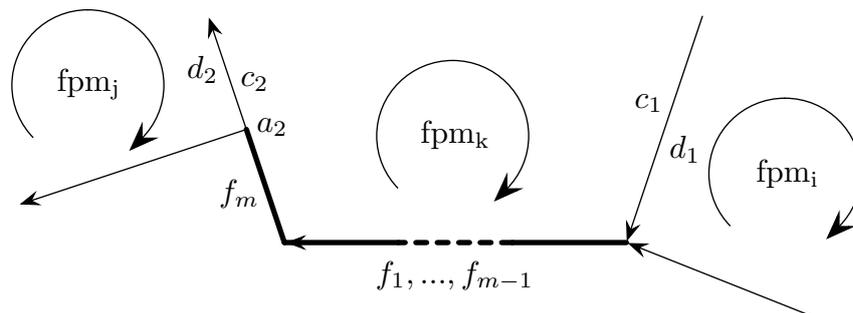


Abbildung 4.8: Merkmal mit zwei Verbindungen, wobei die linke dominant ist.

4. Das Merkmal ist an beiden Enden über Verbindungen zu weiteren Modulen abgeschlossen, wobei die rechte dominant ist (Abbildung 4.9).

Merkmal $F = f_1, \dots, f_m$ mit

$$\begin{aligned} \text{fpm}_k, \text{fpm}_i, \text{fpm}_j &\in F, \quad k \neq i, j \\ (\text{fpm}_k, \text{fpm}_i, c_1, d_1, a_1, b_1) &\in C_{ee} \\ (\text{fpm}_j, \text{fpm}_k, d_2, c_2, a_2, b_2) &\in C_{ee} \\ f_2, \dots, f_m &\in E_k \\ f_1 &= c_1[b_1, 1]. \end{aligned}$$

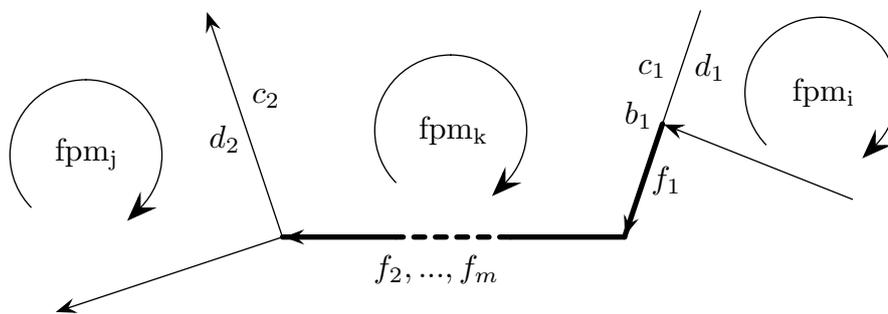


Abbildung 4.9: Merkmal mit zwei Verbindungen, wobei die rechte dominant ist.

5. Das Merkmal ist an beiden Enden über nicht-dominante Verbindungen zu weiteren Modulen abgeschlossen (Abbildung 4.10).

Merkmal $F = f_1, \dots, f_m$ mit

$$\begin{aligned} \text{fpm}_k, \text{fpm}_i, \text{fpm}_j &\in F, \quad k \neq i, j \\ (\text{fpm}_i, \text{fpm}_k, d_1, c_1, a_1, b_1) &\in C_{ee} \\ (\text{fpm}_j, \text{fpm}_k, d_2, c_2, a_2, b_2) &\in C_{ee} \\ f_1, \dots, f_m &\in E_k. \end{aligned}$$

6. Das Merkmal liegt zwischen zwei Verbindungen (Abbildung 4.11).

Merkmal $F = f$ mit

$$\begin{aligned} \text{fpm}_k, \text{fpm}_i, \text{fpm}_j &\in F, \quad k \neq i, j, \quad i \neq j \\ (\text{fpm}_k, \text{fpm}_i, c_1, d_1, a_1, b_1) &\in C_{ee} \\ (\text{fpm}_k, \text{fpm}_j, c_1, d_2, a_2, b_2) &\in C_{ee} \\ f &= c_1[b_1, a_2]. \end{aligned}$$

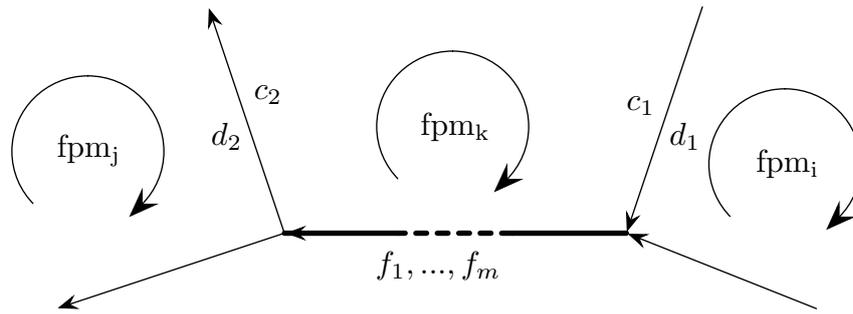


Abbildung 4.10: Merkmal mit zwei Verbindungen, wobei keine dominant ist.

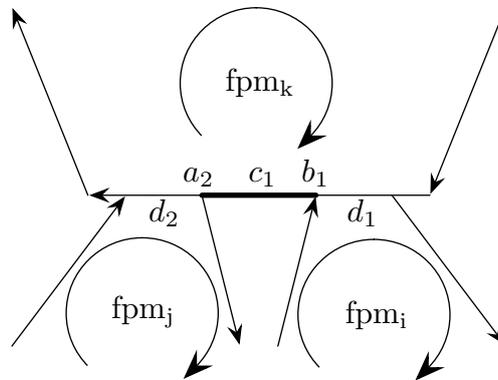


Abbildung 4.11: Merkmal zwischen zwei Verbindungen.

7. Das Merkmal beinhaltet eine PP-Verbindung, an der ein (oder mehrere) Modul über eine KK-Verbindung angebracht ist (Abbildung 4.12).

Merkmal $F = f_1, f_2, \dots, f_{m-1}, f_m$ mit

$$\begin{aligned}
 fpm_k, fpm_i, fpm_j &\in F, \quad k \neq i, j, \quad i \neq j \\
 (fpm_i, fpm_j, c_1, d_1, a_1, b_1) &\in C_{pp} \\
 g &= (\mathbf{r}_{a_1}, \mathbf{r}_{b_1}) \\
 (g, fpm_k, c_2, a_2, b_2) &\in C_{ee} \\
 f_2, \dots, f_{m-1} &\in E_k \\
 f_1 &= g[0, a_2] \\
 f_m &= g[b_2, 1].
 \end{aligned}$$

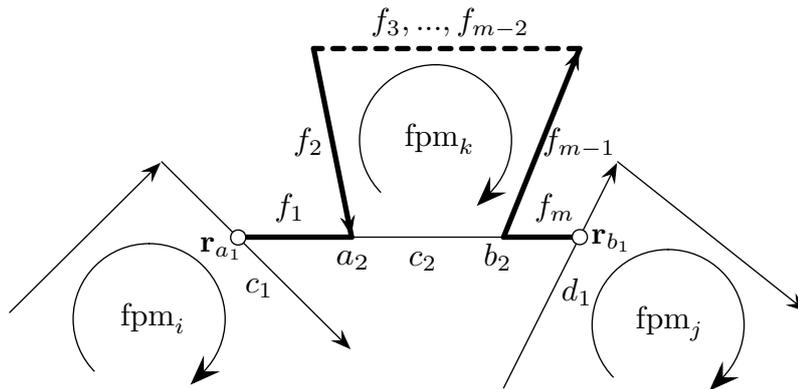


Abbildung 4.12: Merkmal an einer PP-Verbindung mit angedocktem Modul.

8. Das Merkmal ist eine PP-Verbindung, ohne weitere Verbindungen (Abbildung 4.13).

Merkmal $F = f$ mit

$$\begin{aligned} \text{fpm}_i, \text{fpm}_j &\in F, \quad i \neq j \\ (\text{fpm}_i, \text{fpm}_j, c, d, a, b) &\in C_{\text{pp}} \\ f &= (\mathbf{r}_a, \mathbf{r}_b). \end{aligned}$$

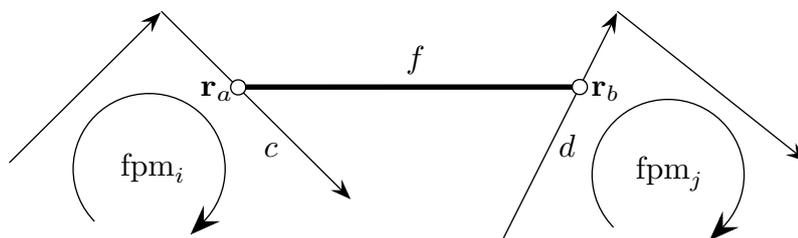


Abbildung 4.13: PP-Verbindung als Merkmal.

9. Das Merkmal ist an beiden Enden über PP-Verbindungen zu weiteren Modulen abgeschlossen (Abbildung 4.14).

Merkmal $F = f_1, f_2, \dots, f_{m-1}, f_m$ mit

$$\begin{aligned} \text{fpm}_k, \text{fpm}_i, \text{fpm}_j &\in F, \quad k \neq i, j \\ (\text{fpm}_k, \text{fpm}_i, c_1, d_1, a_1, b_1) &\in C_{ee} \\ (\text{fpm}_k, \text{fpm}_j, c_2, d_2, a_2, b_2) &\in C_{ee} \\ f_2, \dots, f_{m-1} &\in E_k \\ f_1 &= c_1[b_1, 1] \\ f_m &= c_2[0, a_2] . \end{aligned}$$

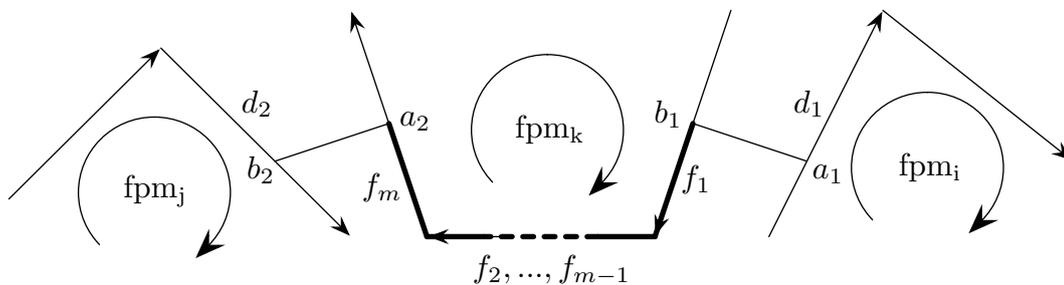


Abbildung 4.14: Merkmal zwischen zwei PP-Verbindungen.

Mit Merkmalen, die an beiden Enden über jeweils eine PP- oder KK-Verbindung abgeschlossen sind, wird analog zu den Merkmalen unter Punkt 1 bis 6 verfahren.

Ein Algorithmus zur Berechnung aller Merkmale eines Grundrisses läuft in zwei Phasen ab. Im ersten Schritt werden alle Merkmale, verursacht von KK-Verbindungen, ermittelt. Anschließend werden Merkmale der PP-Verbindungen hinzugenommen. Der Algorithmus läuft wie folgt ab:

1. Phase: Merkmale der KK-Verbindungen

(a) Suche eines Startpunkts

Ausgehend von einem beliebigen Eckpunkt des Polygons eines Grundrissmoduls wird der Beginn des nächsten Merkmals, hervorgerufen durch eine KK-Verbindung, gesucht.

(b) Ablauf des Kantenzugs

Nun wird der Kantenzug im Uhrzeigersinn abgelaufen und die Merkmalsübergänge der KK-Verbindungen markiert.

2. Phase: Merkmale der PP-Verbindungen

(a) Startpunkt

Begonnen wird mit dem Startpunkt aus der 1. Phase.

(b) Ablauf des Kantenzugs, der in der 1. Phase ermittelt wurde

Der Kantenzug wird im Uhrzeigersinn abgelaufen und die Merkmalsübergänge der PP-Verbindungen markiert. Dabei werden die zwischen zwei Merkmalsübergängen ermittelten Kantenzüge, in Abhängigkeit zu ihren Modulen, ausgegeben.

Das Verfahren durchläuft alle Kanten eines Grundrisses und endet, wenn der Punkt, mit dem begonnen wurde, wieder erreicht wird.

Abschließend lässt sich die Beschreibung der Grundrisse wie folgt kurz zusammenfassen. Die Grundlage bilden einzelne konvexe Polygone mit fester Umlaufrichtung. Diese werden über zwei Verbindungstypen (PP- und KK-Verbindung) miteinander verknüpft und ergeben so den Grundriss.

4.2.6 Einschränkungen und Sonderfälle

Mit der vorgestellten Technik lassen sich nur polygonale Grundrisse erzeugen. Runde Grundrisse, wie bei Türmen, müssen zurzeit über Kantenzüge angenähert werden. Diese Einschränkung kann leicht behoben werden, indem die Verbindungstypen entsprechend erweitert werden.

Bei der Bestimmung der Umrisslinie können die in den Abbildungen 4.15 und 4.16 gezeigten Sonderfälle auftreten. Die zwei Module fpm_1 und fpm_2 in Abbildung 4.15 sind über drei PP-Verbindungen miteinander verbunden. Wie gefordert überschneiden sich die PP-Verbindungen nicht. Die Skizze erweckt den Anschein, als würde die innere, fein gestrichelt gezeichnete PP-Verbindung eine Wand im Inneren des Grundrisses ergeben. Dank des oben genannten Algorithmus kann dieses Problem erst gar nicht auftreten, da hierbei die fein gestrichelte PP-Verbindung nicht beachtet wird. Durch die Umlaufrichtung der Module wird eine Richtung für den gesamten Kantenzug des Umrisses definiert. Folglich wird (ausgehend von fpm_1) beim Erreichen der ersten PP-Verbindung gleich auf fpm_2 verzweigt. Ausgehend von fpm_2 gibt es – wegen der ersten Phase – keine Möglichkeit mehr die zweite, gestrichelte PP-Verbindung zu erreichen. In Abbildung 4.16 wird über die beiden Module fpm_1 , fpm_2 und die beiden PP-Verbindungen ein konkaves Polygon erzeugt, das nicht als PP-Modul aufgenommen werden kann.

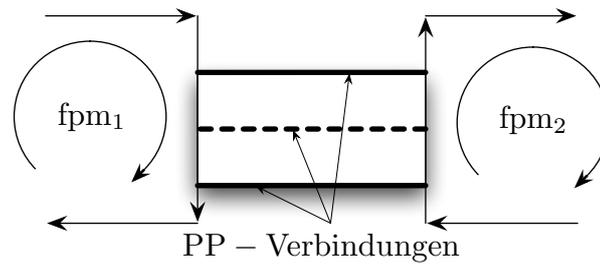


Abbildung 4.15: 1. Sonderfall bei PP-Verbindungen: Die gestrichelte PP-Verbindung verläuft innerhalb des Stockwerks.

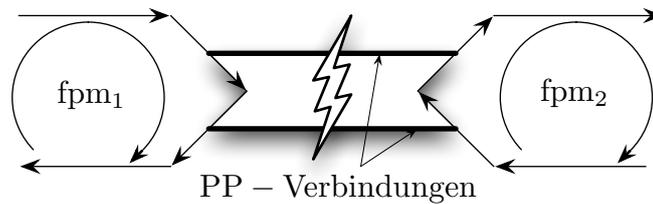


Abbildung 4.16: 2. Sonderfall bei PP-Verbindungen: Die PP-Verbindungen erzeugen zusammen mit den beiden Modulen ein konkaves Polygon, das nicht als PP-Modul verwendet werden kann.

4.3 Erweiterung auf Stockwerke

Nachdem die Grundlagen für die Beschreibung eines beliebigen Grundrisses für ein einzelnes Stockwerk definiert sind, gehen wir einen Schritt weiter und stellen in diesem Kapitel Methoden vor, die eine Erweiterung von Stockwerken zu ganzen Gebäuden erlauben.

4.3.1 Unterteilung

Bei der Erweiterung eines Stockwerks um ein weiteres Stockwerk wird von einem vollständigen Grundriss, bestehend aus Grundrissmodulen ausgegangen. Dieser Grundriss wird unterteilt, wobei eine grobe Unterteilung bereits durch seine Module gegeben ist. Die Module selbst werden in kleinere Polygone unterteilt.

Definition 4.7. Stockwerk

Die Module bzw. deren Unterteilungen bilden die Grundlage der Module des neuen Stockwerks.

Der Einfachheit halber wird im Folgenden der Unterteilungsprozess nur für ein Modul ($fpm = (V, E)$) beschrieben, da mit den weiteren Modulen analog verfahren wird. Um ein Modul zu unterteilen, werden im ersten Schritt seine Kanten

e_1, \dots, e_n beliebig oft unterteilt. Jede Kante e_i wird dabei n_i -mal unterteilt, wobei pro Kante e_i die neuen Punkte $\mathbf{q}_{i,j}$ mit $j = 1, \dots, n_i$ entstehen. Damit erhalten wir eine neue Menge von Punkten

$$V' = \{\mathbf{q}_{1,1}, \dots, \mathbf{q}_{1,n_1}, \dots, \mathbf{q}_{i,1}, \dots, \mathbf{q}_{i,n_i}, \dots, \mathbf{q}_{n,1}, \dots, \mathbf{q}_{n,n_n}\}.$$

Die Punkte der Menge $V' \cup V$ werden untereinander verbunden und es entsteht eine neue Menge von Kanten

$$E' = \{d_k = (\mathbf{r}, \mathbf{s}) \mid \mathbf{r}, \mathbf{s} \in V' \cup V, \mathbf{r} \neq \mathbf{s}, k = 1, \dots, o\}.$$

Dabei dürfen sich die Kanten d_1, \dots, d_o schneiden. Über die Schnittpunkte ergibt sich eine weitere Menge von Punkten

$$V'' = \{d_k \cap d_l \mid d_k, d_l \in E', d_k \neq d_l\}.$$

Im Folgenden sind nur die Teilpolygone von Interesse, die durch keine weitere Kante unterteilt werden. Diese konvexen Teilpolygone eines jeden Moduls, die beim Prozess der Unterteilung entstehen, bilden die Grundlage für neue Grundrissmodule des neuen Stockwerks.

Nachfolgender Abschnitt erläutert den Sachverhalt anhand eines vereinfachten Beispiels. Hierzu wird als Grundriss das Grundrissmodul fpm_1 aus Abschnitt 4.2.1 herangezogen. Dieses Modul wird nun in kleinere Polygone unterteilt. In Abbildung 4.17 ist das Modul zu sehen. Wie man erkennt, ist das ursprüngliche Polygon, gegeben durch die Punkte $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ in folgende vier Teilpolygone unterteilt:

1. Polygon: $\{\mathbf{p}_1, \mathbf{q}_3, \mathbf{q}_5, \mathbf{q}_2\}$,
2. Polygon: $\{\mathbf{q}_3, \mathbf{p}_5, \mathbf{q}_1, \mathbf{q}_5\}$,
3. Polygon: $\{\mathbf{q}_5, \mathbf{q}_1, \mathbf{p}_4, \mathbf{q}_4\}$,
4. Polygon: $\{\mathbf{q}_2, \mathbf{q}_5, \mathbf{q}_4, \mathbf{p}_3, \mathbf{p}_2\}$.

Wenn wir wieder alle Module des Grundrisses betrachten, so haben wir pro Modul eine Menge von konvexen Polygonen. Alle zulässigen Kombinationen aus diesen konvexen Polygonen können als Module für das nächste Stockwerk verwendet werden. Zulässige Kombinationen sind die, die wieder ein konvexes Polygon ergeben.

Soll z. B. ein Stockwerk im nachfolgenden Stockwerk wiederholt werden, so kann man die Unterteilung der einzelnen Module einfach entfallen lassen und die neuen Module des neuen Stockwerks gehen direkt aus den Modulen des alten Stockwerks hervor. Ebenso werden die Verbindungen auf Stockwerkebene des alten Grundrisses für die Verbindungen der Module des neuen Stockwerks verwendet. Der Vorteil

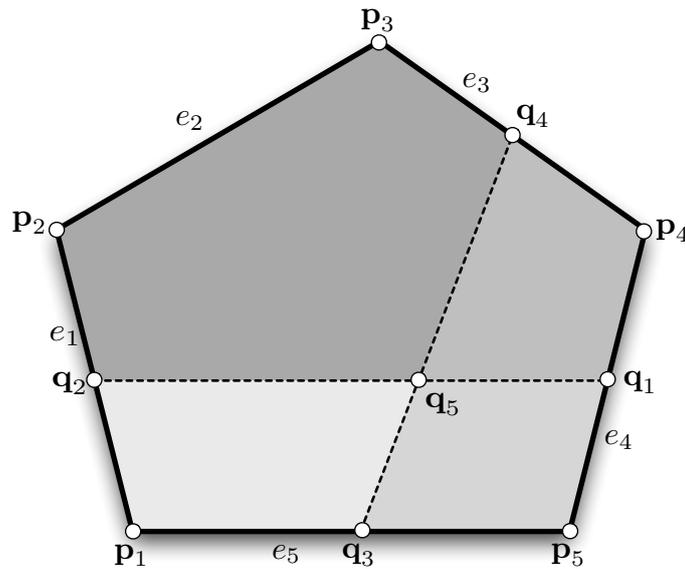


Abbildung 4.17: Unterteiltes Grundrissmodul: Die vier Teilpolygone sind grau hinterlegt. Ebenso sind die internen (gestrichelt) und externen (fett) Verbindungsmöglichkeiten gezeigt.

liegt klar auf der Hand: Die Merkmale des neuen Stockwerks sind die gleichen wie im unterliegenden Stockwerk und müssen somit nicht neu erzeugt werden.

Für das Beispiel ergeben sich mit der Unterteilung maximal neun mögliche neue Grundrissmodule. Eine Auswahl daraus kann als Modul für das neue Stockwerk herangezogen werden. Dabei darf aber jeweils nur eine Kombination aus Modulen genommen werden, die sich nicht gegenseitig überlappen. Für das Modul $fpm_{2,1}$ gilt:

$$\begin{aligned} fpm_{2,1} &= (V_{2,1} = \{p_1, q_3, q_5, q_2\}, \\ &E_{2,1} = \{(p_1, q_3), (q_3, q_5), (q_5, q_2), (q_2, p_1)\}) \end{aligned}$$

Es entspricht dem ersten Polygon. Für alle weiteren Module wird analog verfahren.

Die Module des neuen Stockwerks können, ebenso wie die Module des unteren Stockwerks, Verbindungen zu weiteren Modulen aufweisen. Dabei treten zwei Fälle auf:

1. Extern:

Die Verbindungen betreffen eine Kante d_k , die aus einer Kante e_l des ursprünglichen Moduls fpm_1 hervorgeht: $d_k = e_l[g, h]$. g und h sind Parameter, die die Kante e_l unterteilen. Alle Module, die über diese Kanten angebunden werden, liegen außerhalb des Polygons des ursprünglichen Moduls.

2. Intern:

Verbindungen, die über Kanten erfolgen, die im Inneren des ursprünglichen Polygons liegen. Über diese Kanten können Module, die durch die Unterteilung entstanden sind oder auch beliebige Polygone angebunden werden. Dabei ist darauf zu achten, dass sich die Module im Inneren nicht überlappen und nicht über das Polygon des unteren Stockwerks hinausragen.

In Abbildung 4.17 sind die Kanten der beiden Fälle nochmals grafisch veranschaulicht. Die Kanten für externe Verbindungen (1. Fall) sind fett und die Kanten für interne Verbindungen (2. Fall) sind gestrichelt gezeichnet.

Ein Beispiel für eine Gebäudestruktur, mit unterschiedlichen Grundrissen pro Stockwerk, ist in Abbildung 4.18 zu sehen. Die Module, aus denen jeder Grundriss eines Stockwerks aufgebaut ist, sind in unterschiedlichen Rottönen eingefärbt. Der erste Stock besitzt an seiner linken Seite einen Anbau, der nur in diesem Stockwerk zu finden ist.

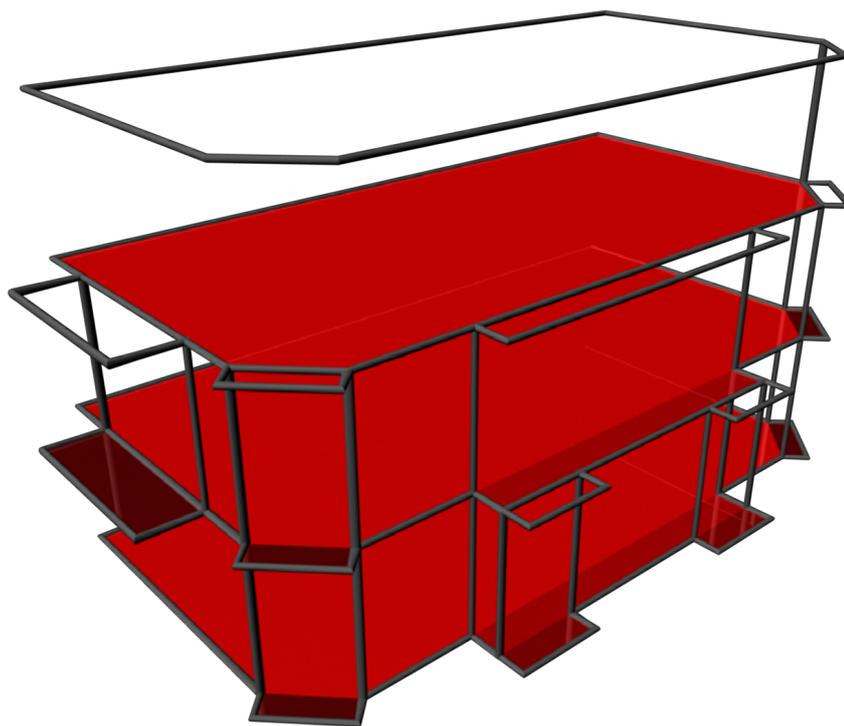


Abbildung 4.18: Gebäudestruktur, die aus verschiedenen Stockwerken aufgebaut ist.

4.3.2 Merkmale

Analog zu Abschnitt 4.2.5 – dort wurden Merkmale innerhalb eines Grundrisses bestimmt – werden auch zwischen den Stockwerken und auf Stockwerken Merkmale definiert. Im Gegensatz zu den Grundrissmerkmalen, werden bei den *Stockwerkmerkmalen* vier Klassen von Merkmalen unterschieden:

Definition 4.8. Freie Außenkanten

Freie Außenkanten verlaufen an der Außenseite eines Stockwerks und gehören freien Flächen an. Freie Flächen sind Umrisslinien auf einem Stockwerk, die nicht mit Modulen für das nächsthöhere Stockwerk belegt sind.

Abbildung 4.19 zeigt ein Beispiel für zwei Grundrissmodule in aufeinander folgenden Stockwerken. Man erkennt das ursprüngliche Polygon, gegeben durch die Punkte p_1, \dots, p_5 , und das Polygon des neuen Moduls fpm' mit den Punkten q_2, q_5, q_4, p_3, p_2 . Die zugehörige freie Fläche ist durch die Punkte $q_2, p_1, p_5, p_4, q_4, q_5$ gegeben. Damit ergibt sich als freie Außenkante q_2, p_1, p_5, p_4, q_4 (in der Abbildung fein gestrichelt dargestellt).

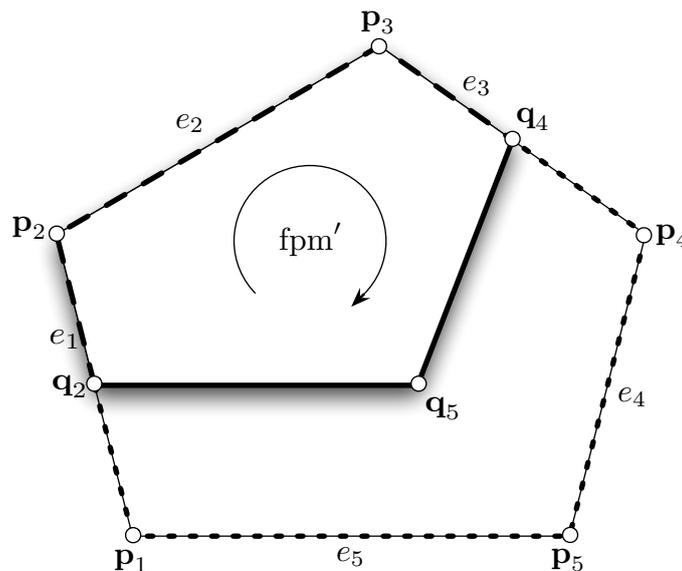


Abbildung 4.19: Drei Merkmale bei Stockwerken: freie Außenkanten (fein gestrichelt), Innenkanten (fett), externe Verbindungen (gestrichelt).

Definition 4.9. Innenkanten

Innenkanten verlaufen innerhalb eines Stockwerks und gehören freien Flächen an. Die Merkmale freie Außenkanten und Innenkanten ergeben zusammen genau die Umrisslinien der freien Flächen.

Abbildung 4.19 zeigt ein Beispiel für zwei Grundrissmodule in aufeinander folgenden Stockwerken. Das ursprüngliche Polygon ist durch die Punkte p_1, \dots, p_5

gegeben und das neue Polygon des neuen Moduls fpm' über die Punkte $\mathbf{q}_2, \mathbf{q}_5, \mathbf{q}_4, \mathbf{p}_3, \mathbf{p}_2$. Die verbundene Innenkante wird über die Punkte $\mathbf{q}_4, \mathbf{q}_5, \mathbf{q}_2$ beschrieben und ist in der Abbildung fett gezeichnet.

Definition 4.10. Externe Verbindungen

Externe Verbindungen sind Verbindungen zu weiteren Modulen, die über Außenkanten erfolgen.

In Abbildung 4.20 ist das externe Modul fpm'' über die Kante $(\mathbf{p}_1, \mathbf{q}_3)[g, h]$ mit dem internen Modul fpm' verbunden.

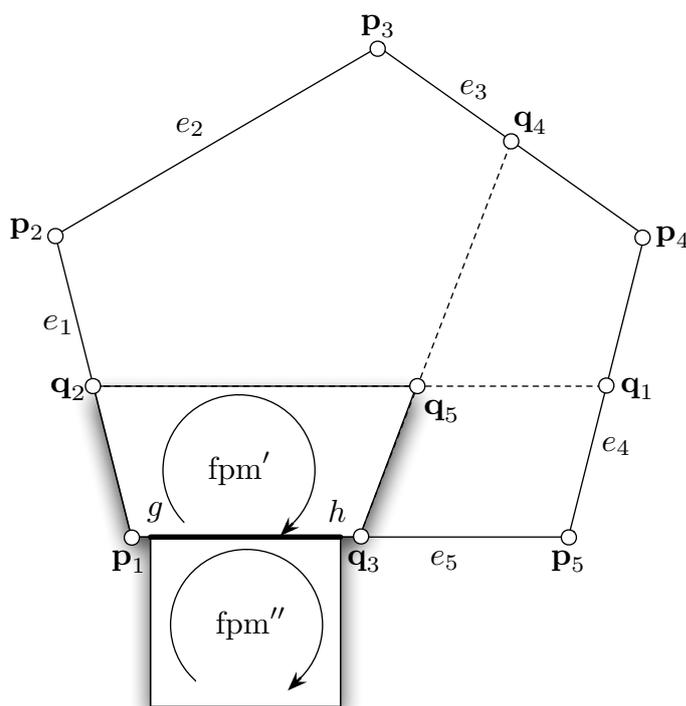


Abbildung 4.20: Merkmal: externe Verbindung.

Definition 4.11. Nahtstellen aufeinander liegender, aneinander grenzender Stockwerke

Nahtstellen aneinander grenzender Stockwerke sind die Außenkanten, an welchen zwei Stockwerke bzw. die zugehörigen Module aneinander stoßen.

Wie Abbildung 4.19 zeigt, ist das die Schnittmenge der Außenkanten der beiden Polygone $\mathbf{p}_1, \dots, \mathbf{p}_5$ (des unterliegenden Moduls) und $\mathbf{q}_2, \mathbf{q}_5, \mathbf{q}_4, \mathbf{p}_3, \mathbf{p}_2$ (des Moduls fpm' des neuen Stockwerks). Das Polygon ist in der Abbildung gestrichelt.

In Abbildung 4.21 sind die vier verschiedenen Merkmale in einem 3D-Modell wiedergegeben. Das Merkmal 1 *freie Außenkanten* ist gelb gekennzeichnet. Merk-

mal 2 *Innenkanten* ist rot, Merkmal 3 *externe Verbindungen* ist blau und Merkmal 4 *Nahtstellen* ist grün dargestellt.

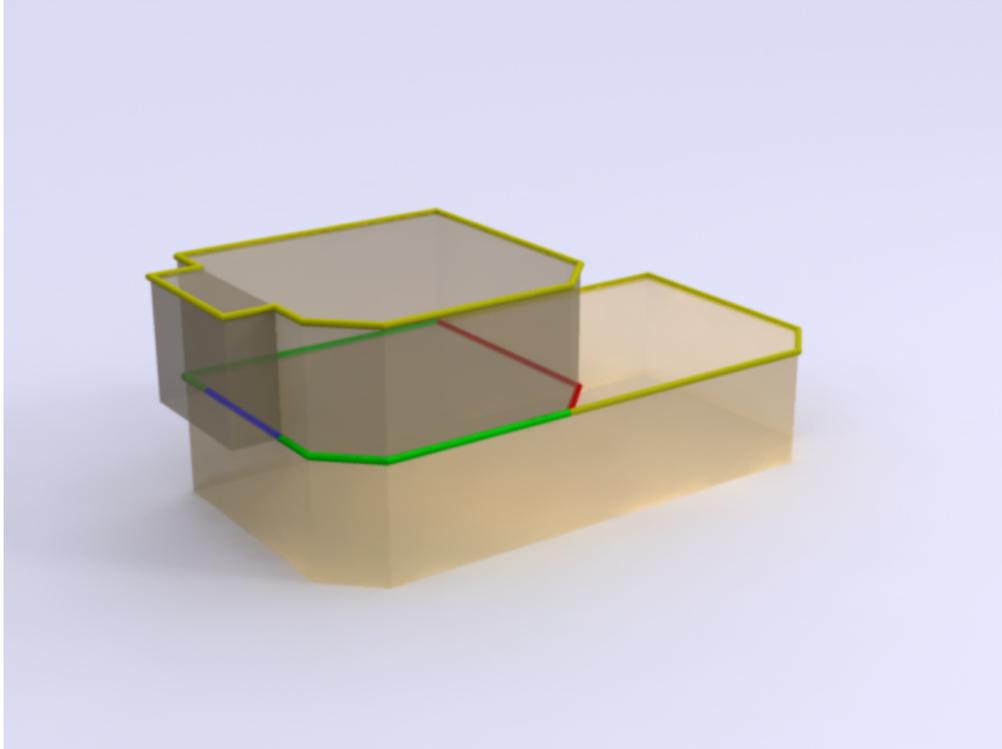


Abbildung 4.21: Merkmale bei Stockwerken.

4.4 Dachrepräsentation

Dieser Abschnitt widmet sich der Repräsentation von Dächern und deren Integration in die Beschreibung von Grundrissen und Stockwerken. Wie in Abschnitt 3.3 beschrieben, lassen sich die gängigen Methoden zur Erzeugung von Dachgeometrien in dieser Arbeit nicht sinnvoll einsetzen. Einerseits können Sattel- und Pultdächer mit frei wählbarem Dachfirst erzeugt werden (wie bei Laycock et al. [LD03] und Müller et al. [MWH⁺06]), wobei aber der Designer auf rechteckige Grundrisse beschränkt ist. Andererseits kann, wie bei der Technik von Felkel et al. [FO98] zwar der Grundriss beliebig sein, aber der Designer hat hierbei keinen Einfluss auf den Dachfirst. Ebenso arbeiten diese Techniken nur auf Polygonen. Außerdem berücksichtigen die Techniken die zugrunde liegende Struktur der Gebäudegrundrisse nur unzureichend.

4.4.1 Dachformen

In unserem Fall ist die automatische Generierung von Dächern auf Basis beliebiger Grundrisspolygone nicht sinnvoll. Dadurch, dass ein Grundrisspolygon aus einzelnen Grundrissmodulen aufgebaut ist, kann schon in einem frühen Stadium der Gebäudegenerierung die Struktur des Daches festgelegt werden. Dabei kommt uns die Grundrissbeschreibung eines Stockwerks (siehe Abschnitt 4.2) zu Hilfe. Jedes Grundrissmodul, das keine Fortsetzung im nächsten Stockwerk erfährt und deswegen als Dach in Frage kommt, erhält weitere Daten. Diese Daten werden je nach Dachtyp unterschieden. Ein Dach kann somit aus verschiedenen, aneinander grenzenden Modulen bestehen. Da über die Verbindungen der Module eine Ordnungsrelation gegeben ist, können die angrenzenden Dächer automatisch aneinander angepasst werden. Andere an die Dächer angrenzende Strukturen (z. B. Wände) müssen ebenfalls angebunden werden. Im Fall der Dächer sind diese Strukturen entweder Module oder Innenkanten der Stockwerkmerkmale. Deshalb kann jeder Dachtyp Schnittstellen besitzen, über die er an angrenzende Module angekoppelt wird. Allgemein sind Schnittstellen bei Dächern Flächen, die auf der Umrisslinie des Moduls senkrecht stehen.

Im Folgenden wird erläutert, welche Dachtypen unterstützt werden, welche Daten sie besitzen, wie sich ihr First berechnet und wie sie aneinander angepasst werden. Zur besseren Veranschaulichung ist zu jedem Dachtyp ein Beispiel angegeben.

Von den von Neufert in [NN96] und von Koch in [Koc90] aufgelistet Dächern, werden von diesem Verfahren die Typen Flach-, Walm-, Pult-, Sattel- und Zelt Dach ermöglicht. Außerdem lassen sich durch die vorgestellte Technik die Formen Walm-, Sattel-, Zelt- und Pultdach als Spezialfall einer allgemeinen Dachform darstellen.

Definition 4.12. Flachdach

Ein Flachdach für ein Modul fpm erhält an seinen Stockwerkmerkmalen „Nahtstellen“, die keine Verbindungen zu weiteren Modulen aufweisen, eine kleine Mauer mit definierte Höhe h und Breite w :

$$\text{flat_roof} = (fpm, h, b)$$

In Abbildung 4.22 ist ein Beispiel für ein Flachdach angegeben. Die Schnittstelle des Flachdaches in dieser Abbildung ist die hintere Kante, an der die Attika (kleine Mauer) fehlt. Wie bereits erwähnt, können an Schnittstellen angrenzende Strukturen entweder andere Module oder Innenkanten von Stockwerkmerkmalen sein.

Definition 4.13. Allgemeines Dach

Ein allgemeines Dach wird über das 8-Tupel

$$\text{common_roof} = (fpm, e_i, e_j, c, d, a, b, h)$$

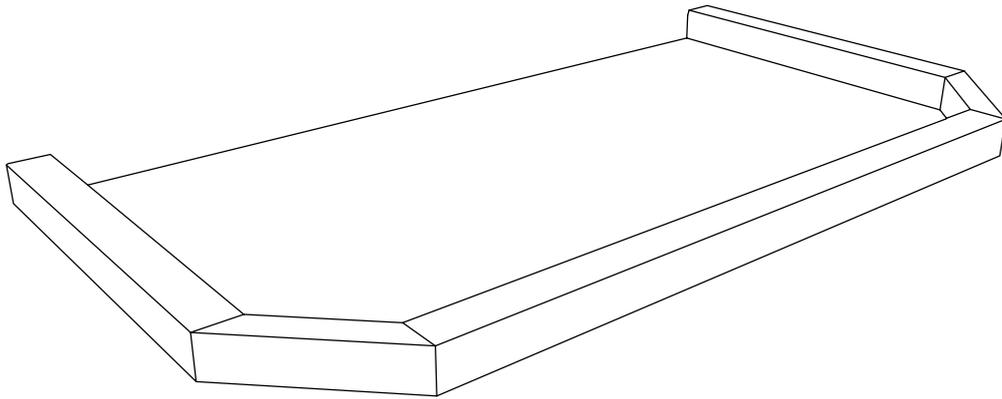


Abbildung 4.22: Flachdach.

definiert. Es kann an jeder Seite Dachschrägen besitzen, die vom Umriss eines Moduls ausgehen und sich nach oben hin in einer Kante, dem First, treffen (siehe Abbildung 4.23). Der First $(\mathbf{r}_a, \mathbf{r}_b)$ wird dabei wie folgt bestimmt: Es wird auf zwei Kanten e_i und e_j des Umrisspolygons jeweils ein Punkt \mathbf{q}_{e_i} bzw. \mathbf{q}_{e_j} bestimmt, mit $\mathbf{q}_{e_i} = (1 - c) \cdot \mathbf{p}_i + c \cdot \mathbf{p}_{i+1}$ und $\mathbf{q}_{e_j} = (1 - d) \cdot \mathbf{p}_j + d \cdot \mathbf{p}_{j+1}$. Über die beiden Punkte wird eine neue Kante definiert, die zusätzlich über die beiden Skalare a bzw. b parametrisiert wird. h definiert schließlich die Höhe des Daches. Damit ergibt sich $\mathbf{r}_a = a(\mathbf{q}_{e_j} - \mathbf{q}_{e_i}) + \mathbf{e}_y h$ und $\mathbf{r}_b = b(\mathbf{q}_{e_j} - \mathbf{q}_{e_i}) + \mathbf{e}_y h$.

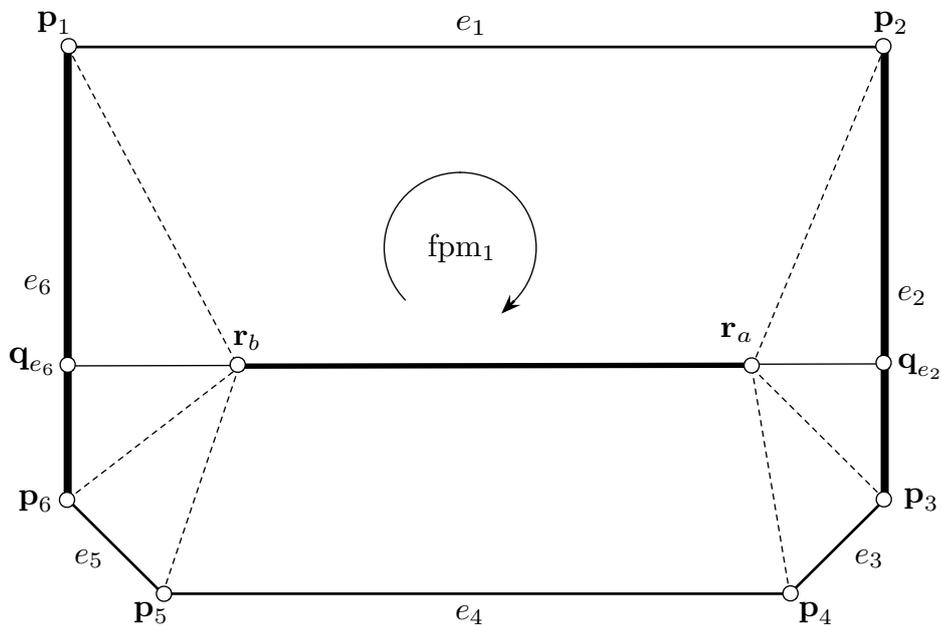


Abbildung 4.23: Schematische Darstellung der allgemeinen Dachform.

In Abbildung 4.23 ist ein Beispiel für ein allgemeines Dach angegeben. Das Po-

lygon des Grundrissmoduls besteht aus $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6)$. Für die Erzeugung des Firstes werden die Kanten e_2 und e_6 unterteilt und es entstehen die Punkte \mathbf{q}_{e_2} und \mathbf{q}_{e_6} . Über die beiden Parameter a und b wird schließlich auf Höhe h der First $(\mathbf{r}_a, \mathbf{r}_b)$ bestimmt. Je nachdem wie die Parameter c, d und a, b gewählt werden, existieren Schnittstellen zu weiteren Modulen.

Im Folgenden werden nun die Varianten Walm-, Sattel-, Zelt- und Pultdach beschrieben, wobei stets das selbe Grundrisspolygon verwendet wird.

Das **Walmdach** besitzt zu allen Seiten Dachschrägen die im First zusammenlaufen. Dabei ist zu beachten, dass für die Parameter a und b gilt: $0 < a < b < 1$. Das Beispiel in Abbildung 4.24 zeigt ein Walmdach. Da ein Walmdach keine senkrechten Dachelemente beinhaltet, existieren auch keine Schnittstellen.

Beim **Satteldach** werden die Parameter $a = 0$ und $b = 1$ gewählt. Abbildung 4.25 zeigt ein Satteldach. Das Satteldach besitzt an den beiden Giebeln $(\mathbf{p}_6, \mathbf{p}_1, \mathbf{r}_b)$ und $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{r}_a)$ Schnittstellen zu weiteren Modulen.

Beim **Pultdach** sitzt der First genau über einer Kante des Umrisspolygons. Üblicherweise werden $a = 0$ und $b = 1$ gewählt. Weiterhin muss gelten: $c = 0$, $d = 1$ und für $e_i, e_j, i = j$. In Abbildung 4.26 ist für bekanntes Umrisspolygon ein Pultdach zu sehen. Das Pultdach besitzt genau eine Schnittstelle bei der Kante e_1 mit dem Polygon $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{r}_a, \mathbf{r}_b)$.

Beim **Zeltdach** fällt der First zu einem einzigen Punkt zusammen. Damit gilt, dass $a = b$. Ein Zeltdach ist in Abbildung 4.27 dargestellt. Da beim Zeltdach, wie auch beim Walmdach, keine senkrechten Dachelemente existieren, gibt es keine Schnittstellen.

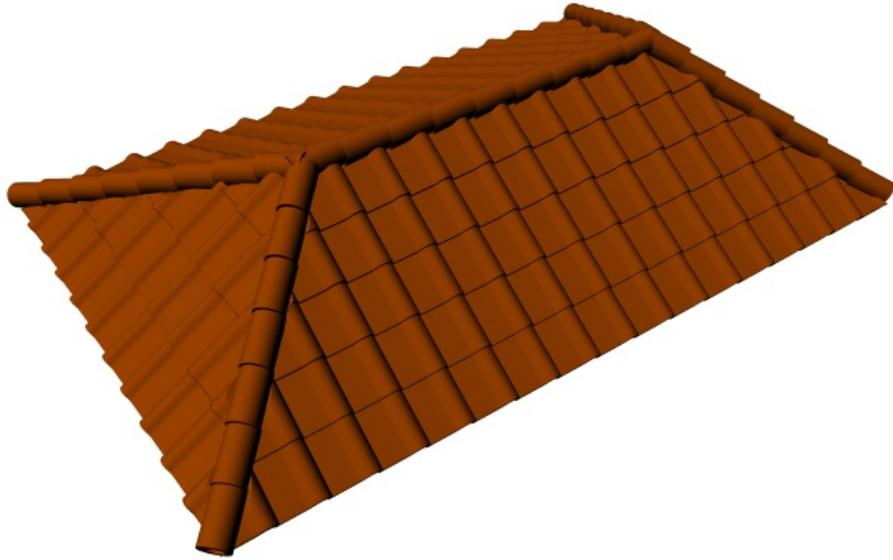


Abbildung 4.24: Walmdach.

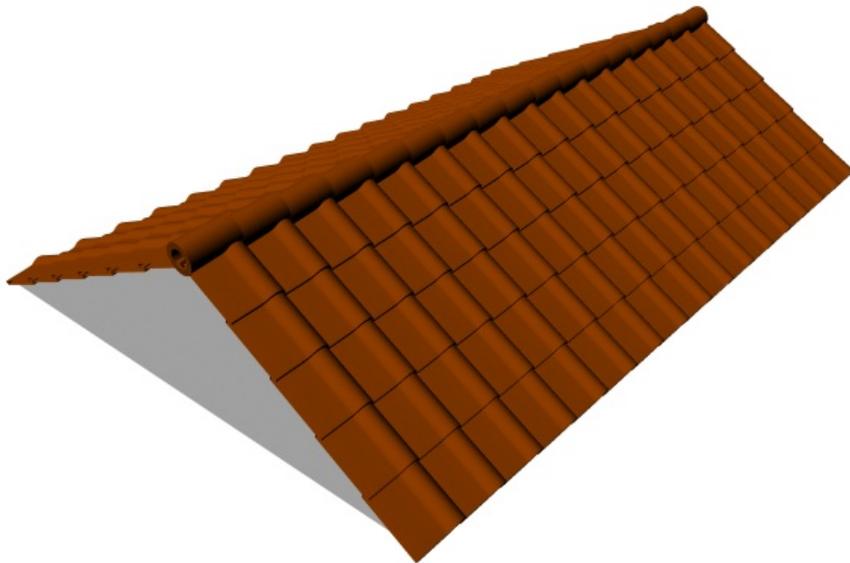


Abbildung 4.25: Satteldach.

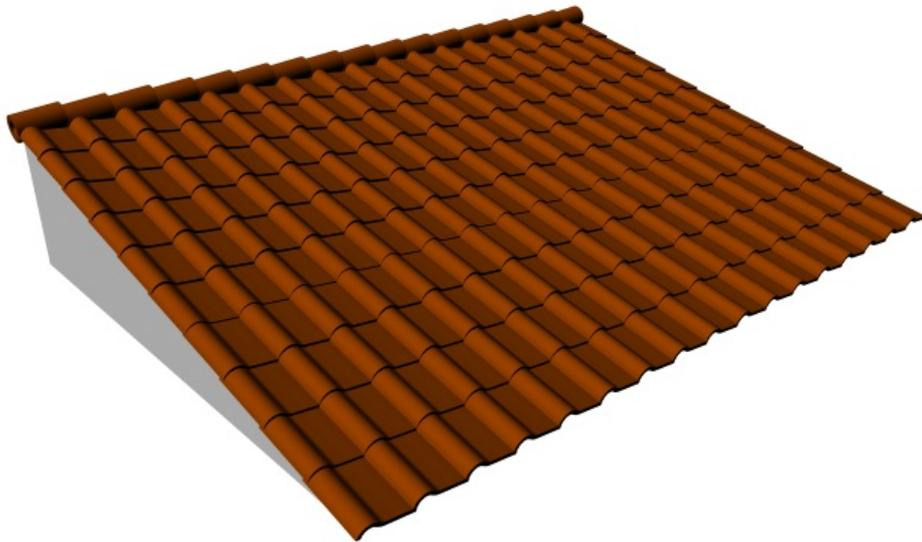


Abbildung 4.26: Pultdach.

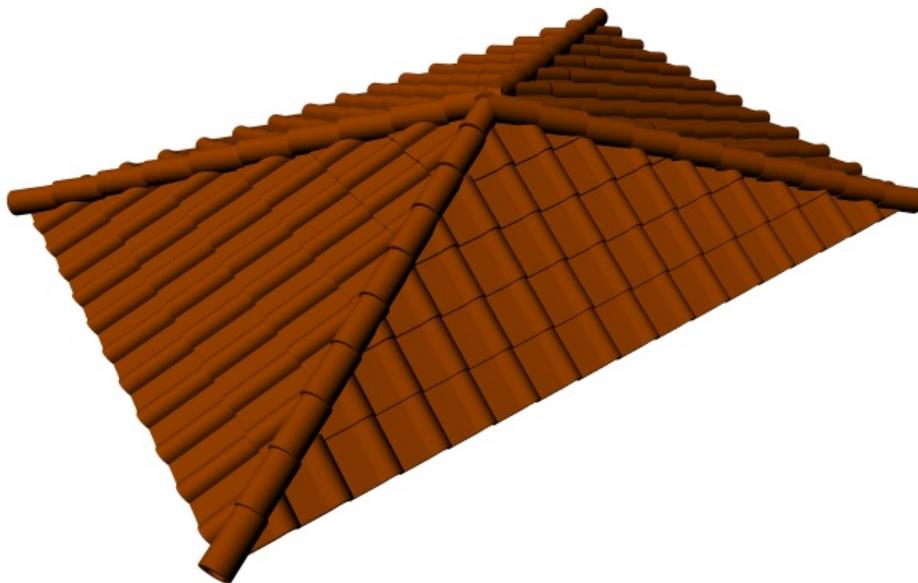


Abbildung 4.27: Zeltdach.

4.4.2 Kombination von Dächern

In diesem Abschnitt wird diskutiert, wie einzelne Dächer zu einem Dach verschmolzen werden.

Voraussetzung für die Verschmelzung einzelner Dächer ist zum einen, dass ihre zugehörigen Grundrissmodule miteinander verbunden sind. Zum anderen muss wenigstens ein Dach an der Verbindung eine Schnittstelle besitzen. Andernfalls können die einzelnen Dächer unabhängig von einander erzeugt werden. Formal bedeutet das:

Gegeben sind:

- eine Verbindungskante $k = (\mathbf{r}_{a'}, \mathbf{r}_{b'})$ (KK- oder PP-Modul-Verbindung),
- die zwei beteiligten Module fpm_i und fpm_j mit $i \neq j$,
- die beiden Firste $(\mathbf{r}_{a_i}, \mathbf{r}_{b_i})$ und $(\mathbf{r}_{a_j}, \mathbf{r}_{b_j})$.

Folgende drei Fälle können nun auftreten:

- $(\mathbf{r}_{a_i} \in k \vee \mathbf{r}_{b_i} \in k) \wedge (\mathbf{r}_{a_j} \in k \vee \mathbf{r}_{b_j} \in k)$

Die Firste beider Module schneiden die Kante k . In diesem Fall müssen die Dächer nicht angepasst werden.

- $(\mathbf{r}_{a_i} \notin k \wedge \mathbf{r}_{b_i} \notin k) \wedge (\mathbf{r}_{a_j} \notin k \wedge \mathbf{r}_{b_j} \notin k)$

Keiner der beiden Firste schneidet die Kante k . Auch in diesem Fall müssen die Dächer nicht angepasst werden.

- $((\mathbf{r}_{a_i} \in k \vee \mathbf{r}_{b_i} \in k) \wedge (\mathbf{r}_{a_j} \in k \wedge \mathbf{r}_{b_j} \notin k)) \vee$
 $((\mathbf{r}_{a_i} \notin k \wedge \mathbf{r}_{b_i} \notin k) \wedge (\mathbf{r}_{a_j} \notin k \vee \mathbf{r}_{b_j} \in k))$

Ein First schneidet die Kante k . Das Modul, dessen First die Kante schneidet, wird für eine Anpassung vorgemerkt.

Ist ein Dach für eine Anpassung vorgemerkt, so wird überprüft, ob der verlängerte Dachfirst des anzupassenden Daches das anliegende Dach schneidet. Wird das Dach geschnitten, so wird der First bzw. das gesamte Dach bis zum Schnittpunkt verlängert. Wenn das Dach nicht geschnitten wird, erfolgt keine Anpassung.

In Abbildung 4.28 sind zwei Module mit ihren zugehörigen Dächern zu sehen. Der First ist jeweils fett gezeichnet, wohingegen die Verbindung gestrichelt dargestellt ist. An den Punkten, an denen nun ein Dachfirst eine Verbindung schneidet, befindet sich eine Dachschnittstelle und das zugehörige Dach wird ggf. angepasst. Im Beispiel wird Dach₂ an Dach₁ angepasst. Abbildung 4.28 zeigt schematisch die ursprünglichen Dächer der beiden Module. Das Ergebnis der Verschmelzung der zwei Dächer ist in Abbildung 4.29 dargestellt. Wie man sehen kann, wurde der First von Dach₂ verlängert, womit beide Dächer zusammen eine Einheit bilden.

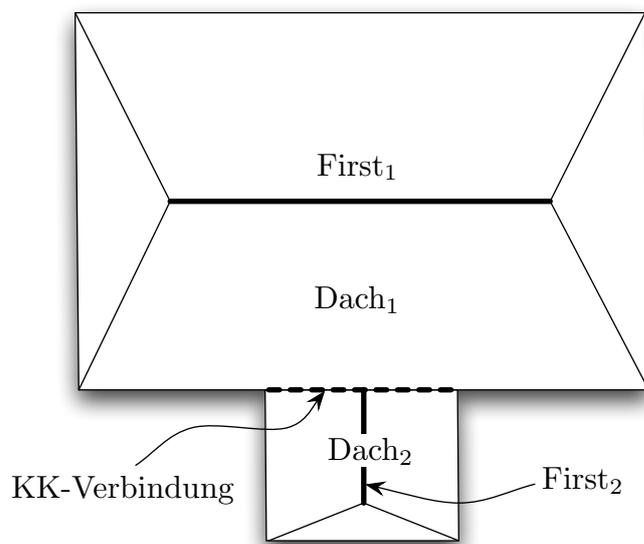


Abbildung 4.28: Zwei Grundrissmodule mit Dächern (schematische Darstellung).



Abbildung 4.29: Die zwei Grundrissmodule mit angepassten Dächern.

Kapitel 5

Fassadenbeschreibung

Im vorangegangenen Kapitel wurden Methoden zur Beschreibung der groben Gebäudestruktur vorgestellt. In diesem Kapitel werden Verfahren beschrieben, wie die einzelnen groben Strukturen sukzessive mit immer detailreicheren Fassadenelementen angereichert werden.

In den folgenden Abschnitten werden Verfahren zur Beschreibung und Erstellung verschiedener architektonischer Strukturen wie Wände, Hausecken, Gesimse (Sockel-, Brüstungs- und Hauptgesimse), Fenster und Türen besprochen.

Eine wesentliche Anforderung an die Beschreibung bzw. Gliederung der Fassaden ist, dass die Beziehung zwischen aneinander grenzenden Strukturen und Strukturen, die ineinander enthalten sind, abgebildet werden können. Das heißt die Strukturen besitzen selber Kenntnis darüber, mit welchen anderen Strukturen sie in Beziehung stehen. Die Berechnung der exakten Geometrie der einzelnen Strukturen in deren Abhängigkeit zueinander ist Thema des Kapitels 7, der Geometrie-Fabrik.

5.1 Gebäudegrundstruktur

Zuerst wird mit den Methoden aus Kapitel 4 die grobe Gebäudestruktur festgelegt. Auf dieser Ebene erfolgt bereits eine Unterteilung in architektonische Elemente, wie Mauervorsprünge (Risalite), Erker und Balkone. Die gesamte Gebäudestruktur eines Beispiels ist in Abbildung 4.18 zu sehen. Die schmalen Rechtecke nach vorne und zu den Seiten repräsentieren Mauervorsprünge. An der linken Wand ist mittig ein Erker angebracht.

Im Weiteren wird aufgezeigt, wie eine grobe Gebäudestruktur durch einzelne Verfeinerungsschritte in feine Strukturen aufgebrochen wird, die später detailierte architektonische Elemente aufnehmen. Es wird aber hauptsächlich auf die

dabei entstehende hierarchische Struktur eingegangen. Wie derartige Strukturen automatisch bzw. halbautomatisch erzeugt werden und wie sich daraus konkrete Geometrie gewinnen lässt, ist Gegenstand der Kapitel 6 und 7.

Dieses Kapitel ist eng mit den beiden anschließenden Kapiteln verknüpft. Basierend auf der Beschreibung der Repräsentation der Fassaden werden im nächsten Kapitel das semantische Modell der groben Gebäudestruktur und der Fassadenstile definiert. Beide Kapitel spielen für die Geometrie-Fabrik eine entscheidende Rolle.

5.2 Basiswände und Hausecken

Basiswände und Hausecken entstehen aus Punkten und Strecken der Kantenzüge der Merkmale eines Grundrisses. Allgemein hat der Umriss für ein Stockwerk folgende Gestalt: $G = \{F_1, \dots, F_n\}$ sei ein Grundriss, bestehend aus den Merkmalen F_1, \dots, F_n . Wie in Abschnitt 4.2.5 beschrieben, repräsentieren die Merkmale Basiswände und Ecken bereits architektonische Strukturen. In Abbildung 5.1 sind für das Beispiel die groben architektonischen Strukturen in Form verschiedenfarbiger Kantenzüge für jedes Stockwerk gezeigt. Hier sind bereits aneinander grenzende Strukturen, innerhalb eines Stockwerks und stockwerkübergreifend, farbig markiert. Die blauen, violetten und türkisen Kanten zeigen Nahtstellen innerhalb eines Stockwerks an. Rote bzw. grüne Kanten beschreiben als stockwerkübergreifende Nahtstellen den Boden bzw. die Decke eines Stockwerks.

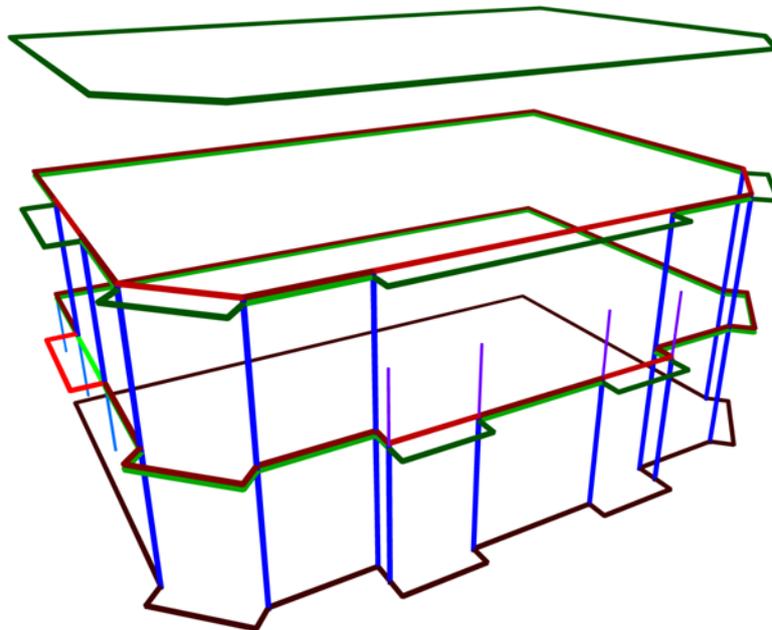


Abbildung 5.1: Nahtstellen der groben Gebäudestrukturen.

Definition 5.1. Basiswände

Basiswände entstehen direkt aus den Kanten der Merkmale. Sei $F_i = f_1, f_2, \dots, f_{m-2}, f_{m-1}$ ein Merkmal mit $F_i \in G$, dann ist jede Kante $f_j \in F_i$ eine Basiswand.

Hausecken werden nur an konvexen Punkten des Umrisspolygons erzeugt. Da auf Ebene der Merkmale begonnen wird und diese gemäß der Definition von Grundrissen konvex sind, gehen die Hausecken direkt aus den inneren Punkten der Merkmale hervor.

Definition 5.2. Hausecken

Sei $F_i = f_1, f_2, \dots, f_{m-2}, f_{m-1}$ ein Merkmal mit $f_j = (\mathbf{p}_j, \mathbf{p}_{j+1})$ für $j = 1, \dots, m-1$, dann befinden sich Hausecken an den Punkten $\mathbf{p}_2, \dots, \mathbf{p}_{m-1}$.

An den Punkten \mathbf{p}_1 und \mathbf{p}_m muss überprüft werden, ob über die anschließenden Merkmale eine konvexe oder konkave Ecke beschrieben wird. Eine konkave Ecke bedarf keiner Behandlung. Für eine konvexe Ecke wird eine Hausecke erzeugt.

5.2.1 Wände und Zwischenwandelemente

Basiswände werden weiter in Wände und ggf. Zwischenwandelemente (kurz ZWE) unterteilt. Die Breite der Teilwände und Zwischenwandelemente kann frei gewählt werden. Die Anzahl der Zwischenwandelemente ist stets um eins kleiner als die Anzahl Teilwände. Allgemein wird für die Unterteilung einer Basiswand nicht zwischen Wand- und Zwischenwandelementen unterschieden, sondern sie werden zu Teilkanten zusammengefasst.

Definition 5.3. Teilkanten

Teilkanten (Wände und ggf. Zwischenwandelemente) entstehen durch Unterteilung der Basiswand $f = (\mathbf{q}_1, \mathbf{q}_2)$. Dabei entsteht die Menge

$$T = \{f[a_1, a_2], f[a_2, a_3], \dots, f[a_{n-1}, a_n]\}$$

mit $0 = a_1 < a_2 < \dots < a_n = 1$.

Sind Wand- wie auch Zwischenwandelemente gewünscht, so gliedert sich die Menge der Teilkanten T in zwei Mengen auf, nämlich die der Wandelemente $W = \{f[a_1, a_2], f[a_3, a_4], \dots, f[a_{n-1}, a_n]\}$ und die der Zwischenwandelemente $Z = \{f[a_2, a_3], f[a_4, a_5], \dots, f[a_{n-2}, a_{n-1}]\}$.

5.2.2 Wand-, Zwischenwand- und Eckpartitionen

Wand-, Zwischenwandelemente, wie auch Ecken, können vertikal in sog. Partitionen unterteilt werden. Diese Partitionen dienen dazu, Gesimse oder spezielle Texturen aufzunehmen. Werden alle Wände, Zwischenwandelemente und Ecken eines

Stockwerks auf gleicher Höhe unterteilt, kann damit eine die Fassade übergreifende horizontale Struktur (wie z. B. ein Gesims) angebracht werden.

Definition 5.4. Partitionen

Partitionen unterteilen Wand-, Zwischenwandelemente und Hausecken horizontal. Sei h eine beliebige Höhe einer der eben genannten Elemente. Die Menge der Partitionen P hat dann folgende Gestalt:

$$P = \{P_1, \dots, P_n\} \text{ mit } P_i = \{h \cdot a_i, h \cdot a_{i+1}\}$$

für $i = 1, \dots, n$ und $0 = a_1 < a_2 < \dots < a_{n+1} = 1$.

5.3 Mauerwerke

Bisher haben wir uns ausschließlich mit der Beschreibung der groben Wände beschäftigt. Feinere Strukturen, wie Mauerwerke, sollen aber auch für Wände zur Verfügung stehen. Nach Koch [Koc90] ist ein Mauerwerk eine Konstruktion aus natürlichen Steinen entweder mit oder ohne Bindemittel¹ aus Lehm bzw. Kalk oder Trass (Mörtelmauerwerk). Wie wir später im Kapitel 7, der Geometrie-Fabrik, sehen werden, ist es nicht sinnvoll, jeden einzelnen Stein als Geometrie zu erzeugen, da sonst die Anzahl Dreiecke drastisch in die Höhe schnellen würde. Deshalb werden Texturen für Mauerziegelverbände und Zyklopenmauerwerke verwendet.

Üblich sind vorgefertigte Texturen, die einfach auf die Geometrie aufgebracht werden. Meist handelt es um kachelbare Texturen, die nahtlos aneinander gefügt werden können. In Abbildung 5.2 ist ein Bildschirmfoto des Online-Rollenspiels *World of Warcraft* [BE04] zu sehen. Die gewöhnliche Texturierung ist zwar einfach handhabbar, birgt aber einige gravierende Nachteile. Wie in der Abbildung gut zu sehen ist, erfolgt keine Anpassung an die darunter liegenden Geometrie. Die Nahtstelle zwischen den Steinen der Wand und dem hölzernen Balkon wirkt aufgesetzt, unnatürlich. Aufgrund des enormen Speicherbedarfs können nur begrenzt viele Texturen verwendet werden, weswegen Texturen wiederholt eingesetzt werden müssen. Ebenso liegen die Texturen nur in festen Auflösungen vor. Eine dynamische Anpassung zur Laufzeit ist damit nur begrenzt möglich.

Die folgenden Verfahren lösen das Problem hinsichtlich Mauerwerkstexturen. In unserem Fall beschäftigen wir uns rein mit der Texturierung der Wandflächen. Hausecken werden als herausragende Merkmale separat modelliert, weswegen sie nicht in die Texturierung, wie bei der Arbeit von Legakis et al. [LDG01], miteinbezogen werden. Die Texturen der Mauerwerke passen sich entsprechend an die hineinragenden Elemente an. Aus einer einfachen textuellen Beschreibung werden verschiedene Typen von Mauerwerken erzeugt. Dabei besitzen die Verfahren die

¹Dann handelt es sich um ein Trockenmauerwerk.



Abbildung 5.2: Bildschirmfoto aus dem Online-Spiel *World of Warcraft* [BE04]. Die Nahtstelle zwischen dem Balkon und der Wand ist klar zu erkennen. Die Mauertextur hat keinen Bezug zum Balkon.

Eigenschaft, dass sie für die Steine eine automatische Anpassung an die zu texturierende Wand zulassen. Aus der allgemeinen Beschreibung für das Mauerwerk – unter Einbezug der darunter liegenden Geometrie – wird schließlich für jede Mauer eine Datei im RenderMan-Format erzeugt. Damit lassen sich die Texturen der Mauerwerke in beliebigen Auflösungen berechnen. Das bedeutet:

- Eine enorme Verringerung des Speicherbedarfs.
- Texturen sind exakt in die Geometrie der Wand eingepasst.
- Es sind vielfältige Texturen möglich, wobei keine der anderen gleicht (bzw. gleichen muss).
- Für Echtzeitanwendungen kann die Auflösung der Textur in Abhängigkeit vom Betrachter gewählt werden.

In diesem Abschnitt werden zwei Techniken vorgestellt, mit denen sich Mauerziegelverbände und Zyklopenmauerwerke erzeugen lassen.

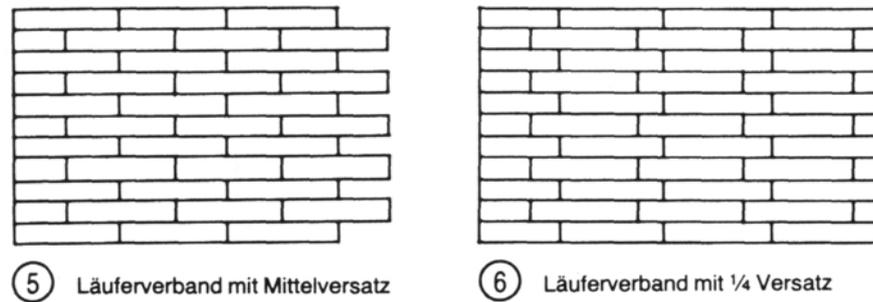


Abbildung 5.3: Beispiele für Läuferverbände [NN96].

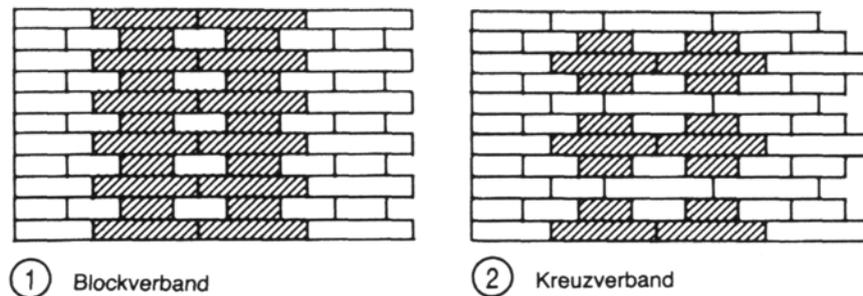


Abbildung 5.4: Beispiele für Kreuz- und Blockverbände [NN96].

5.3.1 Mauerziegelverbände

Ein Mauerziegelverband beschreibt eine feste Anordnung von Steinen durch Versatz der Stoßfugen, um eine höhere Stabilität der Mauer zu gewährleisten. Dabei wird die Mauer schichtweise aufgebaut, wobei die Steine einer Schicht im Regelfall die selbe Höhe haben. Die Breite kann durchaus variieren, wobei sich das Muster der aufeinander folgenden Steine einer Schicht stets wiederholt. Ebenso folgen die übereinander liegenden Schichten einem Muster, so dass in einer höheren Schicht exakt die gleiche Steinreihe einer früheren Schicht wiederholt wird.

Über einen festen, immer wiederkehrenden Versatz der Steine lassen sich verschiedene Muster im Mauerwerk erzeugen. Hier sind die beiden Formen Läufer- und Kreuzverband zu unterscheiden. Bei einem Läuferverband wird eine feste Steingröße verwendet. Ausschlaggebend ist hier die Größe des Versatzes der Stoßkanten von Schicht zu Schicht. In Abbildung 5.3 sind ein Läuferverband mit Mittelversatz² und ein Läuferverband mit 1/4 links und rechts steigendem Versatz zu sehen. Beim Kreuzverband oder Blockverband wechseln sich zusätzlich noch die Steingrößen innerhalb einer Schicht ab, womit sich komplizierte Muster erzeugen lassen. Beispiele für Kreuz- und Blockverbände sind in Abbildung 5.4 zu sehen. Weiterhin ist es noch möglich, abwechselnd verschieden hohe Steine pro Schicht zu verwenden.

²Mittelversatz bezeichnet den Versatz der Stoßkante um eine halbe Steinslänge.

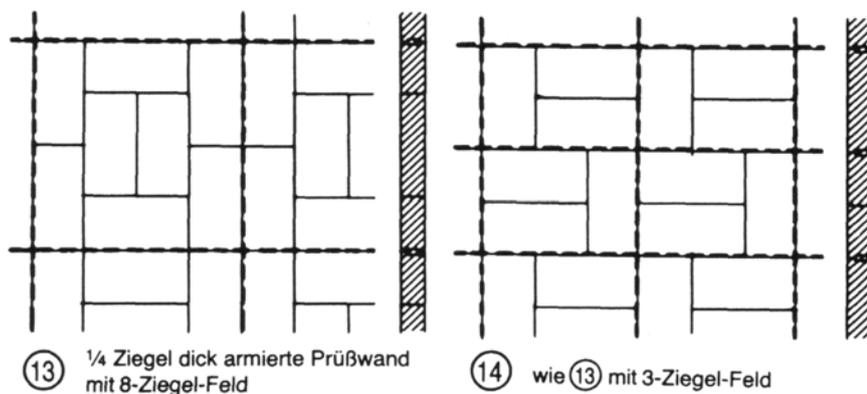


Abbildung 5.5: Beispiele für armierte Mauerwerke [NN96].

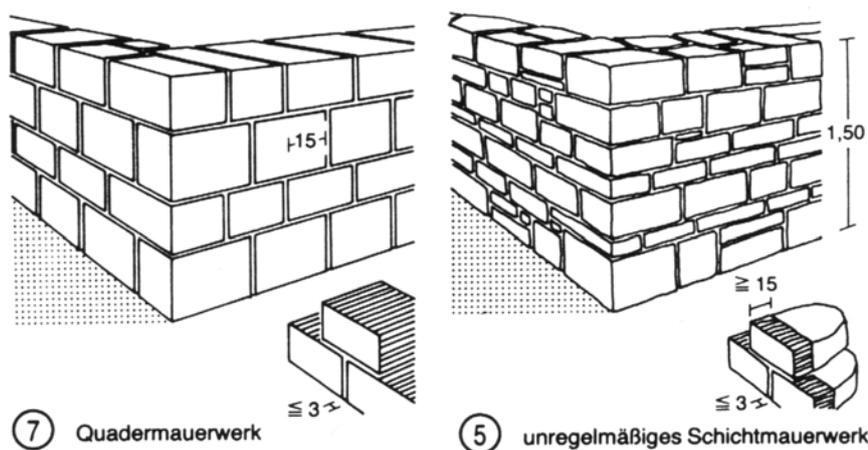


Abbildung 5.6: Beispiele für Quaderstein- und unregelmäßige Schichtmauerwerke (aus [NN96]).

den. Dabei setzen sich die Höhen der Steine aus den Höhen der Schichten (ggf. mit Mörtelschicht) zusammen, die sie umspannen. In [Abbildung 5.5](#) sind zwei armierte Mauerwerke dargestellt. Quadersteinmauerwerke (siehe [Abbildung 5.6](#)) zeichnen sich dadurch aus, dass zwar in einer Schicht die Höhe gleich bleibt, aber die Steinbreite zufällig variiert. Ein unregelmäßiges Schichtmauerwerk wird aus verschiedenen großen Quadersteinen aufgebaut, wobei die regelmäßige Anordnung der Steine aufgebrochen wird.

Im Folgenden werden einfache Beschreibungsformen vorgestellt, mit denen die erwähnten Mauerwerke erzeugt werden können. Dabei wird zwischen regelmäßigen Mauerziegelverbänden und Quadersteinmauerwerken unterschieden.

Die Beschreibung eines Mauerziegelverbands gliedert sich in drei Stufen. Erst werden auf unterster Stufe die Mauerziegel beschrieben. Die Definitionen für alle Beschreibungen werden in einer EBNF (EBNF für XML [[W3C04a](#)]) gegeben:

Definition 5.5. Mauerziegel
$$\begin{aligned}
\text{mauerziegel} &::= \mathbf{Brick}\{ \text{groesse } \text{startfarbe } \text{endfarbe} \} \\
\text{groesse} &::= \text{laenge, hoehe, tiefe}; \\
\text{startfarbe} &::= \text{rot, gruen, blau}; \\
\text{endfarbe} &::= \text{rot, gruen, blau}; \\
\text{laenge} &::= \text{zahl} \\
\text{hoehe} &::= \text{zahl} \\
\text{tiefe} &::= \text{zahl} \\
\text{rot} &::= \text{ganzzahl} \\
\text{gruen} &::= \text{ganzzahl} \\
\text{blau} &::= \text{ganzzahl} \\
\text{zahl} &::= \text{ziffer}^+ [.]\text{ziffer}^* \\
\text{ganzzahl} &::= \text{ziffer}^+ \\
\text{ziffer} &::= 0-9
\end{aligned}$$

Jeder Ziegelstein wird durch seine Größe und seinen Farbbereich beschrieben. Der Parameter *tiefe* beschreibt in diesem Fall den Überstand des Ziegels gegenüber dem Mörtel bzw. der Wand. Die Farbe des Ziegels wird zufällig aus dem über *startfarbe* und *endfarbe* definierten Bereich bestimmt. Dabei liegen die Werte für *rot*, *gruen* und *blau* zwischen 0 und 255.

Als Nächstes wird das Muster des Mauerwerks durch den Verband verschiedener Ziegel definiert. Läufer-, Block und Kreuzverbände werden über eine einheitliche Beschreibung repräsentiert.

Definition 5.6. Verband
$$\begin{aligned}
\text{verband} &::= \mathbf{Pattern}\{ (\text{versatz}, (\text{mauerziegel} \mid \text{luecke}) \\
&\quad (, (\text{mauerziegel} \mid \text{luecke}))^* ;)^+ \} \\
\text{versatz} &::= [-]\text{zahl} \\
\text{luecke} &::= \text{zahl} \\
\text{zahl} &::= \text{ziffer}^+ [.]\text{ziffer}^* \\
\text{ziffer} &::= 0-9
\end{aligned}$$

Jede Schicht wird über ihren Versatz und eine Folge von Ziegeln beschrieben. Der Versatz gibt an, um welche Distanz der Anfang der Ziegelreihe nach links bzw. rechts verschoben beginnt. Dabei wird der Versatz relativ zur vorigen Ziegelschicht vorgenommen. Eine Ausnahme stellt die erste Schicht dar, da sie sich auf sich selbst bezieht. Jede Ziegelfolge besteht aus Ziegeln oder Platzhaltern bzw. Lücken (*luecke*). Über die Platzhalter werden armierte Ziegelverbände ermöglicht,

indem geeignet platzierte höhere Ziegel aus der unteren Schicht in die Platzhalter ragen. In einem Verband können beliebig viele Schichten von Ziegelfolgen definiert werden. Bei der Erzeugung der Textur werden die Schichten von oben nach unten abgelaufen und in umgekehrter Reihenfolge als Textur erzeugt. Wurden alle Schichten abgearbeitet, so wird die Folge der Schichten wiederholt, bis die Wand vollständig mit Ziegeln gefüllt ist.

Als letzter Schritt werden die Beschreibungen für Ziegel und Verbände zum Mauerziegelverband zusammengefasst und mit Parametern zu Mörtel Eigenschaften versehen.

Definition 5.7. Mauerziegelverband

```
mauerziegelverband ::= BrickMasonry{ verband moerteldicke moertelfarbe }
moerteldicke ::= zahl
moertelfarbe ::= rot, gruen, blau;
    rot ::= ganzzahl
    gruen ::= ganzzahl
    blau ::= ganzzahl
    zahl ::= ziffer+ [. ]ziffer*
ganzzahl ::= ziffer+
ziffer ::= 0-9
```

Über *moerteldicke* wird der Abstand zwischen den Ziegeln angegeben. Die Farbe des Mörtels ist über *moertelfarbe* festgelegt. Der in Abbildung 5.7 dargestellte Mauerziegelverband hat folgende Gestalt:

```
Brick1 { 2.0, 0.5, 0.1;           # Länge, Höhe, Tiefe
        206,78,51;              # Farbe: r,g,b
        255,147,124; }
Brick2 { 2.0, 1.0, 0.1;
        206,78,51;
        255,147,124; }
Pattern { -1.0, "Brick1", "Brick2"; # Versatz, Ziegel, Ziegel
         0.0, "Brick1", 2.0;       # Versatz, Ziegel, Lücke
         1.0, "Brick1"; }         # Versatz, Ziegel
BrickMasonry { "Pattern";
              0.05;                # Mörteldicke
              102,102,102; }
```

Damit sind alle Schritte erbracht, um Mauerziegelverbände darstellen zu können. Die Beschreibung von Quadersteinmauerwerken wird in einem einzigen Schritt durchgeführt und ist wie folgt definiert:

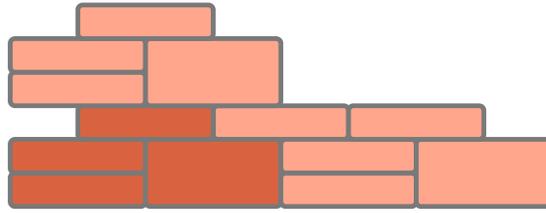


Abbildung 5.7: Beispiel für einen Mauerziegelverband. Das zugrunde liegende Muster ist in dunkelrot gehalten.

Definition 5.8. Quadersteinmauerwerk

```

quadersteinmauerwerk ::= AshlarMasonry{ min_groesse max_groesse
                                     startfarbe endfarbe moerteldicke moertelfarbe }
min_groesse ::= laenge, hoehe, tiefe;
max_groesse ::= laenge, hoehe, tiefe;
startfarbe ::= rot, gruen, blau;
endfarbe ::= rot, gruen, blau;
moerteldicke ::= zahl
moertelfarbe ::= rot, gruen, blau;
laenge ::= zahl
hoehe ::= zahl
tiefe ::= zahl
rot ::= ganzzahl
gruen ::= ganzzahl
blau ::= ganzzahl
zahl ::= ziffer+[.]ziffer*
ganzzahl ::= ziffer+
ziffer ::= 0-9
  
```

Nachdem die Grundlagen für die Beschreibung der verschiedenen Mauerwerke definiert sind, widmen wir uns der passgenauen Einfügung in Wände. Die Wände liegen in Form einfacher Polygone vor, die wiederum einfache Polygone als Löcher (Fenster- oder Türöffnungen) aufweisen dürfen. Die Berechnung der exakten Wandpolygone ist Gegenstand des Kapitels 7. Wir gehen im Weiteren davon aus, dass die exakten Polygone bereits vorliegen. Die grundlegende Technik für die Erzeugung der Texturen basiert auf der Arbeit von Legakis et al. [LDG01]. Hier wird ein Grauwertbild (siehe Abbildung 5.8) geringer Auflösung verwendet, das die aktuelle Belegung der Wand widerspiegelt. Dem Grauwertbild fällt für Mauerziegelverbände eine leicht andere Bedeutung zu als für Quadersteinmauerwerke.

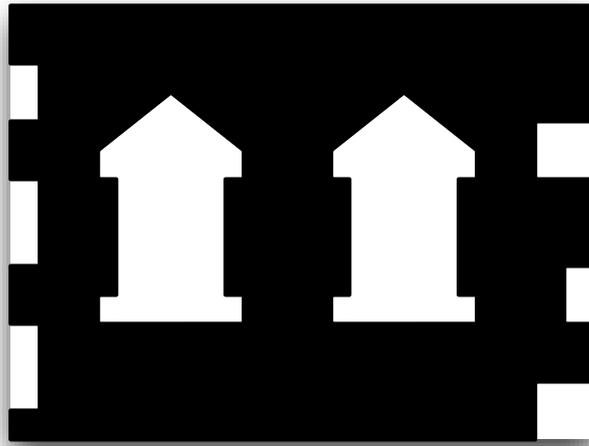
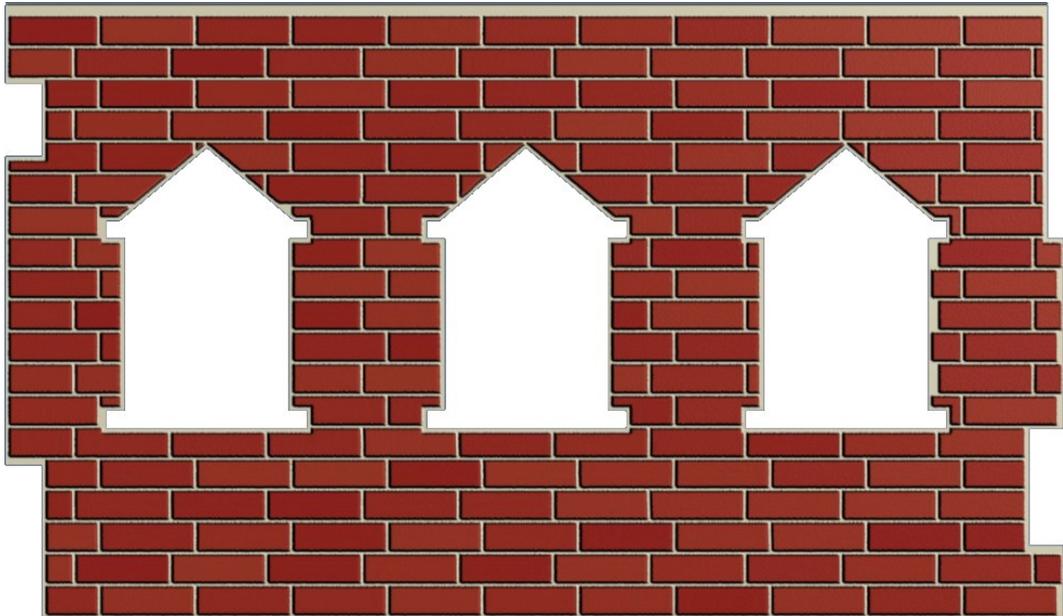


Abbildung 5.8: Beispiel für ein Grauwertbild, das ein Wandpolygon mit Fensteröffnungen zeigt.

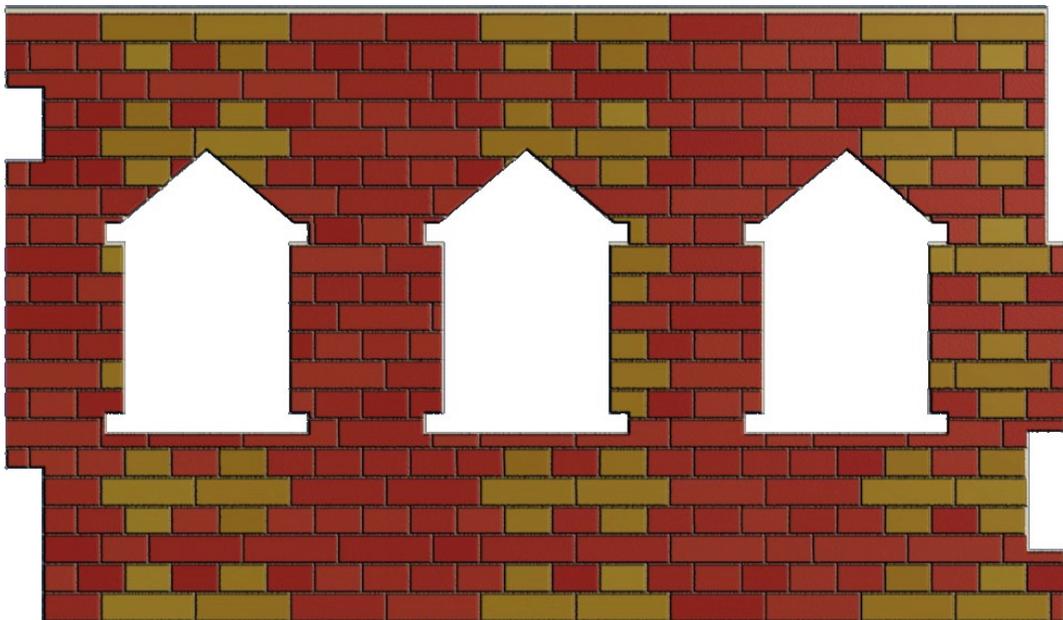
Für Mauerziegelverbände dient das Grauwertbild rein der Schnittberechnung zwischen Ziegeln und Wandpolygon. Im ersten Schritt wird das achsenparallele Hüllrechteck des Wandpolygons berechnet. Anschließend werden anhand der angegebenen Mörteldicke Parallelkurven für Wandpolygon und Löcher erzeugt. Aus Hüllrechteck und Parallelkurven wird schließlich das Grauwertbild generiert. Weiße Bereiche dürfen dabei von den Ziegeln nicht belegt werden. Nun werden die schwarzen Bereiche mit dem Ziegelmuster gefüllt, wobei die Ziegel über das Grauwertbild mit den weißen Bereichen wie folgt geschnitten werden: Erst wird der mittlere Farbwert der vom Ziegel überlagerten Pixel bestimmt. Anhand seines Farbwerts, also wie weit der Farbwert von weiß entfernt ist, kann entschieden werden, ob der Ziegel entweder vollständig gezeichnet (komplett schwarz) oder gar nicht gezeichnet (nahe bei weiß oder vollständig weiß) oder ob er zur exakten Schnittberechnung markiert wird. Wurde das gesamte Grauwertbild mit Ziegeln gefüllt, so werden die zum Schnitt markierten Ziegel mit den Wand- bzw. Lochpolygonen unter Berücksichtigung der Mörteldicke geschnitten. Anschließend werden die Farbe der Ziegel und des Mörtels festgelegt. Der Mauerziegelverband einschließlich Wand werden als Polygone ausgegeben. Das hat den Vorteil, dass die Mauerwerkstexturen in beliebiger Auflösung berechnet werden können. In den Abbildungen 5.9 und 5.10 sind Beispiele für Mauerziegelverbände dargestellt.

Bei Quadersteinmauerwerken werden über das Grauwertbild freie Bereiche gesucht, in denen Quadersteine platziert werden können.³ Im Unterschied zu den Mauerziegelverbänden werden die Steine hier schon beim Platzieren in die zu texturierende Wandfläche eingepasst. Die Größe des Steins wird über lineare Interpo-

³Diese Vorgehensweise ist sehr stark an die Arbeit von Legakis et al. [LDG01] angelehnt.

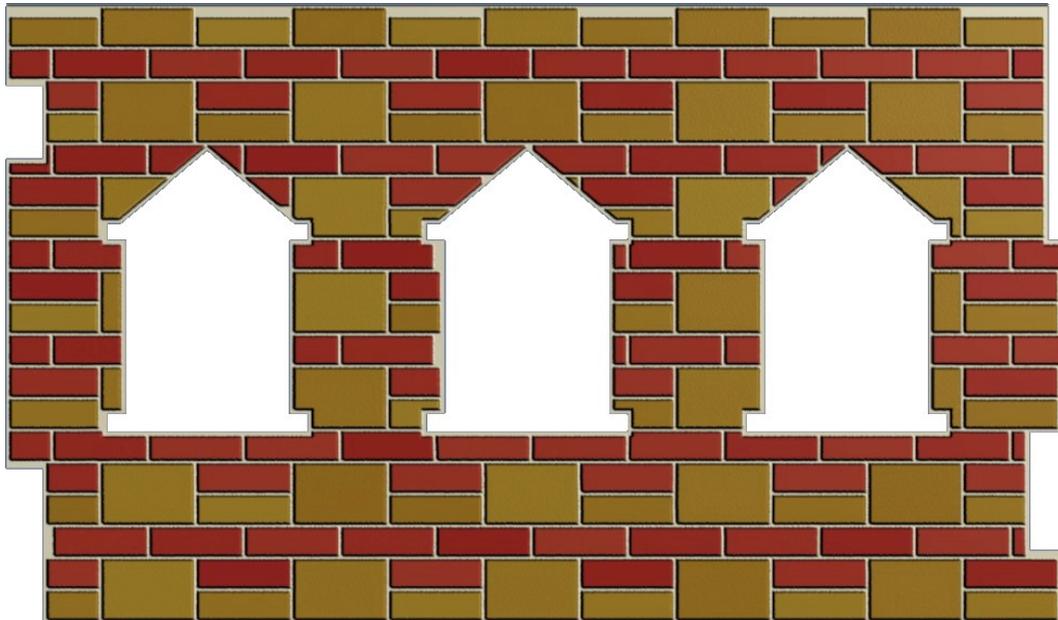


(a) Läuferverband mit $1/4$ links- und rechtssteigendem Versatz.

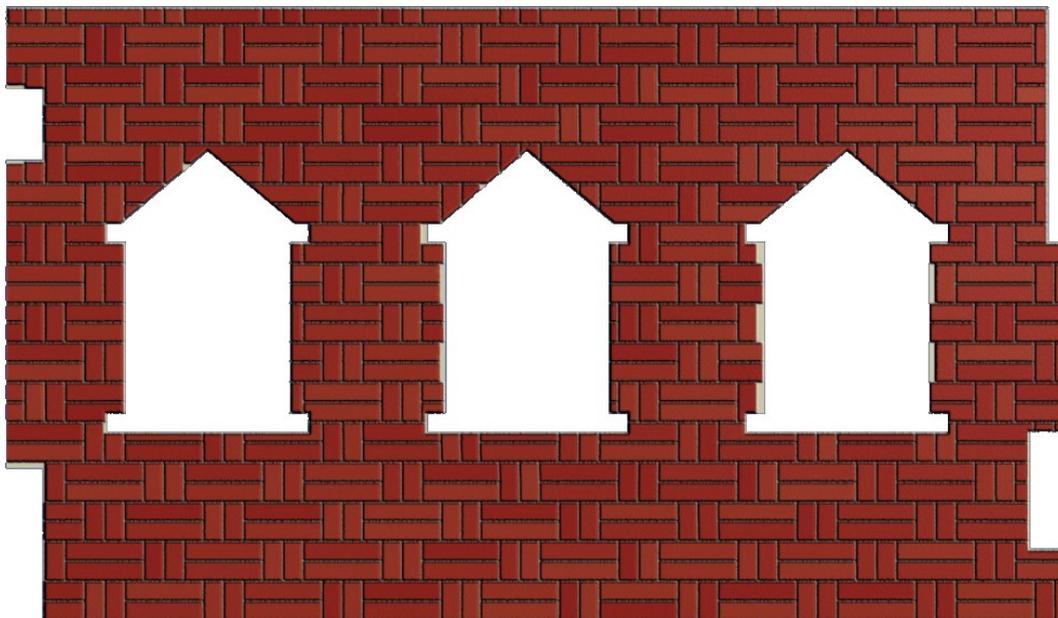


(b) Kreuzverband.

Abbildung 5.9: Verschiedene einfache Mauerziegelverbände.



(a) Armierter Mauerziegelverband.



(b) Ornamental wirkender Mauerziegelverband.

Abbildung 5.10: Verschiedene überlappende Mauerziegelverbände.

lation anhand eines Zufallswertes zwischen den minimalen und maximalen Größen ermittelt. Passt der Stein in den freien Bereich, so wird der Vorgang wiederholt, bis selbst der kleinstmögliche Stein nicht mehr passt. Verbleibende Bereiche werden mit Quadersteinen gefüllt, wobei sich die Steine mit den Polygonen überlappen dürfen. Anschließend werden diese Quadersteine mit den Polygonen geschnitten. Sollten konkave Steine entstehen, so werden diese in konvexe unterteilt. Die neu entstandenen Steine werden nur dann verwendet, wenn sie eine gewisse Mindestgröße nicht unterschreiten. In Abbildung 5.11(a) ist beim Schnitt ein konkaver Stein entstanden, der wie in Abbildung 5.11(b) zu sehen ist, in drei konvexe Steine unterteilt wurde, wobei der schmale Streifen links im Bild, weil zu klein, entfernt wurde.

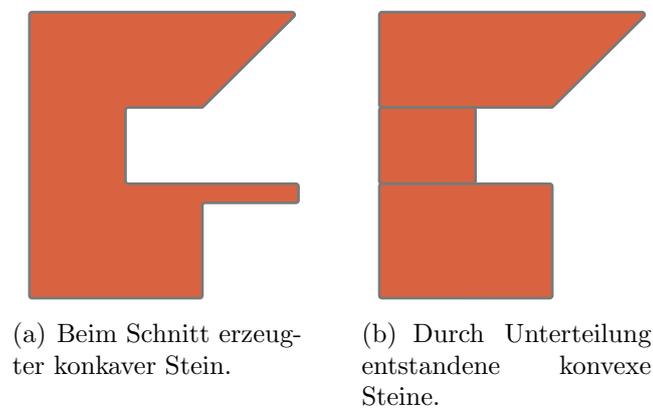
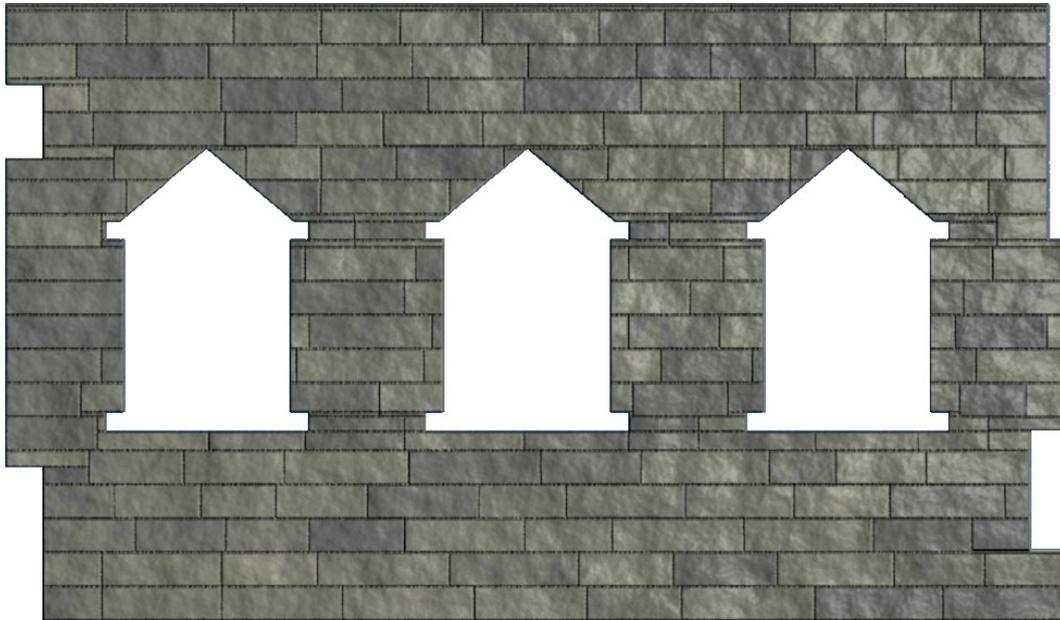


Abbildung 5.11: Quaderstein, der am Randgebiet eines Wandpolygons konkav geschnitten und anschließend in drei konvexe Steine zerlegt wurde. Der fehlende schmale Streifen links ist zu klein und wird weggelassen.

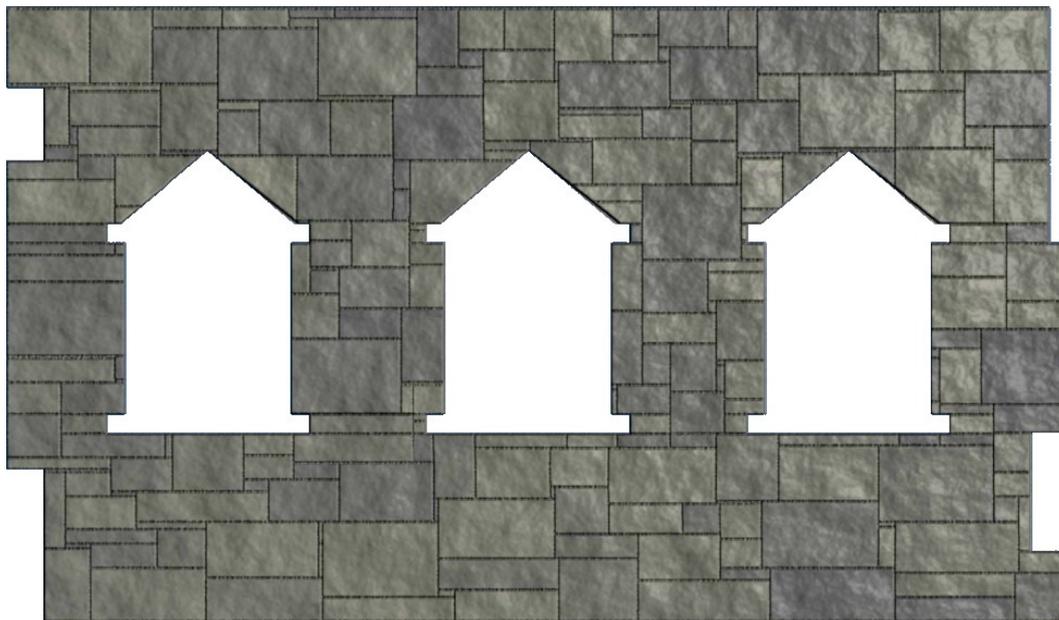
Beispiele zu Quaderstein- und unregelmäßigen Schichtmauerwerken sind in Abbildung 5.12 dargestellt. Die Beschreibung des Quadersteinmauerwerks ist wie folgt:

```
AshlarMasonry {
  0.5, 0.5, 0.05; # min. Steingröße
  1.5, 0.5, 0.1; # max. Steingröße
  80, 80, 80; # Startfarbe der Steine
  100, 100, 90; # Endfarbe der Steine
  0.05; # Mörteldicke
  110, 110, 110; # Mörtelfarbe }
```

Wie wir in Abschnitt 5.4.2 sehen werden, bilden Grauwertbilder die Grundlage bei der Erzeugung von Ornamenten.



(a) Quadersteinmauerwerk. Die Steine einer Reihe besitzen alle die gleiche Höhe.



(b) Unregelmäßiges Schichtmauerwerk.

Abbildung 5.12: Beispiele für Quaderstein- und unregelmäßige Schichtmauerwerke.

5.3.2 Zyklopenmauerwerke

Bei Zyklopenmauerwerken handelt es sich, wie bei Schicht- und Quadermauerwerken, auch um Mauern aus natürlichem Gestein (siehe Neufert [NN96]). Insbesondere ist es eine Sonderform des Bruchsteinmauerwerkes, bestehend aus großen, unregelmäßigen konvexen Steinen. Die Fugen verlaufen unregelmäßig, mit variabler Dicke ohne waagrechte Lagerfugen. In Abbildung 5.13 ist ein Beispiel für ein Zyklopenmauerwerk gezeigt, das als Grundlage für das vorgestellte Verfahren herangezogen wurde.

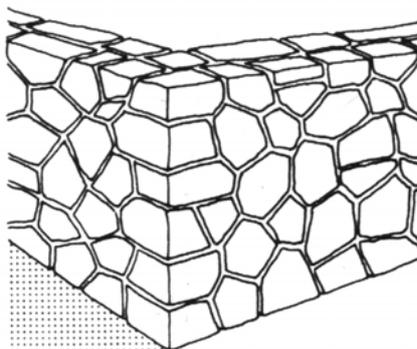


Abbildung 5.13: Beispiel für ein Zyklopenmauerwerk [NN96].

Wie auch die Texturen für Mauerziegelverbände, werden die Texturen für Zyklopenmauerwerke in Wandflächen eingefügt, wobei auch sie sich an die Grenzen der Fläche anpassen. Wie Abbildung 5.13 vermuten lässt, eignen sich Voronoi-Regionen zur Erzeugung der Zyklopenmauerwerke.

Voronoi-Regionen gehen auf Georgy Fedoseevich Voronoi zurück, ein ukrainischer Mathematiker, der um 1907 folgende Dirichletsche Problemstellung von 1850 wieder aufgegriffen hat (nach *Numerical recipes* [PTVF07]): für N Punkte in einer Ebene beschreibt jeder Punkt $\mathbf{p} \in N$ einen Bereich (oder Region), der dichter an \mathbf{p} liegt als an jedem anderen Punkt aus $N - 1$. Bildlich gesprochen wären das z. B. die Einzugsbereiche der Supermärkte einer Stadt, wenn jeder Einwohner in seinem nächstgelegenen Supermarkt einkaufen ginge. Ein Beispiel eines Voronoi-Diagramms ist in Abbildung 5.14 zu sehen.

Die Herausforderung besteht in diesem Fall aber nicht in der Berechnung der Voronoi-Regionen, sondern in der geeigneten Platzierung der Punkte. Ganz allgemein muss „nur“ die Anforderung erfüllt werden, dass die Wandfläche gleichmäßig mit ausschließlich konvexen Steinen gefüllt wird. Dazu sind zwei Probleme zu lösen. Zum einen müssen die Punkte im Wandpolygon gleichmäßig verteilt werden und zum anderen müssen die Punkte an kritischen Regionen des Polygons (bzw. der Polygone, da die Öffnungen für Türen wie auch Fenster zu berücksichtigen sind)

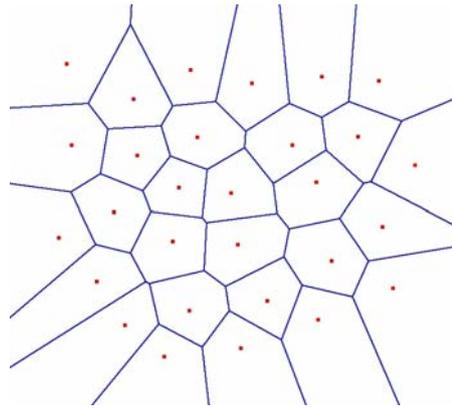


Abbildung 5.14: Beispiel für ein Voronoi-Diagramm.

derart gesetzt werden, dass die erzeugten Voronoi-Regionen – geschnitten mit den Wand- und Öffnungspolygonen – nur konvexe Regionen ergeben.

Vorgegeben wird als mittlere Größe der Steine der Radius r und die zu füllende rechteckige Wandfläche mit den Kantenlängen a und b . Betrachtet man nun die Steine als gleich große Kreisscheiben, so berechnet sich die Anzahl n der Steine, bei optimaler Packdichte, wie folgt:

In horizontaler Richtung werden zwei Schichtungen unterschieden:

$$\begin{aligned} \text{in der 1. Schicht : } n_1 &= \left\lfloor \frac{a}{2r} \right\rfloor, \\ \text{in der 2. Schicht : } n_2 &= \left\lfloor \frac{a-r}{2r} \right\rfloor \in \{n_1, n_1 - 1\}. \end{aligned}$$

Die Anzahl Steine bzw. Kreise in vertikaler Richtung ist

$$n_v = \left\lfloor \frac{b-2r}{\sqrt{3}r} \right\rfloor + 1.$$

Daraus ergibt sich die Anzahl der Steine zu

$$n = \left\lceil \frac{n_v}{2} \right\rceil n_1 + \left\lfloor \frac{n_v}{2} \right\rfloor n_2.$$

Da für die Verteilung der Steine nicht von einer optimalen Packdichte ausgegangen wird, wird n um etwa 20% reduziert. Die Verteilung lässt sich nun mit verschiedenen Methoden des Supersamplings⁴ bewerkstelligen. Dazu zählen u. A. folgende Techniken:

⁴Supersampling wird in der Computergraphik u. A. zur Reduzierung von *Treppeneffekten* bei der Bilderzeugung verwendet.

- **Zufälliges Verteilen**

Dabei werden die Punkte über Pseudozufallszahlen ermittelt. Diese Technik brachte keinen Erfolg, da sehr häufig einerseits eine Ballung von Punkten und andererseits große freie Areale auftraten.

- **Geordnetes Raster**

Die Punkte werden in einem festen Raster angeordnet. Dabei entstehen sehr künstlich wirkende Voronoi-Regionen, die je nach Raster sogar zu Quadraten degenerieren.

- **Geordnetes Raster mit Streuung (Jittering)**

Die Punkte werden wie beim geordneten Raster gesetzt, aber zusätzlich um einen Zufallswert, der innerhalb der Rastergrenzen liegt, verschoben. Auch diese Technik brachte nicht den gewünschten Erfolg.

- **Poisson-Disk-Verteilung**

Hier werden die Punkte auch zufällig in der Ebene verteilt, wobei zwischen den Punkten ein gewisser Mindestabstand eingehalten werden muss. Zur Verteilung werden üblicherweise der *Dart-Throwing*- oder der erweiterte *Dart-Throwing*-Algorithmus verwendet. Diese Technik zeigte die besten Ergebnisse, ist aber sehr rechenintensiv.

In unserem Fall erreichen wir die Verteilung der Punkte über ein Zellrasterverfahren, gefolgt von einer Lloyds-Relaxation. Im ersten Schritt werden in jeder Zelle zufällig gleich viele Punkte verteilt, wobei aber nur Zellen verwendet werden, die vollständig oder größtenteils in der Wandfläche liegen. Im folgenden Schritt wird iterativ die Lloyds-Relaxation angewendet (nach Hiller et al. [HHD03]). Dabei wird erst das Voronoi-Diagramm für die Punkte berechnet. Anschließend werden die Schwerpunkte der Voronoi-Regionen berechnet und als neue Punkte benutzt. Das Verfahren kann nach einer bestimmten Anzahl von Iterationen abgebrochen oder terminiert werden, wenn die Unterschiede in der Distanz zwischen einem Punkt im Voronoi-Diagramm und seinem Schwerpunkt unterhalb eines Schwellwerts liegen. Dadurch lassen sich entweder unregelmäßige Zyklopenmauerwerke erzeugen, wenn das Verfahren vorzeitig abgebrochen wird, oder sehr regelmäßige Strukturen, wenn erst bei der Unterschreitung des Schwellwerts terminiert wird.

Werden die Voronoi-Regionen (sie verkörpern später die Steine) mit dem Wandpolygon und den Polygonen der Öffnungen geschnitten, so können konkave Regionen nur an konkaven Ecken der Wand bzw. an konvexen Ecken der Öffnungen entstehen. Dieses Problem kann leicht umgangen werden, indem die zwei nächsten Punkte so gesetzt werden, dass die gemeinsame Kante durch die problematische Ecke verläuft.

Seien durch die Punkte \mathbf{p}_1 , \mathbf{p}_2 und \mathbf{p}_3 eine konkave bzw. konvexe Ecke gegeben. Die minimale Punktdistanz PD entspricht etwa $2r$. Alle Punkte, die im Umkreis PD um \mathbf{p}_2 liegen, werden entfernt und durch zwei neue Punkte ersetzt. Die beiden Punkte werden – wie in Abbildung 5.15 gezeigt – symmetrisch um die Winkelhalbierende mit einem Abstand von $PD/2$ zu \mathbf{p}_2 und einem Abstand von PD zueinander angeordnet. Anschließend wird erneut die Lloyds-Relaxation auf die verbleibenden Punkte angewendet. Damit gleichen sich die verbliebenen Regionen an die beiden festen Regionen an. Es werden für alle kritischen Ecken neue Voronoi-Zentren berechnet. Dabei muss überprüft werden, ob sich die Kreise kritischer Punkte überschneiden. Gegebenenfalls werden die Radien der Kreise so angepasst, dass sie sich genau berühren. Damit einhergehend werden die Abstände der neuen Zentren angeglichen. In Abbildung 5.16 sind die Ergebnisse zweier Beispiele zu sehen.

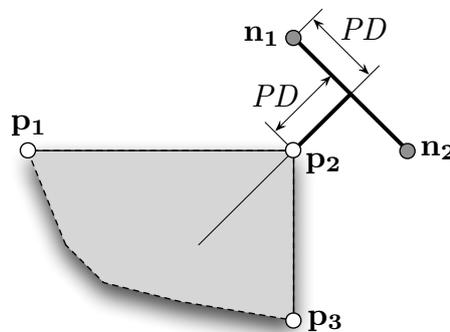
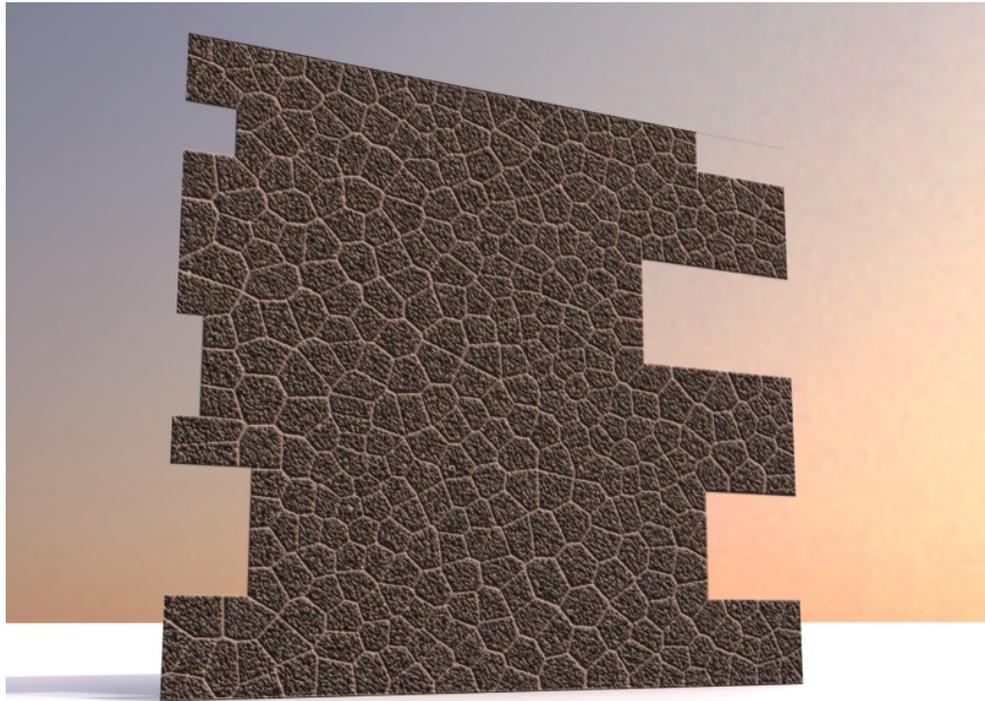
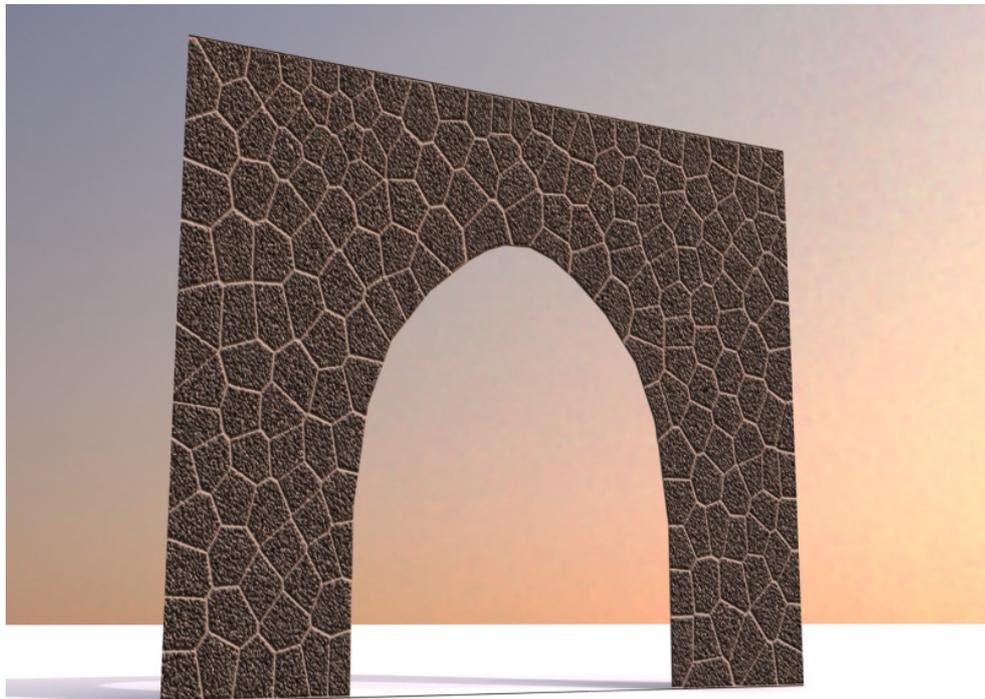


Abbildung 5.15: Beispiel einer problematischen Ecke. Die über die Punkte \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 angedeutete Fläche stellt eine Öffnung mit einer Problemecke in \mathbf{p}_2 dar. Die Punkte \mathbf{n}_1 und \mathbf{n}_2 sind die neuen Zentren der Voronoi-Regionen, die symmetrisch um die Winkelhalbierende durch \mathbf{p}_2 mit dem Abstand PD angeordnet sind.



(a) Zyklopenmauerwerk mit unregelmäßigen Seiten.



(b) Zyklopenmauerwerk mit Toröffnung.

Abbildung 5.16: Beispiele für Zyklopenmauerwerke.

5.4 Dekorationen

In den vergangenen Abschnitten wurde die Fassade von einer groben hin zu einer relativen feinen Struktur unterteilt. An diesem Punkt angekommen, wird die Fassade mit ausschmückenden Elementen, wie Gesimse, Ornamente usw. versehen. Allgemein werden diese Ausschmückungen als Dekorationen bezeichnet. Koch [Koc90] fasst die Dekorationen der Fassaden wie folgt zusammen:

Dekoration: Gesamtheit aller zur Ausschmückung dienenden Gegenstände und Ornamente, auch die Gesamtheit der schmückenden Einzelmotive, die für einen Einzelgegenstand wie eine bestimmte Fassade [...] angeordnet werden.

In unserem Fall dienen Gesimse, Ornamente und Friese zur Ausschmückung der Fassade. Der Einsatz dieser Dekorationen ist vielfältig, so werden z. B. horizontal verlaufende Strukturen (Gurt-, Sockelgesimse usw.) der Fassade, wie auch Fenster (Fensterstöcke und -rahmen) und Türen, damit verfeinert.

5.4.1 Gesimse, Konsolen und Friese

In diesem Abschnitt werden Gesimse, Konsolen und Friese besprochen. Zu jedem der drei Bereiche wird anfangs das Gebiet kurz skizziert und ein Überblick gegeben. Die drei Themenbereiche werden hier zusammengefasst, da man anhand der Gesimse sehr gut auf die Konsolen und Friese überleiten kann. In unserem Fall tauchen Konsolen als Elemente der Gesimse und als Stützen auf. Der Fries wird in Form texturierter Gesimse eingeführt.

5.4.1.1 Gesimse

Ein Gesims oder Sims ist nach Koch [Koc90] und Reid [Rei90] ein aus der Mauer hervortretender waagrechter Streifen zur Betonung waagrechter Bauabschnitte. Nach Koch, Reid und Krauth [KM96] unterscheiden wir für diese Arbeit fünf Typen von Gesimsen:

- **Sockelgesims**

Oberes Abschlussprofil am Sockel der Fassade. In dieser Arbeit wird der gesamte Sockel als Gesims modelliert, um eine einheitliche Beschreibung der Sockel zu erreichen.

- **Gurtgesims**

Dieses Gesims befindet sich zwischen den Stockwerken und grenzt somit Stockwerke voneinander ab.

- **Kranz- oder Hauptgesims**

Dieses Gesims verläuft zwischen dem oberen Ende der obersten Wand und dem Dach. Es wird oft durch Kragsteine (Konsolen) gestützt und beinhaltet meist auch eine Tropfleiste, die die Fassade vor Wasser schützt.

- **Band bzw. Bänder**

Ähnlich zu Gurtgesimsen, nur dass dieser Typ deutlich kleiner ausfällt als Sockel- und Hauptgesimse. Sie verlaufen entlang der Höhe der Fensterbänke, beziehen diese auch meist mit ein.

- **Tür- und Fenstergesims**

Darunter versteht man entweder eine Fenster- oder Türverdachung die als Mittel zur Belebung der Fassade durch Schattenwirkung dient oder ein Gesims auf Höhe der Fensterbank an der Außenwand des Gebäudes. In unserem Fall werden Gesimse für den gesamten Tür- und Fensterstock eingesetzt.

Die Sockel- und Hauptgesimse sind die größten Gesimse, wobei die Hauptgesimse meist sehr ausladend und reichhaltig verziert sind, wie die schematische Darstellung in Abbildung 5.17(a) veranschaulicht. Abbildung 5.17 zeigt verschiedene Sockel-, Gurt- und Hauptgesimse, die aus dem Steinhauerbuch von Krauth und dem Baumformenbuch von Brauswetter [KM96, Bra95] entnommen sind und als Vorgabe für die Entwicklung einer Beschreibungssprache für Gesimse herangezogen wurden. Beispiele für Konsolen der Hauptgesimse und Balkonträger sind in Abbildung 5.17(b) dargestellt.

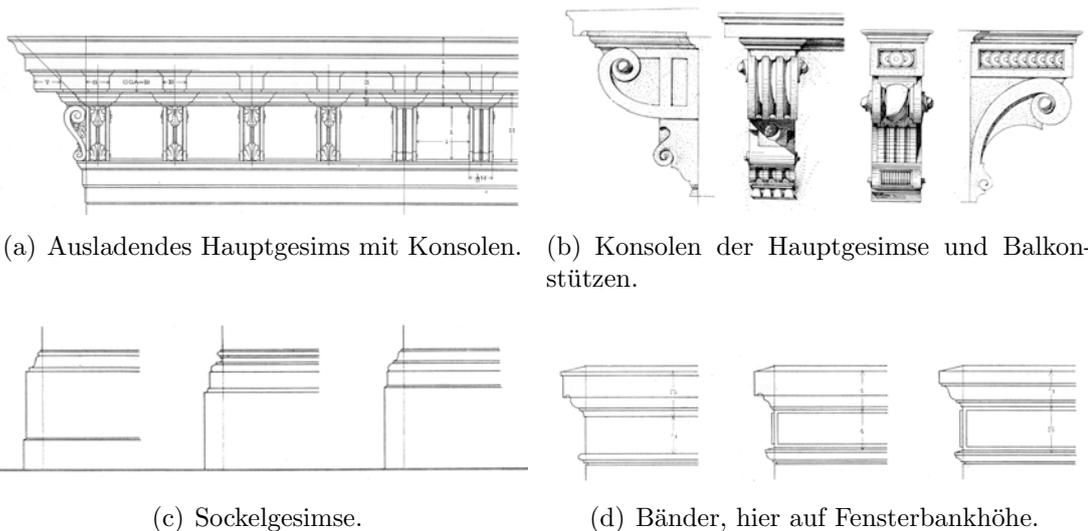


Abbildung 5.17: Gesimse und Konsolen [KM96, Bra95].

Die Beschreibung einer Gesimsstruktur gliedert sich in drei Bereiche: Gesimsprofile, Konsolen und Texturen. Im Weiteren werden wir uns den Gesimsprofilen zuwenden.

Um eine Gesimsstruktur als Gesims oder als Fenster- bzw. Türstock an der Fassade anzubringen, wird erst das Profil des Gesimses in textueller Form angegeben. Anschließend wird das Profil entlang einer oder mehrerer Partitionen angebracht. Dazu wird das Profil skaliert, damit die gesamte Partitionshöhe abgedeckt ist. Ebenso werden die Gesimsstrukturen zur Verfeinerung der Fenster- und Türprofile wie auch der Rahmen herangezogen (siehe Abschnitt 5.5.1). Um die Beschreibung beliebiger Gesimsprofile möglichst einfach zu halten, wurde eine der Turtle-Graphik der Programmiersprache Logo [MAA84] ähnliche Beschreibungssprache entworfen. Die Sprache kennt nur die beiden Kommandos *line* und *arc*, mit denen Linien bzw. Kreisbögen definiert werden. Ein Gesimsprofil wird über eine Liste dieser Kommandos beschrieben.

Definition 5.9. Gesimsprofil

$$\begin{aligned}
 \textit{gesimsprofil} & ::= \mathbf{Modifications}\{ (\textit{arc} \mid \textit{line})^+ \} \\
 \textit{arc} & ::= \mathbf{arc, rel} \mid \mathbf{abs, winkel, winkel, radius} \mid \mathbf{[, texturen]}; \\
 \textit{line} & ::= \mathbf{line, rel} \mid \mathbf{abs, winkel, laenge} \mid \mathbf{[, texturen]}; \\
 \textit{winkel} & ::= [-]\textit{zahl} \\
 \textit{radius} & ::= \textit{zahl} \\
 \textit{laenge} & ::= \textit{zahl} \\
 \textit{texturen} & ::= \textit{tex_index}, \textit{tex_index}, \textit{tex_index} \\
 \textit{tex_index} & ::= [-]\textit{ziffer}^+ \\
 \textit{zahl} & ::= \textit{ziffer}^+ \mid \textit{ziffer}^* \\
 \textit{ziffer} & ::= 0\text{--}9
 \end{aligned}$$

Das Profil wird in einer Ebene erzeugt, die von den beiden Vektoren **up** und **normal**, der Profildbasis, aufgespannt wird. Die beiden Vektoren definieren zugleich auch das globale Koordinatensystem. **up** bezeichnet die aktuelle Bezugsrichtung des Profilverlaufs, wobei **normal** stets orthogonal zu **up** ist. Die Kommandos *line* und *arc* erwirken eine Änderung im Verlauf des Profils und führen gleichzeitig zu einer Änderung der Bezugs- und Normalenrichtung. Die Startrichtung eines Kommandos entspricht stets der des **up**-Vektors. Der erste *winkel*-Parameter beider Kommandos gibt eine Drehung in Richtung **normal** an. Jedes Kommando besitzt einen Startpunkt, der zu Beginn der Ursprung ist. Nach der Ausführung entsteht ein Endpunkt der zugleich den neuen Startpunkt für das nächste Kommando darstellt. Werden die Kommandos der Liste nacheinander abgearbeitet, so entsteht ein zusammenhängendes Profil.

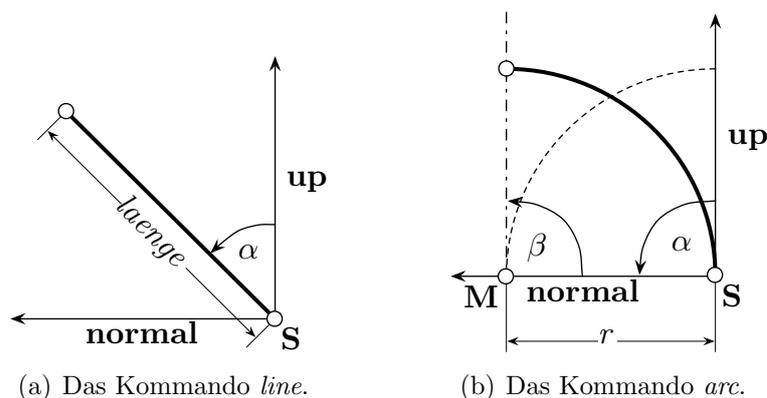


Abbildung 5.18: Illustrationen zur Profildefinition.

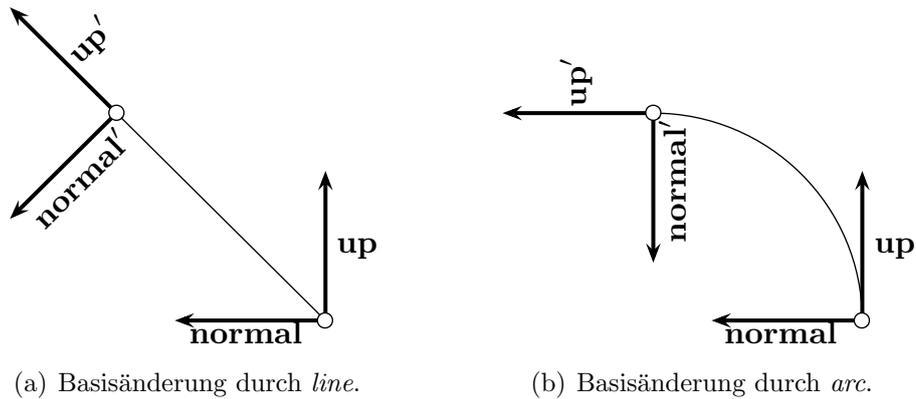
Die Funktionsweisen der beiden Kommandos sind in Abbildung 5.18 dargestellt. Bei *line* (Abb. 5.18(a)) gibt der erste Parameter α den Winkel der zu erzeugenden Strecke relativ zur Profilbasis an. Vom Ursprung **S** der Profilbasis wird dann eine Strecke der Länge *laenge* gezeichnet.

Abbildung 5.18(b) veranschaulicht das Kommando *arc*, das einen Kreisbogen beschreibt. Der erste Parameter α legt die Position des Kreismittelpunkts **M** fest, der sich auf einem Kreis (in der Abbildung fein gestrichelter Viertelkreis) mit dem Radius *radius* um den Ursprung der aktuellen Profilbasis befindet. Der zweite Parameter β gibt das Bogenmaß des Kreisausschnittes an. Damit wird ein Kreisbogen ausgehend vom Startpunkt **S** mit dem Radius *radius* definiert. Über den Parameter *textures* können optional für jeden *line*- bzw. *arc*-Abschnitt bis zu drei Referenzen auf Texturen vergeben werden. Die Funktionsweise der Texturen wird ausführlich in Abschnitt 5.4.1.3 behandelt.

Durch Anwenden von *line* oder *arc* entsteht ein neuer Startpunkt für das nächste Kommando. Jedoch ist die Profilbasis des neuen Punkts abhängig vom zuvor ausgeführten Kommando.

Bei *line* verläuft die neue Profilbasis entlang der Richtung der erzeugten Strecke und befindet sich an deren Endpunkt. Bei *arc* entspricht die Richtung und Position der neuen Basis der Tangente im Endpunkt des Kreisbogens. An einem Beispiel in Abbildung 5.19 wird die Änderung von Ausgangsbasis zur neuen Basis für beide Kommandos veranschaulicht.

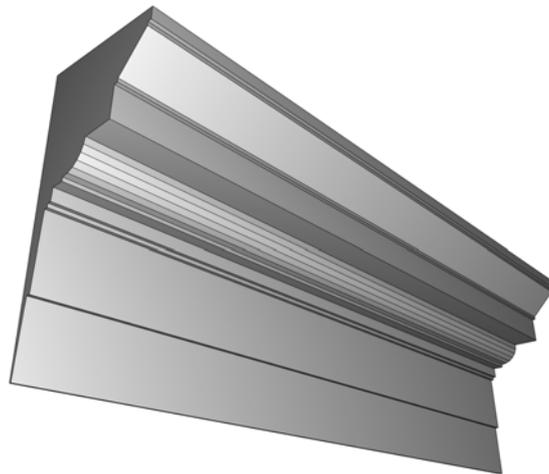
Ein Kommando kann relativ zur neuen Basis oder absolut, also zur ursprünglichen Basis, ausgeführt werden. Über den gemeinsamen Parameter *rel* bzw. *abs* wird festgelegt, ob sich der anschließende Parameter *winkel* auf das globale oder auf das lokale Koordinatensystem bezieht. Dabei ist zu beachten, dass durch eine Änderung von *rel* zu *abs*, ein völlig anderes Ergebnis erzielt wird.

(a) Basisänderung durch *line*.(b) Basisänderung durch *arc*.**Abbildung 5.19:** Transformation der Profilbasen.

In [Abbildung 5.20\(b\)](#) ist ein Gesims dargestellt, das mit den Kommandos aus [Abbildung 5.20\(a\)](#) erzeugt wurde.

Modifications

```
{
  line, rel, 90, 0.3;
  line, rel, -90, 10.0;
  line, rel, 90, 0.6;
  line, rel, -90, 11.0;
  line, rel, 90, 0.6;
  line, rel, -90, 0.6;
  line, rel, 90, 0.6;
  line, rel, -90, 2.0;
  line, rel, 90, 2.0;
  arc, rel, 0, -90, 4.0;
  line, rel, 90, 5.0;
  line, rel, -90, 5.0;
  line, rel, 90, 0.6;
  line, rel, -90, 0.6;
  line, rel, 45, 7.0;
  line, rel, -45, 0.6;
  line, rel, 90, 0.6;
  line, rel, -90, 2.0;
  line, rel, -84, 22.0;
}
```



(a) Kommandos.

(b) Beispiel für ein Gesims.

Abbildung 5.20: Beispiel für ein Gesims mit Beschreibung. In diesem Fall handelt es sich um das Profil eines Hauptgesimses.

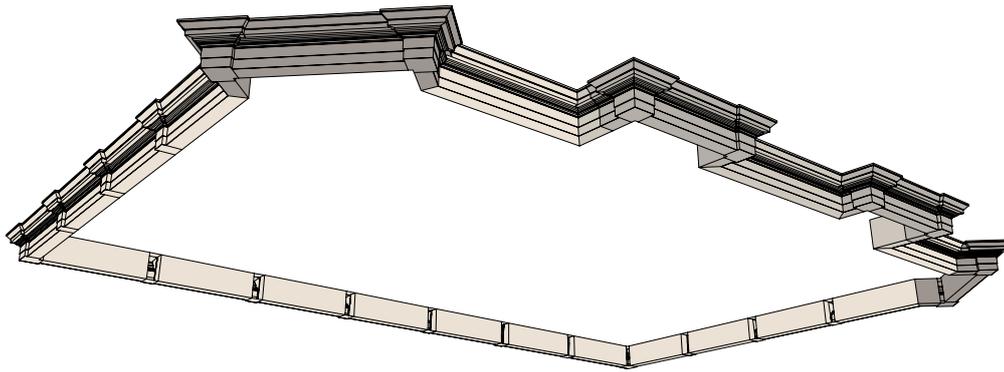


Abbildung 5.21: Beispiel für ein Gurtgesims.

Wie bereits erwähnt, werden die Gesimse entlang der Fassade, also um das Gebäude herum, appliziert. Abbildung 5.21 zeigt ein Gurtgesims, das entlang eines Gebäudeumrisses verläuft.

5.4.1.2 Konsolen

Koch [Koc90] definiert Konsolen als aus der Mauer hervorspringende Tragsteine für Balkone, Figuren, Balken usw., die oft ornamental oder mit Figuren geschmückt sind. In Abbildung 5.17(b) sind Gesimse zu sehen, in die Konsolen eingearbeitet sind. Wie man gut erkennen kann, variiert die Form der Konsolen innerhalb eines Gesimses nicht bzw. nur sehr selten. Da die Motive der Formen der Konsolen von einfachen Voluten bis hin zu sehr komplizierten Darstellungen aus der Pflanzen-, Tier- und vor allem Fabelwelt reichen, wird hier nur die Möglichkeit bereit gestellt, vorgefertigte 3D-Objekte einzubinden. Konsolen erweitern damit die Gesimsbeschreibung. Sie werden als äquidistante Objekte, auf deren rechteckiger Grundfläche sich ein 3D-Objekt befindet, in das Gesims eingefügt.

Definition 5.10. Konsolen

$$\begin{aligned}
 \textit{konsolen} &::= \mathbf{Consoles}\{ (\textit{abstand konsole})^+ \} \\
 \textit{abstand} &::= \textit{zahl}; \\
 \textit{konsole} &::= \textit{”objektdatei”}, \textit{breite}, \textit{hoehe}; \\
 \textit{objektdatei} &::= \textit{zeichenkette} \\
 \textit{breite} &::= \textit{zahl} \\
 \textit{hoehe} &::= \textit{zahl} \\
 \textit{zahl} &::= \textit{ziffer}^+ [\cdot] \textit{ziffer}^* \\
 \textit{ziffer} &::= 0-9 \\
 \textit{zeichenkette} &::= [a-z A-Z 0-9 _ + - .]^+
 \end{aligned}$$

Die Parameter der Konsolenbeschreibung haben folgende Bedeutung. Über *abstand* wird die Distanz zwischen den Mittelpunkten der Konsolenrechtecke angegeben. *objektdatei* enthält das 3D-Objekt, welches auf das Rechteck platziert wird. Die beiden letzten Parameter, *breite* und *hoehe*, bestimmen die Ausmaße des Rechtecks. Das Rechteck selber wird in der Ebene ausgerichtet, die aus dem Kreuzprodukt der Vektoren der Profilbasis und dem **up**-Vektor aufgespannt wird. Die Ausmaße des Rechtecks werden relativ angegeben und das Rechteck wird entsprechend der Größe bzw. der Höhe des Profils skaliert. In Abbildung 5.22 sind die Parameter und die Platzierung der Konsole im Gesims graphisch veranschaulicht. In der Graphik ist der Verlauf des Gesimses durch die beiden horizontal verlaufenden gestrichelten Linien angedeutet.

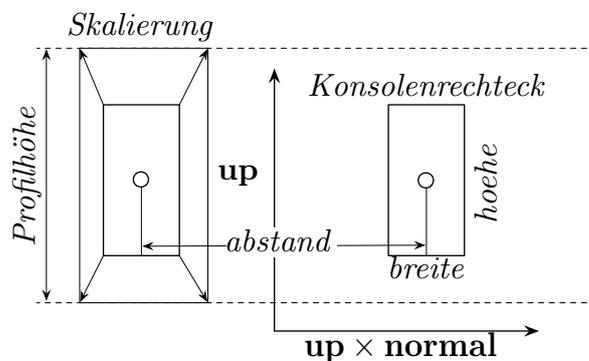


Abbildung 5.22: Platzierung der Konsolen im Gesims.

Das mit Konsolen versehene Gesims in Abbildung 5.28 wurde mit den vorgestellten Techniken erzeugt. Wie man sieht, ist das Gesims auch mit Texturen versehen, die Gegenstand des nächsten Abschnitts sind.

5.4.1.3 Friese

Koch [Koc90] beschreibt einen Fries als einen schmalen Streifen, der Flächen begrenzt oder auch teilt und häufig ornamentiert ist. Beispiele für ornamentierte Friese sind in Abbildung 5.23 zu sehen.

In unserem Fall werden diese schmalen Streifen mit den Gesimsen erzeugt. Friese werden also nicht als eigenständige Objekte beschrieben, sondern sie gehen direkt aus mit Texturen versehenen Gesimsen hervor. Für die Texturierung können beliebige Graphiken oder die im nächsten Abschnitt vorgestellten Ornamente verwendet werden. Die Texturierung der Gesimse stellt speziell an Nahtstellen bzw. Verkröpfungen eine gewisse Herausforderung dar. Die allgemeine Definition für die Texturierung wird analog zu den Gesimsen und Konsolen geführt. Sie stellt damit, wie die Konsolen, eine Erweiterung der Gesimsbeschreibung dar.

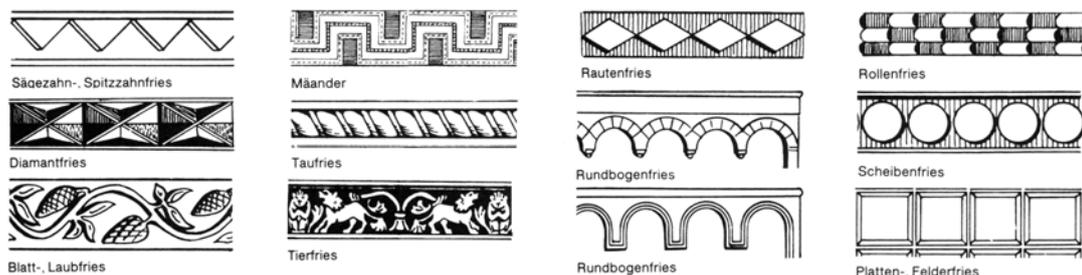


Abbildung 5.23: Beispiele für Frieße [Koc90].

Definition 5.11. Texturenverbund

$$\begin{aligned}
 \text{texturenverbund} &::= \mathbf{TexturePool}\{ (texture)^+ \} \\
 \text{texture} &::= \text{"texturdatei", links, oben, rechts, unten}; \\
 \text{texturdatei} &::= \text{zeichenkette} \\
 \text{links} &::= \text{zahl} \\
 \text{oben} &::= \text{zahl} \\
 \text{rechts} &::= \text{zahl} \\
 \text{unten} &::= \text{zahl} \\
 \text{zahl} &::= \text{ziffer}^+ [.] \text{ziffer}^* \\
 \text{ziffer} &::= 0-9 \\
 \text{zeichenkette} &::= [a-zA-Z0-9_+ -.]^+
 \end{aligned}$$

Wie man sieht, können in einer Beschreibung gleich mehrere Texturen zusammengefasst werden. Im Anschluss an die Beschreibung der Parameter wird dargestellt, weshalb mehrere Texturen in einem Fries sinnvoll sind.

Über jeden *texture*-Eintrag wird genau eine Textur definiert. Dabei wird zum einen der Dateiname (*texturdatei*) und zum anderen der zu verwendende Ausschnitt der Textur in $u-v$ -Koordinaten angegeben, der mit den Parametern *links*, *oben* und *rechts*, *unten* festgelegt wird. Der Wertebereich für die Parameter entspricht den der üblichen Texturkoordinaten im Intervall $[0, 1]$. Dadurch ist es möglich, dass eine Datei bereits mehrere Texturen enthält.

Bis zu drei Texturen können, wie in Definition 5.9 beschrieben, zu jedem durch *line* bzw. *arc* erzeugten Gesimsabschnitt angegeben werden. Bestimmt über ihre Reihenfolge werden sie in Anfangs-, Mittel- und Endtextur unterschieden. Die Aufteilung auf einen Gesimsabschnitt erfolgt, wie in Abbildung 5.24 dargestellt. Wird als Referenz ein negativer Wert angegeben, so erhält der entsprechende Gesimsteil keine Textur. Die mittlere Textur überdeckt den größten Teil des Gesimses. Dabei wird das Seitenverhältnis der Textur so gut wie möglich beibehalten, was dazu

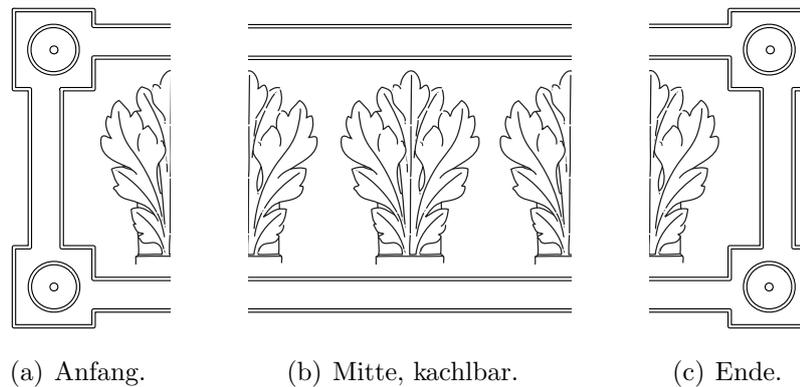
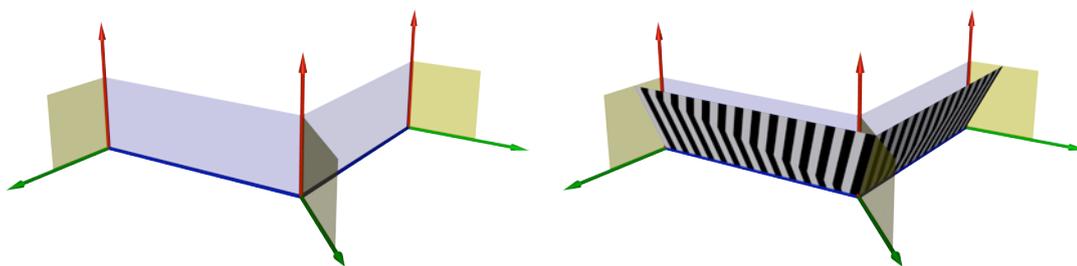


Abbildung 5.24: Beispiel für eine Fries-Textur mit Blattwerk-Ornament.

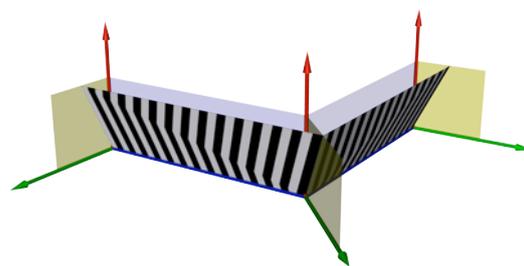
führt, dass die Textur wiederholt aneinander gefügt wird, um die gesamte Gesimsbreite zu füllen. Um Artefakte an den Texturübergängen zu vermeiden, sollte die mittlere Textur in horizontaler Richtung kachelbar sein.

Soll die Textur auf einem Gesims, entlang eines Kantenzugs, aufgebracht werden, so können verschiedene Probleme auftreten. Um die Probleme besser erfassen zu können, sind in [Abbildung 5.25\(a\)](#) die Ebenen, die maßgeblich für die Erzeugung der Geometrie des Gesimses notwendig sind, aufgeführt. Damit greifen wir ein wenig der Geometrieerzeugung voraus. Bei den Ebenen handelt es sich zum einen um die Profilebenen (gelblich dargestellt), in der die Gesimsprofile definiert werden, und zum anderen um die Wandebenen (blaue Vierecke), an denen das Gesims angebracht wird. Es wird zwar immer nur ein Profil entlang eines Kantenzugs geführt, aber zur Erzeugung der Geometrie wird an jedem Scheitelpunkt des Kantenzugs eine Profilebene definiert. Besteht der Kantenzug aus nur einer Kante, so kann die Textur ohne Probleme aufgetragen werden. In diesem Fall wird über *line* ein Rechteck und über *arc* viele Rechtecke (der Bogen wird über Kantenzüge angenähert) erzeugt. Stoßen aber zwei Kanten zusammen, ergibt sich einer der folgenden Fälle:

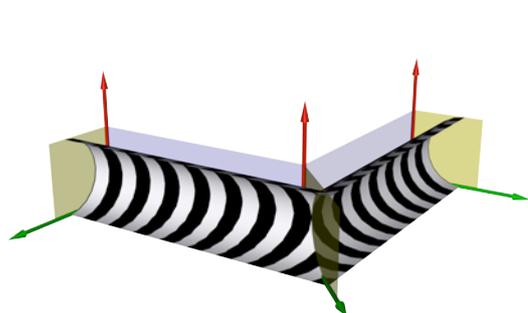
- Die Kanten sind parallel.
In diesem Fall kann das Gesims bzw. das zugehörige Profilsegment jeder Kante ohne Bedenken texturiert werden.
- Die Kanten treffen sich in einem Winkel $0^\circ < \alpha < 180^\circ$ und bilden am Stoßpunkt einen Knick.
Handelt es sich um ein Profilsegment, das über *line* erzeugt wurde, so tritt ein Problem auf, wenn die dadurch beschriebene Fläche nicht parallel zur Wandebene verläuft. Das zu texturierende Rechteck degeneriert zu einem Trapez und die Textur wird verzerrt dargestellt. In [Abbildung 5.25\(b\)](#) ist an der linken Textur gut zu sehen, dass die schwarzweißen Balken einen Knick



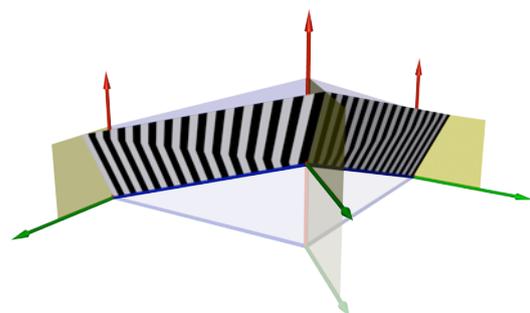
(a) Teil eines Grundrisskantenzugs mit Profil- und Wandebenen.



(b) Problem am Knick bei *line*.



(c) Problem am Knick bei *arc*.



(d) Kantenzug liegt nicht in einer Ebene.

Abbildung 5.25: Probleme der Texturierung bei Gesimsen.

aufweisen. Bei *arc* hingegen werden am Stoßpunkt der Umrisskanten sogar Teile der Textur abgeschnitten (siehe Abbildung 5.25(c)).

- Der Kantenzug liegt nicht in einer Ebene.
Hier werden die Rechtecke ähnlich wie beim vorherigen Punkt verzerrt, aber auch noch geschert und liegen dazu nicht mehr in einer Ebene (siehe Abbildung 5.25(d)).

Das eigentliche Problem liegt aber nicht in der Verzerrung der Texturen, sondern an den unsauberen Übergängen der Texturen an den Stoßpunkten. Hierfür wurde in dieser Arbeit eine einfache und effiziente Lösung entwickelt. Sie bewältigt zwar nicht die Probleme aller aufgeführten Fälle (so bleiben beispielsweise Verzerrungen, die durch Scherungen hervorgerufen werden, bestehen), aber der Übergang an den Stoßpunkten von einer Textur auf die andere wird visuell deutlich ansprechender gestaltet. Erreicht wird das Ganze über die optionalen Anfangs- und Endtexturen. Dafür wird für jeden Gesimsabschnitt einer Umrisskante an seinen Enden zusätzlich Geometrie erzeugt, die jeweils genau die Anfangs- bzw. Endtextur aufnimmt. Dazwischen wird weiterhin die mittlere Textur aufgetragen. In Abbildung 5.26 ist der Schritt von der anfänglichen Geometrie zur erweiterten

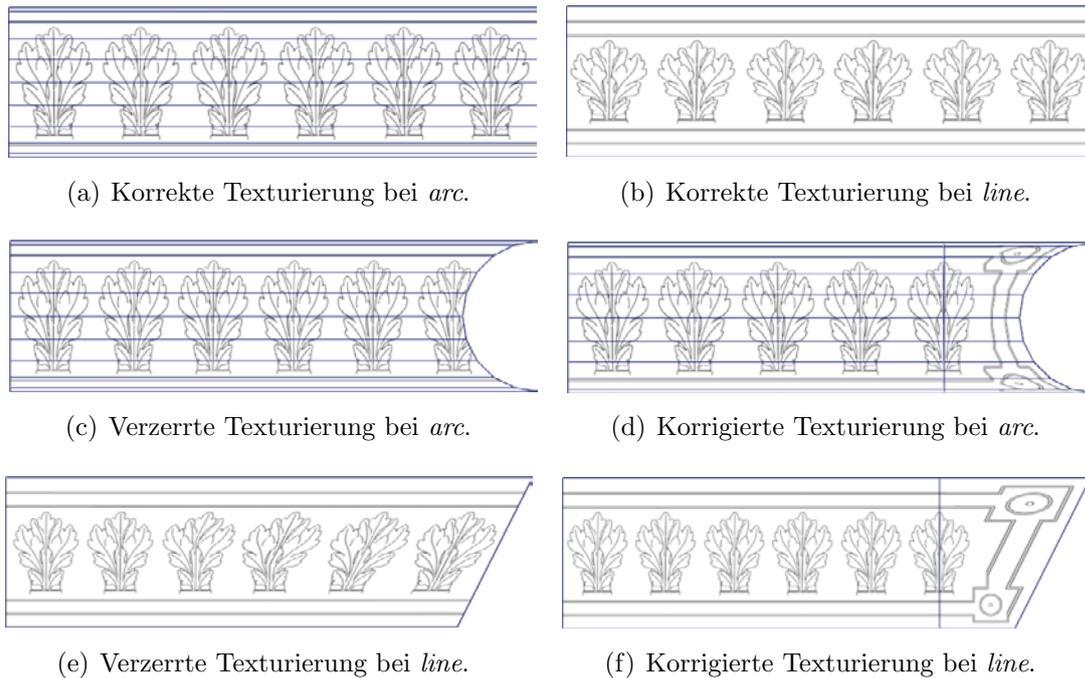


Abbildung 5.26: Fehlerhafte und korrigierte Texturierungen.

Geometrie für *line* wie auch *arc* dargestellt. Gezeigt werden jeweils die Frontalansichten in Parallelprojektion mit Texturen und Drahtgittermodell. Wie man sieht, werden zum einen nur noch die Texturen an den Enden verzerrt dargestellt. Zum anderen sind Teile der Texturen nicht mehr abgeschnitten, wie es bei *arc* der Fall war.

Abbildung 5.27 zeigt das Ergebnis der Vorgehensweise anhand eines Beispiels für ein Bogensegment. Beide Gesimse sind dreigeteilt, wobei die kleinen Segmente die Anfangs- und Endtextur enthalten. Das große Segment in der Mitte ist mit der kachelbaren Textur versehen, die beiden Enden enthalten dazu passende seitlich abschließende Texturen.



Abbildung 5.27: Beispiel für die Lösung des Texturierungsproblems mit Anfangs-, Mittel- und Endtexturen.

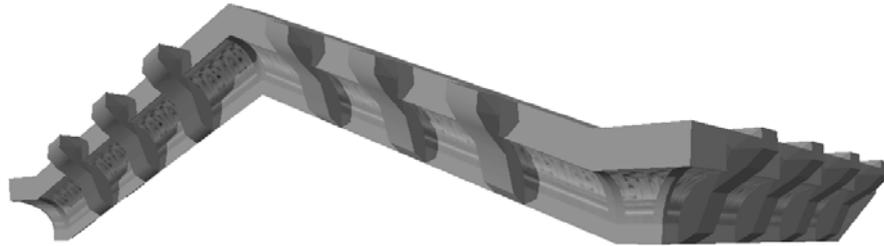


Abbildung 5.28: Lösung für das Texturierungsproblem mit Anfangs-, Mittel- und Endtexturen.

Ein weiteres Beispiel, das neben Texturen, noch Konsolen und ein leicht komplizierteres Gesims beinhaltet, ist in [Abbildung 5.28](#) zu sehen und wurde mit dem in dieser Arbeit entwickelten Prototypen erstellt. Die vollständige Beschreibung des Beispielgesimses, das nicht nur das Gesimsprofil, sondern auch Konsolen und Texturen enthält, sieht wie folgt aus:

```

Modifications {
  line, rel, 0, 0.2;
  arc, abs, 90, 45, 0.5, 0, 1, 2;
  line, rel, -90, 0.4;
}

Consoles {
  1.5;
  "objects/simple_console.obj", 0.35, 0.45;
}

TextureDescs {
  "textures/frieze.bmp", 0.0, 0.625, 0.04875, 0.25;
  "textures/frieze.bmp", 0.0, 1.0, 0.5, 0.625;
  "textures/frieze.bmp", 0.953125, 0.625, 1.0, 0.25;
}

```

Im nächsten Abschnitt werden Methoden zur automatischen Erzeugung von Texturen (in diesem Fall eine spezielle Klasse der Ornamente) vorgestellt.

5.4.2 Ornamente

In diesem Abschnitt machen wir zuerst einen Exkurs in die Welt der Ornamente, um genauer zu verstehen, wie sich Ornamente einordnen lassen und wozu sie eingesetzt werden. Anschließend wird das in dieser Arbeit abgedeckte Teilgebiet der Ornamente besprochen und die Methoden zur Erzeugung derartiger Ornamente vorgestellt.

5.4.2.1 Übersicht der Ornamente

Meyer gibt in seinem *Handbuch der Ornamentik* [Mey27] eine umfassende und an Illustrationen reichhaltige Abhandlung über die Technik und Vielfalt der Ornamente. Im Folgenden wird die Materie der Ornamentik nur grob umrissen, um wenigstens ein Gefühl für den Umfang und die Komplexität zu vermitteln.

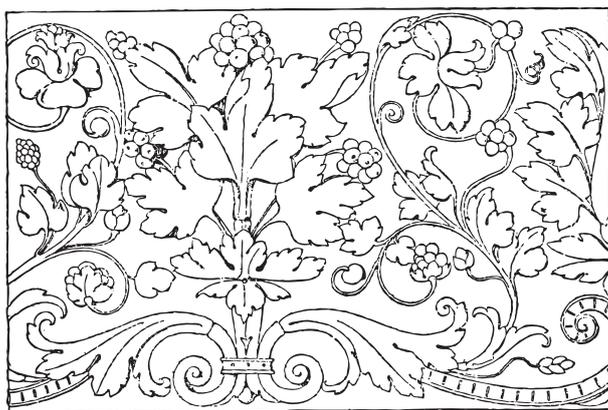
Die Bezeichnung Ornament, Ornamentik, ornamental usw. leiten sich vom lateinischen *ornare* ab, das gleichbedeutend ist mit *schmücken*. Ornamente dienen allgemein der künstlerischen Ausschmückung bzw. Verzierung zahlloser Dinge. In unserem Fall liegt das Interesse auf der Ausschmückung von Gebäudefassaden bzw. deren Elemente. Ornamente lassen sich anhand verschiedener Kriterien klassifizieren.

Eine Form der Klassifizierung orientiert sich an den dargestellten Motiven. Das sind zum einen die *geometrischen Motive*. Sie ergeben sich durch eine gleichförmige Aneinanderreihung von Punkten und Linien, durch regelmäßige Winkelteilung usw. Ein weiteres Motiv sind die *Naturformen*. Darunter fallen die Pflanzen- und Tierwelt, wie auch der menschliche Organismus. Als letztes folgt das sog. *künstliche Motiv*. Hier finden sich entlehnte Formen der Kunst, der Technik und der Wissenschaft. Sie treten meist in Emblemen, Wappen oder Trophäen auf.

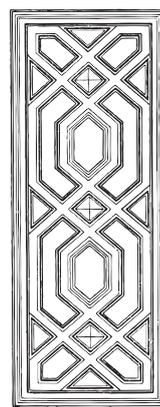
Eine andere Möglichkeit der Unterscheidung erfolgt über die Form des Ornaments, unabhängig von seinem Motiv. Darunter versteht man Einzelformen in Bezug auf ihre Funktionen und nach der Beziehung der Einzelformen untereinander bzgl. der Bildung des Ornaments. Die Ausbildung der Ornamente wird in fünf Gruppen aufgeteilt: Bänder, freie Endigungen, Stützen, begrenzte und unbegrenzte Flachornamente.

Bänder sind, ähnlich wie Friese, schmale Streifen und beinhalten meist geometrische und pflanzliche Motive. Sie haben einen säumenden, einfassenden und verknüpfenden Charakter, weshalb man sie meist als Elemente der Umrandung oder Trennung vorfindet.

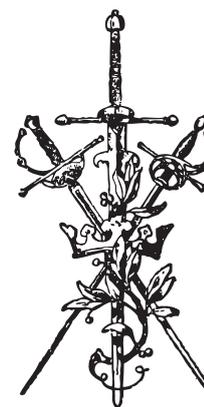
Freien Endigungen bezeichnen diejenigen Ornamentformen, die in ihrer Anwendung ein Enden, ein Aufhören in einer stilistisch korrekten Form zum Ausdruck bringen. Mit dem Zusatz *frei* wird angedeutet, dass es sich hierbei nicht um ein zwingendes Abschließen handelt, sondern um eine Art stilistisch richtige Lösung.



(a) Ein Pflanzenmotiv.



(b) Ein geometrisches Motiv.



(c) Ein sog. künstliches Motiv.

Abbildung 5.29: Ornamente, klassifiziert nach ihren Motiven [Mey27].

Je nach Richtung der Endigung – oben, unten oder seitwärts – können hier verschiedene Motive eingesetzt werden. So eignen sich pflanzenähnliche Motive sehr gut für Endigungen nach oben. Quasten und Fransen hingegen unterstützen eher die Richtung nach unten und Wasserspeier seitwärts.

Die *Stützen* umschließen all diejenigen Elemente, die einen stützenden, tragenden Charakter haben. Darunter fallen alle Teile der Säulen (Fuß, Schaft und Kapitäl), Pfeiler, Pilaster, wie auch die von den Gesimsen bekannten Konsolen.

Das *begrenzte Flachornament* hat zum Ziel, Flächen zu schmücken bzw. dekorativ zu überziehen. Dabei können beliebige Techniken, wie z. B. Malerei, Einlegearbeiten, Gravierungen usw. verwendet werden. Dieser Typ von Ornamenten wird benutzt, wenn beliebig geformte Flächen (in unserem Fall sind das einfache Polygone) gefüllt werden sollen. Dabei wird das Ornament genau in die Fläche eingepasst. Überlicherweise werden meist einfache Formen wie Rechtecke, regelmäßige Vielecke, Kreise, Ellipsen usw. gefüllt. Die gesamte Bandbreite an Motiven steht für die Füllung zur Verfügung. Sie werden entweder zufällig, in symmetrischer Weise oder sich an der Form orientierend eingebracht.

Die *unbegrenzten* oder *endlosen Flachornamente* hingegen können sich in alle Richtungen unbegrenzt ausbreiten. Dazu gehört es, dass das Motiv eine Wiederholung zulässt (also kachelbar ist). In den meisten Fällen sind die Motive rein geometrischer oder organischer Art oder Vereinigungen beider. Beispiele für endlose Ornamente sind Tapeten, Parkett-, Mosaik- und Fliesenmuster.

Abbildung 5.29 zeigt einige typische Motive der Ornamente. Beispiele für die nach der Form klassifizierte Ornamente sind in Abbildung 5.30 zu sehen.

Aufgrund der Vielfalt an möglichen Ornamentformen bzw. -motiven treffen wir

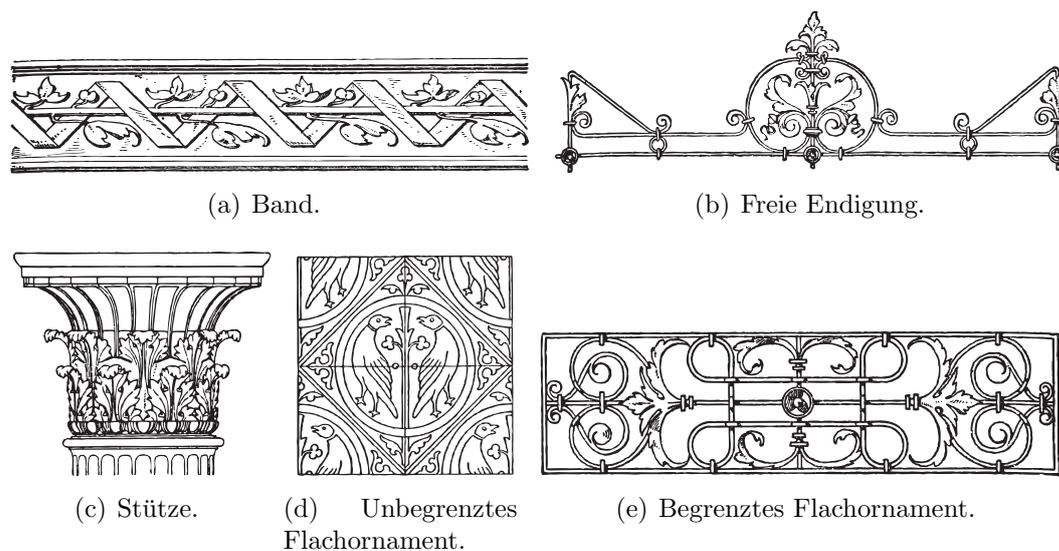


Abbildung 5.30: Ornamente, klassifiziert nach ihrer Form [Mey27].

hier eine Einschränkung auf nur einen Teilbereich der Ornamente. Ebenso lassen sich nicht alle Ornamentstile mit genau einer Methode beschreiben. Methoden für die Beschreibung geometrischer Strukturen sind sicher für Naturformen schlecht geeignet.

Deshalb widmen wir uns in dieser Arbeit den begrenzten Flachornamenten mit pflanzenähnlichen Motiven (siehe Abbildung 5.29(a)). Aus dem Bereich der Naturformen können Motive wie Früchte, Blüten, Blätter und Stängel relativ einfach umgesetzt werden. Eine parametrische einfache Form zur Beschreibung tierischer oder menschlicher Formen, die sich auch noch in beliebige Flächen einpassen lassen, dürfte hingegen recht schwierig sein. Ziel ist es, eine beliebige Fläche mit pflanzenähnlichen Motiven zu füllen. Dabei kommt uns die Natur der Pflanzen mit ihrem Wachstum entgegen. Über geeignete generative Methoden sollen sich die pflanzlichen Motive – ausgehend von einem Startpunkt – in die freien Flächen ausbreiten. Im Fall der Gebäudefassaden werden Ornamente z. B. unter Fensterbänken, in Giebeln oder über Türen erzeugt.

Die Basis für die Erzeugung pflanzlicher Motive liefert die von Wong et al. [WZS98] entwickelte Methode. Die Hauptaufgabe liegt darin, effizient die noch verbleibenden freien Bereiche in der zu füllenden Fläche zu finden. Ebenso werden Methoden und Strukturen zur Verwaltung bereits platzierter Ornamentelemente benötigt, wie auch Methoden zur Auswahl von Elementen, die neue Motive ausbilden können. Um möglichst viele unterschiedliche Ornamente aus einer festgelegten Menge von Motiven erzeugen zu können, müssen die Elemente eine eigene Logik zur Steuerung des Wachstums und der Auswahl der neu zu generierenden Elemente besitzen. Daraus ergibt sich, dass den Elementen die Funktionalität, eigene

Programmfragmente einzubringen und auszuführen, bereitgestellt werden muss. Es gibt also für die Ornamentierung ein Grundgerüst, das ausschließlich die Aufgabe zur Findung freier Bereiche und die Auswahl geeigneter Elemente für ein weiteres Wachstum übernimmt. Die Art und Weise des Wachstums wird aber den Elementen weitestgehend selbst überlassen.

5.4.2.2 Erzeugung pflanzenähnlicher Ornamente

In diesem Abschnitt wird das Prinzip zur Erzeugung der pflanzenähnlichen Ornamente erläutert.

Die mit dem Ornament zu füllende Fläche wird über ein Binärbild angegeben und in einem Grauwertbild (auch als *Backbuffer* bezeichnet) gespeichert. Somit können nahezu beliebige Formen mit Ornamenten gefüllt werden.

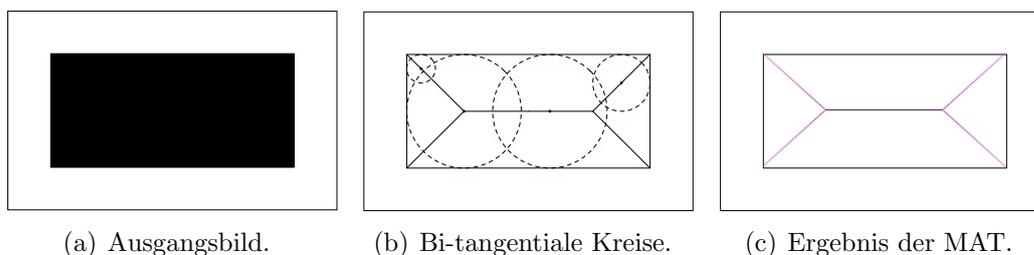
Das Kernstück der Ornamentierung ist ein Algorithmus, der die Verwaltung des Grauwertbildes übernimmt. Das beinhaltet zum einen die Möglichkeit der Bearbeitung und Analyse des Grauwertbildes. Damit spiegelt das Grauwertbild in jedem Erzeugungsschritt die aktuelle Belegung der Ornamentfläche wider. Zum anderen führt der Algorithmus noch eine Liste über die bereits platzierten Elemente. Mit Hilfe des Grauwertbildes und der Elementliste entscheidet der Algorithmus über das weitere Wachstum des Ornaments. Dazu wird einerseits bestimmt, welche Flächen noch für weitere Ornamentteile zur Verfügung stehen und andererseits welche bereits vorhandenen Elemente neue Motive in den freien Bereichen erzeugen dürfen. Die Art und Weise der Auswahl der Elemente obliegt hierbei nicht dem Algorithmus, sondern den Elementen selbst. Er bekommt lediglich das Ergebnis der Aktion mitgeteilt, das eine Liste der neu erstellten Elemente und deren Grundformen enthält. Jedes der Elemente beschreibt seine Form über elementare geometrische Grundformen, die wir später noch genauer kennen lernen werden. Für die Darstellung im Grauwertbild wird aber die tatsächliche Form der Elemente nur angenähert, damit später Abstandsberechnungen im Grauwertbild schneller durchgeführt werden können.

Im Grauwertbild geben die Grauwerte der Pixel direkt die Belegung der Ornamentfläche wieder. Dabei wird der freie Bereich durch schwarze Pixel und die nicht belegbaren Bereiche durch weiße Pixel repräsentiert. Bereits platzierte Elemente werden über graue Pixel markiert und dürfen, ebenso wie die weißen Pixel, nicht überschrieben werden. Die Auflösung des Grauwertbildes unterliegt einem Zielkonflikt. Zum einen sollte die Auflösung nicht zu hoch sein, da darauf aufwendige Rechenoperationen ausgeführt werden. Zum anderen bestimmt die Auflösung die maximale Anzahl zu platzierender Elemente, denn es kann nur höchstens ein Element pro Pixel abgelegt werden.

5.4.2.3 Berechnung der freien Breiche

Der größte Berechnungsaufwand entfällt auf die Bestimmung der freien Bereiche. Mit einem naiven Verfahren müssten erst die freien (schwarzen) Pixel gesucht werden. Um die maximale Größe des von diesem Pixel erreichbaren freien Bereichs zu ermitteln, müssten alle direkt umliegenden Pixel überprüft werden, ob auch diese frei sind. Durch Ausweiten des Testradius um jeweils ein Pixel würde somit der maximal freie Bereich ermittelt. Aber aufgrund seines enormen Rechenaufwands kann dieses Verfahren nicht eingesetzt werden.

Eine schnellere Methode bietet die Mittelachsentransformation (*medial axis transform*, kurz: MAT). Die MAT arbeitet dabei auf einem Binärbild und reduziert die Vordergrundpixel (bei uns die schwarzen Pixel) zu einer Art Skelett bzw. Graph, wobei die Ausmaße und die Zusammenhänge des ursprünglichen Bereichs erhalten bleiben. Eine anschaulichere Erklärung – wie Abbildung 5.31(b) zeigt – für die MAT ist, dass alle Mittelpunkte derjenigen Kreise gesucht werden, die zum einen vollständig in den Vordergrundbereich passen und zum anderen an mindestens zwei Punkten den Randbereich tangential berühren. Die gefundenen Mittelpunkte definieren das Skelett des Vordergrundbereichs. Jedes Pixel des Skeletts hat die Eigenschaft, dass er zu mindestens zwei Punkten des Randbereichs gleichen Abstand hat. Färbt man nun die Mittelpunkte (Pixel des Skeletts) nach aufsteigendem Radius von fast weiß bis nach fast schwarz ein, so wird noch der Abstand der Pixel zum Rand kodiert, wie in Abbildung 5.31(c) zu sehen ist. Die MAT bietet also eine sehr vereinfachte Darstellung des Originalbildes, aus der sich aber genau jenes Originalbild vollständig rekonstruieren lässt.



(a) Ausgangsbild.

(b) Bi-tangentiale Kreise.

(c) Ergebnis der MAT.

Abbildung 5.31: Mittelachsentransformation.

Die Berechnung der MAT erfolgt mit folgendem Verfahren.⁵ Hierbei wird die MAT über die Abstandstransformation (distance transform; [GW87]) des Bildes berechnet. Dabei wird erst ein Grauwertbild erzeugt, das für jedes Pixel seinen Abstand zum Randbereich speichert. Die reduzierte MAT wird bestimmt, indem wir für jedes Pixel die Abstandswerte seiner Nachbarn untersuchen. Das Pixel

⁵Genau genommen wird mit diesem Verfahren nicht die vollständige MAT, sondern nur ein Teil berechnet und zwar der Graph, der die Mittelpunkte der lokal größten bi-tangentiale Kreise repräsentiert.

wird dann zur MAT hinzugenommen, wenn sein Wert größer oder gleich seiner Nachbarn ist. Diese Pixel bilden in ihrer Nachbarschaft somit ein lokales Maximum. Das Ergebnis, die reduzierte MAT, selbst wird nicht mehr als Pixelbild gespeichert, sondern als nach den Kreisradien sortierte Liste. Wird ein neues Element platziert, so wird die Abstandstransformation nicht vollständig neu berechnet, sondern es werden nur die Pixel neu berechnet, deren Wert größer ist als der Abstand zum eben gesetzten Element.

5.4.2.4 Grundformen der Ornamentelemente

Im Folgenden wollen wir uns den Grundformen der Ornamentelemente widmen, also denjenigen geometrischen Formen, die ein einzelnes Element aufbauen, um später aus einer Vielzahl von Elementen ein Gesamtornament entstehen zu lassen. Jedes Element kann mehrere Gruppen von Grundformen beinhalten. Das sind zum einen Grundformen, die genau das Element bzw. sein Aussehen beschreiben (z. B. eine Blüte) und zum anderen Grundformen, die die detaillierte Darstellung nur grob annähern. Eine Blüte mit vielen Blättern kann z. B. durch eine Kreisscheibe angenähert werden. Diese groben Formen, auch Vertreter genannt, dienen dazu, die von einem Element belegten Stellen im Grauwertbild zu markieren. Des Weiteren werden die Vertreter im Grauwertbild herangezogen, um diejenigen Elemente zu finden, die dem aktuell gewählten freien Bereich im Grauwertbild am nächsten liegen.

Für die Darstellung der Ornamentelemente stehen fünf Grundformen zur Verfügung:

- **Punkt**
Das ist die einfachste Grundform und repräsentiert genau einen Punkt gegeben durch eine Koordinate. Im Grauwertbild belegt diese Form genau ein Pixel.
- **Linie**
Hierüber wird eine Strecke mit einem Anfangs- und einem Endpunkt festgelegt, definiert über zwei Koordinaten. Die Darstellung im Grauwertbild erfolgt über eine ein Pixel breite Linie.
- **Kreisbogen**
Der Kreisbogen wird durch die Parameter für die Koordinate seines Mittelpunkts, den Radius, den Start- und den Endwinkel beschrieben. Im Grauwertbild wird der Kreisbogen mit einer ein Pixel breiten Linie gezeichnet.
- **Kreis**
Der Kreis ist über zwei Parameter, seinen Mittelpunkt und seinen Radius gekennzeichnet. Im Grauwertbild wird der Kreis als gefüllte Fläche dargestellt.

- **Bézier**

Mit dieser Grundform werden kubische Bézier-Kurven beschrieben. Dabei kann die Kurve aus mehreren Bézier-Segmenten zusammengesetzt sein. Die erste Kurve wird über acht Parameter, die Koordinaten ihrer Kontrollpunkte, bestimmt. Jede weitere Kurve benötigt nur noch sechs Parameter, da stets an den letzten Kontrollpunkt der Vorgängerkurve angeknüpft wird. Im Grauwertbild kann diese Grundform entweder als eine ein Pixel breite Linie oder, im Falle geschlossener Bézier-Kurven, als gefüllte Fläche gezeichnet werden.

5.4.2.5 Ornamentelemente

Bisher haben wir die geometrischen Grundformen, aus denen die Ornamentelemente aufgebaut werden, kennen gelernt. Nun werden die Ornamentelemente als Ganzes vorgestellt.

Wie eingangs beschrieben, werden über einen Algorithmus freie Bereiche, in denen ein Ornament entstehen soll, bestimmt. Die Art und Weise des Wachstums der einzelnen Elemente des Ornaments ist den Elementen vorbehalten d. h. sie entscheiden, wann welche Elemente erzeugt werden. Deswegen sind Ornamentelemente eigenständige Objekte, die Daten wie auch Methoden enthalten.⁶

Jedes Element bietet wenigstens eine Wachstumsmethode. Als Informationen bzw. Parameter erhält die Methode das Grauwertbild, den nächstgelegenen freien Bereich, im Allgemeinen ein Kreis, und seinen Radius. Anhand dieser Daten entscheidet schließlich die Methode, ob in diesem Kreis weitere Elemente beliebigen Typs erzeugt werden können. Ebenso kann die Methode Graphiken direkt in das Grauwertbild zeichnen, wie auch Graphiken in das Ornament selbst ausgeben. Als Rückgabewert meldet die Wachstumsmethode, ob sie erfolgreich Ausgaben für das Ornament erzeugen konnte. Wurden dabei neue Elemente generiert, werden diese in einer Liste übergeben. Neben der Wachstumsmethode kann das Element eine Zeichenmethode bereitstellen, um sich selbst in das Ornament auszugeben. Diese Methode wird aber erst ausgeführt, nachdem eine vollständige Platzierung der Elemente erfolgte. Illustration 5.32 veranschaulicht allgemein ein Ornamentelement mit seinen Eigenschaften, wie auch Ein- und Ausgaben.

5.4.2.6 Platzieren der Ornamentelemente

Die Platzierung der Ornamentelemente startet mit einem oder mehreren Wurzelementen im freien Ornamentbereich. Diese Wurzelemente sind die Ausgangspunkte für das Wachstum des Ornaments. Für die Platzierung neuer Elemente

⁶Der Objektcharakter der Ornamentelemente ist stark an die objektorientierte Programmierung angelehnt.

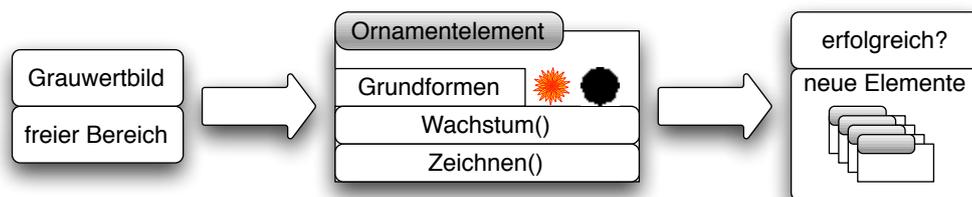
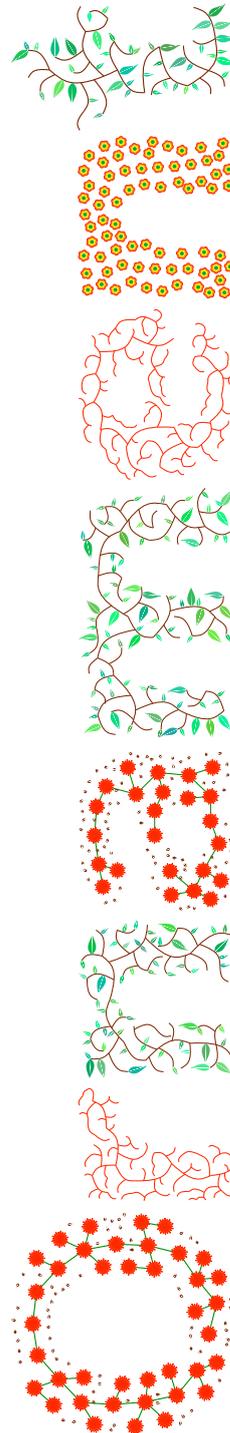


Abbildung 5.32: Ein-, Ausgaben, Eigenschaften und Funktionsweisen des Ornamentelements.

werden über die reduzierte MAT freie Bereiche im Graustufenbild gesucht, die absteigend nach ihrer Größe sortiert werden. Ausgehend vom größten Bereich werden diejenigen bereits gesetzten Elemente, die einen Mindestabstand zu diesem Bereich einhalten, in eine Liste aufgenommen. Die Wachstumsmethode für das erste Element der Liste wird schließlich für den ihm nächsten freien Bereich ausgeführt. Bei Erfolg wird das Grauwertbild aktualisiert, die neuen Elemente in die Liste aufgenommen und die MAT aktualisiert. Um nicht erneut eine vollständige Abstandstransformation durchführen zu müssen, werden nur die Pixel aktualisiert, deren aktueller Wert größer ist als ihr Abstand zum eben verwendeten Kreis. Der Vorgang wird mit einer neuen Liste freier Bereiche wiederholt. Sollte zum aktuell gewählten freien Bereich keines der nahe gelegenen Elemente erfolgreich neue Elemente gesetzt haben, so wird der nächstgrößere Bereich gewählt. Der Vorgang endet, wenn keine freien Bereiche mehr zur Verfügung stehen oder wenn kein Element erfolgreich neue Elemente platzieren konnte. Nach Abschluss des Wachstumsvorgangs, zeichnen sich die gesetzten Ornamentelemente selber. Zurzeit wird das fertige Ornament als Vektorgraphik im PDF-Format ausgegeben. Somit steht die Ornamentgraphik in beliebiger Qualität und Auflösung zur Verfügung. Die Graphik kann später bei der Bilderzeugung der Gebäudefassade als Höhenmodell zur Erzeugung eines Reliefs (engl. *bump mapping*) oder als Textur eingesetzt werden. Dadurch, dass sich jedes Ornamentelement selber zeichnet, sind natürlich der Art und Weise der Ausgabe keine Grenzen gesetzt und es können beliebige Formate erzeugt oder die Technik sogar in andere Programme eingebunden werden. [Abbildung 5.33](#) zeigt zum einen ein Binärbild – die zu füllenden Ornamentflächen – und zum anderen als Ergebnis das fertige Ornament. In diesem Beispiel wurden als Ornamentelemente Blüten, Blätter und einfache Kurven verwendet.

Ornament

(a) Binärbild.



(b) Fertiges Ornament.

Abbildung 5.33: Beispiel für ein Ornament.

5.5 Fenster und Türen

In diesem Abschnitt wird die Repräsentation der Fenster und Türen beschrieben. Als Basis dienen die in Abschnitt 5.2.1 definierten Wände. Für Fenster wie auch für Türen, wird ein Rechteck, die Fenster-/Türöffnung in einer Wand, definiert. Die Dimension der Öffnung wird relativ zur Wandhöhe und -breite angegeben, wie in Abbildung 5.34 zu sehen ist. Allgemein ist eine grobe, rechteckige Öffnung für ein Fenster oder eine Tür in einer Wand folgendermaßen definiert:

Definition 5.12. Grobe Fenster-/Türöffnung

Gegeben sind die folgenden Daten: Die Ausmaße der Wand hinsichtlich Höhe und Breite sind über w_h und w_b spezifiziert. $(\mathbf{w}_1^b, \mathbf{w}_r^b, \mathbf{w}_r^t, \mathbf{w}_1^t)$ beschreiben die Eckpunkte des Wandpolygons. Die Breite der Öffnung wird über die Parameter v_s, v_e , mit $v_s, v_e \in [0, 1]$ und $v_s < v_e$ so wie die Höhe über h_s, h_e , mit $h_s, h_e \in [0, 1]$ und $h_s < h_e$, gegeben.

Das resultierende Polygon der Öffnung ist $(\mathbf{f}_1^b, \mathbf{f}_r^b, \mathbf{f}_r^t, \mathbf{f}_1^t)$, wobei $\mathbf{f}_1^b = \mathbf{w}_1^b + v_s(\mathbf{w}_1^t - \mathbf{w}_1^b) + h_s(\mathbf{w}_r^b - \mathbf{w}_1^b)$. $\mathbf{f}_r^b, \mathbf{f}_r^t, \mathbf{f}_1^t$ berechnen sich analog.

Damit ergeben sich für die Öffnung die folgenden vier Kanten im Gegenuhrzeigersinn und von unten angefangen:

Kante	Punkte
unten	$(\mathbf{b}_l, \mathbf{b}_r)$, mit $\mathbf{b}_l = \mathbf{f}_l^b$ und $\mathbf{b}_r = \mathbf{f}_r^b$
rechts	$(\mathbf{r}_l, \mathbf{r}_r)$, mit $\mathbf{r}_l = \mathbf{f}_r^b$ und $\mathbf{r}_r = \mathbf{f}_r^t$
oben	$(\mathbf{t}_l, \mathbf{t}_r)$, mit $\mathbf{t}_l = \mathbf{f}_r^t$ und $\mathbf{t}_r = \mathbf{f}_1^t$
links	$(\mathbf{l}_l, \mathbf{l}_r)$, mit $\mathbf{l}_l = \mathbf{f}_1^t$ und $\mathbf{l}_r = \mathbf{f}_1^b$

5.5.1 Verfeinerung der Fenster- und Türöffnungen

Noch sind Fenster und Türen einfache rechteckige Öffnungen in Wänden. Um eine möglichst große Vielfalt verschiedener Fenster- und Türöffnungen zu erzielen, erhält jede Kante einen Satz von Parametern, mit denen sie modifiziert wird. Dabei wird zwischen den vier Kanten (unten, rechts, oben, links) keine Unterscheidung getroffen. Somit ist eine einheitliche Repräsentation gewährleistet, was zur Folge hat, dass die Beschreibung bzw. die Verfeinerung der Kanten und die Erzeugung der Geometrie erheblich vereinfacht wird.

Im Folgenden wird ein Verfahren vorgestellt, um jede der vier Kanten durch einen beliebigen Kantenzug zu verfeinern. Die Verfeinerung hat folgende Gestalt:

Definition 5.13. Verfeinerung durch einen Kantenzug

Gegeben ist die Kante $(\mathbf{f}_l, \mathbf{f}_r)$. Eine Verfeinerung der Kante $(\mathbf{f}_l, \mathbf{f}_r)$ durch einen

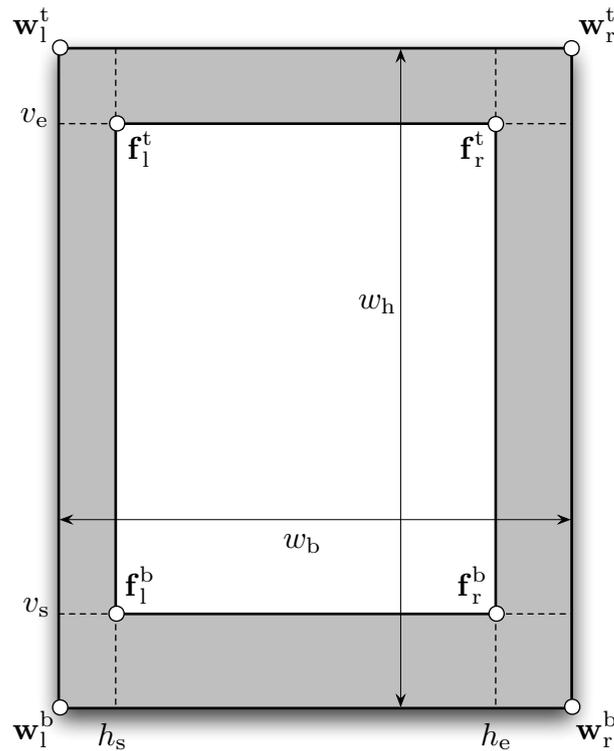


Abbildung 5.34: Definition einer Fenster- oder Türöffnung.

beliebigen Kantenzug wird über die Parameter $dist_l$, $dist_r$, $dist_b$, $size$ und der Beschreibung des Kantenzugs mit der Gesimsbeschreibung nach Definition 5.9 erzielt. Wertebereich der Parameter: $dist_l, dist_r, dist_b, size \in [0, 1]$

Mit den Parametern $dist_l$, $dist_r$ und $dist_b$ werden die Größe und der Abstand einer Kante ($\mathbf{f}_l, \mathbf{f}_r$) relativ zu ihrem Ursprung variiert und man erhält eine neue Kante (\mathbf{l}, \mathbf{r}). Der über den Parameter $size$ und der Kante (\mathbf{l}, \mathbf{r}) bestimmten Fläche $F_{\text{Kantenzug}}$ (in Abbildung 5.35 grau hinterlegt) fällt eine ganz besondere Aufgabe zu. Denn bei allen Kanten, die diesen Parameter gesetzt haben, kann die Kante innerhalb dieser Fläche zusätzlich über einen beliebigen Kantenzug verfeinert werden. Für den Verlauf des Kantenzugs wird die in Abschnitt 5.4.1 eingeführte Beschreibungssprache herangezogen. Anschließend wird der Kantenzug so skaliert, dass er ausschließlich in der Fläche $F_{\text{Kantenzug}}$ verläuft. Startpunkt bzw. Endpunkt des Kantenzugs ist \mathbf{l} bzw. \mathbf{r} . In Abbildung 5.35 ist ein Beispiel für eine mögliche Verfeinerung einer Kante dargestellt. Der resultierende Kantenzug ist gestrichelt gezeichnet und verläuft innerhalb der grau hinterlegten Fläche $F_{\text{Kantenzug}}$.

Bei der Verfeinerung entstehen aus den Kanten Flächen, die sich nach innen richten, also die Öffnung verkleinern. Damit die überlappenden, aneinander grenzenden Flächen aufgelöst werden können, werden die Kanten in zwei Gruppen eingeteilt: in *dominante* und *nicht-dominante* Kanten. Dominante Kanten erstre-

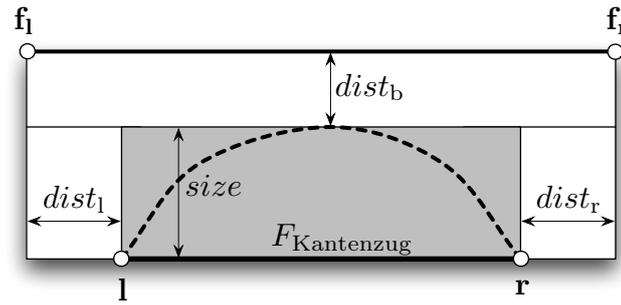


Abbildung 5.35: Verfeinerung der Kante durch einen Kantenzug.

cken sich über die gesamte Breite (f_l, f_r). Ihre anliegenden Kanten sind automatisch nicht-dominant – kurzum, die Kanten sind alternierend dominant und nicht-dominant.

In [Abbildung 5.36](#) ist ein Beispiel einer Verfeinerung einer Fensteröffnung für alle vier Kanten zu sehen. Die den Kanten zugehörigen vier Bereiche sind grau hinterlegt. Die Kanten oben (t_l, t_r) und unten (b_l, b_r) sind dominant. Im Beispiel ist nur für die obere Kante eine neue Fläche entstanden, die durch ein dunkleres Grau gekennzeichnet ist. Die Bereiche der nicht-dominanten Kanten links (l_1, l_r) und rechts (r_1, r_r) hingegen sind in einem helleren Grau gehalten. Ebenso besitzt die Kante oben (t_l, t_r) als einzige den Parameter *size*. Der gesamte Kantenzug, der durch diese Form der Verfeinerung hervorgegangen ist, ist gestrichelt dargestellt. In nachfolgender Liste ist gezeigt, wie für das Beispiel aus den vier Ausgangskanten bzw. Punkten der groben Fensteröffnung die neuen Kanten bzw. Punkte berechnet werden. Man beachte, dass die Parameter deren Wert gleich 0 ist, der besseren Übersicht wegen weggelassen wurden.

$$\begin{aligned}
 b_l &= f_l^b + d_l^{\text{unten}} \cdot (f_r^b - f_l^b) \\
 b_r &= f_r^b + d_r^{\text{unten}} \cdot (f_l^b - f_r^b) \\
 r_l &= f_r^b + d_b^{\text{rechts}} \cdot (f_l^t - f_r^t) \\
 r_r &= f_r^t + d_b^{\text{rechts}} \cdot (f_l^t - f_r^t) + size \cdot (f_l^b - f_l^t) \\
 t_l &= f_r^t + d_r^{\text{oben}} \cdot (f_l^t - f_r^t) + size \cdot (f_l^b - f_l^t) \\
 t_r &= f_l^t + d_l^{\text{oben}} \cdot (f_r^t - f_l^t) + size \cdot (f_l^b - f_l^t) \\
 l_1 &= f_l^t + d_b^{\text{links}} \cdot (f_l^t - f_r^t) + size \cdot (f_l^b - f_l^t) \\
 l_r &= f_l^b + d_b^{\text{links}} \cdot (f_l^t - f_r^t)
 \end{aligned}$$

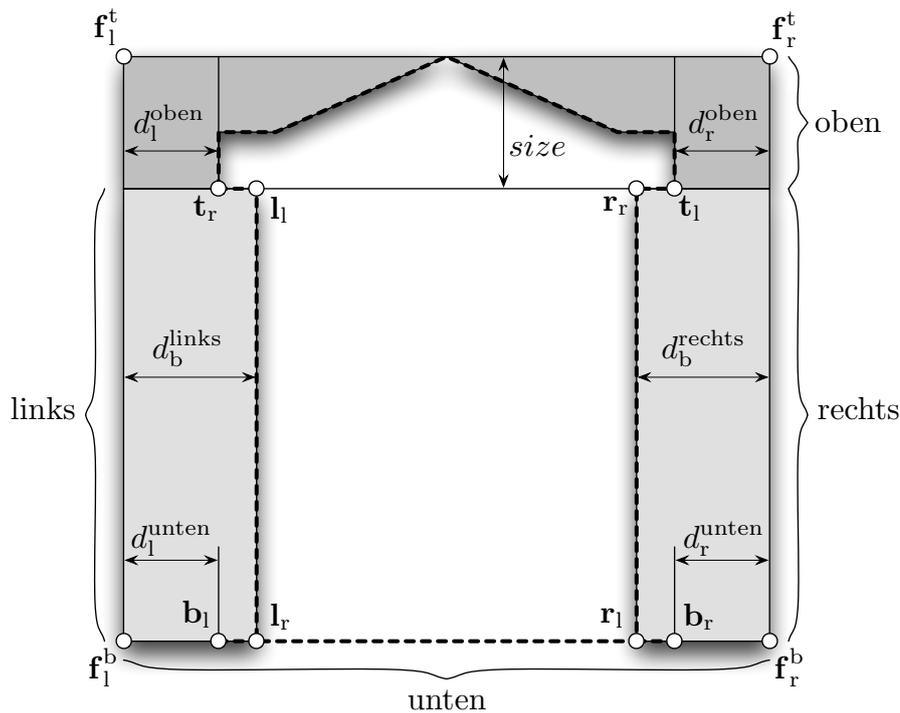


Abbildung 5.36: Beispiel für die Verfeinerung einer Fensteröffnung.

5.5.2 Fenster- und Türstöcke

Bisher wurden nur die Öffnungen in der Wand für Fenster und Türen betrachtet. Im nächsten Schritt werden für jede (verfeinerte) Kante Fenster- bzw. Türstöcke definiert. In diesem Fall handelt es sich um eine Art Rahmen mit bestimmter Dicke, der entlang einer Kante verläuft und über die Wand nach vorne hinausragen kann. Weiterhin muss unterschieden werden, ob ein Rahmen entlang einer Kante verläuft, deren Parameter $size$ gesetzt ist. Bei diesen Kanten verläuft der Rahmen ausschließlich innerhalb der Fläche $F_{Kantenzug}$. Bei allen anderen Kanten wird der Rahmen mit der angegebenen Dicke nach innen gezeichnet und muss auf Überschneidung mit angrenzenden Kanten geprüft werden. Dabei kommt uns wieder die Dominanz der Kanten zu Hilfe. Entsprechend seiner Nachbarn wird eine nicht-dominante Kante in ihrer Ausdehnung so angepasst, dass sie sich nahtlos in den Gesamtrahmen einfügt. Konkret wird ihre Ausdehnung um die Dicke der angrenzenden Kanten verringert. Zusätzlich kann der Kante noch ein rechteckiger Bereich ($F_{Ornament}$) hinzugefügt werden, der ein Ornament aufnehmen kann.⁷ Dazu werden die Parameter $dist_{orn,l}$, $dist_{orn,r}$ und $dist_{orn,b}$ benötigt. Mit ihnen wird, wie in Abbildung 5.37 dargestellt, das Rechteck $F_{Ornament}$ definiert. Zusätzlich kann für das

⁷Später, wenn das fertige Fenster mit Hilfe von Bilderzeugungsverfahren berechnet wird, kann das Ornamentbild als Relief und auch als Textur verwendet werden.

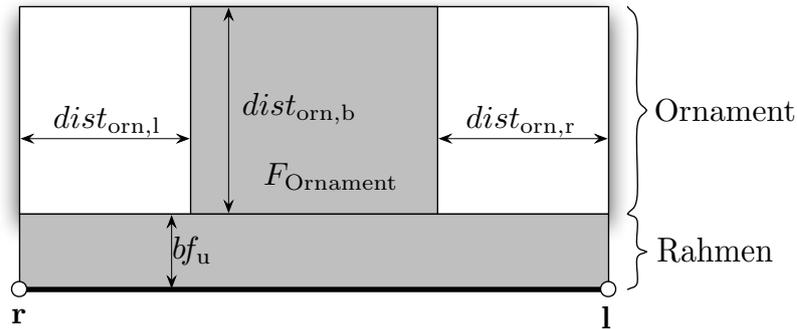


Abbildung 5.37: Zusammenspiel der Fenster-/Türstöcke mit dem Ornamentrechteck.

Rechteck noch ein Rahmen angegeben werden, dessen Profil unter Zuhilfenahme der Gesimsbeschreibung festgelegt wird. Die allgemeine Definition ist wie folgt:

Definition 5.14. Fenster-/Türstock mit optionalem Ornamentrechteck
 Gegeben ist die bereits verfeinerte Kante (\mathbf{l}, \mathbf{r}) . Ein Fenster- oder Türstock für die gegebene Kante bzw. deren Verfeinerung besitzt eine Dicke $bf \in [0, 1]$, einen Wandüberstand $o \in \mathbb{R}$ und hat die Länge der Kante bzw. des Kantenzugs.

Das optionale Rechteck für das Ornament wird über die Parameter $dist_{orn,l} \in [0, 1]$, $dist_{orn,r} \in [0, 1]$ und $dist_{orn,b} \in [0, 1]$ beschrieben. Das Ornament selbst wird mit der Technik aus Abschnitt 5.4.2 erzeugt. Ist zusätzlich noch ein Rahmen gewünscht, so wird dessen Profil über die Gesimsbeschreibung definiert.

Das Zusammenspiel der Parameter für die Beschreibung der Fenster- bzw. Türstöcke und des Ornamentrechtecks ist in Abbildung 5.37 dargestellt.

Abbildung 5.38 zeigt ein Beispiel, für ein Fenster mit einer verfeinerten Fensteröffnung und Fensterstock. Die Kante oben $(\mathbf{t}_l, \mathbf{t}_r)$ hat den Parameter *size* gesetzt und ist dominant. Die Kanten links $(\mathbf{l}_l, \mathbf{l}_r)$ und rechts $(\mathbf{r}_l, \mathbf{r}_r)$ sind nicht-dominant. Ihre Ausdehnungen werden entsprechend der Dicke der Kante oben (und zwar um den Wert von *size*) und der Dicke bf_u des Rahmens der Kante unten $(\mathbf{b}_l, \mathbf{b}_r)$ angeglichen.

In der Abbildung repräsentieren die pastellfarbenen Flächen die Rahmen. Wie im Vergleich zu Abbildung 5.36 gut zu sehen ist, wurden die Kanten links und rechts, entsprechend der Ausdehnung der Kanten oben und unten, angepasst. Ebenso gut ersichtlich ist, dass in beiden Abbildungen die Öffnung (die weiße Fläche innen) durch den Rahmen deutlich kleiner wurde.

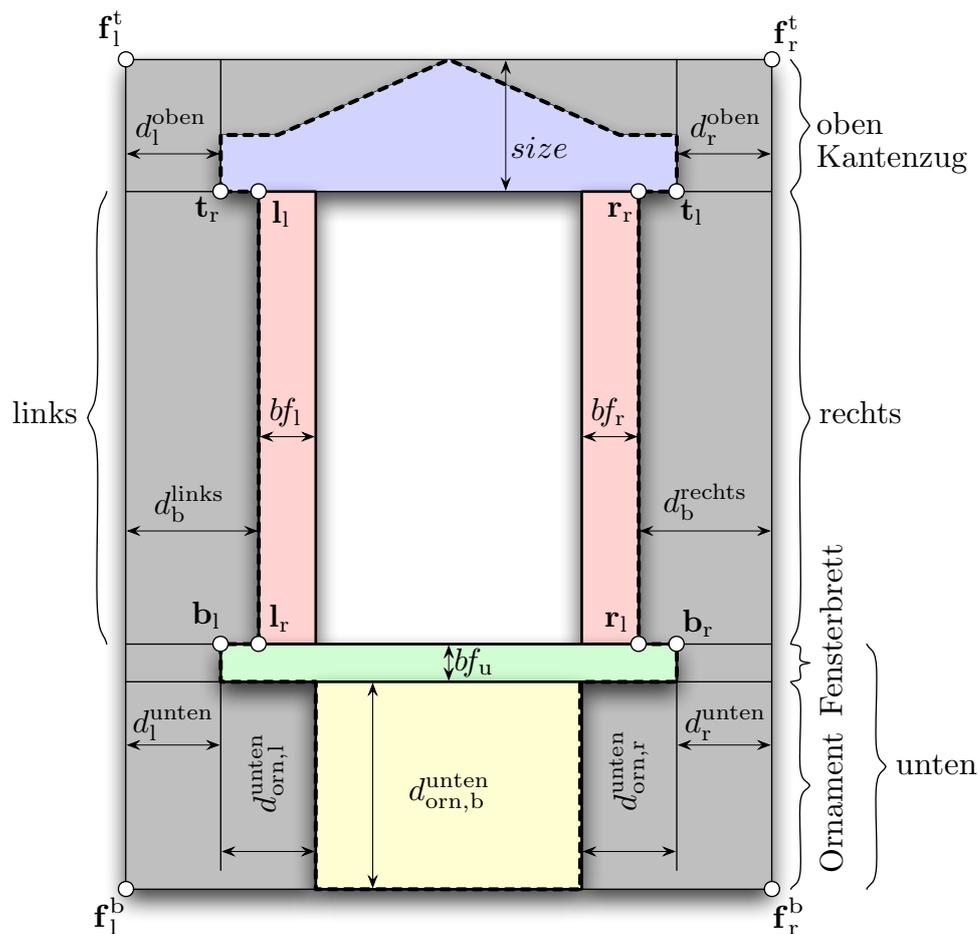


Abbildung 5.38: Beispiel für die Verfeinerung einer Fensteröffnung mit Fensterstock und Ornamentrechteck.

Für die Rahmen wurden drei verschiedene Stile implementiert. Sie alle besitzen die selben Grundparameter aus Definition 5.14 und unterscheiden sich nur hinsichtlich ihrer zusätzlichen Parameter, weshalb wir hier auf zusätzliche Definitionen verzichten. Bei den Rahmentypen handelt es sich um die folgenden vier Typen:

- **Einfach**

Dieser Stil ist gleich dem Stil, der mit der allgemeinen Definition gegeben ist.

- **Gesims**

Hier wird statt des einfachen Rahmens ein gleich dickes Gesims gezeichnet. Als zusätzlicher Parameter wird die Beschreibung des Gesimsprofils benötigt.

- **Doppelgesims**

Hier wird auf einem einfachen geraden Gesims noch ein weiteres Gesims in Form eines Giebels oder Bogens aufgebracht. Die Form des zweiten Gesimses ist über den verfeinerten Kantenzug vorgegeben. Als Parameter sind hier die beiden Gesimsbeschreibungen erforderlich.

- **Klinker**

Dieser Stil ist ähnlich zum ersten, einfachen Stil, nur dass zusätzlich entlang des Rahmens, der nun den Mörtel darstellt, einzelne, gleich große Steine eingebracht werden. Als Parameter sind die Steinbreite und der Überstand gegenüber dem Mörtel anzugeben.

In den Abbildungen 5.39 und 5.40, am Ende des nächsten Abschnitts, werden Beispiele für Fenster mit unterschiedlichen Öffnungen und Rahmen gezeigt.

5.5.3 Fenster- und Türrahmen

Nachdem die Fenster- und Türstöcke definiert sind, werden die eigentlichen Fenster mit Fensterrahmen und Fensterscheibe bzw. Türrahmen mit einfachen Türen eingefügt.

Der Unterschied zwischen Fensterstock und Fensterrahmen besteht darin, dass der Fensterstock eher als zur Wand zugehörig gesehen werden kann, wo hingegen der Fensterrahmen eher ein Bestandteil des Fensters selber ist. Für ein Fenster verläuft der Rahmen entlang des gesamten Kantenzugs des Fensterstocks und wird durch zwei Streben unterteilt. Zusätzlich wird der Kantenzug des Fensterstocks für das Polygon der Fensterscheibe verwendet. Ebenso verläuft bei der Tür der Rahmen entlang der (verfeinerten) Kanten. Als Tür wird entweder ein einzelnes Türpolygon oder zwei Türpolygone, die durch horizontales Teilen des Einzelpolygons hervorgehen, verwendet. Für die Rahmenform stehen die Stile *einfach* und *Gesims* zur Verfügung. Entsprechend bescheiden fällt die allgemeine Definition für Fenster- bzw. Türrahmen aus.

Definition 5.15. Fenster-/Türrahmen

Gegeben ist der fertige Fenster- bzw. Türstock.

Ein Fenster- oder Türrahmen besitzt eine Dicke $bf \in [0, 1]$, einen Wandüberstand $o \in \mathbb{R}$ und hat die Länge der Innenkante des Fenster- bzw. Türstocks. Optional kann für das Profil des Rahmens eine Gesimsbeschreibung angegeben werden.

In Abbildung 5.39 werden Beispiele zu verschiedenen Fensterformen und Fensterstocktypen dargestellt. Alle vier Fenster sind als Vektormodell berechnet worden. Der Fensterstock unten links besteht aus einfachen Rahmen zu seiner linken und rechten Seite. Sein Fensterbrett wie auch sein Giebel überragen die Seitenteile. Der Giebel besitzt ein Doppelgesims, bestehend aus einem geraden Gesims

unten und einem dachförmigen, dreifach gewinkeltem Gesims oben. Die Form unten rechts besitzt oben eine Rundung und gerade Seiten in Gesimsform, mit einfachem Fensterbrett. Der Fensterstock oben links ist fast wie der unten links, nur dass die einfachen Seitenteile durch ein Ziegelwerk ersetzt wurden. Die Fensterform oben rechts wurde an ihrer Oberseite durch eine Rundung mit geraden Seiten verfeinert. Die Seitenteile links und rechts, wie auch das Fensterbrett sind gerade. Außer dem Fensterbrett sind alle Teile in Ziegelform gehalten. Abbildung 5.40 zeigt zwei Beispiele für Fenster mit Ornamenten.



Abbildung 5.39: Verschiedene Fensterformen als Vektormodell.

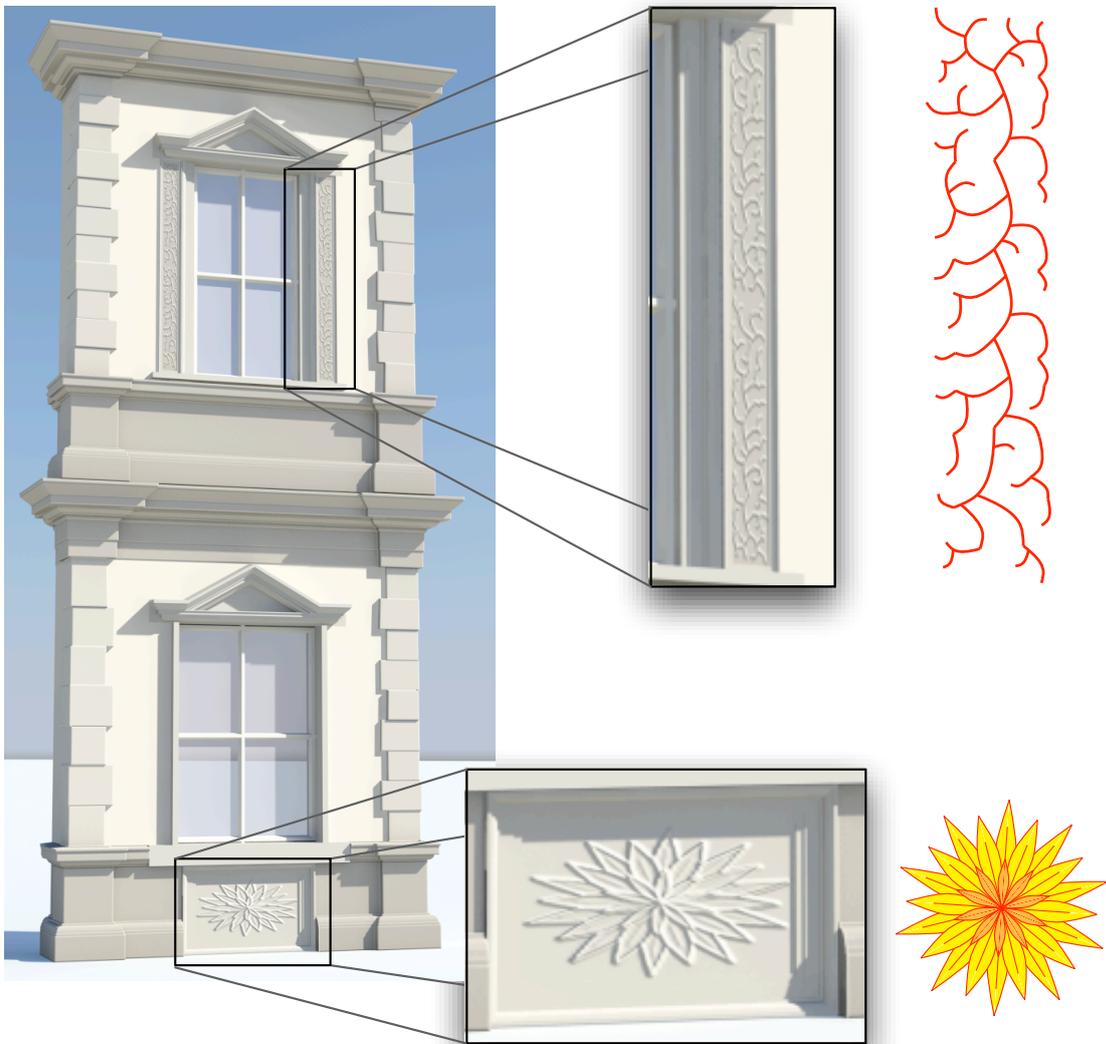


Abbildung 5.40: Ornamentierte Fensterstöcke mit Quellornamenten. Oben: Ornament der Seitenrahmen des oberen Fensters; unten: Ornament des unteren Fensters.

Kapitel 6

Semantik von Gebäudefassaden

In diesem Kapitel widmen wir uns zum einen dem generativen Aspekt dieser Arbeit und zum anderen der symbolischen Darstellung der Gebäudefassaden. In den vergangenen Kapiteln wurde die Strukturierung des groben Gebäudes bzw. der Fassadenelemente vorgestellt. Nun beschäftigen wir uns damit, in welcher Form diese Strukturinformationen zur Repräsentation einer vollständigen Gebäudefassade dargestellt werden müssen, um daraus automatisch die detaillierte Geometrie der Fassade erzeugen zu lassen. Dies führt uns zunächst in den Bereich der prozeduralen Modellierung. Hier werden grundsätzlich zwei Methodiken verwendet, um mit dem Computer Strukturen generativ zu erzeugen. Zum einen können sie, wie in Abschnitt 2.1.1 beschrieben, über Grammatiken konstruiert werden, zum anderen über Prozeduren, also Funktionen bzw. Methoden im Sinne von Programmiersprachen.

Wir entscheiden uns für den Ansatz der Prozeduren. Sie sind zwar fehleranfälliger, aber in ihrem Umfang mächtiger und vielfältiger, da sie Kontrollstrukturen, wie z. B. Schleifen, zulassen. Außerdem führt unser Vorhaben, eine Gebäudefassade zu beschreiben, in den Bereich der Wissensrepräsentation, ein Teilgebiet der Künstlichen Intelligenz. In der Wissensrepräsentation wird ein Ausschnitt der realen Welt modelliert. Dabei wird Wissen über die Realität in symbolischer Form repräsentiert, wobei Mechanismen vorgesehen sind, um Schlussfolgerungen zu ziehen und implizite Informationen aus dem Wissen abzuleiten. Im Umfeld dieser Arbeit beschränkt sich die Realität auf eine umfangreiche, aber nicht vollständige Repräsentation von Gebäudefassaden.

Im weiteren Verlauf des Kapitels werden zunächst allgemeine Anforderungen an die strukturelle Nachbildung der vollständigen Gebäudefassade formuliert. Anschließend wird auf die verwendete Technik eingegangen, mit der Gebäudestrukturen und Stile verwaltet werden. Die Technik definiert im gewissen Sinne die Syntax, in der die Fassaden beschrieben werden.

6.1 Anforderungen

In den vorigen Kapiteln wurden die Grundlagen der Darstellung der groben Gebäudestruktur und der detaillierten Fassade erarbeitet. Nun ist eine geeignete Form zu finden, mit der die erworbenen Grundlagen so umgesetzt werden, dass eine Gebäudefassade (also grobe Gebäudestrukturen wie auch detaillierte Fassaden) einheitlich und vollständig erfasst werden kann. Die Strukturierung detaillierter Fassadenelemente, wie die eines Fensters, Gesimses usw. wird hier zu einem *Stil* zusammengefasst. Ein Stil muss sich dabei nicht nur auf ein einzelnes Fenster, eine Tür oder ein Gesims beziehen, er kann sich über Basiswände und Stockwerke bis hin zum gesamten Gebäude erstrecken. Die wichtigste Anforderung an die Darstellung der Gebäudefassade ist die Trennung der groben *Gebäudestruktur* vom gewählten *Stil*. Damit ist gemeint, dass die Gebäudestruktur und der Stil getrennt voneinander existieren können. Das erfordert eine generische Beschreibung des Gebäudes ohne Stilinformation, die trotzdem die Anwendung und Austauschbarkeit beliebiger Stile gewährleistet. Eine weitere Anforderung ist, dass Änderungen an der groben Gebäudestruktur und an Details der Fassade einfach durchgeführt und schnell sichtbar gemacht werden können. Folglich muss die Fassade parametrisiert vorliegen. Diese Parametrisierung soll aber in Form einer symbolischen Repräsentation der Gebäudefassade erfolgen, womit wir das Gebiet der Wissensrepräsentation betreten. Hier muss Wissen über die Gebäudefassade modelliert werden. Aus dem entwickelten Modell müssen später räumliche Beziehungen leicht ableitbar sein, um aneinander grenzende Fassadenelemente erkennen und ggf. einander anpassen zu können. Dabei ist es wichtig, das Modell in zwei Bereiche zu unterteilen, wie Liebig et al. [LFL01] gezeigt haben:

- Im *Konzeptmodell* ist eine abstrakte, hierarchische Beschreibung der Elemente des Gebäudes und der Fassade als Objekte festzulegen. Ebenso werden hier die Eigenschaften und Beziehungen der Objekte untereinander abgebildet.
- Im *Weltmodell* werden konkrete Instanzen der Objekte des Konzeptmodells zusammengefasst und definieren dadurch eine konkrete Ausprägung einer Fassade.

Eine hierarchische Struktur ist bereits über die Definitionen der groben Gebäudestruktur zur Grundrissbeschreibung und der Stile, die mittels der Fassadenbeschreibung festgelegt werden, gegeben. Eine Baumstruktur zur Abbildung bietet sich deshalb an. In den folgenden drei Abschnitten werden verschiedene Techniken vorgestellt, die sich für eine hierarchische Darstellung des Gebäudemodells prinzipiell eignen. Im Anschluss werden die Techniken auf Tauglichkeit hin evaluiert.

6.2 Typisierte Graphen

Mit Graphen können sehr effizient komplexe Strukturen aufgebaut und abgebildet werden. Erweitert man die Graphen noch um Typen und Attribute, so können damit Datenbasen, wie man sie von Datenbanken her kennt, verwaltet werden. Im Gegensatz zu Datenbanken können mit Graphen Änderungen und Manipulationen in der Struktur schnell durchgeführt werden. Während bei Datenbanken eine Änderung der Struktur (das Löschen von Attributen) meist ein Löschen und neu Erzeugen ganzer Tabellen erfordert – wobei die Inhalte zwischengespeichert, ggf. konvertiert und anschließend wiederhergestellt werden müssen –, können derartige Vorgänge in Graphen durch lokale Modifikationen erfolgen. Hierbei sind nur einzelne Knoten oder Kanten betroffen, die gelöscht, erzeugt oder deren Typ modifiziert werden.

Typisierte Graphen (kurz: TGraphen) sind gerichtete, typisierte und attributierte Graphen. Im Weiteren wollen wir das Konzept der TGraphen näher beleuchten, wobei die Arbeit von Ebert et al. [EF95] als Grundlage dient. Ein TGraph besitzt folgende Eigenschaften (nach Dahm [DW98]):

- Knoten und Kanten sind eigenständige, identifizierbare Objekte;
- mehrere Kanten zwischen den selben Knoten sind möglich (Mehrfachkanten);
- die Kanten sind gerichtet;
- der Graph kann zur Laufzeit verändert werden, d. h. Knoten wie auch Kanten können eingefügt, manipuliert und gelöscht werden;
- über alle Kanten (pro Knoten) ist eine lokale Ordnung definiert;
- zur Unterscheidung von Knoten- und Kantenklassen werden Knoten und Kanten typisiert und
- Knoten und Kanten können mit Attributen versehen werden, wobei das Attributierungsschema vom jeweiligen Knoten- bzw. Kantentyp abhängig ist.

6.2.1 Gerichtete Graphen

Ein gerichteter Graph besteht aus einer endlichen Menge K von Knoten und einer endlichen Menge E gerichteter Kanten. Jede Kante $e \in E$ verbindet genau einen Startknoten $k \in K$ mit genau einem Zielknoten $l \in K$. Für $k = l$ bildet die Kante e eine Schleife.

Die Bezüge der Kanten auf ihre Start- und Zielknoten werden über die zwei Funktionen $\alpha : E \rightarrow K$ und $\omega : E \rightarrow K$ beschrieben. Dabei bezeichnen $\alpha(e)$

den Start- und $\omega(e)$ den Zielknoten der Kante e und es gilt: $\forall e \in E : \exists k, l \in K \wedge \alpha(e) = k \wedge \omega(e) = l$. Gilt $k, l \in K$ und $e, f \in E$ mit $e \neq f$ und $\alpha(e) = \alpha(f) = k$, $\omega(e) = \omega(f) = l$, so sind e und f Mehrfachkanten.

6.2.2 Typen

Knoten wie auch Kanten haben in vielen Anwendungen von Graphen unterschiedliche Bedeutungen. Um diesen Unterschieden gerecht zu werden, wird für Knoten und Kanten ein Typensystem eingeführt. Für Kanten wird damit festgelegt, welche zwei Typen von Knoten sie miteinander verbinden und für Knoten wird vorgegeben, welcher Typ als eingehende und welcher als ausgehende Kante zulässig ist. Dadurch, dass Knoten und Kanten einander bedingen, können zum einen Inkonsistenzen schnell entdeckt und zum anderen kann sicher gestellt werden, dass über ein korrektes Typensystem nur zulässige Graphen erzeugt werden. Zum anderen können Knoten und Kanten ähnlichen Charakters zu (abstrakten) Klassen bzw. Obertypen, wie auch bei der objektorientierten Programmierung, zusammengefasst werden. Es können für ausgehende und eingehende Kanten eines bestimmten Typs noch Kardinalitäten angegeben werden, um die Anzahl eingehender und ausgehender Kanten eines bestimmten Typs für einen Knoten zu beschränken.

6.2.3 Attribute

Im Allgemeinen ist die reine Typisierung der Graphen noch nicht ausreichend. Meist ist es sinnvoll, noch zusätzliche Attribute zu einem Knoten oder einer Kante zu speichern. Attribute bilden hier einen Bezeichner auf einen Wert ab (Schreibweise: *Bezeichner* \mapsto *Wertebereich*). Die Attribute eines Knotens oder einer Kante müssen lokal eindeutig sein. Im Fall von Klassenhierarchien erben die Untertypen eines Knoten oder einer Kante die Attribute ihres Obertyps.

6.3 Wissensbasen

In der Wissensrepräsentation wird Wissen über einen Ausschnitt der realen Welt repräsentiert. Wie schon von den Datenbanken her bekannt, kann auch hier das abgebildete Wissen unter verschiedenen Gesichtspunkten ausgewertet werden. Was Wissensbasen gegenüber herkömmlichen Datenbanken auszeichnet, ist die Eigenschaft, aus vorhandenen Daten durch Inferenzmechanismen zuvor implizite Kenntnisse bzw. Informationen abzuleiten. Im Weiteren beschränken wir uns auf die terminologische Wissensrepräsentation. Nach Baader et al. [BBH⁺92] stellt sie eine Erweiterung der semantischen Netze dar bzw. hat sich daraus entwickelt und besitzt eine durch die terminologischen Logiken geprägten Charakter. Damit war

die Grundlage gelegt, Schlussfolgerungen aus den Daten zu ziehen bzw. Schlussfolgerungsalgorithmen zu erstellen und zu untersuchen. Als klassische Vertreter sind hier *LOOM* [MB91], *KL-ONE* [BS85] und *CLASSIC* [BBMR89] zu nennen, wohingegen *RacerPro* [RS08] zu den aktuellen Systemen zählt. Im Folgenden gilt unser Interesse den semantischen Netzen, die den Ursprung der genannten Systeme bilden. Nach Sowa [Sow91] bietet ein semantisches Netz Strukturen zur Repräsentation von Wissen. Die Netze werden in Form zusammenhängender Knoten und Kanten dargestellt, womit sich eine gewisse Verwandtschaft zu den TGraphen feststellen lässt. Die Knoten des Netzes verkörpern meist Objekte der realen Welt bzw. die Eigenschaften der Objekte. Über die Kanten werden Beziehungen bzw. Relationen zwischen den Objekten modelliert. Wie auch die TGraphen, besitzen semantische Netze ein Typensystem für Knoten und Kanten. Ebenso trifft man hier das Konzept der Ober- und Untertypen an, wobei diese meist als Typhierarchie oder Taxonomiehierarchie bezeichnet werden. Eigenschaften der Obertypen werden auch den Untertypen vererbt, einschließlich ihrer Relationen zu anderen Objekten. Hier werden hauptsächlich Beziehungen der Form *A ist ein Untertyp von B* (kurz *A ist ein B*) und *A hat als Eigenschaft bzw. besitzt B* (kurz *A hat B*) modelliert, was im Beispiel in Abbildung 6.1 gut zu sehen ist. Da sich die terminologischen Wissensbasen aus den semantischen Netzen entwickelt haben, liegt ihnen noch meist deren Sichtweise zugrunde. Im Unterschied zu den semantischen Netzen und auch den TGraphen stehen hier zusätzlich Inferenzmechanismen und regelbasiertes Schließen auf den Netzen im Vordergrund, um implizites Wissen ableiten zu können.

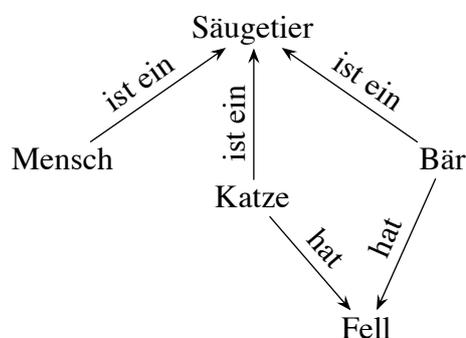


Abbildung 6.1: Beispiel für ein einfaches semantisches Netz.

6.4 Datenbanken

Datenbankverwaltungssysteme dienen der zentralen, sicheren und effizienten Verwaltung sehr großer Datenmengen. Heute werden meist relationale Datenbanken eingesetzt, vor allem in verteilten heterogenen Umgebungen. Ihre Hauptaufgabe

besteht darin, als Dienstgeber hoch verfügbar und zuverlässig viele Abfragen von Dienstnehmern in möglichst kurzer Zeit zu bewältigen (*Client-Server-Modell*). Ihre Stärken sind eine konsistente Speicherung und eine sehr hohe Sicherheit der Daten gegenüber Systemausfällen.

Relationale Datenbanken basieren auf der Relationenalgebra, einer formalen Sprache, die maßgeblich von Edgar F. Codd [Cod70] entwickelt wurde. Wie bei den TGraphen und semantischen Netzen liegt auch hier der Fokus auf einer semantischen Datenmodellierung. In diesem Modell existiert als einzige Datenstruktur nur die Tabelle bzw. Relation. Dadurch gibt es im Gegensatz zu den TGraphen und semantischen Netzen keine explizite Unterscheidung zwischen Objekten und Beziehungen. Tabellen werden über eindeutige Namen referenziert, womit ein Typensystem für Objekte und Beziehungen direkt gegeben ist. Eigenschaften werden über Attribute der Tabellen dargestellt. Attribute müssen atomar sein, d. h. sie besitzen keine weitere Struktur und können auch nicht weiter unterteilt werden.

Relationale Datenbanken unterstützen keine Hierarchien im Sinne der Objektorientierung. Eine Vererbung von Eigenschaften bzw. Attributen ist somit nicht möglich.¹ Die Beschreibung des semantischen Modells erfolgt im *Gegenstands-Beziehungs-Modell* (*ER-Modell*), das zur Klasse der nicht-formalen Modelle gehört (nach Lockemann [LKK93]). Das ER-Modell dient hauptsächlich der anschaulichen Modellierung des abzubildenden Sachverhalts, woraus anschließend das Datenbankschema (also die Tabellen) gewonnen wird. Das semantische Modell im Datenbankschema wird, wenn einmal festgelegt, meist nur statisch verwendet und nicht mehr verändert. Dynamische Anpassungen sind nur recht schwierig durchführbar und bedingen Anpassungen der Software auf Seite der Dienstnehmer. Bezüglich der ER-Modelle stellen die TGraphen eine Erweiterung dar und werden deswegen auch als *erweiterte ER-Modelle* bzw. *EER-Modelle* bezeichnet. Im Gegensatz dazu dienen TGraphen nicht nur der Beschreibung semantischer Modelle, sondern werden direkt als konkrete Implementierung im Computer eingesetzt. Von den bisher vorgestellten Techniken besitzen nur die relationalen Datenbanken ein standardisiertes Protokoll zur Datenmanipulation. Dabei handelt es sich um die deklarative Kommandosprache *SQL* (engl. *Structured Query Language*). Mit ihr können recht einfach Daten abgefragt, verändert und eingepflegt werden.

6.5 Vergleich zwischen TGraphen, Wissensbasen und Datenbanken

In den letzten drei Abschnitten wurden drei Techniken zur semantischen Modellierung vorgestellt. In Anbetracht der in Abschnitt 6.1 gestellten Anforderungen

¹Eine Lösung hierfür sind objektorientierte Datenbanksysteme, siehe Lockemann [LKK93].

werden die drei Techniken anhand einer Reihe von Kriterien miteinander verglichen. Am Ende wird das Ergebnis in Tabelle 6.1 zusammengefasst.

Mit jeder der Techniken kann ein *semantisches Modell* beschrieben werden. *Objekte und Beziehungen* zwischen Objekten werden nur von den TGraphen und den Wissensbasen als voneinander unabhängige Strukturen modelliert. Bei den Datenbanken wird zwischen Objekten und Beziehungen nicht unterschieden, sondern sie werden einheitlich als Tabellen dargestellt. *Ober- und Untertypen mit Vererbung* werden ebenso nur von den TGraphen und Wissensbasen unterstützt. Die *Anpassbarkeit*, also Änderungen am Konzeptmodell, ist bei allen Techniken gegeben, wobei bei den Datenbanken Änderungen mit einem relativ großem Aufwand verbunden sind. *Inferenzmechanismen und regelbasiertes Schließen* sind nur bei den Wissensbasen vorhanden. Die *Netzwerkfähigkeit* ist bei den Datenbanken besonders gut entwickelt. Sie alleine sind für einen sicheren konkurrierenden Mehrbenutzerbetrieb über Netzwerk ausgelegt. Für *LOOM* wurde von Liebig et. al [LFL01] eine Netzwerkerweiterung implementiert, die aber keinen konkurrierenden Zugriff erlaubt. OKBC (*Open Knowledge Base Connectivity*) [CFF⁺98] ist in Anlehnung an ODBC (*Open Database Connectivity*) eine Anstrengung seitens des *SRI*² (*Stanford Research Institute*) und des *Knowledge Systems Laboratory of Stanford University*, einen einheitlichen Standard für den netzwerktransparenten Zugriff für Wissensrepräsentationssysteme zu entwickeln. Für die *Datenkonsistenz* implementieren meist nur die Datenbankverwaltungssysteme eigene Techniken (und umgehen damit die Dateiverwaltung des Betriebssystems), um korrekte Daten auch nach Systemabstürzen zu gewährleisten. Ebenso unterstützen nur die Datenbanken ein die Klasse der Datenbanksysteme umgreifendes *standardisiertes Protokoll zur Datenmanipulation*. Aktuelle Systeme wie *RacerPro* verwenden als Beschreibungssprache *OWL (Web Ontology Language)* [W3C04b], eine Spezifikation des *W3C*, um Ontologien anhand einer formalen Beschreibungssprache austauschen zu können.

Da in dieser Arbeit nur ein Prototyp erstellt wurde, lag das Augenmerk hauptsächlich darauf, ein gut bedienbares System zur Datenverwaltung und -manipulation zu haben. Vor allem musste sich die Integration in den Prototyp einfach gestalten. Weiterhin sollte sich räumliches Schließen leicht realisieren lassen. Die Unterstützung von semantischen Modellen, Objekten und Beziehungen zwischen Objekten war ein weiteres Kriterium, das erfüllt sein musste. Ebenso sollten die semantischen Modelle schnell zu ändern sein, ohne dass bereits vorhandene Daten verloren gingen oder aufwendig umkopiert werden mussten. Somit schieden die Datenbanken schon recht früh aus. Wenn auch reizvoll, da Inferenzmechanismen und regelbasiertes Schließen vorhanden, konnten *LOOM* und *RacerPro* nicht überzeugen. *LOOM*, das unter Lisp läuft, ist schlicht nicht mehr zeitgemäß, da das System relativ umständlich aufzusetzen ist. Außerdem benötigt die von

²Artificial Intelligence Center of SRI International; <http://www.ai.sri.com/~okbc/>

	TGraphen	Wissensbasen	Datenbanken
semantisches Modell	+	+	+
Objekte und Beziehungen	+	+	o
Typensystem	+	+	+
Ober- und Untertypen mit Vererbung	+	+	-
Anpassbarkeit	+	+	o
Inferenzmechanismen und regelbasiertes Schließen	-	+	-
Netzwerkfähigkeit	-	o	+
Datenkonsistenz	-	-	+
standardisiertes Protokoll zur Datenmanipulation	-	o	+

Tabelle 6.1: Vergleich von TGraphen, Wissensbasen und relationalen Datenbanken hinsichtlich der gestellten Anforderungen.

Liebig et al. entwickelte Netzwerkkomponente für Lisp ein spezielles Lisp-System und einen speziellen C-Compiler, die unter aktuellen Unix-Systemen nicht mehr verfügbar sind. *RacerPro* ist ein kommerzielles System, dessen Lizenz für akademische Institutionen nur für sechs Monate Gültigkeit hat – ein Kauf wäre deutlich zu teuer gewesen. Des Weiteren war nicht klar, ob das System und der Prototyp reibungslos miteinander gearbeitet hätten. Aus diesen Gründen wurden die TGraphen gewählt. Anfänglich war geplant, das System *GraLab* [DW98] zu verwenden. Da aber der Prototyp plattformunabhängig in Python implementiert wurde, musste von *GraLab* für Python ein plattformunabhängiges Modul erstellt werden, was aber technisch nicht möglich war. Deswegen wurden die Grundfunktionalität der TGraphen vollständig in Python neu implementiert. Damit war nicht nur eine einfache Integration in den Prototypen gewährleistet, sondern auch eine einfache Benutzung. Für das semantische Modell der Gebäudefassade wurden Methoden entwickelt und implementiert, die räumliche Abhängigkeiten durch Traversierung des Graphen ermitteln, wobei auf formale Mechanismen verzichtet wurde.

6.6 Fassadensemantik mit TGraphen

In diesem Abschnitt wird das *Konzeptmodell* (das semantische Modell) der Gebäudefassade beschrieben. Im *Weltmodell* sind die Instanzen der Objekte des *Konzeptmodells* zusammengefasst. Die Abbildung der beiden Modelle wird vollständig von den TGraphen übernommen. In diesem Fall wird nur eine abgeschwächte Form

der TGraphen benötigt, denn das Konzeptmodell der Gebäudefassade kommt vollständig ohne Unter- und Obertypen aus. Folglich müssen weder eine allgemeine Vererbungsstrategie für die TGraphen implementiert noch eine spezielle Typisierung realisiert werden. Es ist anzumerken, dass eine „einfache Vererbung“ einzelner Attribute vorgesehen ist. Dabei handelt es sich um spezielle Attribute eines Konzepts, die, falls nicht gesetzt, vom Vorgängerkonzept übernommen werden. Kardinalitäten hinsichtlich eingehender und ausgehender Kanten eines bestimmten Typs werden zurzeit nicht unterstützt. Ebenso werden keine Restriktionen auf Wertebereiche der Attribute vorgenommen.

6.6.1 Konzepte und Relationen allgemein

Die Funktionalität der verwendeten TGraphen ist wie folgt gegeben. Die konkrete Realisierung der TGraphen stellt, wie in Abschnitt 6.2 beschrieben, typisierte Knoten und Kanten zur Verfügung, die zusätzlich mit Attributen versehen werden können. Dabei verkörpern die Knoten die Konzepte und die Kanten die Relationen. C sei die Menge der Knoten, R die Menge der Kanten und A die Menge der Attribute. Die Eindeutigkeit der Elemente der Knoten- und Kantenmenge kann in diesem Fall auch als Typisierung der Knoten und Kanten gesehen werden. Die Attribute werden den Knoten und den Kanten durch die Funktion $\tau : (C \cup R) \rightarrow \mathcal{P}(A)$ zugewiesen. Dabei bezeichnet $\mathcal{P}(A)$ die Potenzmenge von A . Zusätzlich kann für jedes Attribut über die Funktion $\sigma : (C \times A) \rightarrow \{\top|\perp\}$ angegeben werden, ob die Instanz bei nicht gesetztem Attribut den Wert des Attributs der Instanz des Vorgängerkonzepts verwenden soll. Sollte der Vorgänger ebenso das Attribut nicht gesetzt haben, so wird rekursiv weiter gesucht. Falls keiner der Vorgänger das Attribut gesetzt hat oder besitzt, so bleibt das Attribut nicht gesetzt. Jede Kante $e \in R$ hat genau einen Startknoten $l \in C$ und einen Zielknoten $k \in C$. Die Zuweisung der Knoten wird über die beiden Funktionen $\alpha : R \rightarrow C$ und $\omega : R \rightarrow C$ beschrieben. Dabei bezeichnen $\alpha(e)$ den Start- und $\omega(e)$ den Zielknoten der Kante e .

Mit den Knoten und Kanten wird das Konzeptmodell der Gebäudefassaden beschrieben. Jeder Knoten repräsentiert dabei ein Element der Gebäudefassade. Da bei der Erzeugung von Knoten und Kanten im TGraphen eine Konsistenzprüfung durchgeführt wird, werden die folgenden Bedingungen automatisch eingehalten:

- Jeder Knoten und jede Kante ist eindeutig.
- Jede Kante verbindet genau einen Startknoten mit einem Endknoten.
- Attribute müssen nur lokal für ihren Knoten bzw. ihre Kante eindeutig sein.

Für konkrete Ausprägungen der Fassaden werden im Weltmodell Instanzen von Knoten und Kanten erzeugt.

6.6.2 Konzepte und Relationen der Gebäudestruktur

In diesem Abschnitt werden die Konzepte und die Beziehungen der groben Struktur des Gebäudes definiert. Es werden aber nur einzelne, ausgewählte Konzepte und Relationen vorgestellt. Eine vollständige Liste der Definition für die Gebäudestruktur und den Fassadenstil ist im Anhang A aufgeführt. Die Darstellung der Konzepte wird über den *Konzeptnamen* und die *Attribute* vorgenommen und die Beziehungen werden über ihren *Namen*, ihre *Start-* bzw. *Zielknoten* und ihre *Attribute* definiert. Die neuen Konzepte, Beziehungen und Attribute werden ihren Mengen hinzugefügt. Zu Beginn sind alle Mengen leer: $C, R, A = \emptyset$.

Definition. Konzept Grundstein

Die grobe Gebäudestruktur startet mit dem Konzept „Grundstein“. Von ihm aus wird ein Stockwerk, bestehend aus einzelnen Grundrissmodulen aufgebaut, wie in Abschnitt 4.2 beschrieben. Seine beiden Attribute „Höhe“ und „Breite“ legen die Höhe des Stockwerks und die Breite seiner Mauern fest.

$$\begin{aligned} C &\cup \text{Grundstein}, \\ A &\cup \{\text{Höhe}, \text{Breite}\}, \\ \tau(\text{Grundstein}) &= \{\text{Höhe}, \text{Breite}\} \end{aligned}$$

Definition. Konzept Grundrissmodul

Als nächstes werden die „Grundrissmodule“ benötigt, aus denen die grobe Gebäudestruktur besteht. Das Modul hat nur das Attribut „Breite“, das die Breite seiner Mauern bezeichnet. Optional kann hier noch der Typ des Moduls angegeben werden, also ob es sich hierbei um den Haupttrakt, einen Nebenflügel, einen Mauer vorsprung oder einen Erker handelt. Die Höhe des Grundrissmoduls entspricht der seines zugehörigen Grundsteins.

$$\begin{aligned} C &\cup \text{Grundrissmodul}, \\ A &\cup \text{Typ}, \\ \tau(\text{Grundrissmodul}) &= \{\text{Typ}, \text{Breite}\}, \\ \sigma(\text{Grundrissmodul}, \text{Breite}) &= \top \end{aligned}$$

Definition. Konzept Dach

Die in den Definitionen 4.12 für ein Flachdach und 4.13 für ein allgemeines Dach beschriebenen Dachformen werden hier in einem einzigen „Dach“-Konzept zusammengefasst. Das Dach beherbergt eine ganz Reihe von Attributen, wobei zusätzlich das Attribut „Typ“ eingeführt wird, um zwischen den beiden grundlegenden Dachformen unterscheiden zu können.

$$\begin{aligned} C &\cup \text{Dach}, \\ A &\cup \{\text{Kante}_i, \text{Kante}_j, \text{c_Kante}_i, \text{d_Kante}_j, \text{a_First}, \text{b_First}\}, \\ \tau(\text{Dach}) &= \{\text{Typ}, \text{Kante}_i, \text{Kante}_j, \text{c_Kante}_i, \text{d_Kante}_j, \\ &\quad \text{a_First}, \text{b_First}, \text{Höhe}\} \end{aligned}$$

Damit sind einige wichtige Konzepte, die für die grobe Gebäudestruktur notwendig sind, definiert. Nun wird ein Auszug aus den Beziehungen zwischen den Konzepten angegeben.

Definition. Relation `initiales_Grundrissmodul`

Diese Relation setzt dem Konzept „Grundstein“ das initiale Grundrissmodul, an das alle weitere Module und Stockwerke angeknüpft werden. Diese Relation hat keine Attribute.

$$\begin{aligned} R \cup & \text{initiales_Grundrissmodul,} \\ \alpha(\text{initiales_Grundrissmodul}) &= \text{Grundstein,} \\ \omega(\text{initiales_Grundrissmodul}) &= \text{Grundrissmodul} \end{aligned}$$

Definition. Relation `naechstes_Stockwerk`

Diese Relation definiert den Grundstein für das nachfolgende Stockwerk und verbindet somit Grundsteine miteinander. Diese Relation hat keine Attribute.

$$\begin{aligned} R \cup & \text{naechstes_Stockwerk,} \\ \alpha(\text{naechstes_Stockwerk}) &= \text{Grundstein,} \\ \omega(\text{naechstes_Stockwerk}) &= \text{Grundstein} \end{aligned}$$

Hier wird ein klassischer Zyklus definiert, der aber so im Weltmodell nicht zustande kommen kann, da der Prototyp im Weltmodell auf unzulässige Zyklen prüft.

Definition. Relation `intra_Verbindung`

Damit werden die Grundrissmodule aneinander gefügt, um den Grundriss eines Stockwerks zu formen. Hier sind die Verbindungstypen Kante-Kante- und die beiden Punkt-Punkt-Verbindungen zusammengefasst. Die Form der Verbindung wird über das Attribut „Typ“ spezifiziert. Alle weiteren Attribute entspringen den Definitionen 4.2, 4.3 und 4.4.

$$\begin{aligned} R \cup & \text{intra_Verbindung,} \\ A \cup & \{Kante_k, a_Kante_i, b_Kante_j\}, \\ \tau(\text{intra_Verbindung}) &= \{Typ, Kante_k, Kante_i, Kante_j, \\ & \quad a_Kante_i, b_Kante_j\}, \\ \alpha(\text{intra_Verbindung}) &= \text{Grundrissmodul,} \\ \omega(\text{intra_Verbindung}) &= \text{Grundrissmodul} \end{aligned}$$

Definition. Relation `inter_Verbindung`

Mit dieser Relation wird eine das Stockwerk übergreifende Verbindung für Grundrissmodule definiert. Dabei wird ein Grundrissmodul mit dem Grundrissmodul des vorigen Stockwerks verbunden. Diese Relation besitzt keine Attribute.

$$\begin{aligned} R \cup & \text{inter_Verbindung,} \\ \alpha(\text{inter_Verbindung}) &= \text{Grundrissmodul,} \\ \omega(\text{inter_Verbindung}) &= \text{Grundrissmodul} \end{aligned}$$

Definition. Relation *hat_Dach*

Hier wird ein Grundrissmodul-Konzept mit einem Dach-Konzept verbunden. Diese Relation hat keine Attribute.

$$\begin{aligned}
 R &\cup \textit{hat_Dach}, \\
 \alpha(\textit{hat_Dach}) &= \textit{Grundrissmodul}, \\
 \omega(\textit{hat_Dach}) &= \textit{Dach}
 \end{aligned}$$

In Abbildung 6.2 ist das semantische Modell der groben Gebäudestruktur graphisch darstellt. Die Abhängigkeiten zwischen den Konzepten sind gut zu erkennen. Die Attribute der Relationen wurden zugunsten der besseren Übersicht weggelassen.

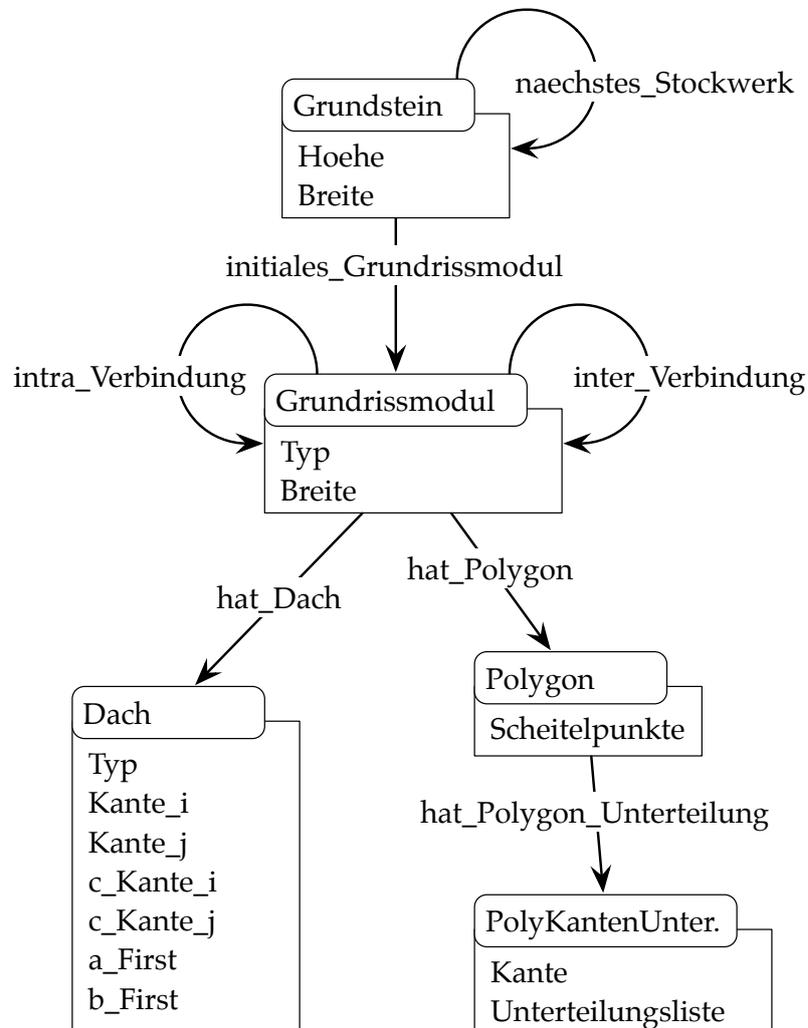


Abbildung 6.2: Semantisches Modell der groben Gebäudestruktur.

6.6.3 Konzepte und Relationen des Fassadenstils

Im Folgenden werden nur die wichtigsten Konzepte und Beziehungen des Fassadenstils vorgestellt. Die vollständige Liste aller Konzepte und Relationen ist im Anhang A angegeben. Die Beschreibung erfolgt analog zur groben Gebäudestruktur, wobei die neuen Konzepte, Beziehungen und Attribute in die bereits vorhandenen Mengen aufgenommen werden.

Definition. Konzept Stil

In diesem Konzept wird der „Stil“ für entweder ein ganzes Gebäude oder für grobe Teilbereiche, wie die Grundrissmodule, festgelegt. Dieses Konzept hat keine Attribute.

$$C \cup \text{Stil}$$

Wie wir sehen, ist das Konzept alleine wenig aussagekräftig. Es wird erst durch die Beziehungen zu seinen Nachfolgern, die sämtliche Details der Fassade beinhalten, zu einer umfangreichen Ansammlung von Informationen. Wie in der Definition angemerkt, kann der Stil (mit all seinen Beziehungen) entweder auf das gesamte Gebäude, einzelne Stockwerke oder Grundrissmodule angewendet werden.

Definition. Konzept Unterteilung

In diesem Konzept werden verschiedene Strategien für die „Unterteilung“ der Basiswände und Hausecken festgelegt. Seine Attribute haben dabei folgende Bedeutung. Über „ZWE“ wird angegeben, ob Zwischenwandelemente erzeugt werden sollen. Im Attribut „allg_Unterteilung“ wird eine allgemeine Vorschrift zur Unterteilung der Basiswände beschrieben. Die beiden Attribute „h_ZWE“ und „h_Hausecken“ bestimmen, ob Zwischenwandelemente bzw. Hausecken horizontal unterteilt werden sollen. Das Konzept der Wandpartitionen wird ggf. auf diese Elemente angewendet.

$$\begin{aligned} C &\cup \text{Unterteilung}, \\ A &\cup \{ZWE, \text{allg_Unterteilung}, h_ZWE, h_Hausecken\}, \\ \tau(\text{Unterteilung}) &= \{ZWE, \text{allg_Unterteilung}, h_ZWE, h_Hausecken\} \end{aligned}$$

Eine Unterteilung enthält als Information das Seitenverhältnis einer Wand und eines Zwischenwandelementes. Bei der Erzeugung der Geometrie werden entsprechend dieser Daten die Basiswände unterteilt.

Definition. Konzept Wandpartition

Mit diesem Konzept können Wände horizontal in einzelne „Wandpartitionen“ unterteilt werden. Eine Ausprägung dieses Konzepts erzeugt genau eine Partition. Sollen weitere erzeugt werden, so sind entsprechend viele Ausprägungen anzugeben, wobei das System eigenständig auf Überlappung überprüft und ggf. Anpassungen vornimmt. Über die beiden Attribute „x_Start“ und „x_Stop“ wird der Bereich der Partition definiert.

$$\begin{aligned}
C &\cup \text{Wandpartition}, \\
A &\cup \{v_Start, v_Stop\}, \\
\tau(\text{Wandpartition}) &= \{v_Start, v_Stop\}
\end{aligned}$$

Definition. Konzept FensterTuer

Mit diesem Konzept wird die grobe rechteckige Öffnung für Fenster und Türen definiert. Ob es sich um ein Fenster oder eine Tür handelt, wird mit „Typ“ festgelegt. Entsprechend wird bei Türen die Verfeinerung der unteren Kante nicht beachtet. Die Öffnung kann relativ zur Wand oder in absoluten Koordinaten angegeben werden. Zu den Attributen für die relative Positionierung zählen „v_Start“, „v_Stop“, „h_Start“ und „h_Stop“. Für die absolute Positionierung werden die Attribute „Breite“, „Hoehe“, „dist_links“ und „dist_rechts“ benötigt. Einer der wichtigsten Parameter ist „id“. Über ihn kann der Ausprägung eine Bezeichnung zugewiesen werden, um von Wandunterteilungen als Fensterstil genutzt werden zu können.

$$\begin{aligned}
C &\cup \text{FensterTuer}, \\
A &\cup \{h_Start, h_Stop, dist_links, dist_rechts, id\}, \\
\tau(\text{FensterTuer}) &= \{v_Start, v_Stop, h_Start, h_Stop, \\
&\quad \text{Breite, Hoehe, dist_links, dist_rechts, id}\}
\end{aligned}$$

Es bleibt zu bedenken, ob nicht weitere Verfeinerungsformen der unteren Kante für Türen eingesetzt werden könnten, die z. B. eine Treppe erzeugen.

Definition. Konzept FTVerfeinerung

Mit diesem Konzept wird eine Kante der groben Fenster- oder Türöffnung über einen Kantenzug verfeinert. Die Verfeinerung ist über das Attribut „Gesimsbeschreibung“ gegeben und beinhaltet eine aus Abschnitt 5.4.1 bekannte Gesimsbeschreibung. Über das Attribut „Unterteilungen“ wird angegeben, wie detailliert der Kantenzug später gezeichnet werden soll.

$$\begin{aligned}
C &\cup \text{FTVerfeinerung}, \\
A &\cup \{\text{Gesimsbeschreibung, Unterteilungen}\}, \\
\tau(\text{FTVerfeinerung}) &= \{\text{Gesimsbeschreibung, Unterteilungen}\}
\end{aligned}$$

Definition. Konzept FTStockDoppelgesims

Über dieses Konzept wird der (verfeinerten) Kante ein Fenster- bzw. Türstock zugewiesen. Zurzeit werden noch drei weitere Typen bereitgestellt, die im Anhang aufgelistet sind. Über das Attribut „dominant“ wird die Dominanz der Kante festgelegt. Die Ausmaße des Fenster- oder Türstocks werden über die Attribute „Groesse“, „Breite“ und „Vorsprung“ bestimmt.

$$\begin{aligned}
C &\cup \text{FTStockDoppelgesims}, \\
A &\cup \{\text{dominant, Groesse, Vorsprung}\}, \\
\tau(\text{FTStockDoppelgesims}) &= \{\text{dominant, Groesse, Breite, Vorsprung}\}
\end{aligned}$$

Definition. Konzept Gesims

In diesem Konzept werden die Daten für ein „Gesims“ definiert. Als Attribute enthält es die „Gesimsbeschreibung“ und in „Unterteilungen“ den Detailgrad der Bögen im Kantenzug.

$$C \cup \text{Gesims},$$

$$\tau(\text{Gesims}) = \{\text{Gesimsbeschreibung}, \text{Unterteilungen}\}$$

Damit sind die wichtigsten Konzepte für den Fassadenstil aufgelistet. Im Folgenden wird ein Ausschnitt der Relationen gegeben. Die vollständige Liste aller Konzepte und Relationen zum Stil ist im Anhang A aufgeführt.

Definition. Relation hat_Unterteilung

Über diese Relation wird einem Stil eine Vorschrift zur Unterteilung der Basiswände zugeordnet. Ein Stil kann nur eine Form der Unterteilung haben.

$$R \cup \text{hat_Unterteilung},$$

$$\alpha(\text{hat_Unterteilung}) = \text{Stil},$$

$$\omega(\text{hat_Unterteilung}) = \text{Unterteilung}$$

Definition. Relation hat_Wandpartition

Damit erhält ein Stil Angaben über Wandpartitionen. Mehrere Wandpartitionen können einem Stil zugewiesen sein.

$$R \cup \text{hat_Wandpartition},$$

$$\alpha(\text{hat_Wandpartition}) = \text{Stil},$$

$$\omega(\text{hat_Wandpartition}) = \text{Wandpartition}$$

Definition. Relation hat_FensterTuer

Hiermit werden die Fenster- und Türstile für einen Stil gesetzt.

$$R \cup \text{hat_FensterTuer},$$

$$\alpha(\text{hat_FensterTuer}) = \text{Stil},$$

$$\omega(\text{hat_FensterTuer}) = \text{FensterTuer}$$

Definition. Relation hat_FTVerfeinerungLinks

Hierüber werden Angaben zur Verfeinerung für die linke Kante einer Fenster- oder Türöffnung gemacht. Es existieren noch drei weitere Relationen, die die Kanten unten, rechts und oben verfeinern.

$$R \cup \text{hat_FTVerfeinerungLinks},$$

$$\alpha(\text{hat_FTVerfeinerungLinks}) = \text{FensterTuer},$$

$$\omega(\text{hat_FTVerfeinerungLinks}) = \text{FTVerfeinerung}$$

Definition. Relation hat_FTDoppelgesims

Mit dieser Relation wird für eine verfeinerte Kante ein Tür- bzw. Fensterstock vom

Typ *Doppelgesims* gesetzt. Für die drei weiteren Formen wird jeweils analog eine eigene Relation definiert.

$$\begin{aligned}
 R \cup \text{hat_FTDoppelgesims}, \\
 \alpha(\text{hat_FTDoppelgesims}) &= \text{FTVerfeinerung}, \\
 \omega(\text{hat_FTDoppelgesims}) &= \text{FTDoppelgesims}
 \end{aligned}$$

Definition. Relation *hat_Gesims*

Das Konzept „*Gesims*“ steht mit einer ganzen Reihe von anderen Konzepten in Verbindung und es existieren entsprechend viele Relationen. Als Startknoten wollen wir in diesem Fall die *Wandpartition* herausgreifen, die mit einem *Gesims* versehen werden kann.

$$\begin{aligned}
 R \cup \text{hat_Gesims}, \\
 \alpha(\text{hat_Gesims}) &= \{ \text{Wandpartition}, \text{Fensterrahmen}, \text{Ornament}, \\
 &\quad \text{FTStockDoppelGesims}, \text{FTStockGesims} \}, \\
 \omega(\text{hat_Gesims}) &= \text{Gesims}
 \end{aligned}$$

Die Konzepte *Fensterrahmen* und *Ornament* sind im Anhang spezifiziert.

In Abbildung 6.3 ist das semantische Modell des Fassadenstils graphisch zusammengefasst. In der Abbildung wurden die Attribute der Konzepte wie auch die Beschriftungen der Relationen weggelassen, um die Struktur besser zu erkennen. Wie man sieht, verbinden genau vier Relationen die beiden Konzepte *FensterTuer* und *FTVerfeinerung*. Damit werden genau die vier Kanten der groben Fenster- bzw. Türöffnung verfeinert. Jede verfeinerte Kante kann als Fensterstock ein *Gesims*, *Doppelgesims*, *Klinker* oder einfache Struktur aufweisen. Zusätzlich kann jede Seite eines Fenster- oder Türstocks mit einem *Ornament* (wie in Abschnitt 5.4.2 beschrieben) versehen werden.

Die Abbildung lässt vermuten, dass einige der in Kapitel 5 eingeführten Elemente fehlen. So sind z. B. weder *Basiswände* und *Wände*, noch *Hausecken* als Konzepte eingetragen. Das ist dadurch bedingt, dass zu diesem Zeitpunkt nur die Vorschriften (in diesem Fall die Unterteilungen) bekannt sind, wie *Basiswände* erstellt bzw. unterteilt werden, diese Vorschriften aber noch nicht ausgeführt wurden. Für dieses Vorgehen gibt es zwei gute Gründe. Zum einen erfordert ein Wechsel des Stils eine umfangreiche neue Strukturierung betroffener Bereiche des Graphen, speziell die Unterteilung der *Wände* und ihre *Fensterstile*, was einen hohen Rechenaufwand bedeutet. Zum anderen würde die klare Trennung von Gebäudestruktur und Stil zerstört werden. Wie wir im übernächsten Abschnitt 6.6.5 sehen, wird dieser Konflikt durch Einführen eines Zwischenmodells, dessen Erzeugung erst angestoßen werden muss, aufgehoben.

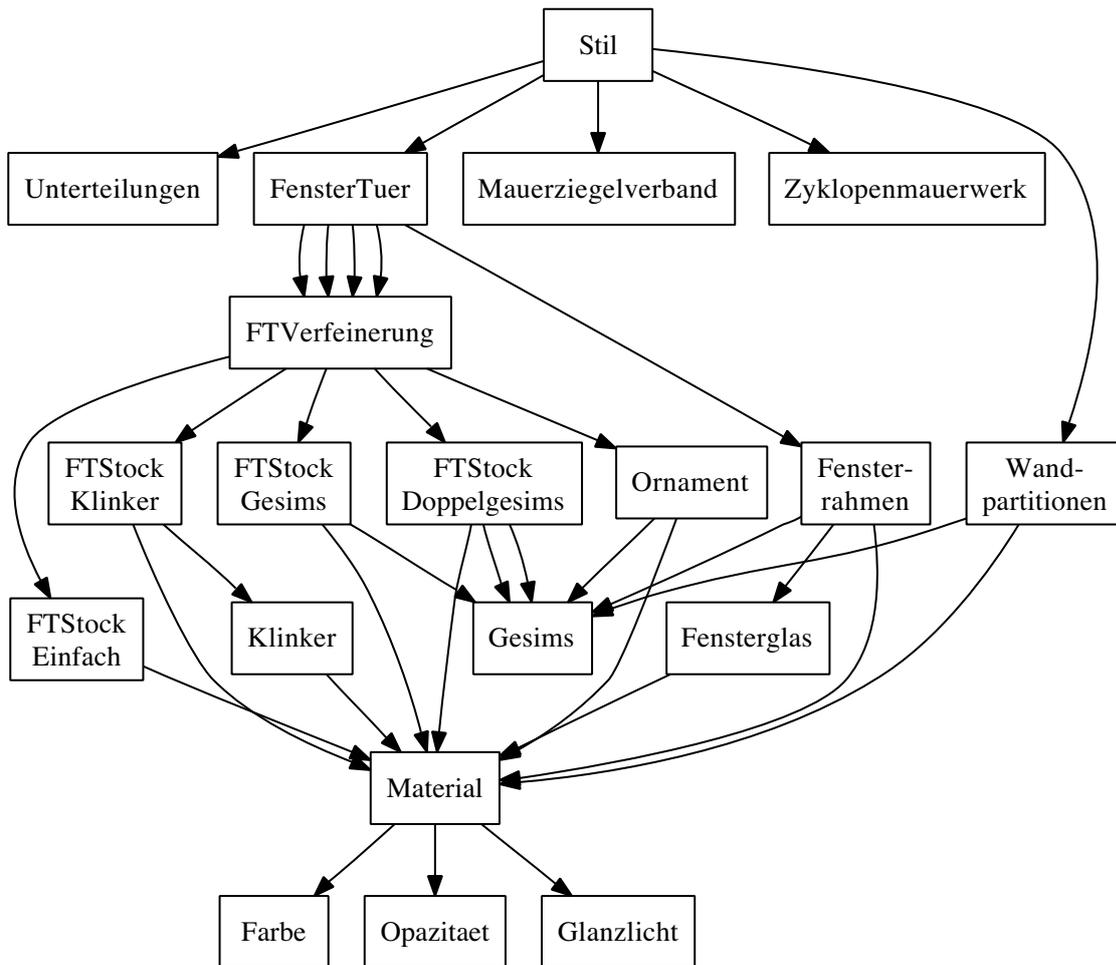


Abbildung 6.3: Semantisches Modell des Fassadenstils.

6.6.4 Anwendung des Stils auf die Gebäudestruktur

In den vergangenen Abschnitten haben wir uns eine semantische Beschreibung zum einen für die grobe Gebäudestruktur und zum anderen für den Fassadenstil erarbeitet. Damit ist die Anforderung, die Gebäudestruktur vom Stil zu trennen, erfüllt. Nun muss nur noch der Stil auf die Gebäudestruktur und zwar reversibel anwendbar sein. Dazu benötigen wir noch die beiden folgenden Relationen zwischen Gebäudestruktur und Fassadenstil.

Definition. Relation *hat_StockwerkStil*

Damit wird für ein ganzes Gebäude bzw. für ein Stockwerk ein Stil definiert.

$$R \cup \text{hat_Stockwerksstil},$$

$$\alpha(\text{hat_Stockwerksstil}) = \text{Grundstein},$$

$$\omega(\text{hat_Stockwerksstil}) = \text{Stil}$$

Definition. Relation `hat_ModulStil`

Mit dieser Relation wird einem Grundrissmodul ein Stil zugewiesen.

$$\begin{aligned} R &\cup \text{ hat_Modulstil,} \\ \alpha(\text{ hat_Modulstil}) &= \text{ Grundrissmodul,} \\ \omega(\text{ hat_Modulstil}) &= \text{ Stil} \end{aligned}$$

Damit haben wir alle notwendigen Strukturen für das Konzeptmodell der Gebäudefassade. Im Weltmodell existieren Ausprägungen (Instanzen der Konzepte) die eine konkrete Fassade, bestehend aus Gebäudestruktur und Fassadenstil, beschreiben. Dank der TGraphen können verschiedene Stile nebeneinander vorliegen. Um einen Stil durch einen anderen zu ersetzen, muss nur der gewünschte Stil in die entsprechende Relation eintragen werden. Damit kann man einen Stil für das gesamte Gebäude setzen, hat aber auch die Möglichkeit, einzelnen Gebäudestrukturen eigene Stile zuzuweisen oder sie wieder zu entfernen.

6.6.5 Vom Weltmodell zum Zwischenmodell

Inzwischen sind wir so weit, dass konkrete Gebäudefassaden im Weltmodell vorliegen. Wie bereits erwähnt, wurden aber noch keine Basiswände, Wände, wie auch Hausecken erstellt und unterteilt. Im folgenden Schritt wird aus dem Weltmodell ein hierarchisches Zwischenmodell erstellt, das aber nach wie vor nur symbolische und minimale geometrische Informationen enthält. Die Prozeduren erzeugen die Elemente Wände, Zwischenwandelemente und Hausecken. Des Weiteren werden von ihnen Partitionen erstellt, die die genannten Elemente übergreifen. Ebenso werden Wandelementen Fensterstile zugewiesen. Die neu erzeugten Elemente werden mit den bereits im Weltmodell enthaltenen Informationen in ein Zwischenmodell eingefügt. Dabei werden sämtliche räumliche Informationen extrahiert und in das Zwischenmodell eingetragen. Elemente des Zwischenmodells kennen damit ihre direkten Nachbarn innerhalb eines Stockwerks, wie auch stockwerkübergreifend. Wände, Zwischenwandelemente und Hausecken besitzen Verweise auf ihre Partitionen, wobei Wände zusätzlich noch Verweise auf ihre Fenster- bzw. Türstile enthalten. Damit können gezielt Details, wie z. B. der Fensterstil und die Materialeigenschaften, auf Wände angebracht werden. Es ist darauf zu achten, dass diese Änderungen keine Auswirkungen auf das Weltmodell haben, da das Weltmodell diese Elemente nicht besitzt. Ist einmal das Zwischenmodell erzeugt und werden danach Änderungen im Weltmodell vorgenommen, so müssen betroffene Teile des Zwischenmodells erneut generiert werden. Ebenso werden Änderungen an Details ggf. überschrieben. In der Geometrie-Fabrik wird aus dem Zwischenmodell das fertige 3D-Modell der Gebäudefassade berechnet.

6.6.6 Vorteile der Technik

Die Aufteilung der Gebäudefassade in ein geteiltes Konzeptmodell, Weltmodell und Zwischenmodell bietet viele Vorteile. So ist im Konzeptmodell die abstrakte Definition aller zulässigen Gebäudefassaden enthalten. Damit ist sichergestellt, dass im Weltmodell nur konsistente Gebäudefassaden existieren. Außerdem ist das Konzeptmodell sehr übersichtlich, womit Änderungen sehr einfach vorgenommen werden können. Gegebenenfalls müssen noch Anpassungen an den Ausprägungen im Weltmodell vollzogen werden.

Eine Gebäudefassade im Weltmodell stellt eine sehr kompakte Beschreibung der späteren Gebäudegeometrie samt Ornamenttexturen dar. Im Augenblick wird eine Fassade bestehend aus Konzeptmodell und Ausprägung im Weltmodell als XML-Datei gespeichert. Dabei hat die Beschreibung eines fünfstöckigen Gebäudes samt Stil einen Speicherbedarf von ca. 200 kB. Für aktuelle Anwendungen, wie z. B. Computerspiele mit virtuellen Städten, bedeutet das, dass zum einen der Speicherbedarf einer gesamte Stadt enorm zurückginge, da alle Gebäude zur Laufzeit erzeugt werden können. Zum anderen wären Techniken denkbar, die zur Laufzeit anhand bestimmter Kriterien und Regeln zufällige Gebäudebeschreibungen generieren, womit eine sich am Spielablauf dynamisch orientierende Umgebung möglich wäre. Es sind natürlich noch eine Vielzahl von Anwendungsszenarien möglich, auf die im Kapitel 8 eingegangen wird.

In Abbildung 6.4 sind alle Schritte die zum fertigen 3D-Modell beitragen, zusammengefasst. Dabei wird in jedem Schritt auf den Abstraktionsgrad, den Detailreichtum, die Anpassbarkeit und den Speicherbedarf Bezug genommen.

Kapitel 7

Geometrie-Fabrik

In der Geometrie-Fabrik wird aus dem Zwischenmodell das detaillierte 3D-Modell der Gebäudefassade erstellt, wie auch die Texturen der Ornamente berechnet. Dabei wird die symbolische Information in Geometrie überführt.

Wie das Konzept- und Weltmodell, ist auch das Zwischenmodell auf TGraphen abgebildet. Im Gegensatz zu Konzept- und Weltmodell, existieren in der Implementierung des Prototyps für jedes Objekt des Zwischenmodells eine eigene Klasse. Die Klassen beinhalten nicht nur die Daten der Fassadenelemente des Zwischenmodells, sondern auch eigene Methoden. Über diese Methoden berechnen die Elemente ihre Geometrie und geben sie aus.

Die Ausgabe erfolgt über eine generische Schnittstelle. Das bedeutet, dass die Ausgabe unabhängig von einem gewählten Zielformat erfolgt. Die Abbildung auf ein konkretes Format wird vom Prototyp übernommen, womit beliebige Ausgabeformate ermöglicht werden. Zurzeit werden RenderMan RIB¹, Maya/MEL² und eine Protokollausgabe angeboten.

7.1 Ordnungsrelationen

Um das vollständige 3D-Modell der Fassade zu erzeugen, wird das Zwischenmodell von der Wurzel bis zu den Blättern durchlaufen. Über diese hierarchische Struktur lassen sich räumliche Abhängigkeiten sehr einfach ableiten. Die grobe Gebäudestruktur, wie Gebäudeflügel, Stockwerke und Mauervorsprünge sind bereits direkt in der Hierarchie abgebildet. Durch die Anwendung der Unterteilungsvorschriften auf die Merkmale der Umrisslinien, sind ebenso die Nachbarschaftsbeziehungen zwischen Wänden, Zwischenwandelementen wie auch Hausecken gegeben. Weiter-

¹<http://renderman.pixar.com/>

²<http://www.autodesk.com/maya/>

hin verfügen Wände über Informationen zu ihren Türen und Fenstern wie auch deren Rahmen. Damit sind zwar alle räumlichen Abhängigkeiten erfasst, aber es ist noch nicht geklärt, wie die Elemente aneinander anzupassen sind, d. h. welche Elemente ihre Geometrie modifizieren müssen und welche nicht verändert werden. Das Problem wird über lokal begrenzte Ordnungsrelationen in der Hierarchie gelöst.

- **Ordnungsrelation nach der Tiefe**

Das bedeutet, dass tiefer gelegene Fassadenelemente eine Verfeinerung ihrer Vorgänger bewirken. Darunter fallen folgende Abhängigkeiten:

- Wände werden durch ihre Fenster bzw. Türen angepasst.
- Fenster- bzw. Türöffnung werden durch ihre Fenster- bzw. Türstöcke und Rahmen verfeinert.

- **Ordnungsrelation auf gleicher Ebene**

Hier sind Wände, Zwischenwandelemente und Hausecken betroffen. Wände haben eine geringer Priorität und passen sich den Profilen der Hausecken und der Zwischenwandelemente an.

Eine Sonderstellung nehmen stockwerkübergreifende Abhängigkeiten ein. Dazu gehören zurzeit nur Dächer, die ihre oben anliegenden Wände, ZWE und Hausecken anpassen. Das kann ggf. soweit führen, dass sich Fenster oder Türen nicht mehr in die Wand einbringen können. Die Abhängigkeiten und deren Auflösung über die Ordnungsrelationen sind in Abbildung 7.1 schematisch dargestellt.

7.2 Geometrieerzeugung

Nachdem die Abhängigkeiten unter den Fassadenelementen geklärt sind, wenden wir uns der Berechnung der Geometrie zu. Dabei wird nur das grundsätzliche Vorgehen vermittelt – auf elementare geometrische Operationen wird verzichtet. Gestartet wird bei der ersten Basiswand des ersten Merkmals des Grundrissmoduls des Grundsteins. Der Basiswand sind ihre Nachbarn nach links, rechts, oben wie unten bekannt. Bei den Nachbarn handelt es sich ebenso um Basiswände. Über die Unterteilungsvorschrift wird sie in Hausecken, Wand- und Zwischenwandelemente untergliedert. Die neu entstandenen Elemente werden räumlich sortiert und von der Basiswand verwaltet. Für die Hausecken und Zwischenwandelemente sind über die Unterteilungen ihre Partitionen und ggf. deren Gesimsformen bekannt. Die Stärke der Partitionen berechnet sich aus der Dicke der angrenzenden Wände bzw. Basiswände. Die Höhe der Partitionen ergeben sich entsprechend ihrer Ausmaße relativ zur Höhe der Wände. Damit können die einzelnen Partitionen der Hausecken und Zwischenwandelemente gezeichnet werden. Anschließend bestimmen die Wände bzw. die Wandpartitionen ihre Geometrie anhand oben genannter Abhängigkeiten. Der Schnitt der Wand oder Wandpartitionen mit einer

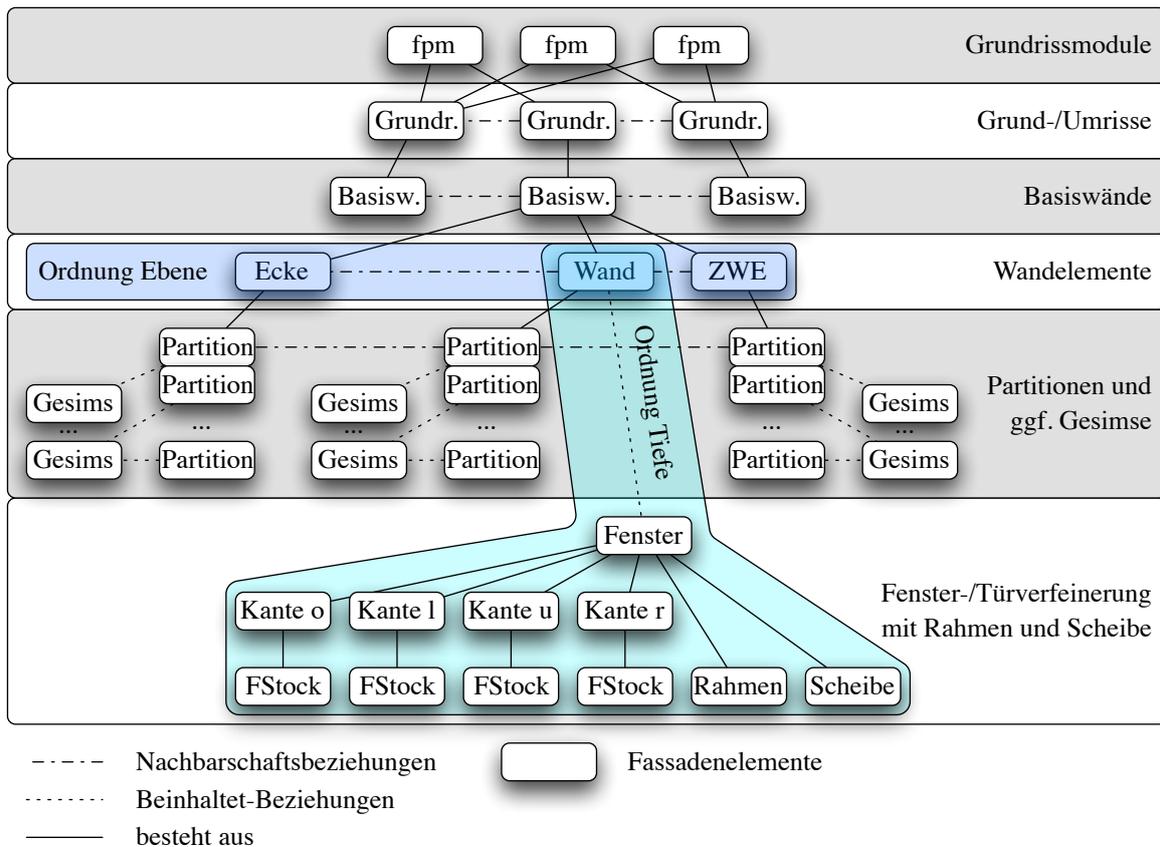


Abbildung 7.1: Abhängigkeiten der Fassadenelemente und deren Ordnungsrelationen.

Fenster-/Türöffnung und den vier angrenzenden Profilen wird über übliche 2D-Schnittverfahren ermittelt. Dabei unterscheiden wir, ob das Schnittergebnis zur Geometrieerzeugung oder zur Erzeugung der Mauerwerkstexturen benötigt wird. Es könnte zwar auch die exakte Geometrie erzeugt und ausgegeben werden, was aber die Anzahl Dreiecke drastisch in die Höhe schnellen ließe. Da nur ein visuell gefälliger Eindruck zu vermitteln ist, reicht eine gröbere Geometrie mit an den Nahtstellen exakt passender Textur. Dadurch bleibt das Datenaufkommen gering und die optisch hohe Qualität erhalten. Für die Geometrie werden die Wandpartitionen und das Polygon der Öffnung auf die Ebene der Basiswand projiziert, geschnitten und die Ergebnisse in den 3D-Raum rücktransformiert. Die resultierenden Wandteile werden noch um die Wanddicke extrudiert, wobei ggf. an der Stoßkante zweier Wände eine Gehrung vorgenommen wird. Wandteile bzw. Partitionen werden auch für horizontale Strukturen herangezogen. Enthalten sie ein Gesims, so wird überprüft, ob diese Partition mit ihren Nachbarpartitionen eine horizontale Struktur bilden. Diese Gesimse werden dann einheitlich erzeugt und

ggf. texturiert, aber nach Fertigstellung wieder getrennt nach Wand, Zwischenwandelement und Hausecke ausgegeben. Für die Mauerwerkstexturen hingegen wird die exakte Geometrie benötigt. In die Schnittberechnung werden hier noch zusätzlich die vier Seitenprofile einbezogen und die Ergebnisse mit den ungenaueren Ergebnissen aus dem vorigen Schritt für die Erzeugung des Mauerwerks zur Verfügung gestellt. Die Wandteile, die kein Gesims aufweisen, werden mit ihren zugehörigen Texturen ausgegeben. Falls zulässig, erzeugen Fenster und Türen ihre Fenster- und Türstöcke und ggf. noch Ornamente. Anhand der Attribute werden Gesimse, einfache Umrandungen oder Ziegelumrandungen generiert. Danach werden die Rahmen, Fenster- bzw. Türpolygone gezeichnet. Im selben Schema werden alle weiteren Basiswände des Stockwerks wie auch alle anderen Stockwerke generiert. Abschließend werden noch für Module mit aneinander grenzenden Dächern deren Dachverbund ermittelt und ausgegeben.

7.3 Prototyp

Der Prototyp, der die Geometrie-Fabrik wie auch die semantischen Modelle enthält, wurde in der Skriptsprache *Python*³ entwickelt. Für die graphische Oberfläche wurde *wxWidgets*⁴ verwendet, das wie auch Python für verschiedene Plattformen erhältlich ist. Dank der Plattformunabhängigkeit ist der Prototyp unter Microsoft Windows XP/Vista, Apple Mac OS X und Linux lauffähig.

³<http://www.python.org/>

⁴<http://www.wxwidgets.org/>

Der Prototyp besitzt ein einziges Fenster, das wahlweise jeweils eine Bildschirmmaske für die Konfiguration des Ausgabeformats, für die Bearbeitung der Konzept- und Weltmodelle oder für die Bearbeitung des Zwischenmodells darstellt. In Abbildung 7.2 ist ein Bildschirmfoto gezeigt, das die Bildschirmmaske für die Konfiguration des Ausgabeformats darstellt. Zu sehen sind Parameter für RenderMan RIB.

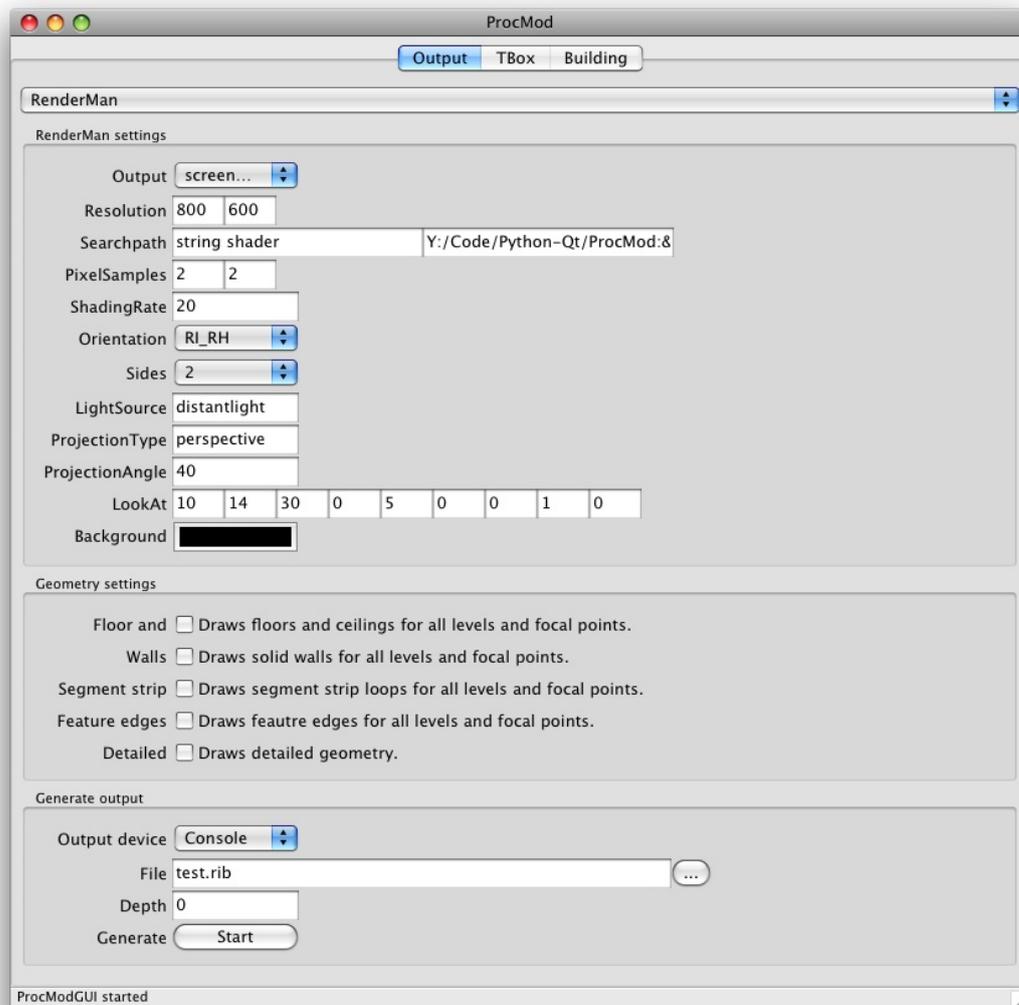


Abbildung 7.2: Konfiguration der Ausgabeformate. Hier sind die Parameter für RenderMan RIB gezeigt.

Die Darstellung des Konzept- und Weltmodells ist in Abbildung 7.3 zu sehen. Die Bildschirmmaske enthält Eingabefelder für Konzepte, Relationen und deren Attribute des Konzeptmodells, zusammengefasst im Containerbaustein *TGraph - Common classes*. Die Instanzen zum Weltmodell sind unter *TGraph - Common instances* vereint. Im untersten Teil der Bildschirmmaske können das aktuelle Konzept- und Weltmodell als Graph ausgegeben werden.

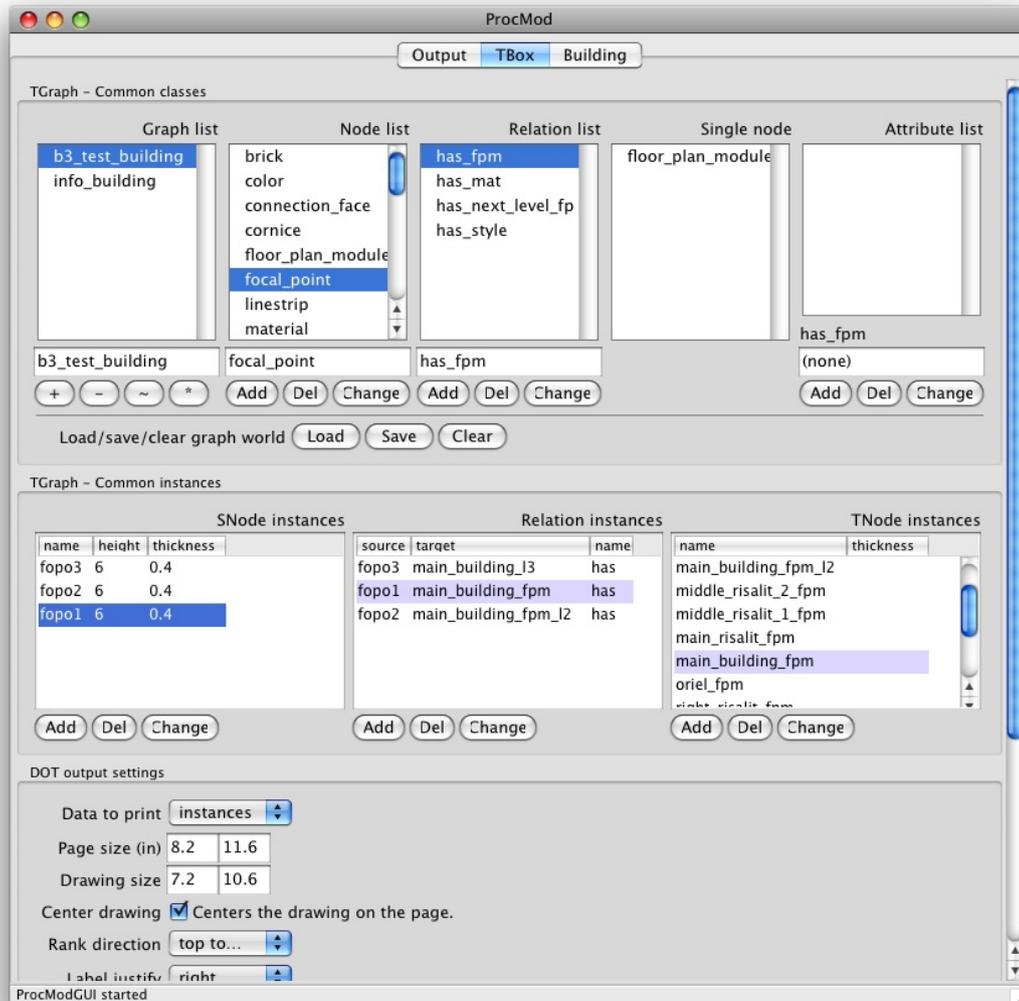


Abbildung 7.3: Konzept- und Weltmodell. Im oberen Abschnitt werden die Konzepte, Relationen und deren Attribute spezifiziert. Darunter werden konkrete Instanzen im Weltmodell angelegt.

Das Zwischenmodell wird als Baumstruktur dargestellt, wie in Abbildung 7.4 zu sehen ist. Im linken Teil der Bildschirmmaske kann der Benutzer die Gebäudefassade hierarchisch traversieren und Knoten im Baum anwählen. Rechts vom Baum werden die Parameter des aktuell gewählten Knotens dargestellt. Hier können Anpassungen, wie z. B. Änderungen des Stils, vorgenommen werden.

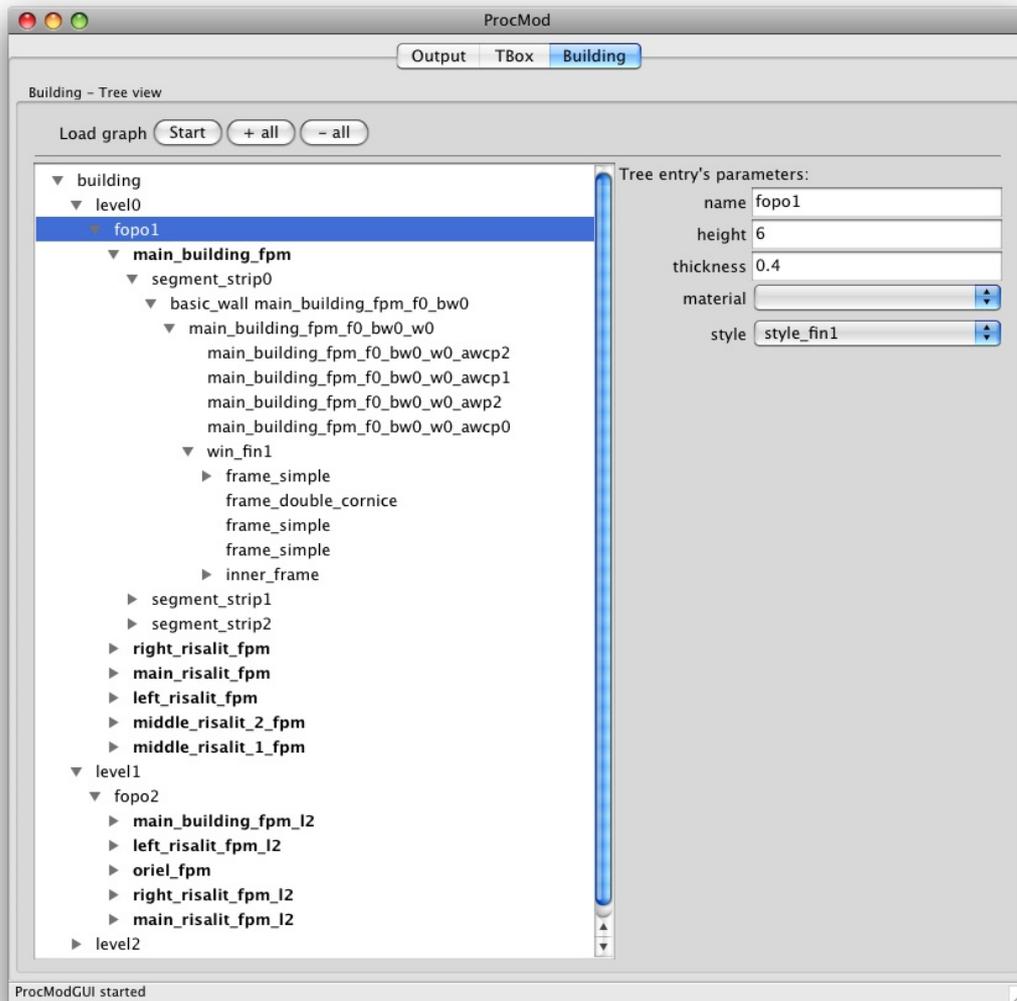


Abbildung 7.4: Das Zwischenmodell, das durch Anwendung von Prozeduren aus dem Weltmodell hervorgeht. Hier können die gesamte Hierarchie der Fassade durchlaufen und Änderungen auf beliebiger Ebene vorgenommen werden.

7.4 Ergebnisse

In diesem Abschnitt werden einige ausgewählte Beispiele vorgestellt, die mit dem Prototyp erzielt wurden. Abbildungen 7.5 und 7.6 zeigen ein freistehendes Wohnhaus. Die Fensterstöcke in den ersten beiden Stockwerken weisen Gesimse und Ornamente auf.



Abbildung 7.5: Freistehendes Haus mit Gesimse und Ornamente.



Abbildung 7.6: Nahaufnahme des Hauses.

Das Gebäude in den Abbildungen 7.7 und 7.8 ist ein Modell des Hauptgebäudes der Fakultät für Informatik der Universität Karlsruhe.



Abbildung 7.7: Gesamtansicht des Modells des Informatikgebäudes.



Abbildung 7.8: Sicht aus dem zweiten Stock.

In den Abbildungen 7.9 und 7.10 ist die selbe Grundstruktur des Informatikgebäudes zu sehen, aber mit klassischem Stil.



Abbildung 7.9: Das Informatikgebäude im klassischen Stil.



Abbildung 7.10: Dieselbe Sicht wie in Abbildung 7.8, wobei das Gebäude in einem klassischen Stil gehalten ist.

Die Grundstruktur der Fakultät, ohne Stil, zeigt Abbildung 7.11.

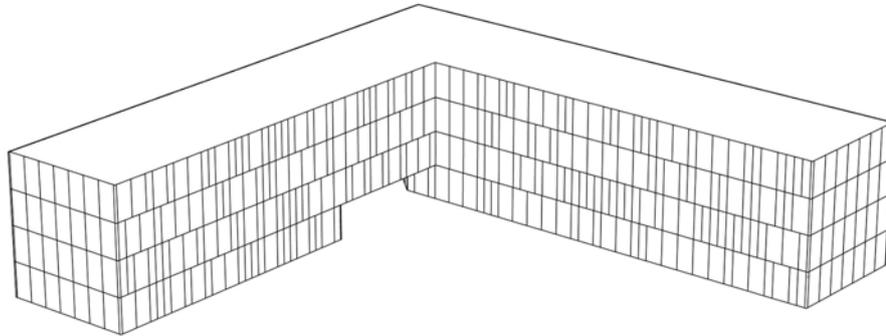


Abbildung 7.11: Die Grundstruktur des Fakultätsgebäudes.

Ein großes Gebäude mit zwei Seitenflügeln ist in Abbildung 7.12 dargestellt.



Abbildung 7.12: Großes Gebäude mit Haupt und Seitenflügel.

Abbildung 7.13 zeigt eine Gebäudezeile, die an eine englische Arbeitersiedlung angelehnt ist. Über das Quadersteinmauerwerk wird der optische Eindruck der Gebäudezeile stark verbessert. Dieselbe Gebäudegeometrie, aber mit unterschiedlichen Mauerwerken, ist in Abbildungen 7.14 zu sehen. Allein durch die Benutzung verschiedener Mauerwerke wird der Eindruck erweckt, dass es sich hier um völlig unterschiedliche Gebäude handelt.



Abbildung 7.13: Eine Gebäudezeile, aus einfachen Häusern aufgebaut.



Abbildung 7.14: Die Gebäudezeile aus voriger Abbildung mit Klinker als Mauerwerkstextur.

Abbildung 7.15 zeigt eine leichte Modifikation der Gebäudezeile.



Abbildung 7.15: Die Gebäudezeile mit einem Kreuzverband. Die gesamte Front besitzt unten an der Wand einen Sockel und die Türen sind mit einer Stufe versehen.

Das Gebäude in Abbildung 7.16 zeigt eine Nachbildung des Karlsruher Schlosses, das mit den in dieser Arbeit erworbenen Techniken erstellt wurde.



Abbildung 7.16: Modell des Karlsruher Schlosses. Es ist die Front des Schlosses zu sehen, wobei aber der Balkon über dem Haupteingang fehlt.

Mit den nächsten beiden Abbildungen wird eine Fotografie⁵ des Schlosses (Abb. 7.17) dessen Modell (Abb. 7.18) gegenüber gestellt. Die Unterschiede sind klar zu erkennen. Im Gegensatz zum Modell weist das Original ein Mansardendach mit Dachgauben auf. Hingegen ist die Grundstruktur des Schlosses im Modell exakt wiedergegeben. Ebenso entsprechen im Modell die Fenster- und Türgestaltungen wie auch die Gesimse stark dem Original, so dass durchaus ein hoher Wiedererkennungswert gegeben ist.



Abbildung 7.17: Foto des Karlsruher Schlosses (Nordansicht).



Abbildung 7.18: Modell des Karlsruher Schlosses.

⁵Die Fotografie wurde mit der Digitalkamera des Autors angefertigt.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

Die computergestützte Modellierung detaillierter Gebäudefassaden wird in der Computeranimation, bei der Produktion von Kinofilmen und bei Computerspielen immer bedeutender. Eine geeignete interaktive Modellierung muss für den Benutzer in der Bedienung mächtig und einfach zugleich sein. Um Modelle in kurzer Zeit erstellen zu können, muss ein hoher Abstraktionsgrad im Modellierungsprozess gewährleistet sein. In der vorliegenden Arbeit wurden Techniken entwickelt, die das breite Spektrum der Anforderungen abdecken. Sie erstrecken sich von der Erstellung einer groben Gebäudestruktur über die Beschreibung detailreicher Fassadenelemente bis hin zur automatischen Geometrie- und Texturerzeugung. Zusammengeführt in einen Prototyp, ist ein System entstanden, das den geforderten Ansprüchen gerecht wird. Es zeichnet sich gegenüber bekannten Techniken durch seine Flexibilität und Vielfalt aus.

Verbesserungen konnten hinsichtlich folgender Gebiete erzielt werden. Eine Unterteilung des Gebäudes in architektonische Abschnitte, unabhängig vom Stil, ist diesem Bereich neu. Dies hat zum einen zur Folge, dass räumliche Abhängigkeiten leicht abzuleiten sind. Bisher mussten hierfür geometrische Schnittoperationen durchgeführt werden. Zum anderen lassen sich durch diese Form der Unterteilung Dächer generieren, die den architektonischen Charakter der Gebäudestruktur widerspiegeln. Verschiedene Dachformen können für jeden Abschnitt angegeben werden. Damit ergeben sich für die Dächer gestalterische Freiheiten, die mit aktuell bekannten Techniken nicht erreicht werden können. Darüber hinaus wurde für eine Vielzahl Stile eine einheitliche und einfache Beschreibungsform entwickelt. Mit der Unterteilung und der Stilbeschreibung sind die Grundlagen für die semantische Repräsentation des gesamten Gebäudes mit Fassade erbracht. Daraus wurde ein semantisches Modell entwickelt, das die Beschreibung der Gebäudefassade auf

einer abstrakten Ebene zulässt. Eine derart detaillierte semantischen Darstellung eines Gebäudes, wobei Gebäudestruktur und Stil voneinander getrennt behandelt werden können, ist völlig neu. Ebenso sind Techniken zur prozeduralen Erzeugung von Mauerwerken im semantischen Modell enthalten. Die Beschreibungen der Mauerziegelverbände wie auch Zyklopenmauerwerke werden zur Laufzeit ausgewertet und es werden von der Geometrie abhängige Texturen generiert. Die vollständige Gebäudefassade mit Texturen benötigt in der semantischen Darstellung nur sehr wenig Speicherplatz. Damit ist das vorgestellte System einzigartig in der Flexibilität der Repräsentation, der Qualität der Ausgabe und der Sparsamkeit hinsichtlich des Speicherbedarfs.

Der Prototyp deckt folgende Bereiche ab:

- **Einfache Modellierung der Gebäudestruktur**

Die Gebäude bestehen aus zusammenhängenden Grundrissmodulen. Ein Modul beschreibt einen Abschnitt des Gebäudes, der als eine Einheit aufgefasst werden kann. Eine solche Einheit ist z. B. der Gebäudehaupttrakt, ein Seitenflügel, ein Mauervorsprung oder ein Erker. Jedes Stockwerk wird aus diesen Modulen aufgebaut. Um weitere Stockwerke zu erhalten, werden die Module auf das nächste Stockwerk ausgedehnt und ggf. mit zusätzlichen Modulen erweitert. Mit diesem Vorgehen lassen sich beliebige Umrisse pro Stockwerk erzeugen. Der größte Vorteil liegt darin, dass aneinander grenzende Elemente sehr einfach ermittelt werden können. Diese Information wird später benötigt, um diese Elemente zu finden und automatisch anzupassen. Außerdem modelliert der Benutzer somit in einem sehr frühen Stadium die groben Strukturen. Zu diesem Zeitpunkt werden auch ihre Eigenschaften festgelegt und der Benutzer weiß, welche Rolle ihnen im Gebäude zufällt.

Für die Beschreibung der Dächer ist die Technik der Untergliederung der Gebäude in Module außerordentlich von Nutzen. Damit erhält jeder Gebäudeabschnitt seine eigene Dachstruktur. Es wurde eine generische Dachform vorgestellt, mit der sich die meisten üblichen Dachformen sehr leicht darstellen lassen. Um die gesamte Dachstruktur zu erzeugen, werden die einzelnen Dächer miteinander verschmolzen. Bisher existiert kein anderes Verfahren, das es dem Benutzer erlaubt, für beliebige Grundrisse die Dachform in Abhängigkeit der darunterliegenden Gebäudeteile zu steuern.

- **Beschreibung des Stils**

Für Fassadenelemente, die gewissen üblichen Konventionen entsprechen, ließ sich eine generische, parametrisierte Beschreibung finden. Diese Beschreibungen ergeben jeweils einen Stil. Dabei werden horizontale Strukturen eingeführt, die Gesimse, Bänder oder Mauerwerkstexturen usw. aufnehmen können. Diese Strukturen erstrecken sich entlang eines Umrisses über das gesamte Stockwerk oder nur über Teilbereiche, die aus den einzelnen Modulen

hervorgehen.

Lokal abgeschlossene Strukturen sind Fenster und Türen, sowie deren Verfeinerungen. Über eine einfache Beschreibungssprache werden Fensteröffnungen, alle Arten von Rahmen und Gesimsen definiert. Damit wird der Benutzer entlastet, da er eine Beschreibung für viele Bereiche der Fassade verwenden kann.

Um Texturen für Ornamente und Mauerwerke zu erzeugen, wurden eigene Beschreibungsformen entwickelt. Sie bieten den Vorteil, dass aus ihnen Texturen in beliebiger Auflösung berechnet werden können und sie kompakt in textueller Form vorliegen.

• Semantische Modelle

Zur semantischen Modellierung der Gebäudefassade wurden drei Modelle vorgestellt, die aufeinander aufbauen, wobei der Abstraktionsgrad stetig ab-, die Detailschärfe aber zunimmt. Die Grundlage der Modelle bilden die Gebäudestruktur und die Stile.

Das *Konzeptmodell* beschreibt eine abstrakte Sicht auf die Gebäudefassade; dieses Modell beinhaltet den grundlegenden Aufbau, die Zusammenhänge der Gebäudestruktur und die Stile.

Im *Weltmodell* existieren Instanzen des Konzeptmodells, die konkrete Fassaden repräsentieren. Ebenso sind Vorschriften zur vertikalen Unterteilung angegeben. Sie bestimmen, in welcher Weise große Wandflächen in Abschnitte zu unterteilen sind. Die Stile können auf der Ebene des gesamten Gebäudes, eines Stockwerks oder eines Moduls angegeben werden.

Das *Zwischenmodell* entsteht, indem die Unterteilungsvorschriften ausgeführt werden. Aus den Wandflächen, die direkt aus den Modulen hervorgehen, entstehen zunächst bei der vertikalen Unterteilung Wände, Zwischenwandelemente und Hausecken. Anschließend erfolgt die Einteilung in horizontale Abschnitte. Zuletzt wird den Wandabschnitten noch der gewünschte Stil zugewiesen. Für vertikale Elemente sind das Fenster, Türen, Ornamente und deren Verfeinerung; Mauerwerke, Gesimse und Bänder werden auf horizontale Elemente angewendet.

Der Benutzer kann Änderungen in allen Modellen vornehmen. Im Konzeptmodell werden strukturelle Änderungen durchgeführt, die die Semantik der Gebäudefassade grundlegend ändern. Nur in diesem Modell kann das zur Folge haben, dass ggf. Programmcode implementiert werden muss, der Fassadenelemente aneinander anpasst. Im Weltmodell werden nur Instanzen konkreter Gebäudefassaden angepasst. Änderungen können nur bis auf die Ebene der Basiswände durchgeführt werden. Man muss beachten, dass Änderungen die erneute Erzeugung betroffener Teile des Zwischenmodells bedingen. Im Zwi-

schenmodell können beliebige Anpassungen vorgenommen werden, die sich bis zu feinsten Details eines einzelnen Wandabschnitts erstrecken.

- **Geometrie- und Texturerzeugung**

Aus dem Zwischenmodell werden schließlich die Gebäudegeometrie und die Texturen erzeugt, wobei räumliche Abhängigkeiten berücksichtigt werden. Sie werden über Ordnungsrelationen aufgelöst und die Geometrie automatisch angepasst. Das kann sogar soweit führen, dass einzelne Elemente, wie Fenster, aus Platzmangel nicht gezeichnet werden. Anschließend werden Texturen in Abhängigkeit der zugehörigen Geometrie berechnet. Das fertige Modell kann in die Formate RenderMan RIB oder Maya/MEL exportiert werden. Da die Geometrie unabhängig vom gewählten Format erzeugt wird, können beliebige Ausgabeformate definiert werden. Die Fassade wird entsprechend der semantischen Modelle hierarchisch ausgegeben. Damit werden direkt hierarchische Strukturen in Maya unterstützt, womit sich das Modell sehr einfach nachbearbeiten lässt.

Insgesamt kann man feststellen, dass die vorgestellten Techniken und das daraus resultierende Gesamtsystem zur schnellen Modellierung detaillierter Gebäudefassaden die gestellten Anforderungen erfüllen.

8.2 Ausblick

In diesem Abschnitt wird ein Überblick über mögliche Erweiterungen und Einsatzgebiete gegeben:

- **Fassadenelemente**

Fensterelemente bieten bisher nur einfache Unterteilungen für einzelne Fensterscheiben. Ein flexiblere vertikale und horizontale Unterteilung ist wünschenswert und kann über das Konzeptmodell auch einfach umgesetzt werden. Um Kirchenfenster realistisch zu gestalten, sind teilweise Bleiglasfenster notwendig. Diese können mit Alejo Hausners Arbeit zur Simulation dekorativer Mosaik [Hau01] erzeugt werden.

Im Bereich der Ornamente sind vielfältige Erweiterungen denkbar. Dabei scheint eine Untersuchung geometrischer Motive lohnenswert, da die Ornamente aus einfachen geometrischen Formen aufgebaut sind. Ein entsprechender Formalismus müsste deshalb gut zu entwickeln sein.

- **Innenräume**

Die Module werden für die Fortführung auf das jeweils nächste Stockwerk

bereits unterteilt. Diese Technik könnte ausgebaut und für Innenräume herangezogen werden. Damit ließen sich vollständige Gebäude nicht nur mit Fassaden, sondern auch einschließlich Räumen im Inneren erzeugen.

- **Erweiterung auf Städte**

Pascal Müller hat in seinen Arbeiten [PM01] und [MWH⁺06] Techniken vorgestellt, die das Straßennetz einer Stadt automatisch erzeugen. Es ist gut möglich, diese Techniken mit denen der vorliegenden Arbeit zu kombinieren. Zurzeit werden eigene Untersuchungen zu einer automatisierten Generierung von Straßennetzen unternommen, die auch geologische Gegebenheiten, verschiedenartige Stadtzentren und Bevölkerungsdichten berücksichtigt.

- **Erweiterung des Konzeptmodells**

Zurzeit werden über das Konzeptmodell nur Relationen, Konzepte, deren Attribute und Unterteilungsvorschriften abgebildet. Der Programmcode, der die einzelnen Fassadenelemente miteinander verbindet bzw. einen korrekten Übergang zwischen den Elementen schafft, ist derzeit separiert. Der nächste Schritt sieht im Konzeptmodell Konstrukte vor, die es erlauben, Programmfragmente direkt in die Modelle zu integrieren, so dass diese während der Modellverarbeitung, also zur Laufzeit, ausgeführt werden. Damit kann der ehemals externe Programmcode direkt zusammen mit den Konzepten und Relationen gespeichert werden. Benötigt wird dafür ein Nachrichtensystem, über das sich Konzepte untereinander verständigen können. Die Funktionalität, Programmcode zur Laufzeit auszuführen, muss nur einmal im Konzeptmodell implementiert werden und steht folglich für alle Instanzen des Weltmodells zur Verfügung. Der Prototyp ist in der Programmiersprache Python programmiert. Python bietet bereits die Möglichkeit neuen Programmcode zur Laufzeit auszuführen, so dass die vorgestellte Erweiterung in den Prototyp integriert werden könnte.

- **Objekterkennung**

Laserabtastungen von Gebäuden erzeugen riesige Datenmengen ohne Semantik. Nun wäre es möglich spezifische Merkmale von Gebäuden, wie regelmäßig verlaufende Fensterreihen, zu extrahieren. Daraus würde man einzelne Fenster gewinnen und deren Form bestimmen. Mit Unterstützung der vorgestellten Techniken könnte man eine erste Parametrisierung einer Fenster-schablone erstellen und diese mit den abgetasteten Daten abgleichen. Das Verfahren würde man solange wiederholen bis die Unterschiede zwischen 3D-Modell und Daten einen Schwellwert unterschreiten. Das Ergebnis wäre zwar nur eine Näherung der Daten, aber dafür hätte man eine semantische Darstellung des Gebäudes. Damit würde eine Klassifizierung des Gebäudes erheblich erleichtert und das Datenaufkommen drastisch reduziert werden.

Außerdem ließe sich ein entsprechender Algorithmus zur semantischen Gebäudeerkennung mit den Techniken dieser Arbeit testen. Eine weitere Form der Gebäudeerkennung wird über Bilddaten vorgenommen. Die Arbeiten von Müller et al. [MZWG07] und Bekins et al. [BA05] bieten bereits gute Ansätze, wobei aber nur rechteckige Strukturen erkannt werden. Zusätzlich könnten die Fenster- und Türbeschreibung dieser Arbeit dazu verwendet werden, um auch beliebig geformte Fenster und Türen zu erkennen.

- **Integration in Autodesk Maya**

Der Prototyp ist vollständig in Python implementiert und unterstützt die Ausgabe in Maya/MEL. Da Maya inzwischen eine Unterstützung für Python bietet, ließe sich der Prototyp relativ einfach in Maya integrieren. Außerdem könnten sogleich die in Maya vorhandenen hierarchischen Strukturen (*Hypergraph*) die TGraphen ersetzen. Damit wäre die Möglichkeit gegeben, in Maya Gebäude graphisch interaktiv zu entwerfen und zu bearbeiten.

Anhang A

Definition des semantischen Modells

Hier sind alle Konzepte und Relationen des semantischen Modells der vollständigen Gebäudefassade aufgeführt. Einige der Konzepte und Relationen wurden bereits exemplarisch in den Abschnitten [6.6.2](#) und [6.6.3](#) beschrieben.

A.1 Konzepte und Relationen der Gebäudestruktur

Definition A.1. Konzept Grundstein

Die grobe Gebäudestruktur startet mit dem Konzept „Grundstein“. Von ihm aus wird ein Stockwerk, bestehend aus einzelnen Grundrissmodulen aufgebaut, wie in Abschnitt [4.2](#) beschrieben. Seine beiden Attribute „Höhe“ und „Breite“ legen die Höhe des Stockwerks und die Breite seiner Mauern fest.

$$\begin{aligned} C &\cup \text{Grundstein}, \\ A &\cup \{\text{Höhe}, \text{Breite}\}, \\ \tau(\text{Grundstein}) &= \{\text{Höhe}, \text{Breite}\} \end{aligned}$$

Definition A.2. Konzept Grundrissmodul

Als nächstes werden die „Grundrissmodule“ benötigt, aus denen die grobe Gebäudestruktur besteht. Das Modul hat nur das Attribut „Breite“, das die Breite seiner Mauern bezeichnet. Optional kann hier noch der Typ des Moduls angegeben werden, also ob es sich hierbei um den Haupttrakt, einen Nebenflügel, einen Mauer vorsprung oder einen Erker handelt. Die Höhe des Grundrissmoduls entspricht der seines zugehörigen Grundsteins.

$$\begin{aligned}
C &\cup \text{Grundrissmodul}, \\
A &\cup \text{Typ}, \\
\tau(\text{Grundrissmodul}) &= \{\text{Typ}, \text{Breite}\}, \\
\sigma(\text{Grundrissmodul}, \text{Breite}) &= \top
\end{aligned}$$

Definition A.3. Konzept Polygon

Das Konzept „Polygon“ beinhaltet den konvexen Kantenzug für ein Grundrissmodul. Sein Attribut „Scheitelpunkte“ enthält eine Liste von Scheitelpunkten (im \mathbb{R}^2), die den Kantenzug beschreiben.

$$\begin{aligned}
C &\cup \text{Polygon}, \\
A &\cup \text{Scheitelpunkte}, \\
\tau(\text{Polygon}) &= \text{Scheitelpunkte}
\end{aligned}$$

Definition A.4. Konzept PolygonKantenUnterteilung

Mit dem Konzept „PolygonKantenUnterteilung“ können optional gezielt einzelne Kanten eines Polygons für die Aufteilung in Wände überlappungsfrei unterteilt werden. Die Attribute sind der Index der zu unterteilenden „Kante“ und die „Unterteilung“ selbst. Hier ist eine Liste von Tupeln anzugeben, wobei der erste Wert den Bereich der Kante angibt und der zweite Wert den Fensterstil.

$$\begin{aligned}
C &\cup \text{PolygonKantenUnterteilung}, \\
A &\cup \{\text{Kante}, \text{Unterteilungsliste}\}, \\
\tau(\text{PolygonKantenUnterteilung}) &= \{\text{Kante}, \text{Unterteilungsliste}\}
\end{aligned}$$

Wie Definition 5.3 in Kapitel 5 angegeben, entstehen aus den Kanten eines Polygons Basiswände und schließlich Wände. Wie die Unterteilung vollzogen wird, kann entweder über den Stil allgemein angegeben oder über Prozeduren gezielt für beliebige Kanten festgelegt werden. Für die Fensterstile in den Unterteilungen wird nicht direkt ein Fensterstil-Konzept angegeben, sondern ein Bezeichner, den der Fensterstil selber als Attribut gesetzt hat. Dadurch dass ein Stil mehrere Fensterstile besitzen kann, ist es möglich durch Austausch des Stils gleich eine ganze Reihe von Fensterstilen zu ersetzen. Wird kein Fensterstil angegeben, so wird erst versucht den Fensterstil des Stil-Konzepts des Grundrissmoduls zu wählen. Sonst wird der Fensterstil des Grundsteins genommen. Eine Liste für die Unterteilung, die in einer Instanz gesetzt sein muss, sieht z. B. folgendermaßen aus: $\{([0,0.5],2), ([0.5,1.0],4)\}$. In diesem Fall bedeutet das, dass die erste Hälfte der Kante mit dem Fensterstil „2“ und die zweite Hälfte mit dem Fensterstil „4“ versehen wird.

Definition A.5. Konzept Dach

Die in den Definitionen 4.12 für ein Flachdach und 4.13 für ein allgemeines Dach beschriebenen Dachformen werden hier in einem einzigen „Dach“-Konzept zusammengefasst. Das Dach beherbergt eine ganz Reihe von Attributen, wobei zusätzlich das Attribut „Typ“ eingeführt wird, um zwischen den beiden grundlegenden Dachformen unterscheiden zu können.

$$\begin{aligned}
C &\cup \text{Dach}, \\
A &\cup \{Kante_i, Kante_j, c_Kante_i, d_Kante_j, a_First, b_First\}, \\
\tau(\text{Dach}) &= \{Typ, Kante_i, Kante_j, c_Kante_i, d_Kante_j, \\
&\quad a_First, b_First, Hoehe\}
\end{aligned}$$

Damit sind alle Konzepte, die für die grobe Gebäudestruktur notwendig sind, definiert. Nun werden die Beziehungen zwischen den Konzepten angegeben.

Definition A.6. Relation `initiales_Grundrissmodul`

Diese Relation setzt dem Konzept „Grundstein“ das initiale Grundrissmodul, an das alle weitere Module und Stockwerke angeknüpft werden. Diese Relation hat keine Attribute.

$$\begin{aligned}
R &\cup \text{initiales_Grundrissmodul}, \\
\alpha(\text{initiales_Grundrissmodul}) &= \text{Grundstein}, \\
\omega(\text{initiales_Grundrissmodul}) &= \text{Grundrissmodul}
\end{aligned}$$

Definition A.7. Relation `naechstes_Stockwerk`

Diese Relation definiert den Grundstein für das nachfolgende Stockwerk und verbindet somit Grundsteine miteinander. Diese Relation hat keine Attribute.

$$\begin{aligned}
R &\cup \text{naechstes_Stockwerk}, \\
\alpha(\text{naechstes_Stockwerk}) &= \text{Grundstein}, \\
\omega(\text{naechstes_Stockwerk}) &= \text{Grundstein}
\end{aligned}$$

Hier wird ein klassischer Zyklus definiert, der aber so im Weltmodell nicht zustande kommen kann, da der Prototyp im Weltmodell auf unzulässige Zyklen prüft.

Definition A.8. Relation `intra_Verbindung`

Damit werden die Grundrissmodule aneinander gefügt, um den Grundriss eines Stockwerks zu formen. Hier sind die Verbindungstypen Kante-Kante- und die beiden Punkt-Punkt-Verbindungen zusammengefasst. Die Form der Verbindung wird über das Attribut „Typ“ spezifiziert. Alle weiteren Attribute entspringen den Definitionen 4.2, 4.3 und 4.4.

$$\begin{aligned}
R &\cup \text{intra_Verbindung}, \\
A &\cup \{Kante_k, a_Kante_i, b_Kante_j\}, \\
\tau(\text{intra_Verbindung}) &= \{Typ, Kante_k, Kante_i, Kante_j, \\
&\quad a_Kante_i, b_Kante_j\}, \\
\alpha(\text{intra_Verbindung}) &= \text{Grundrissmodul}, \\
\omega(\text{intra_Verbindung}) &= \text{Grundrissmodul}
\end{aligned}$$

Definition A.9. Relation `inter_Verbindung`

Mit dieser Relation wird eine das Stockwerk übergreifende Verbindung für Grundrissmodule definiert. Dabei wird ein Grundrissmodul mit dem Grundrissmodul des vorigen Stockwerks verbunden. Diese Relation besitzt keine Attribute.

$$\begin{aligned}
 R &\cup \text{inter_Verbindung}, \\
 \alpha(\text{inter_Verbindung}) &= \text{Grundrissmodul}, \\
 \omega(\text{inter_Verbindung}) &= \text{Grundrissmodul}
 \end{aligned}$$

Definition A.10. Relation hat_Polygon

Mit dieser Relation wird eine das Stockwerk übergreifende Verbindung für Grundrissmodule definiert. Dabei wird ein Grundrissmodul mit dem Grundrissmodul des vorigen Stockwerks verbunden. Diese Relation besitzt keine Attribute.

$$\begin{aligned}
 R &\cup \text{hat_Polygon}, \\
 \alpha(\text{hat_Polygon}) &= \text{Grundrissmodul}, \\
 \omega(\text{hat_Polygon}) &= \text{Polygon}
 \end{aligned}$$

Definition A.11. Relation hat_Polygon_Unterteilung

Diese Relation verbindet ein Polygon-Konzept mit einer oder mehreren PolygonKantenUnterteilung-Konzepten. Diese Relation besitzt keine Attribute.

$$\begin{aligned}
 R &\cup \text{hat_Polygon_Unterteilung}, \\
 \alpha(\text{hat_Polygon_Unterteilung}) &= \text{Polygon}, \\
 \omega(\text{hat_Polygon_Unterteilung}) &= \text{PolygonKantenUnterteilung}
 \end{aligned}$$

Definition A.12. Relation hat_Dach

Hier wird ein Grundrissmodul-Konzept mit einem Dach-Konzept verbunden. Diese Relation hat keine Attribute.

$$\begin{aligned}
 R &\cup \text{hat_Dach}, \\
 \alpha(\text{hat_Dach}) &= \text{Grundrissmodul}, \\
 \omega(\text{hat_Dach}) &= \text{Dach}
 \end{aligned}$$

A.2 Konzepte und Relationen des Fassadenstils

Definition A.13. Konzept Stil

In diesem Konzept wird der „Stil“ für entweder ein ganzes Gebäude oder für grobe Teilbereiche, wie den Grundrissmodulen, festgelegt. Dieses Konzept hat keine Attribute.

$$C \cup \text{Stil}$$

Definition A.14. Konzept Unterteilung

In diesem Konzept werden verschiedene Strategien für die „Unterteilung“ der Basiswände und Hausecken festgelegt. Seine Attribute haben dabei folgende Bedeutung. Über „ZWE“ wird angegeben, ob Zwischenwandelemente erzeugt werden sollen. Im Attribut „allg_Unterteilung“ wird eine allgemeine Vorschrift zur

Unterteilung der Basiswände beschrieben. Die beiden Attribute „h_ZWE“ und „h_Hausecken“ bestimmen, ob Zwischenwandelemente bzw. Hausecken horizontal unterteilt werden sollen. Das Konzept der Wandpartitionen wird ggf. auf diese Elemente angewendet.

$$\begin{aligned} C &\cup \text{Unterteilung}, \\ A &\cup \{ZWE, \text{allg_Unterteilung}, h_ZWE, h_Hausecken\}, \\ \tau(\text{Unterteilung}) &= \{ZWE, \text{allg_Unterteilung}, h_ZWE, h_Hausecken\} \end{aligned}$$

Definition A.15. Konzept Mauerziegelverband

Über dieses Konzept wird ein Mauerziegelverband definiert. Das Attribut „Beschreibung“ enthält die Definition des Mauerziegelverbands nach Abschnitt 5.3.1.

$$\begin{aligned} C &\cup \text{Mauerziegelverband}, \\ A &\cup \{\text{Beschreibung}\}, \\ \tau(\text{Mauerziegelverband}) &= \{\text{Beschreibung}\} \end{aligned}$$

Definition A.16. Konzept Zyklopenmauerwerk

Mit diesem Konzept wird ein Zyklopenmauerwerk festgelegt. Das Zyklopenmauerwerk kann entweder über die Anzahl der Steine („Steine“) oder dem mittleren Abstand der Steinzentren („Abstand“) definiert werden. Über das Attribut „Lloyd“ wird die maximale Anzahl an Lloyd Relaxationen angegeben.

$$\begin{aligned} C &\cup \text{Zyklopenmauerwerk}, \\ A &\cup \{\text{Steine}, \text{Abstand}, \text{Lloyd}\}, \\ \tau(\text{Zyklopenmauerwerk}) &= \{\text{Steine}, \text{Abstand}, \text{Lloyd}\} \end{aligned}$$

Definition A.17. Konzept Wandpartition

Mit diesem Konzept können Wände horizontal in einzelne „Wandpartitionen“ unterteilt werden. Eine Ausprägung dieses Konzepts erzeugt genau eine Partition. Sollen weitere erzeugt werden, so sind entsprechend viele Ausprägungen anzugeben, wobei das System eigenständig auf Überlappung überprüft und ggf. Anpassungen vornimmt. Über die beiden Attribute „v_Start“ und „v_Stop“ wird der Bereich der Partition definiert.

$$\begin{aligned} C &\cup \text{Wandpartition}, \\ A &\cup \{v_Start, v_Stop\}, \\ \tau(\text{Wandpartition}) &= \{v_Start, v_Stop\} \end{aligned}$$

Definition A.18. Konzept FensterTuer

Mit diesem Konzept wird die grobe rechteckige Öffnung für Fenster und Türen definiert. Ob es sich um ein Fenster oder eine Tür handelt, wird mit „Typ“ festgelegt. Entsprechend wird bei Türen die Verfeinerung der unteren Kante nicht beachtet. Die Öffnung kann relativ zur Wand oder in absoluten Koordinaten angegeben werden. Zu den Attributen für die relative Positionierung zählen „v_Start“,

„v_Stop“, „h_Start“ und „h_Stop“. Für die absolute Positionierung werden die Attribute „Breite“, „Hoehe“, „dist_links“ und „dist_rechts“ benötigt. Einer der wichtigsten Parameter ist „id“. Über ihn kann der Ausprägung eine Bezeichnung zugewiesen werden, um von Wandunterteilungen als Fensterstil genutzt werden zu können.

$$\begin{aligned} C &\cup \text{FensterTuer}, \\ A &\cup \{h_Start, h_Stop, dist_links, dist_rechts, id\}, \\ \tau(\text{FensterTuer}) &= \{v_Start, v_Stop, h_Start, h_Stop, \\ &\quad \text{Breite, Hoehe, dist_links, dist_rechts, id}\} \end{aligned}$$

Definition A.19. Konzept FTVerfeinerung

Mit diesem Konzept wird eine Kante der groben Fenster- oder Türöffnung über einen Kantenzug verfeinert. Die Verfeinerung ist über das Attribute „Gesimsbeschreibung“ gegeben und beinhaltet eine aus Abschnitt 5.4.1 bekannte Gesimsbeschreibung. Über das Attribut „Unterteilungen“ wird angegeben, wie detailliert der Kantenzug später gezeichnet werden soll.

$$\begin{aligned} C &\cup \text{FTVerfeinerung}, \\ A &\cup \{\text{Gesimsbeschreibung, Unterteilungen}\}, \\ \tau(\text{FTVerfeinerung}) &= \{\text{Gesimsbeschreibung, Unterteilungen}\} \end{aligned}$$

Definition A.20. Konzept FTStockGesims

Über dieses Konzept wird der (verfeinerten) Kante ein Fenster- bzw. Türstock in Gesimsform zugewiesen. Mit dem Attribut „dominant“ wird die Dominanz der Kante festgelegt. Die Ausmaße des Fenster- oder Türstocks werden über die Attribute „Groesse“, „Breite“ und „Vorsprung“ bestimmt.

$$\begin{aligned} C &\cup \text{FTStockGesims}, \\ A &\cup \{\text{dominant, Groesse, Vorsprung}\}, \\ \tau(\text{FTStockGesims}) &= \{\text{dominant, Groesse, Breite, Vorsprung}\} \end{aligned}$$

Definition A.21. Konzept FTStockDoppelgesims

Über dieses Konzept wird der (verfeinerten) Kante ein Fenster- bzw. Türstock in Form eines Doppelgesimses zugewiesen. Über das Attribut „dominant“ wird die Dominanz der Kante festgelegt. Die Ausmaße des Fenster- oder Türstocks werden über die Attribute „Groesse“, „Breite“ und „Vorsprung“ bestimmt.

$$\begin{aligned} C &\cup \text{FTStockDoppelgesims}, \\ \tau(\text{FTStockDoppelgesims}) &= \{\text{dominant, Groesse, Breite, Vorsprung}\} \end{aligned}$$

Definition A.22. Konzept FTStockKlinker

Über dieses Konzept erhält eine (verfeinerte) Kante Ziegelsteine als Fenster- bzw. Türstock. Über das Attribut „dominant“ wird die Dominanz der Kante festgelegt.

Die Ausmaße des Verlaufs der Mörtelschicht werden über die Attribute „Groesse“, „Breite“ und „Vorsprung“ bestimmt.

$$C \cup FTStockKlinker,$$

$$\tau(FTStockKlinker) = \{ \text{dominant, Groesse, Breite, Vorsprung} \}$$

Definition A.23. Konzept FTStockEinfach

Mit diesem Konzept erhält eine (verfeinerte) Kante einen schlichten Fenster- bzw. Türstock. Über das Attribut „dominant“ wird die Dominanz der Kante festgelegt. Die Ausmaße des Verlaufs der Ziegelsteine werden über die Attribute „Groesse“, „Breite“ und „Vorsprung“ bestimmt.

$$C \cup FTStockEinfach,$$

$$\tau(FTStockEinfach) = \{ \text{dominant, Groesse, Breite, Vorsprung} \}$$

Definition A.24. Konzept FTOrnament

Mit diesem Konzept kann ein Fenster- oder Türstock um ein „FTOrnament“ mit Rahmen bereichert werden. Die Attribute „Abstand_links“ und „Abstand_rechts“ bestimmen die Breite des Ornaments bzgl. der Breite des Fensterstocks. Mit „Hoehe“, „Vorsprung“ und „Breite“ werden die Ausmaße des Rahmens, falls vorhanden, festgelegt. Die Form des Ornaments ist in „Ornamentbeschreibung“ gegeben.

$$C \cup FTOrnament,$$

$$A \cup \{ \text{Abstand_links, Abstand_rechts, Ornamentbeschreibung} \}$$

$$\tau(FTOrnament) = \{ \text{Groesse, Breite, Vorsprung, Abstand_links,} \\ \text{Abstand_rechts, Ornamentbeschreibung} \}$$

Definition A.25. Konzept Fensterrahmen

Dieses Konzept beschreibt den inneren Rahmen eines Fensters, also den Teil, der nicht zum Fensterstock gehört. Das Attribut „Vorsprung“ bestimmt in welchem Abstand relativ zur Wand die Rahmen und das Fenster gezeichnet werden. Mit „Breite“ wird die Rahmenbreite festgelegt.

$$C \cup Fensterrahmen,$$

$$\tau(Fensterrahmen) = \{ \text{Breite, Vorsprung} \}$$

Definition A.26. Konzept Fensterglas

Mit diesem Konzept wird das „Fensterglas“ definiert. Dieses Konzept hat keine Attribute.

$$C \cup Fensterglas$$

Definition A.27. Konzept Gesims

In diesem Konzept werden die Daten für ein „Gesims“ definiert. Als Attribute enthält es die „Gesimsbeschreibung“ und in „Unterteilungen“ den Detailgrad der Bögen im Kantenzug.

$$C \cup Gesims,$$

$$\tau(Gesims) = \{ \text{Gesimsbeschreibung, Unterteilungen} \}$$

Definition A.28. Konzept Klinker

Dieses Konzept beschreibt die Form der Ziegelsteine für das Konzept „FTStock-Klinker“. Mit den Attributen „Breite“, „Vorsprung“ und „Abstand“ werden die Breite eines Ziegels, sein Vorsprung gegenüber der Mörtelschicht und der Abstand zwischen zwei Ziegeln bestimmt.

$$C \cup \text{Klinker},$$

$$\tau(\text{Klinker}) = \{\text{Breite, Vorsprung, Abstand}\}$$

Definition A.29. Konzept Material

Mit dem Konzept „Material“ werden klassische Materialeigenschaften festgelegt. Darunter fallen „Ambientanteil“, „Diffusanteil“, „Glanzanteil“ und „Rauheit“.

$$C \cup \text{Material},$$

$$A \cup \{\text{Ambientanteil, Diffusanteil, Glanzanteil, Rauheit}\},$$

$$\tau(\text{Material}) = \{\text{Ambientanteil, Diffusanteil, Glanzanteil, Rauheit}\}$$

Definition A.30. Konzept Farbe

Mit diesem Konzept wird einem „Material“ seine diffuse Farbe zugewiesen. Das Attribut „rgb“ bestimmt den Rot-, Grün- und Blauanteil.

$$C \cup \text{Farbe},$$

$$A \cup \{\text{rgb}\},$$

$$\tau(\text{Farbe}) = \{\text{rgb}\}$$

Definition A.31. Konzept Opazitaet

Dieses Konzept bestimmt die Lichtdurchlässigkeit eines Materials. Das Attribut „rgb“ bestimmt den Rot-, Grün- und Blauanteil.

$$C \cup \text{Opazitaet},$$

$$\tau(\text{Opazitaet}) = \{\text{rgb}\}$$

Definition A.32. Konzept Glanzlicht

Über das Konzept „Glanzlicht“ wird die Farbe des Glanzlichtes festgelegt. Das Attribut „rgb“ bestimmt den Rot-, Grün- und Blauanteil.

$$C \cup \text{Glanzlicht},$$

$$\tau(\text{Glanzlicht}) = \{\text{rgb}\}$$

Damit sind alle Konzepte des Stils definiert und können über Relationen miteinander verbunden werden. Unter dem Konzept *Stil* sind alle anderen Konzepte der Fassadenelemente zusammengefasst.

Definition A.33. Relation hat_Unterteilung

Über diese Relation wird einem Stil eine Vorschrift zur Unterteilung der Basiswände zugeordnet. Ein Stil kann nur eine Form der Unterteilung haben.

$$R \cup \text{hat_Unterteilung},$$

$$\alpha(\text{hat_Unterteilung}) = \text{Stil},$$

$$\omega(\text{hat_Unterteilung}) = \text{Unterteilung}$$

Definition A.34. Relation `hat_Wandpartition`

Damit erhält ein Stil Angaben über Wandpartitionen. Mehrere Wandpartitionen können einem Stil zugewiesen sein.

$$\begin{aligned} R \cup \text{hat_Wandpartition}, \\ \alpha(\text{hat_Wandpartition}) &= \text{Stil}, \\ \omega(\text{hat_Wandpartition}) &= \text{Wandpartition} \end{aligned}$$

Definition A.35. Relation `hat_Mauerziegelverband`

Damit erhält ein Stil Angaben über sein Mauerwerk. Ein Stil kann immer nur einen Typ von Mauerwerk besitzen. Werden Mauerziegelverbände und Zyklopenmauerwerke angegeben, so wird nur das erste Mauerwerk berücksichtigt. Später, im Zwischenmodell, kann selbst einer einzelnen Wandpartition ein Mauerwerk zugewiesen werden.

$$\begin{aligned} R \cup \text{hat_Mauerziegelverband}, \\ \alpha(\text{hat_Mauerziegelverband}) &= \text{Stil}, \\ \omega(\text{hat_Mauerziegelverband}) &= \text{Mauerziegelverband} \end{aligned}$$

Definition A.36. Relation `hat_Zyklopenmauerwerk`

Über diese Relation kann dem Stil alternativ ein Zyklopenmauerwerk zugewiesen werden. Sonst verhält sich diese Relation wie „`hat_Mauerziegelverband`“.

$$\begin{aligned} R \cup \text{hat_Zyklopenmauerwerk}, \\ \alpha(\text{hat_Zyklopenmauerwerk}) &= \text{Stil}, \\ \omega(\text{hat_Zyklopenmauerwerk}) &= \text{Zyklopenmauerwerk} \end{aligned}$$

Definition A.37. Relation `hat_FensterTuer`

Hiermit werden die Fenster- und Türstile für einen Stil gesetzt.

$$\begin{aligned} R \cup \text{hat_FensterTuer}, \\ \alpha(\text{hat_FensterTuer}) &= \text{Stil}, \\ \omega(\text{hat_FensterTuer}) &= \text{FensterTuer} \end{aligned}$$

Definition A.38. Relation `hat_Fensterrahmen`

Ein Fenster erhält damit seine inneren Rahmen.

$$\begin{aligned} R \cup \text{hat_Fensterrahmen}, \\ \alpha(\text{hat_Fensterrahmen}) &= \text{FensterTuer}, \\ \omega(\text{hat_Fensterrahmen}) &= \text{Fensterrahmen} \end{aligned}$$

Definition A.39. Relation `hat_FTVerfeinerungLinks`

Hierüber werden Angaben zur Verfeinerung für die linke Kante einer Fenster- oder Türöffnung gemacht. Es existieren noch drei weitere Relationen, die die Kanten unten, rechts und oben verfeinern.

$$\begin{aligned} R \cup \text{hat_FTVerfeinerungLinks}, \\ \alpha(\text{hat_FTVerfeinerungLinks}) &= \text{FensterTuer}, \\ \omega(\text{hat_FTVerfeinerungLinks}) &= \text{FTVerfeinerung} \end{aligned}$$

Definition A.40. Relation *hat_FTDoppelgesims*

Mit dieser Relation wird für eine verfeinerte Kante ein Tür- bzw. Fensterstock vom Typ *Doppelgesims* gesetzt. Für die drei weiteren Formen wird jeweils analog eine eigene Relation definiert.

$$\begin{aligned} R &\cup \textit{hat_FTDoppelgesims}, \\ \alpha(\textit{hat_FTDoppelgesims}) &= \textit{FTVerfeinerung}, \\ \omega(\textit{hat_FTDoppelgesims}) &= \textit{FTDoppelgesims} \end{aligned}$$

Definition A.41. Relation *hat_Klinker*

Diese Relation weist einem Fenster- oder Türstock, der aus Ziegelsteinen besteht, die Beschreibung der Ziegel zu.

$$\begin{aligned} R &\cup \textit{hat_Klinker}, \\ \alpha(\textit{hat_Klinker}) &= \textit{FTStockKlinker}, \\ \omega(\textit{hat_Klinker}) &= \textit{Klinker} \end{aligned}$$

Definition A.42. Relation *hat_Gesims*

Das Konzept *Gesims* steht mit einer ganzen Reihe von anderen Konzepten in Verbindung und es existieren entsprechend viele Relationen.

$$\begin{aligned} R &\cup \textit{hat_Gesims}, \\ \alpha(\textit{hat_Gesims}) &= \{ \textit{Wandpartition}, \textit{Fensterrahmen}, \textit{Ornament}, \\ &\quad \textit{FTStockDoppelGesims}, \textit{FTStockGesims} \}, \\ \omega(\textit{hat_Gesims}) &= \textit{Gesims} \end{aligned}$$

Definition A.43. Relation *hat_Material*

Das Konzept *Material* ist wie das *Gesims* bei vielen Konzepten über die Relation „*hat_Material*“ vertreten.

$$\begin{aligned} R &\cup \textit{hat_Material}, \\ \alpha(\textit{hat_Material}) &= \{ \textit{Wandpartition}, \textit{Fensterrahmen}, \textit{Fensterglas}, \textit{Ornament}, \\ &\quad \textit{FTStockDoppelGesims}, \textit{FTStockGesims}, \textit{FTStockKlinker}, \\ &\quad \textit{FTStockEinfach} \}, \\ \omega(\textit{hat_Material}) &= \textit{Material} \end{aligned}$$

Damit sind alle Relationen des Stils beschrieben.

Literaturverzeichnis

- [Arc00] ARCON VISUELLE ARCHITEKTUR. *ArCon*, 2000. <http://www.arcon-software.com/> (gesehen 2006).
- [Aut08] AUTODESK. *Maya*, 2008. <http://www.autodesk.com/maya> (gesehen 2008).
- [BA05] DANIEL BEKINS und DANIEL G. ALIAGA. *Build-by-number: rearranging the real world to visualize novel architectural spaces*. In: *Proceedings of the IEEE Visualization Conference*, Seiten 143–150, 2005.
- [Bau72] BRUCE G. BAUMGART. *Winged-edge Polyhedron Representation*. Technischer Bericht STAN-CS-320, Computer Science Department, Stanford University, Stanford, CA, 1972.
- [BBH⁺92] FRANZ BAADER, HANS-JÜRGEN BÜRCKERT, BERNHARD HOLLUN-
DER, ARMIN LAUX und WERNER NUTT. *Terminologische Logiken*.
KI, 6(3):23–33, 1992.
- [BBJ⁺01] PETER J. BIRCH, SHAUN P. BROWNE, VINCE J. JENNINGS, AN-
DREW M. DAY und DAVID B. ARNOLD. *Rapid procedural-modelling
of architectural structures*. In: *Proceedings of the 2001 conference on
Virtual reality, archeology, and cultural heritage*, Seiten 187–196. ACM
Press, 2001.
- [BBMR89] ALEXANDER BORGIDA, RONALD J. BRACHMAN, DEBORAH L. MC-
GUINNESS und LORI ALPERIN RESNICK. *CLASSIC: a structural data
model for objects*. SIGMOD Rec., 18(2):58–67, 1989.
- [BE04] BLIZZARD-ENTERTAINMENT. *World of Warcraft*, 2004. <http://www.worldofwarcraft.com> (gesehen 2008).
- [BFO04] JAN BENDER, DIETER FINKENZELLER und PETER OEL. *HW3D: A
tool for interactive real-time 3D visualization in GIS supported flood
modelling*. In: *Proceedings of the 17th international conference on
computer animation and social agents*, Seiten 305–314, Geneva (Swit-
zerland), 2004.

- [BJDA01] PETER J. BIRCH, VINCE J. JENNINGS, ANDREW M. DAY und DAVID B. ARNOLD. *Procedural Modelling of Vernacular Housing for Virtual Heritage Environments*. In: *Proceedings of the 19th Eurographics UK Conference*, 2001.
- [Bra95] A. BRAUSEWETTER. *Das Bauformenbuch*. Verlag von E. A. Seemann, Leipzig, 1895.
- [BS85] RONALD J. BRACHMANN und JAMES G. SCHMOLZE. *An Overview of the KL-ONE Knowledge Representation System*. *Cognitive Science*, 9(2):171–202, April 1985.
- [CFF⁺98] VINAY K. CHAUDHRI, ADAM FARQUHAR, RICHARD FIKES, PETER D. KARP und JAMES P. RICE. *Open Knowledge Base Connectivity*. Artificial Intelligence Center SRI International and Knowledge Systems Laboratory Stanford University, 1998.
- [Cod70] EDGAR FRANK CODD. *A Relational Model of Data for Large Shared Data Banks*. *Communications of the ACM*, 13(6):377–387, 1970.
- [CSHD03] MICHAEL F. COHEN, JONATHAN SHADE, STEFAN HILLER und OLIVER DEUSSEN. *Wang Tiles for image and texture generation*. *ACM Transactions on Graphics (TOG)*, 22(3):287–294, 2003.
- [DHL⁺98] OLIVER DEUSSEN, PAT HANRAHAN, BERND LINTERMANN, RADOMÍR MĚCH, MATT PHARR und PRZEMYSŁAW PRUSINKIEWICZ. *Realistic modeling and rendering of plant ecosystems*. In: *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, Seiten 275–286, New York, NY, USA, 1998. ACM Press.
- [DW98] PETER DAHM und FRIEDBERT WIDMANN. *Das Graphenlabor*. Fachberichte Informatik 11/98, Universität Koblenz-Landau, Institut für Informatik, Koblenz, 1998.
- [EF95] JÜRGEN EBERT und ANGELIKA FRANZKE. *A Declarative Approach to Graph Based Modeling*. In: E. MAYR, G. SCHMIDT und G. TINHOFER (Herausgeber): *Graphtheoretic Concepts in Computer Science*, Nummer 903 in *LNCS*, Seiten 38–50, Berlin, 1995. Springer Verlag.
- [EMP⁺03] DAVID S. EBERT, F. KENTON MUSGRAVE, DARWYN PEACHEY, KEN PERLIN und STEVEN WORLEY. *Texturing & Modeling. A Procedural Approach*. Morgan Kaufmann Publishers, Inc., 2003.
- [Fau02] MICHAEL FAUTZ. *Objekt- und Texturrekonstruktion mit einer robotergeführten Kamera*. Doktorarbeit, Universität Karlsruhe (TH), Fakultät für Informatik, 2002.

- [FO98] PETR FELKEL und STEPAN OBDRZALEK. *Straight Skeleton Implementation*. In: LASZLO SZIRMAY KALOS (Herausgeber): *14th Spring Conference on Computer Graphics (SCCG'98)*, Seiten 210–218, 1998.
- [GG71] L. S. GARANIN und A. YA. GURDIZ. *A Heuristic Method of Forming up the Schemes of Layouts of Industrial Complexes*. In: *Proceedings of the 5. Information Processing (IFIP) Congress 71*, 5, Seiten 1066–1075. North-Holland Publishing Company, 1971.
- [Goo08] GOOGLE. *SketchUp*, 2008. <http://www.sketchup.com/> (gesehen 2008).
- [GPSL03a] STEFAN GREUTER, JEREMY PARKER, NIGEL STEWART und GEOFF LEACH. *Real-time procedural generation of 'pseudo infinite' cities*. In: *GRAPHITE '03: Proc. of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, Seiten 87–94, New York, NY, USA, 2003. ACM Press.
- [GPSL03b] STEFAN GREUTER, JEREMY PARKER, NIGEL STEWART und GEOFF LEACH. *Undiscovered Worlds - Towards a Framework for Real-Time Procedural World Generation*. In: *Proceedings of the Fifth Intern. Digital Arts and Culture Conference*, 2003.
- [GSL07] STEFAN GREUTER, NIGEL STEWART und GEOFF LEACH. *Procedural Urban Streets in Real-Time*. In: *Proceedings of the 20th international conference on computer animation and social agents*, Seiten 119–128, Hasselt (Belgium), 2007.
- [GW87] RAFAEL C. GONZALES und PAUL WINTZ. *Digital image processing*. Addison-Wesley Publishing Company, Inc., 1987.
- [Hau01] ALEJO HAUSNER. *Simulating Decorative Mosaics*. In: EUGENE FIUME (Herausgeber): *SIGGRAPH 2001, Computer Graphics Proceedings*, Seiten 573–578, 2001.
- [HF01] SVEN HAVEMANN und DIETER FELLNER. *A versatile 3D model representation for cultural reconstruction*. In: *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, Seiten 205–212. ACM Press, 2001.
- [HHD03] STEFAN HILLER, HEINO HELLWIG und OLIVER DEUSSEN. *Beyond Stippling - Methods for Distributing Objects on the Plane*. *Computer Graphics Forum*, 22(3):515–515, 2003.
- [Hof89] CHRISTOPH M. HOFFMANN. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers, Inc., San Mateo, California 94403, 1989.

- [HWB95] MIKAKO HARADA, ANDREW WITKIN und DAVID BARAFF. *Interactive physically-based manipulation of discrete/continuous models*. In: *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, Seiten 199–208, New York, NY, USA, 1995. ACM Press.
- [KM96] THEODOR KRAUTH und FRANZ SALES MEYER. *Das Steinhauerbuch*. Verlag von E. A. Seemann, Leipzig, 1896.
- [Koc90] WILFRIED KOCH. *Baustilkunde*. Orbis Verlag, 1990.
- [LD03] ROBERT G. LAYCOCK und ANDREW M. DAY. *Automatically Generating Roof Models from Building Footprints*. In: *The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003. Journal of WSCG - FULL Papers*, Band 11, 2003.
- [LDG01] JUSTIN LEGAKIS, JULIE DORSEY und STEVEN GORTLER. *Feature-based cellular texturing for architectural models*. In: *Proceedings of the 28th conference on Computer graphics and interactive techniques*, Seiten 309–316. ACM Press, 2001.
- [LFL01] THORSTEN LIEBIG, DIETER FINKENZELLER und MARKO LUTHER. *KB-VISION: A Tool for Graphical Manipulation and Visualization of Domain Models*. In: E. KAWAGUCHI H. KANGASSALO, H. JAAKKOLA (Herausgeber): *Proceedings of the Tenth European-Japanese Conference on Information Modelling and Knowledge Bases (EJC 2000)*, Frontiers in Artificial Intelligence and Applications, Seiten 161–168. IOS Press, 2001.
- [LKK93] PETER C. LOCKEMANN, GERHARD KRÜGER und HEIKO KRUMM. *Telekommunikation und Datenhaltung*. Carl Hanser Verlag München Wien, 1993.
- [MAA84] ANNE MACDOUGALL, TONY ADAMS und PAULINE ADAMS. *Learning Logo on the Apple II*. Simon & Schuster, 1984.
- [Män88] MARTTI MÄNTYLÄ. *An Introduction to Solid Modeling*. Computer Science Press, Maryland, 1988.
- [MB91] ROBERT MACGREGOR und MARK H. BURSTEIN. *Using a Description Classifier to Enhance Knowledge Representation*. *IEEE Expert*, 6(3):41–46, 1991.
- [Mey27] FRANZ SALES MEYER. *Handbuch der Ornamentik*. Verlag von E. A. Seemann, Leipzig, 1927.

- [Miy90] KAZUNORI MIYATA. *A method of generating stone wall patterns*. In: *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, Seiten 387–394, New York, NY, USA, 1990. ACM Press.
- [MWH⁺06] PASCAL MÜLLER, PETER WONKA, SIMON HAEGLER, ANDREAS ULMER und LUC VAN GOOL. *Procedural modeling of buildings*. ACM Transactions on Graphics (TOG), 25(3):614–623, 2006.
- [MZWG07] PASCAL MÜLLER, GANG ZENG, PETER WONKA und LUC VAN GOOL. *Image-based Procedural Modeling of Facades*. ACM Transactions on Graphics (TOG), 26(3), 2007.
- [NMMK05] ANDRE NICOLL, JAN MESETH, GERO MÜLLER und REINHARD KLEIN. *Fractional Fourier Texture Masks: Guiding Near-Regular Texture Synthesis*. Computer Graphics Forum, 24(3):569–579, September 2005.
- [NN96] ERNST NEUFERT und PETER NEUFERT. *Bauentwurfslehre, 34. überarbeitete Auflage*. Vieweg Friedr. + Sohn Verlag, 1996.
- [PL96] PRZEMYSŁAW PRUSINKIEWICZ und ARISTID LINDENMAYER. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, USA, 1996.
- [PM01] YOAV I. H. PARISH und PASCAL MÜLLER. *Procedural modeling of cities*. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, Seiten 301–308. ACM Press, 2001.
- [PTVF07] WILLIAM H. PRESS, SAUL A. TEUKOLSKY, WILLIAM H. VETTERLING und BRIAN P. FLANNERY. *Numerical recipes*. Cambridge University Press, Cambridge, Third Auflage, 2007. The art of scientific computing.
- [Rei90] RICHARD REID. *Bauwerke - Ein Reiseführer*. Weidbild Verlag, 1990.
- [RS08] RACER-SYSTEMS. *RacerPro*, 2008. <http://www.racer-systems.com/> (gesehen 2008).
- [Sal73] ARTO SALOMAA. *Formal Languages*. ACM Press, 1973.
- [Sch92] UWE SCHÖNING. *Theoretische Informatik kurz gefasst*. BI Wissenschaftsverlag, 1992.
- [SG71] GEORG STINY und JAMES GIPS. *Shape grammars and the generative specification of painting and sculpture*. In: *Proceedings of the 5. Information Processing (IFIP) Congress 71*, 5, Seiten 1460–1465. North-Holland Publishing Company, 1971.

- [Sow91] JOHN F. SOWA. *Toward the Expressive Power of Natural Language*. In: JOHN F. SOWA (Herausgeber): *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Seiten 157–189. Kaufmann, San Mateo, 1991.
- [W3C04a] W3C. *EBNF for XML*, 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/#sec-notation> (gesehen 2007).
- [W3C04b] W3C. *OWL Web Ontology Language*, 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/> (gesehen 2007).
- [WWSR03] PETER WONKA, MICHAEL WIMMER, FRANÇOIS SILLION und WILLIAM RIBARSKY. *Instant architecture*. ACM Transactions on Graphics (TOG), 22(3):669–677, 2003.
- [WZS98] MICHAEL T. WONG, DOUGLAS E. ZONGKER und DAVID H. SALE-SIN. *Computer-generated floral ornament*. In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, Seiten 423–434. ACM Press, 1998.

Die computergestützte Modellierung detaillierter Gebäudefassaden wird in der Computeranimation, bei der Produktion von Kinofilmen und bei Computerspielen immer bedeutender. Eine geeignete interaktive Modellierung muss für den Benutzer in der Bedienung mächtig und einfach zugleich sein. Um Modelle in kurzer Zeit erstellen zu können, muss ein hoher Abstraktionsgrad im Modellierungsprozess gewährleistet sein.

In dieser Arbeit werden neue Techniken vorgestellt, die das breite Spektrum der Anforderungen abdecken. Sie erstrecken sich von der Erstellung einer groben Gebäudestruktur über die Beschreibung detaillierter Fassadenelemente bis hin zur automatischen Geometrie- und Texturerzeugung. Zusammengeführt ist ein System entstanden, das den geforderten Ansprüchen gerecht wird. Es zeichnet sich gegenüber bekannten Techniken durch seine Flexibilität und Vielfalt aus.

Die Arbeit ist mit zahlreichen Illustrationen und Graphiken ausgeschmückt, um die verschiedenen Techniken zu veranschaulichen.