

DEEP CROSS-DOMAIN BUILDING EXTRACTION FOR SELECTIVE DEPTH ESTIMATION FROM OBLIQUE AERIAL IMAGERY

B. Ruf^{a,b}, L. Thiel^a, M. Weinmann^b

^aFraunhofer IOSB, Video Exploitation Systems, 76131 Karlsruhe, Germany -
{boitumelo.ruf, laurenz.thiel}@iosb.fraunhofer.de

^bInstitute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology,
76131 Karlsruhe, Germany - {boitumelo.ruf, martin.weinmann}@kit.edu

Commission I, ICWG I/II

KEY WORDS: aerial oblique imagery, object detection, building extraction, deep learning, convolutional neural networks, transfer learning, depth estimation, semi-global matching

ABSTRACT:

With the technological advancements of aerial imagery and accurate 3d reconstruction of urban environments, more and more attention has been paid to the automated analyses of urban areas. In our work, we examine two important aspects that allow online analysis of building structures in city models given oblique aerial image sequences, namely automatic building extraction with convolutional neural networks (CNNs) and selective real-time depth estimation from aerial imagery. We use transfer learning to train the Faster R-CNN method for real-time deep object detection, by combining a large ground-based dataset for urban scene understanding with a smaller number of images from an aerial dataset. We achieve an average precision (AP) of about 80% for the task of building extraction on a selected evaluation dataset. Our evaluation focuses on both dataset-specific learning and transfer learning. Furthermore, we present an algorithm that allows for multi-view depth estimation from aerial image sequences in real-time. We adopt the semi-global matching (SGM) optimization strategy to preserve sharp edges at object boundaries. In combination with the Faster R-CNN, it allows a selective reconstruction of buildings, identified with regions of interest (RoIs), from oblique aerial imagery.

1. INTRODUCTION

In recent years, more and more attention has been paid to the automated analyses of urban areas due to an increase in urbanization and the need for more efficient urban planning and sustainable development. While the consideration of urban trees in planning processes can provide measurable economic, environmental, social and health benefits (Kelly, 2011), the monitoring and analysis of the given buildings e.g. allows to create and update city models (Kolbe, 2009), finding suitable roof planes for solar energy installations (Schuffert et al., 2015) and reasoning about a diversity of processes.

The basis for such automated analyses is typically given with data acquired from aerial platforms. The automated extraction of objects from such data has been a topic of great interest over decades (Mayer, 2008). To reason about possible updates or extract objects of interest, the 2d outlines of buildings as well as a reconstruction of the 3d shape of buildings may be compared to the information given in model data. To foster research on both of these issues, the ISPRS benchmark on urban object classification and 3d building reconstruction (Rottensteiner et al., 2012) has been initialized and addressed by a diversity of approaches.

Due to technological advancements, aerial oblique imagery nowadays provides a ubiquitous tool for an accurate 3d reconstruction of urban environments (Cavegn et al., 2014). Based on the 3d reconstruction, city models with different level of detail (LOD) can be created. Particularly LOD2 building models with distinctive roof structures and larger building installations like balconies and stairs and LOD3 building models with detailed wall and roof structures, doors and windows (Kolbe, 2009) are

desirable for a variety of applications. Recent work in the field of 3d change detection (Palazzolo and Stachniss, 2017; Ruf and Schuchert, 2016; Taneja et al., 2015) suggests to use images for an efficient verification or improvement of such models.

In our work, we aim at methodologies that allow online change detection and analysis of building structures in city models given oblique aerial image sequences captured from commercial off-the-shelf (COTS) drones. Hereby, we use "online" to indicate that our algorithms should allow a direct annotation and processing of the input sequence. A key aspect in this process is the real-time image-based depth estimation, giving us information on the current structure of the scene. However, the model data estimated from imagery typically differs greatly from city models in its detail and the objects it holds. While city models only depict buildings, image-based models also contain vegetation and dynamic objects such as cars and pedestrians. Apart from being potential sources for errors, the reconstruction of such objects is not necessary for finding changes in building structures.

In this paper, we examine two aspects to remedy this, namely extraction of buildings from aerial imagery and selective real-time image-based depth estimation for the identified objects. As no dataset exists that allows to train and test a method for building extraction from imagery captured by COTS drones, we use transfer learning by using a large ground-based dataset intended for semantic urban scene understanding and adopting a small aerial dataset to our needs. The assumption is that oblique imagery captured from low altitudes show building parts that are also depicted in ground-based imagery, which e.g. holds for building façades.

In summary, our main contribution is a methodology for building extraction and selective 3d reconstruction from oblique aerial imagery which

- relies on cross-domain training of a convolutional neural network (CNN) to deliver object proposals corresponding to buildings,
- allows real-time image-based depth estimation that preserves strong discontinuities at object boundaries, giving buildings sharp edges, and
- is evaluated on different datasets, whereby the focus is put on both dataset-specific learning and transfer learning to obtain appropriate object proposals as the basis for 3d reconstruction.

This paper is structured as follows: We briefly summarize related work in Section 2 giving an overview of the recent advances in the fields of object detection and image-based 3d reconstruction. In Section 3, we give a detailed description of our methodology for deep building extraction and depth estimation. The training process together with the datasets used, as well as the experimental results are described in Section 4. We give a short discussion of the achieved results in Section 5 before we conclude our paper in Section 6.

2. RELATED WORK

In the following, we provide an overview on the advances in the field of object detection, followed by a short outline on the related work of image-based 3d reconstruction.

2.1 Object Detection

In the past, a rich variety of approaches have been presented for object detection in given imagery. Thereby, several approaches focus on instance search, i.e. the detection of specific types of objects by generating candidate windows around regions of interest (RoIs) and thus delivering object proposals. In general, this can be achieved with *window scoring methods* and *grouping methods*. For a more detailed discussion of such methods, we refer to (Sommer et al., 2016), and we instead only briefly summarize the main ideas.

The *window scoring methods* typically generate candidate windows by applying either a sliding window approach or a random sampling technique. For each candidate window, a score is calculated which allows to rank or discard these windows w.r.t. their score. To define the scoring of candidate windows, different cues may be taken into account such as a generic objectness measure quantifying how likely it is for an image window to contain an object of any class (Alexe et al., 2012). This measure combines several image cues such as multiscale saliency, color contrast, edge density or superpixels straddling. A different objectness measure relies on the number of edges that exist in the window and those edges that are members of contours overlapping the window's boundary (Zitnick and Dollár, 2014).

The *grouping methods* typically perform an image segmentation followed by a grouping of segments to generate multiple (possibly overlapping) segments that are likely to correspond to objects. Thereby, the grouping is typically based on a diverse set of cues including superpixel shape, appearance cues and boundary estimates (Hosang et al., 2016; Sommer et al., 2016). A commonly used approach is known as selective search (Uijlings et

al., 2013). This approach relies on a hierarchical grouping to get a set of small starting regions forming the basis of a selective search. Subsequently, a greedy data-driven algorithm is used to iteratively merge regions based on a variety of complementary grouping criteria and a variety of complementary color spaces with different invariance properties. This yields a small set of high-quality object locations.

With the great success of modern deep learning in a variety of research domains, such techniques have also been introduced in recent years for object detection in the form of deriving object proposals. The combination of a region proposal method with deep CNNs has been proposed with the R-CNN (Girshick et al., 2014). This approach makes use of selective search (Uijlings et al., 2013) to generate category-independent region proposals, but is generally agnostic to the particular region proposal method. The derived proposals are provided as input to a CNN that extracts a feature vector of fixed length for each region. Finally, the feature vectors characterizing the derived RoIs are used to classify the objects located within these regions using a support vector machine (SVM). The R-CNN approach outperformed classical methods on public benchmarks for object detection. To improve efficiency, the Fast R-CNN (Girshick, 2015) has been proposed. Amongst others, the introduced innovations also allow using the very deep VGG-16 network (Simonyan and Zisserman, 2014) which performed quite well in the ImageNet Localization+Classification Challenge 2014 (Russakovsky et al., 2015). At this point, the computation of region proposals was identified as bottleneck in terms of runtime.

Instead of using conventional approaches to derive region proposals (Felzenszwalb et al., 2010; Uijlings et al., 2013), the use of a region proposal network (RPN) has been proposed (Ren et al., 2017) which takes an image (of any size) as input and delivers a set of candidate windows, i.e. rectangular object proposals, each with an objectness score. These region proposals in turn are used by a Fast R-CNN for object detection and, due to the increase in computational efficiency, the resulting approach was dubbed Faster R-CNN (Ren et al., 2017). This improvement allows real-time object detections on a GPU achieving state-of-the-art results of over 70% mean average precision (mAP) on public benchmarks.

2.2 Image-based 3d Reconstruction

In recent years, a lot of attention has been paid to dense image-based depth estimation and model reconstruction. With the ever increasing computational power of modern hardware, it is possible to perform an online reconstruction from imagery acquired with a single moving camera (Newcombe and Davison, 2010; Newcombe et al., 2011; Stühmer et al., 2012). The presented 3d models are of great detail, but typically depict only a small-scale scenery.

With semi-global matching (SGM) (Hirschmüller, 2008), an optimization strategy was introduced which can be used for depth estimation from a stereo setup or a multi-image matching (Rothermel et al., 2012). Since its announcement, SGM has been used and adopted for a wide variety of applications in the field of dense depth estimation and 3d reconstruction as it provides a good trade-off between accuracy and computational effort. As a result, state-of-the-art methods for aerial image-based 3d reconstruction also rely on the SGM optimization (d'Angelo and Kuschik, 2012; Haala et al., 2015; Rothermel et al., 2012).

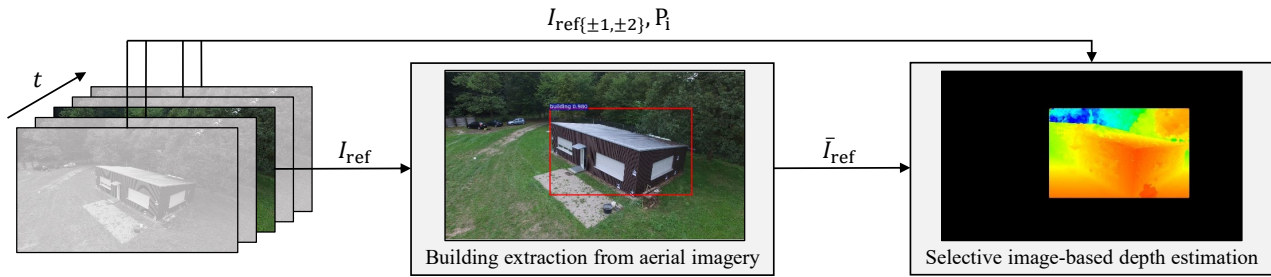


Figure 1. Overview of our methodology for building-aware selective image-based depth estimation.

3. METHODOLOGY

The processing pipeline of our approach is composed of two parts: building extraction and selective depth estimation, as outlined in Figure 1. For each iteration of the presented pipeline, we choose a bundle of five consecutive images of an input sequence which depict the scene of interest from five slightly different viewpoints. We select the center image of the input bundle as our reference image and pass it on to the first processing step in which we use an object detection algorithm to identify buildings and mark these with axis-aligned RoIs. The annotated reference image is then passed on to the second step in which it is combined with the remaining four images of the input bundle to estimate the depth for the previously identified RoIs. In the following, both parts of the pipeline will be discussed in more detail.

3.1 Building Extraction from Aerial Imagery

For the task of building extraction from aerial imagery, we have adopted and trained the Faster R-CNN (Ren et al., 2017) based on the VGG-16 network of Simonyan and Zisserman (2014). As depicted in Figure 2, the Faster R-CNN first passes the input image to a set of convolutional layers creating a feature map. The VGG-16 based Faster R-CNN holds 13 convolutional layers and produces a feature map of size $14 \times 14 \times 512$. The second processing step is made up of the RPN which computes numerous proposals by sliding differently sized windows over the input. As the RPN is sharing its convolutional layers with the rest of the network, it takes the previously computed feature map as input. The resulting proposals are recombined with the initial feature map in the third step and fed into a RoI pooling layer which uses max-pooling to convert the feature map within each proposal into a spatially confined feature map of size $7 \times 7 \times 512$. Two fully connected layers then produce feature vectors of size $1 \times 1 \times 4096$ which are passed to the bounding box regression to compute four coordinates for each of the bounding boxes and to the softmax classifier to compute the corresponding class scores. We have adjusted the output size of the classifier to two, as we only want to distinguish if an identified object is a building or not. As the

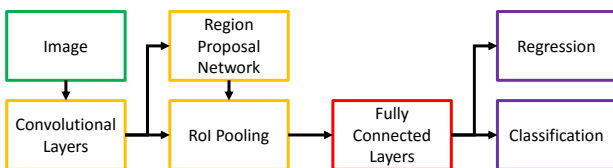


Figure 2. Structural overview of the Faster R-CNN. Based on the deep VGG-16 network, it is composed of 13 convolutional layers and two fully connected layers. The object proposals are classified by a softmax classifier and their coordinates are computed by the regression layer.

regression layer computes the four coordinates of each bounding box, its output size is set to eight accordingly.

Furthermore, as we aim to detect buildings in oblique aerial imagery, which might result in a partial occlusion of buildings by other buildings, we have substituted the conventional non-maximum suppression (NMS) by the Soft-NMS as presented by Bodla et al. (2017). Instead of removing all bounding boxes of an object that lie up to a certain threshold within the bounding box with the highest score \mathcal{M} , the Soft-NMS decays the score of all non-maximum boxes according to the amount of overlap w.r.t. \mathcal{M} . This results in better detections of objects which are partially occluded by other objects of their kind, as the proposals are not eliminated immediately.

3.2 Selective Image-based Depth Estimation

For image-based depth estimation, we employ a multi-view plane-sweep algorithm. As input, our algorithm takes a bundle of five consecutive images, one reference image I_{ref} and four matching images $I_{ref\{\pm 1, \pm 2\}}$, two to either side of I_{ref} in terms of image acquisition time. To this end, we assume that the input images are provided together with corresponding camera projection matrices $\mathbf{P}_i = \mathbf{K} [\mathbf{R}_i^T \quad -\mathbf{R}_i^T \mathbf{C}_i]$, which consist of the rotation matrix $\mathbf{R}_i^T \in SO(3)$ and the position of the camera centers $\mathbf{C}_i \in \mathbb{R}^3$, relative to a reference coordinate system. As we typically consider an image sequence, we assume the same intrinsic calibration matrix \mathbf{K} for all images.

The algorithm samples the scene by using multiple planes $\Pi_r \in \mathbb{R}^3$, parameterized by their normal vector \mathbf{n}_r and distance d_r relative to \mathbf{C}_{ref} . For each plane, the four matching images $I_{ref\{\pm 1, \pm 2\}}$ are warped into I_{ref} according to the plane-induced homography

$$\mathbf{H}_{i \rightarrow ref}^r = \mathbf{K} \cdot \frac{\mathbf{R} - \mathbf{t} \cdot \mathbf{n}_r^T}{d_r} \cdot \mathbf{K}^{-1}, \quad (1)$$

with \mathbf{R}, \mathbf{t} : relative rotation and translation between the cameras,
 \mathbf{n}_r : plane normal vector,
 d_r : plane distance from the reference camera.

For each set of planes that share the same normal vector, we select the distances d_r so that the scene is sampled inversely between two bounding planes Π_{min} and Π_{max} (Ruf et al., 2017). As similarity measure in the process of image matching, we choose a 9×7 Census Transform (Zabih and Woodfill, 1994). In order to account for occlusions, we accumulate the matching cost within the left and right subset of the matching images and select the



Figure 3. Example figures of the datasets used. (a) Cityscapes dataset showing an urban street scene with building façades. (b) ISPRS benchmark dataset comprising oblique aerial imagery of a suburban area. (c) Exemplary scene of the GoogleEarth dataset depicting different building kinds and dense vegetation. (d) Excerpt of our own dataset used to demonstrate the local image-based depth estimation.

minimum of the two as suggested by Kang et al. (2001). This yields a per-pixel matching cost for each plane which is stored in a three-dimensional cost volume \mathcal{C} of size $W \times H \times D$, where W and H represent the image size and D is the number of planes with which the scene is sampled.

Given the cost volume, we employ an edge-aware SGM optimization (Hirschmueller, 2008) to extract the per-pixel minimum and with it for each pixel the distance d_r of the plane Π_r which best approximates the structure of the scene. The adapted energy function of the SGM optimization is as follows:

$$E(d_r) = \sum_p \mathcal{C}(p, d_r(p)) + \sum_{q \in N_p} P_1 \cdot T[|\text{Idx}(d_r(p)) - \text{Idx}(d_r(q))| = 1] + \sum_{q \in N_p} P_2 \cdot T[|\text{Idx}(d_r(p)) - \text{Idx}(d_r(q))| > 1]. \quad (2)$$

With the first term, the matching costs of all pixels are accumulated for a given d_r . With the second term, all neighborhood pixels q with a neighboring plane parameterization, i.e. the index of $d_r(p)$ and $d_r(q)$ only changes by a maximum of one, are penalized with P_1 . Neighboring pixels with other plane parameterizations are penalized with P_2 . The optimization of Equation 2 is done by dynamic programming along eight concentric paths as suggested by Hirschmueller (2008).

In order to enforce discontinuities at object boundaries, we use the line segment detector (LSD) introduced by Von Gioi et al. (2012) to create a binary line image $I_{\text{ref}}^{\text{line}}$ of the reference image, where $I_{\text{ref}}^{\text{line}}(p) = 1$ indicates the existence of a line segment at a given pixel p . With this line image, we adjust P_2 according to:

$$P_2 = \begin{cases} P_1 & , \text{ if } I_{\text{ref}}^{\text{line}}(p) = 1 \\ P_2 & , \text{ otherwise.} \end{cases} \quad (3)$$

Thus we reduce P_2 to P_1 , if a given pixel p is on a line segment of $I_{\text{ref}}^{\text{line}}$, allowing neighboring pixels with strong discontinuities to influence the selection of the optimal d_r . Dependent on the parameterization of the LSD, most line segments are detected at object boundaries, which allows us to enforce strong edges in the depth image.

Having found the optimal per-pixel plane parameterization $\hat{\Pi}(p)$, the final depth image is calculated by intersecting a viewing ray through pixel p with $\hat{\Pi}(p)$.

For a selective reconstruction that is confined to identified RoIs, we perform the per-plane image warping on the complete images, yet the matching and the subsequent SGM optimization is

only done within the given RoIs, treating each bounding box individually. We optimized our algorithm with CUDA, achieving real-time performance when aiming to estimate depth maps for every key-frame which are typically generated at 1 Hz - 2 Hz by state-of-the-art SLAM systems. This enables the online analysis of the depicted scene.

4. EXPERIMENTS

Before presenting the results achieved by the Faster R-CNN and our algorithm for depth estimation, we introduce the datasets used for training and evaluation.

4.1 Datasets

Deep learning based approaches, such as the Faster R-CNN, require a large amount of training data. To the best of our knowledge, there exists no suitable training dataset for object detection from oblique aerial imagery. Assuming that such images depict both the façade and the roof of buildings, we have adopted two different datasets for training the CNN, namely the Cityscapes dataset for semantic urban scene understanding (Cordts et al., 2016) and the dataset given with the ISPRS benchmark for multi-platform photogrammetry (Nex et al., 2015).

The Cityscapes dataset (cf. Figure 3(a)) consists of a large number of images which were captured from a car driving through urban areas. It provides a pixel-wise semantic ground truth. We generated a ground truth for building extraction by converting the semantic labeling of the building class into axis-aligned bounding boxes, needed for the training of the Faster R-CNN. The dataset provides disjoint subsets used for training and evaluation. The *training* subset consists of approx. 2900 images, whereas the *validation* subset contains 500 images.

The Dortmund-Zeche-Zollern subset of the ISPRS benchmark (cf. Figure 3(b)) consists of few large images with approx. 48 mega pixels captured from nadir and oblique viewpoints. Due to the large image size, we have cropped each input image into 64 equal-sized subimages with a size of 1022×766 pixels. Because this benchmark does not contain any annotated ground truth, we have selected 700 oblique images and annotated the depicted buildings with bounding boxes. We have divided the annotated images into a *training* and a *validation* subset containing 500 and 200 images, respectively.

To further evaluate the performance of the Faster R-CNN to detect buildings on a different dataset, we have generated a semi-synthetic dataset from GoogleEarth (cf. Figure 3(c)). We have selected seven different scenes with underlying model data in order

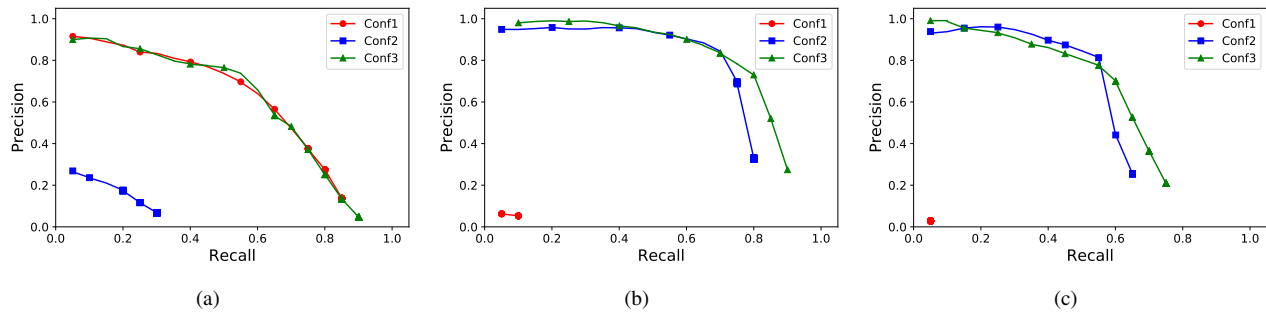


Figure 4. Precision-recall curves for different validation datasets. (a) Results achieved by the three training configurations evaluated on the Cityscapes validation set. (b) Results achieved on the ISPRS validation set. (c) Precision-recall values corresponding to an evaluation on the GoogleEarth dataset.

to render different viewpoints. The scenes differ in the amount and the kind of buildings they show. We have rendered each scene with three off-nadir angles (0° , 30° and 60°) and from five distances (100 m, 150 m, 200 m, 250 m and 300 m), where each image has a size of 1920×1080 pixels.

A fourth dataset was used to demonstrate the performance of our approach on a real-world scenario (cf. Figure 3(d)). It represents an own dataset of a single building which was captured with a DJI Phantom 3 from three different heights (2 m, 8 m and 15 m) while flying around the front of the building. This dataset fits to the use-case of a selective 3d reconstruction as the building only covers a small amount of the input images. The input images are of size 1920×1080 pixels.

4.2 Building Extraction from Aerial Imagery

4.2.1 Training We have selected three configurations to train the Faster R-CNN for building extraction. The configurations differ in the training datasets used:

Conf1: Cityscapes

Conf2: ISPRS Dortmund-Zeche-Zollern

Conf3: Cityscapes + ISPRS Dortmund-Zeche-Zollern

For each dataset, the available training subset was used. As a training method of the Faster R-CNN, we have chosen the presented approximate joint training which alternates between training the Fast R-CNN network and the additional RPN. Ren et al. (2017) suggest that it results in an approx. $1.5 \times$ speedup and produces a similar accuracy as when using the alternating training method. For all configurations, we have initialized the weights with a VGG-16 model which was pretrained on the ImageNet dataset. We have used 250,000 training iterations.

In order to increase the amount of training data and to oppose overfitting, we have augmented the initial training sets with affinely transformed copies of the input images. First, we horizontally flipped the input data to increase the database and generalize the model. Second, due to the perspective of the Cityscapes image data, most bounding boxes surrounding buildings are of slim horizontal nature. In order to oppose overfitting to such geometry, we have rotated the images by 90° and added them to the dataset. Third, to accommodate for different object sizes, the input images were randomly downsampled by three different scaling factors. As the image size of the Cityscapes and the ISPRS datasets differ, we used different scaling factors for each dataset.

Table 1. Quantitative results achieved by the three configurations evaluated on the three validation datasets. The underlined values represent the baselines as the training and evaluation was done on the same kind of data.

	<i>Conf1</i>	<i>Conf2</i>	<i>Conf3</i>
Train.	Cityscapes	ISPRS	Cityscapes + ISPRS
Eval.	Cityscapes	ISPRS	Cityscapes + ISPRS
Cityscapes	<u>60.57 %</u>	6.62 %	60.76 %
ISPRS	1.01 %	<u>73.43 %</u>	80.81 %
GoogleEarth	0.45 %	56.81 %	60.70 %

In terms of the Cityscapes dataset, we have downsampled the image data to 60%, 40% or 20% of its original size. For the ISPRS dataset, we used images scaled to 100%, 80% or 60% of their original image size.

4.2.2 Results We have evaluated the performance of the Faster R-CNN to detect buildings on the validation subsets of the Cityscapes and ISPRS datasets as well as the generated GoogleEarth dataset. We have evaluated models of the Faster R-CNN that were trained with the three previously described training configurations. For evaluation on the Cityscapes dataset, we have downsampled the test images to 40% of their original size. The test images of the ISPRS dataset were used with the original image size, while the GoogleEarth images were scaled to 80%. These values were determined empirically, giving the best results for each dataset. We have set the RPN to generate 12,000 proposals for each image.

Table 1 shows the average precision (AP) achieved by the three training configurations in detecting buildings on the three evaluation datasets. For the Soft-NMS, we have used a linear decay function with a NMS threshold of 0.3. Furthermore, in Figure 4(a), we have plotted the precision-recall curves of *Conf1-3*, evaluated on the validation set of the Cityscapes dataset. Figure 4(b) shows the performance of the training configurations w.r.t. the ISPRS validation dataset and Figure 4(c) shows the results of the evaluation on the GoogleEarth dataset.

Figure 5 shows an excerpt of the results achieved by the Faster R-CNN trained with *Conf3* and combined with Soft-NMS. This configuration was chosen as it quantitatively achieves the best results on all three datasets. While Figures 5(a) - 5(c) hold the results for the ISPRS dataset, Figures 5(d) - 5(f) show the detections for images of the GoogleEarth dataset. Each detection is marked with a red bounding box. With it, the Faster R-CNN computes a confidence score in $[0.0, 1.0]$ with which it identifies an object as

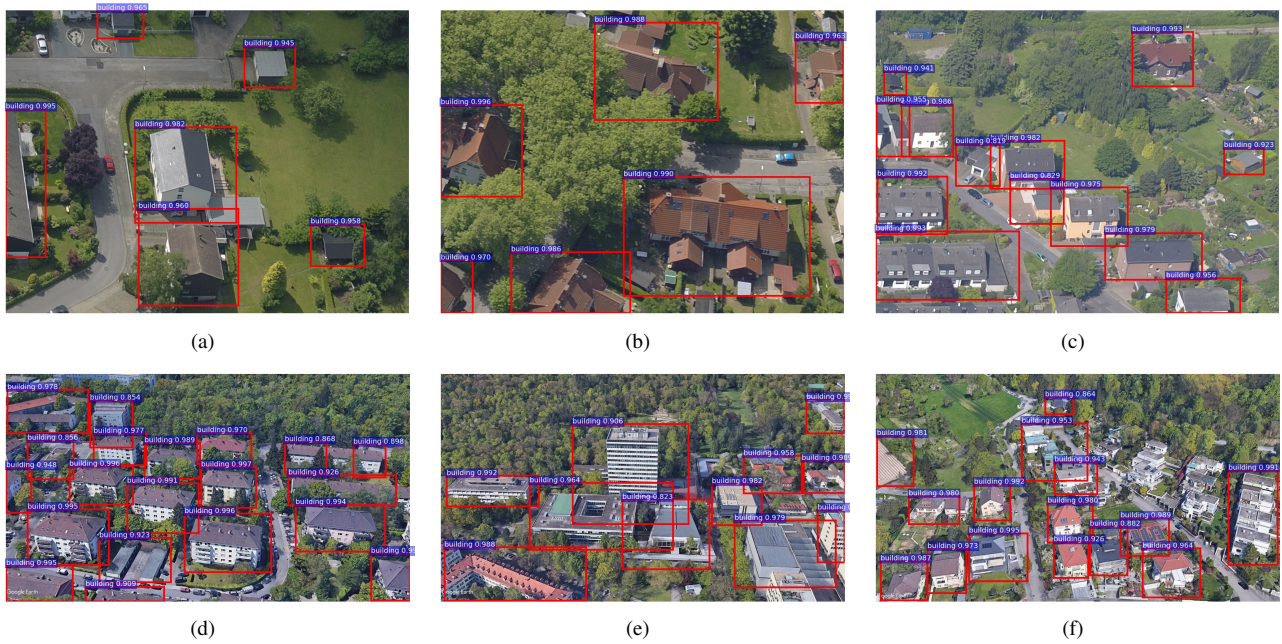


Figure 5. Excerpt of the qualitative results achieved by the Faster R-CNN trained with *Conf3*. Figures (a), (b) and (c) depict results from the ISPRS dataset. Figures (d), (e) and (f) show an excerpt of the detection from the GoogleEarth dataset. Only detections with a confidence score > 0.8 are highlighted.

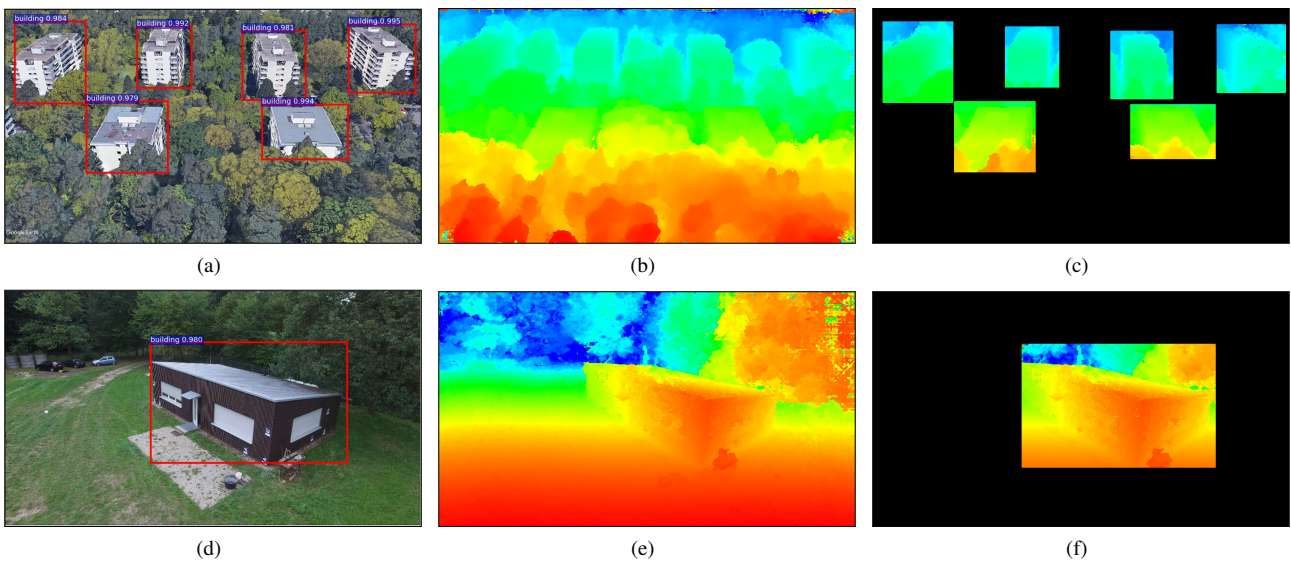


Figure 6. Results of selective image-based depth estimation. Figures (a) and (d) show the reference frames with corresponding detections. Figure (a) was captured at a distance of 150 m with an off-nadir angle of 60° . Figure (d) was captured at an altitude of 15 m. Full depth maps corresponding to the reference frames are shown in Figures (b) and (e). The results of a selective image-based depth estimation are shown in Figures (c) and (f), where the depth is only computed for the detected objects. Depth values are color-coded going from red (near) to blue (far).

a building. These scores are displayed above each RoI together with the corresponding class name of the detected object. In Figure 5, only objects with a score > 0.8 are displayed.

All training and experiments were performed on a NVIDIA Titan X. The runtime to process one image for the Cityscapes dataset is approx. 140 ms, for the ISPRS dataset approx. 160 ms and for the GoogleEarth dataset approx. 285 ms. The differences arise from the different image sizes which were used for the experiment.

4.3 Selective Image-based Depth Estimation

We have evaluated our approach for selective image-based depth estimation on two datasets. First, we have generated an image set, made up of five different viewpoints around a reference image (cf. Figure 6(a)), from one of the scenes of the GoogleEarth dataset. The scene shows a few isolated buildings surrounded by lots of vegetation, which is one of the key motivations to use an object detection algorithm to extract ROIs prior to the depth estimation.

Further experiments are done on imagery taken from our own dataset (cf. Figures 3(d) and 6(d)). Again, it shows an isolated object of interest surrounded by lots of scenery not relevant for the task of building reconstruction. It was captured from 15 m altitude.

We have parameterized our algorithm to use 128 frontoparallel planes. The penalties of our SGM optimization were set to $P_1 = 5$ and $P_2 = 50$ in order to enforce smooth reconstructions within object boundaries. The LSD was used with the presented default configurations and the input images were downsampled to half of their size. The camera projection matrices were computed using multicore bundle adjustment (Wu et al., 2011). We employ a final 3×3 median filter to reduce outliers.

Figure 6 holds an excerpt of our experiments for selective depth estimation. Figures 6(a) and 6(d) show the respective reference images with the RoIs generated by the Faster R-CNN. Again, we use the model trained with *Conf3*. In Figures 6(b) and 6(e), the full depth maps are shown, in which the depth values are visually encoded, going from red (near) to blue (far). In Figures 6(c) and 6(f), the results of the selective depth estimation are shown. Hereby, the depth values were only computed for the RoIs identified by the Faster R-CNN and fused together into one depth map corresponding to I_{ref} .

Again, all experiments were performed on a NVIDIA Titan X. The depth estimation for the full image took approx. 510 ms. The selective depth estimation reduced the complete runtime to an average of 226 ms.

5. DISCUSSION

We consider the underlined numbers in Table 1 as baseline, because they represent the performance of the Faster R-CNN in detecting buildings when respectively trained and evaluated on the same kind of data. Concerning these baselines, we achieve equal or better results when training the Faster R-CNN with a combination of the Cityscapes and the ISPRS datasets, i.e. *Conf3*. Even when evaluated on a dataset, which it has not seen during training, i.e. the GoogleEarth dataset, the *Conf3* model achieves equal results as the *Conf1* model when evaluated on the Cityscapes dataset.

The loss in performance w.r.t. the GoogleEarth dataset can be attributed to different factors. First, the fact that it is a different dataset which was not used during training. We assume that the model does not generalize well to building structures which are atypical w.r.t. the training data. Figure 5(f) shows that the model does not identify the building blocks on the right, as well as the individual buildings in the center. Furthermore, the field on the left is falsely detected as a building with a significantly high score. A second factor may be the quality of the underlying 3d model of GoogleEarth and the projected textures. A poor quality in the model or the texture may result in false detections. Nonetheless, the results achieved by *Conf3* are significantly better than the ones achieved by the other models on this dataset.

The precision-recall curves reveal how the different training datasets influence the results achieved. Figure 4(a) shows that due to the size of the Cityscapes dataset, in comparison to the annotated subset of the ISPRS dataset, the *Conf3* model delivers very similar results as the *Conf1* model. Figure 4(b) shows that the additional use of the Cityscapes dataset improves the results compared to the baseline of the ISPRS dataset, i.e. *Conf2*.

Nonetheless, in order to detect buildings from aerial imagery, the use of the ISPRS training data is needed, as Figures 4(b) and 4(c) show that *Conf1* does not achieve any reasonable results.

The training process of the Faster R-CNN keeps the first two convolutional layers of the VGG-16 network fixed (Girshick, 2015). Changing this, so that the weights of all layers are trained worsens the results by 1%-2%. A similar effect was observed when a dense-sparse-dense training (Han et al., 2017) was performed in order to regularize the model. In further experiments, we have evaluated the use of standard NMS instead of Soft-NMS. As expected, the results achieved were inferior to those presented.

The results in Figure 6 show that our algorithm for image-based depth estimation allows to reconstruct building structures while preserving sharp edges at object boundaries. Our algorithm achieves real-time performance for the full reconstruction. Nonetheless, the runtime can be reduced by 50% if only objects of interest are considered. The results also reveal that there seems to be no significant loss in quality of the selective reconstruction compared to the full depth map. As our main focus in this paper is the use of Faster R-CNN to extract buildings from aerial imagery, an elaborate performance evaluation of our image-based depth estimation algorithm is done in other work.

As the building detection currently only extracts axis-aligned RoIs, it happens that parts of irrelevant objects are encapsulated in the bounding boxes. To overcome this, one can use a semantic segmentation that attributes a specific object class to each pixel. However, as our depth estimation algorithm is parallelized on the GPU, it needs to be invoked on a regular grid. The recently published Mask R-CNN (He et al., 2017) would allow an extraction of bounding boxes prior to reconstruction and a pixel-wise post-filtering of the resulting depth images.

Altogether, this paper shows that we can use transfer learning of the Faster R-CNN for building extraction from oblique aerial imagery, by using a combination of a large ground-based dataset and a smaller aerial dataset. The results show that the use of the large Cityscapes dataset is necessary to learn building-related features. Nonetheless, a small aerial training dataset is vital to generalize the model to perform detections in aerial imagery. With our approach, we achieve a real-time extraction and selective reconstruction of building structures. However, a registration of the reference frame w.r.t. the city model is needed in order to perform a 3d change detection and analysis.

6. CONCLUSION & FUTURE WORK

In summary, we present a methodology for cross-domain building extraction from aerial imagery with CNNs. We combine a large ground-based dataset for urban scene understanding with a smaller number of images from an aerial dataset to train the Faster R-CNN method for real-time deep object detection. We achieve an AP of about 80% for the task of building extraction on a selected evaluation dataset. In combination with a presented algorithm for real-time image-based depth estimation, it allows us to selectively reconstruct buildings from aerial imagery, preserving sharp edges. These reconstructions can be used for 3d scene analysis or change detection in city models.

As axis-aligned bounding boxes do not always appropriately approximate the building outlines, we aim to incorporate a semantic segmentation for post-filtering the depth map. To this end, in

the image-based depth estimation, we perform the same image warpings for all identified RoIs. In future work, different homographies are to be used for each RoI, which allows to further adjust the sampling process to the object of interest. Finally, we plan to combine it with an algorithm to register the image data against textureless model data, allowing us to match the depth map against the model data for structural change detection.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the provision of the datasets for multi-platform photogrammetry by ISPRS and EuroSDR, released in conjunction with the ISPRS scientific initiative 2014 and 2015, led by ISPRS ICWG I/Vb.

REFERENCES

- Alexe, B., Deselaers, T. and Ferrari, V., 2012. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(11), pp. 2189–2202.
- Bodla, N., Singh, B., Chellappa, R. and Davis, L. S., 2017. Soft-NMS – Improving object detection with one line of code. In: *Proc. IEEE International Conference on Computer Vision*, pp. 5562–5570.
- Cavegn, S., Haala, N., Nebiker, S., Rothermel, M. and Tutzauer, P., 2014. Benchmarking high density image matching for oblique airborne imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-3, pp. 45–52.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B., 2016. The Cityscapes dataset for semantic urban scene understanding. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223.
- d'Angelo, P. and Kusch, G., 2012. Dense multi-view stereo from satellite imagery. In: *Proc. IEEE International Geoscience and Remote Sensing Symposium*, pp. 6944–6947.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D. and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9), pp. 1627–1645.
- Girshick, R., 2015. Fast R-CNN. In: *Proc. IEEE International Conference on Computer Vision*, pp. 1440–1448.
- Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587.
- Haala, N., Rothermel, M. and Cavegn, S., 2015. Extracting 3d urban models from oblique aerial images. In: *Proc. IEEE Joint Urban Remote Sensing Event*, pp. 1–4.
- Han, S., Pool, J., Narang, S., Mao, H., Gong, E., Tang, S., Elsen, E., Vajda, P., Paluri, M., Tran, J. et al., 2017. DSD: Dense-sparse-dense training for deep neural networks. *arXiv preprint arXiv:1607.04381*.
- He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask R-CNN. In: *Proc. IEEE International Conference on Computer Vision*, pp. 2980–2988.
- Hirschmüller, H., 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), pp. 328–341.
- Hosang, J., Benenson, R., Dollár, P. and Schiele, B., 2016. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(4), pp. 814–830.
- Kang, S. B., Szeliski, R. and Chai, J., 2001. Handling occlusions in dense multi-view stereo. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 103–110.
- Kelly, M., 2011. Urban trees and the green infrastructure agenda. In: *Proc. Urban Trees Research Conference*, pp. 166–180.
- Kolbe, T. H., 2009. Representing and exchanging 3d city models with CityGML. In: *3D Geo-Information Sciences*, pp. 15–31.
- Mayer, H., 2008. Object extraction in photogrammetric computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing* 63(2), pp. 213–222.
- Newcombe, R. A. and Davison, A. J., 2010. Live dense reconstruction with a single moving camera. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1498–1505.
- Newcombe, R. A., Lovegrove, S. J. and Davison, A. J., 2011. DTAM: Dense tracking and mapping in real-time. In: *Proc. IEEE International Conference on Computer Vision*, pp. 2320–2327.
- Nex, F., Gerke, M., Remondino, F., Przybilla, H., Bäumker, M. and Zurhorst, A., 2015. ISPRS benchmark for multi-platform photogrammetry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3/W4, pp. 135–142.
- Palazzolo, E. and Stachniss, C., 2017. Change detection in 3d models based on camera images. In: *Proc. 9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles at the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–6.
- Ren, S., He, K., Girshick, R. and Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6), pp. 1137–1149.
- Rothermel, M., Wenzel, K., Fritsch, D. and Haala, N., 2012. SURE: Photogrammetric surface reconstruction from imagery. In: *Proc. LC3D Workshop*, pp. 1–9.
- Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S. and Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3, pp. 293–298.
- Ruf, B. and Schuchert, T., 2016. Towards real-time change detection in videos based on existing 3d models. In: *Proc. SPIE 10004, Image and Signal Processing for Remote Sensing XXII*, pp. 100041H:1–14.
- Ruf, B., Erdnuess, B. and Weinmann, M., 2017. Determining plane-sweep sampling points in image space using the cross-ratio for image-based depth estimation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W6, pp. 293–322.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115(3), pp. 211–252.
- Schuffert, S., Voegtle, T., Tate, N. and Ramirez, A., 2015. Quality assessment of roof planes extracted from height data for solar energy systems by the EAGLE platform. *Remote Sensing* 7(12), pp. 17016–17034.
- Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sommer, L. W., Schuchert, T. and Beyerer, J., 2016. A comprehensive study on object proposals methods for vehicle detection in aerial images. In: *Proc. 9th IAPR Workshop on Pattern Recognition in Remote Sensing*, pp. 1–6.
- Stühmer, J., Gumhold, S. and Cremers, D., 2012. Parallel generalized thresholding scheme for live dense geometry from a handheld camera. In: *Trends and Topics in Computer Vision*, Springer, pp. 450–462.
- Taneja, A., Ballan, L. and Pollefeys, M., 2015. Geometric change detection in urban environments using images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(11), pp. 2193–2206.
- Uijlings, J. R. R., Van De Sande, K. E. A., Gevers, T. and Smeulders, A. W. M., 2013. Selective search for object recognition. *International Journal of Computer Vision* 104(2), pp. 154–171.
- Von Gioi, R. G., Jakubowicz, J., Morel, J.-M. and Randall, G., 2012. LSD: a Line Segment Detector. *Image Processing On Line* 2, pp. 35–55.
- Wu, C., Agarwal, S., Curless, B. and Seitz, S. M., 2011. Multicore bundle adjustment. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3057–3064.
- Zabih, R. and Woodfill, J., 1994. Non-parametric local transforms for computing visual correspondence. In: *Proc. European Conference on Computer Vision*, Springer, pp. 151–158.
- Zitnick, C. L. and Dollár, P., 2014. Edge boxes: locating object proposals from edges. In: *Proc. 13th European Conference on Computer Vision*, Vol. 5, pp. 391–405.