Window Detection in Facades for Aerial Texture Files of 3D CityGML Models

Franziska Lippoldt
franziska.lippoldt@fraunhofer.sg flippoldt@ntu.edu.sg

Abstract

The author inspects the optimal way to extract geometric facade features of windows from aerial texture files of CityGML models. The following method can be integrated and used for aerial texture modifications or 3D modeling details of 3D CityGML models. The author uses the Mask R-CNN with different configurations and backbone graphs to be tested on two data sets. As to improve the scores on the data sets, two traditional solutions to adjust the results are used: The author tests to integrate the more traditional approach of dbscan clustering to correct the results. Further the author also uses the texture coordinates available from the 3D CityGML file to correct our predictions.
As those 3D model textures origin from aerial photos, but are essentially smaller crops of a bigger image, facing typical challenges associated with low-level vision problems and bad image resolution and quality. This application can detect windows and facades from the Berlin CityGML model, extract the windows and doors and adjust the 3D model to integrate those. In addition, it is possible to replace the original windows and doors and insert black counterparts or standard models. The latter procedure will play a crucial role in privacy, as those elements might reveal private objects or persons next to the windows and can be automatically replaced.

1. Introduction

A complete list of 26 international cities has been publishing their city 3D models online to be available for public, most of them available in CityGML format as established in [7]. While several of those also contain textures, the 3D city models have a simplified shape: in CityGML terms they are at LOD2, providing the basic shape of the build but omitting facade details.
The idea of window detection and facade segmentation originates from the idea of automatically integrating the 3D features into the 3D models. In order to do so, it is necessary to analyze the given texture images through image segmentation.
Due to the strong correspondence of the texture files of the Berlin 3D model of [4] with satellite images, the results of this paper can also be applied to satellite or drone captures. This is especially interesting since satellite images are provided in super resolution and need to be cropped into suitable parts. In contrast to those "selected crops", the crops of the CityGML model are determined in advance and cannot be rearranged to a whole image.
In order to segment the facade and analyze the windows, the author proposes to use a semantic neural network. The network's main components are a masking part of the facade and a facade segmentation. While this structure has been deployed in the Mask R-CNN [8] originally to detected humans and deliver fast region proposal for semantic segmentation, the author transfers the usage to a different area of application. The original "mask" was deployed on the image to suggest regions faster and more accurate object detection, here the "mask" is used to build a facade aware segmentation to find windows.

2. Related Work

Since 2015, semantic image segmentation has been tackled as one of the main parts of neural networks. Segnet [2] was one of the first to tackle the problem of semantic segmentation of the whole image. Up to this year, several implementations for image segmentation exist. Recent challenges of this year deal with the semantic segmentation of satellite data. The winning networks of the satellite changes were mostly combined neural networks, which assign different task to different neural networks in pipelines. In general, image segmentation tasks can be more accurate when separately training networks for the objects and then combining those networks together, see [10].
As an applied example of semantic segmentation, facade segmentation has been studied in several works. Facade segmentation from street view style photos has been reduced to the task of finding repetitive objects or grid structures on the facade, depending on the architectural style. Major works were done by [13] and [11].
More traditional approaches that feature regularity constrains on traditional image recognition algorithsm for facade detection have been evaluted by [12] and [15]. Both of those works are provided however are provided with more

Figure 1. Three sample aerial images from the 256 Data set in original quality slightly shrunken

information than pure texture images, the first with complete satelite images and the latter with additional street views of the same object.

More recently this year, the DeepGlobe challenge [3], a set of three contests for satellite data object recognition, has been started. Whereas segmentation methods such as the SegNet proposed a single network with one training data set, those recent DeepGlobe solutions offer higher flexibility and directed training towards weak points of the object recognition.

The current state-of-the-art neural network for semantic segmentation for detection of the exact shape of humans is the Mask R-CNN, which shows both improvements in speed as well as a very good accuracy. Its main components are a Region Proposal Network (RPN) for recommending appropriate regions in the image and the main detection and segmentation network, which segments the image into objects pixel-wise. It is the extension of Fast R-CNN [6] and Faster R-CNN [14], which have been focusing on region detection enhancement.

On the other hand, there has been a recent work on integrations of 3D CityGML modeling and machine learning, most of all recently a paper on 'A Data-driven Approach for Adding Facade Details to Textured LoD2 CityGML Models' [16]. However, unlike this work, the author proposes to use the original texture image from the CityGML file, instead of rendered building facades, that are orthogonal to the facade. Using original texture images is a challenge, as the perspective distortion varies within each image and the texture may as well contain a lot of undesired information, such as neighboring houses, trees, roof tops, that will lower the prediction quality.

3. Texture image analysis

In this section, the author evaluates the image quality and properties as far as possible through data. A subsection of the Berlin CityGML data set with 203211 texture images has been selected. The author's main motivation for restricting our work to one district was to select buildings

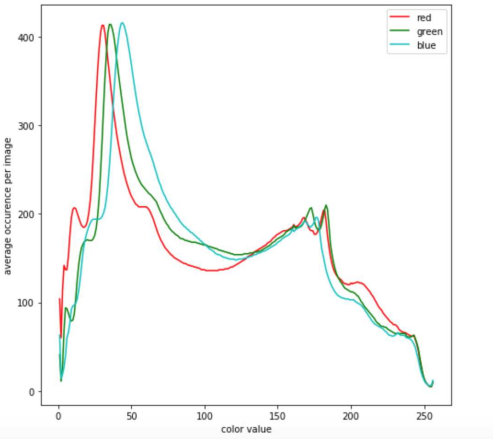

Figure 2. Average color values for RGB channels on selected region

that are not extravagant, i.e. monuments or buildings with specific architecture. Of those texture images, the average width and height per texture are 163px and 181px respectively. The average (rounded) RGB color value is 160 160 160.The average histogram over all of those texture images can be seen in figure 2. There is a slight valley in around 90, and a strong bias towards the left side of the graph, i.e. the black values. In order to check for color and lightening settings, the average HSV histogram has been plotted in figure 3. While the value channel is as indicated before relatively equal with a peak around value 90, the hue and saturation channels show high fluctuations. The saturation curve is strongly shifted towards the left side, indicating weak saturation.

The texture image files show the following variations:

1. Varying exposure settings (Figure 3)
2. Perspective distortion
3. No fixed orientation(i.e. the roof top of the building can be on any side of the aerial image)

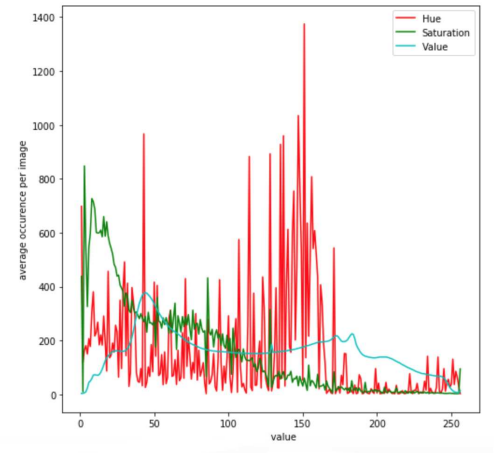

Figure 3. Average values for HSV channels on selected region

While technically perspective distortion can be calculated and inverted, it is non trivial to do so in combination with varying exposure and no fixed orientation. In Figure 1, a subset of windows is displayed. For highly underexposed or overexposed images, the border of the windows is lost. All together, those issues pose already significant problems for human eyes to spot the windows. The analysis of possible window candidates can only be confirmed by detecting features on the image, i.e. rooftop tiles or facades. While facades are basically a grid of windows, windows are part of the facade. While it is clear that the set of all windows of a house define the facade and the facade contains the set of all windows, the given input images will leave neither one or the either completely resolved.

4. Method

In this section, the author states initial assumptions and challenges on the aerial image data set and the detection of windows and doors and will explain methods to find a solution to those challenges. The author will discuss methods to check for the following assumptions:

A1 The number of possible features to be retrieved from the data set is restricted, hence the depth of the neural network is not required to be very deep

A2 The windows on the facade of those areal textures can be described as a cluster

A3 The lack of features is a cause for a low recall value

Each of the following subsection will try to to evaluate one of those assumptions and try to find the reasoning behind it with statistical experiments. The author is aware that, as much as neural networks in general only provide statistical experiments or scores to ”prove” claims, our argumentation is valid at first for this specific data set and the aerial images as described in the pages before. Whether or not our claims can be generalized to all aerial images needs to be proven, even though there is a strong indication that this could be the case.

4.1. Data sets and format

In the process of optimizing the trained network, the training set has been constantly adjusted and modified. Two different data sets of images of fixed size have been labeled and been used for experimenting with the neural network. While the texture image size is between 100px and 300px, the author chose to select images which can be cropped to size 128 and 256 respectively. (Sizes are in terms of the power of two as the convolutional blocks halve the size of the input and double the features.)

Data set 128 The first data set contains crops of the texture images of size 128x128. The author has labeled the whole texture file and then cropped them adaptively such that each cropped image has a maximal overlay of 10% with the previous one. Those results have been normalized with respect to the histogram. Further augmentation has been done in terms of 90 degrotations without loss of quality. Further, the images have been randomly shuffled and divided into the dataset for training, validation and testing with ratio 6:2:2. Overall this data set contains around 6,000 images.

Data set 256 The second data set contains adaptive crops of hand-labeled images of size 256x256. The authors have selected the best texture files from a set of more than 1,000 images in terms of resolution, image dimensions and exposure settings. Most of those chosen images are not much larger in terms of side length than 300px. While the same adaptive cropping and augmentation as in the 128 data set has been used, the resulting training and evaluation set is slightly smaller in terms of the total number of images. Accordingly, the augmented images were divided randomly into training, validation and test sets the same way as the other data set.
The data set contains 1,000 images.

4.2. Detection process

In order to detect windows from the labeled data sets, the author chose to use masked detections in contrast to more general box detection outputs, for the following reasons: For one, even though windows typically do have a box shaped outline, the perspective distortions are relatively high, i.e. the area of the bounding box often appears one third bigger than the area of the window element itself. Therefor mask detection can be much more accurate. For the other, masks are useful for future enhancements, mask detections add new possibilities to the interpretation of output and adjustment of aerial textures, as in the case of covering objects in the window for privacy reasons for example.
The author first started with using the standard Mask R-

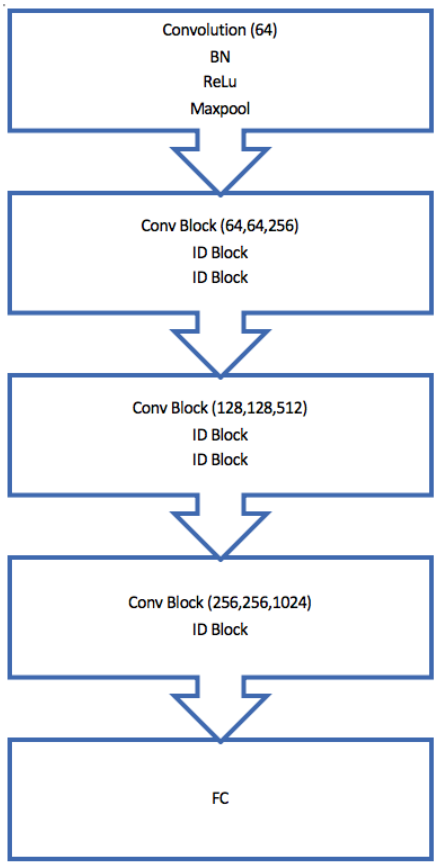

Figure 4. Modified backbone structure, originating from Resnet101, seperated into stages one to four, but with a cut off fourth stage and the fully connected layers in the end

CNN [8] in combination with the Resnet101 backbone [9] and the standard Imagenet weights. Then configuration settings have been adjusted piece by piece: those include the region proposal parameters, the anchor parameters, the detection threshold, the loss weights and the region proposal depth. Later on the author changed even the basic structure of the network graph and the depth of the network, changing from the typical deep Resnet101 network to a more shallow structure. The main motivation for using the Resnet structure from the start was the basic structure of the identity modules, that will prevent deeper layers from collapsing to zero. That is to say, even though not expecting the best possible outcome, the high depth of the network would not, at least, worsen the prediction results.

Due to our initial assumption that there is a restricted number of possibly recognizable features, the author tested to trim the Resnet101 backbone significantly. More than 22 identity blocks have been removed, but the first layers and their structure has been preserved, modifying the graph from stage 4 on, see figure 4.

4.3. Facade elements as clusters

As first results show that the trained network still has a high imbalance between precision and recall, the author checked possible error sources through threshold changes and considered traditional clustering as an approach to filter predictions from the output of the network.

As proposed in various papers, regularity constrains can be used to filter facade elements. Our approach is to test regularity assumptions that were made in the work of Wolff [15] for our data. In this approach, the facade regularity is detected by comparing aerial and street view image of the same building. As this data set consists of aerial images containing only partial facades,the regularity of the window facade is not as strong as used by Wolff, but the author decides to test the window detection on a cluster filter. The clustering is based on the assumption that common facade architectures are regular and in fact windows often lie on a regular grid structure. This test checks whether or not the perspective distortions still allow a minimal improvement through clustering. As this work is on aerial texture images from a huge over all data set, it is not suitable to check for image dependent (user input requiring) approaches but to look for global parameters to adjust the overall results.
Evaluating the correctness of Assumption two, namely that a facade can be expressed as a cluster, the author is testing cluster methods to detect outliers and wrong detections. There is a subset of aerial texture images that contains both a roof and a facade. This might be either the roof from the neighboring house that is covering the facade partially or the roof of the same house. However, inspection of the detections reveals that most of the wrong detections origin in window detection on windows and chimneys on the roof top. When the author initially labeled the data set, the roof top windows appeared to be of slightly different shape and successful detection was not expected.
That is why our data sets do not contain roof windows at this point in time, nor chimneys.
Seeing that a facade consists of several windows, with a certain regularity in their distance to each other and side length, clustering seemed a natural approach to try.
The author has used the DBSCAN algorithm as proposed by [5] provided through the python package scikit learn. It's crucial parameters include the minimal number of points per cluster and the maximal distance points can be apart to be still considered in one cluster (epsilon distance). Different configurations were used on either one,two or three points per window. The one point configuration uses the center point, the two point configuration uses opposite edges and the three point configuration uses both center and opposite attaches for the cluster criteria.
The other close at hand approach to improve the output was to consider the facade labeling as provided through the code of the CityGML file and is described in section 4.4.

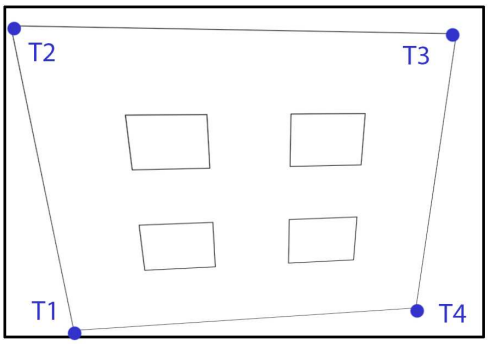

Figure 5. Illustration of an aerial texture image (bold border) with texture coordinates labeling a facade (thin border)

4.4. Texture coordinate retrieval as a facade label

While the inspection results show several wrong detections of windows on roof tops, chimneys and other rectangular elements, restricting the prediction results through data available through the CityGML file was an intuitive step to take. Technically, one building consists of several surfaces. A roof surface, a wall surface and a ground surface. Each surface is described by a sequence of points in 3D space. The ordering of those points is crucial to the orientation of the surface. Assuming ordinary polygonal surfaces, a surface is closed if the last and the first point of this sequence are equal. Textures in CityGML are defined on so called Ring Surfaces, i.e. closed surfaces with a certain degree of topological simplicity, i.e. orientable surfaces with measurable surface area. For an image with width w and height h, can define texture coordinates as 2D points, where each coordinate is in the range of 0 and 1. For a Ring Surface with a sequence $P_1, P_2, P_3, \cdots, P_n$ of n points and texture with texture coordinates $T_1, T_2, T_3, \cdots, T_n$ the texture mapping projects the texture onto the object by projecting edge points of the image to the edge points of the surface and the interior accordingly. An illustration of a texture coordinate labeling can be seen in Figure 5.
The detections masked with the polygon mask obtained from the points $T_1, T_2, T_3, \cdots, T_n$ through point-wise multiplication of the texture coordinate mask.

5. Results

For the assumptions made in Section 4, this section will provide the test results to verify or disprove those statements.

A1 In Section 5.3, the author has tested a much shallower version of a backbone. Despite instability during the training processed, the results achieved an average precision only three percent lower than with the original backbone with 101 layers. This is a strong argument for the fact that the depth as provided by the "standard backbone" is not required.

A2 In Section 5.4, the possibility of using clustering to improve the prediction outcome was discussed. The author tested various parameters on the test data set, but are unable to find one set of parameters that matches all aerial texture images. As the distance from the camera to the buildings itself is not significantly different per building, the perspective needs to be incorporated as a parameter for successful clustering.

A3 While the assumption on the restricted number of features is hard to prove or disprove, the author concludes that either using a completely different model or tusing another training data set is the only solution to possibly further improve the prediction results. The author has tested different parameters, different configurations (Section 5.1), different backbone structures (Section 5.3), output clustering (Section 5.4) and texture coordinates output filtering (Section 5.6) but still failed to significantly improve the recall value.

The author used slightly different hardware and software for the section 5.1 (GTX 1070s and Ubuntu 16) and 5.4 / 5.6 (GTX 1080s and Ubuntu 18), which explains the slight decrease in score from section 5.1.

5.1. Training and accuracy scores

The author originially started with training the Mask R-CNN [1] on the pre-trained Resnet101 backbone [9] modifying parameters such as anchors, region proposals and more parameters, having run more than 30 configurations of the neural network. Out of those, the author selected the best five parameter configurations for each data set. The following results are discussed on the training of two epochs and fine tuning of 3 epochs, where one epoch contains all training set data and validation steps are set to the number of validation images.

Configurations and Test	Recall	Precision	$AP_{@IoU0.5}$
A. 128 - standard	0.53	0.85	0.85
B. 128 - optimised	0.60	0.82	0.87
C. 256 - standard	0.51	0.94	0.91
D. 256 - optimised	0.58	0.90	0.93

Table 1. Models and scores

In order to use the networks inference results for detecting windows and analysing the architecture of a facade, the main tasks lie in detecting every existing window as well as reducing the number of wrongly detected windows.
In other words, the recall is a crucial value to improve. As can be seen in Table 1, the optimisation of the AP goes hand

Data set	Δ Recall	ΔPrecision	Δ $AP_{@IoU50}$
A. 128	0.07	-0.02	0.02
B. 256	0.07	-0.04	0.02

Table 2. Overview of improvements made through configuration and parameter optimisation

in hand with the optimisation of the recall. However, the author has also realised that for any type of recall and AP improvement, the precision value has slightly decreased by at least 0.02, see Table 2. In none of the AP improved configurations has the author seen an improvement of the precision, however the recall has always been higher by at least 0.01 and at most 0.07. During the configuration testing, the author has tried to increase the precision by changing the loss function parameters such that the mask score is three times higher than the other scores. However, this approach did not improve the precision. Henceforth suggesting that the precision of the mask results depends strongly on the complete structure of the network and each layer and feature size. It is likely that the lack of information in the image in terms of image size and resolution leads to a restriction in the possible features and hence a threshold for the possible precision values.

While the results in table 1 show a high precision score, the recall value is low. As it is common to consider the detection threshold as the origin for a bad balance between precision and recall, testing on the score was also done on thresholds other than the used 0.7. For all the tests of threshold 0.6, 0.5 and 0.4, the recall has not been improved by more than one percent.

The author therefor argues that the bad recall and precision ratio arises from either the quality of the images (missing information in terms of low vision problems) or the mask detection model itself. The author therefor implemented a significantly shorter version of the network graph, dropping a lot of identity layers of the Resnet101 network in the second next section.

5.2. Anchors, ROIs and AP scores

Due to the small image and object size, the anchor size of region proposal in the neural network has been modified in most of the configurations. The author has chosen significantly smaller anchor scales and different anchor ratios.

Another crucial parameter to the improvement of the $AP_{@IoU50}$ score is the number of trained regions per image. The author found out that adjusting the number of trained regions per image to reflect the number of windows per image improves the recall values. If chosen improperly, several windows are either not detected at all or windows are double detected, i.e. several regions span over one window and classify it as three objects. An example of missing windows is shown in Figure 6.

Regarding the Region Proposal Network (RPN), the number of regions for training showed a correlation to the recall value and the under and over fit of the network with respect to a single image evaluation. In our test case, the missing windows often occur towards the center of the image and the dimension of the window as were not indifferent from other detected windows.

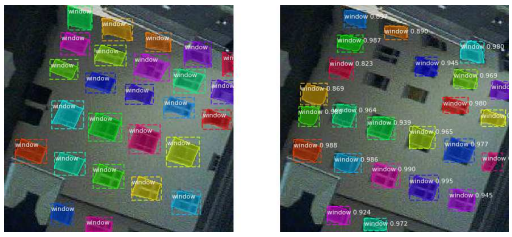

Figure 6. Window labeling (left) and detection results (right) for wrong parameter selection in terms of region proposal threshold filtering, leading to "missing windows"

5.3. Network depth and precision

Data set	Recall	Precision	$AP_{@IoU50}$
256	0.5278	0.9135	0.8927

Table 3. Scores for the modified graph of Figure 4

The modified graph was tested on the best configuration of the previous result, see Table 3. While the author also planned on translating all the results on the "normal backbone" to the cut off version. However, he training process was found to be unstable and required to minimize the size of training steps in order to save the weights in between.

The author tried different configurations, however found the specific best configuration 256 too unstable to complete successful training. Overall, the best network and configuration achieved an average precision score of 89%. Given the fact that more than two third of all identity blocks from the network were removed, only around three percent in average precision $AP_{@IoU50}$ were lost.

5.4. Clustering Improvements

The author expected two and three point configurations to be more effective in clustering, as otherwise there will be no awareness w.r.t. the window area: small windows far apart from each other and large windows with center points far away but actually close will both be excluded/included in the same manner. However, this was not the case. As shown in Figures 7 and 8, the graphs for the two and three point approach per window are nearly identical in terms of AP score and maximal distance epsilon.

Without doubt one could tune the clustering parameters on each image separately, but all test parameters were performed on the whole set of test images at once. Our results

Clustering set	Recall	Precision	$AP_{@IoU50}$
No Clustering	0.5418	0.9181	0.9175
Center point	0.5358	0.9237	0.9186
Opposite edges	0.5358	0.9237	0.9185
Three point	0.5357	0.9191	0.9237

Table 4. Overview of improvements made through dbscan clustering

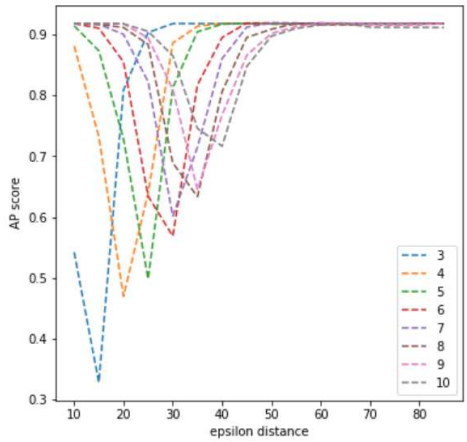

Figure 7. AP score variation with respect to the maximal cluster element distance epsilon (in px) for the two point model, each graph for a cluster with minimum of 3 to 10 window elements according to the label

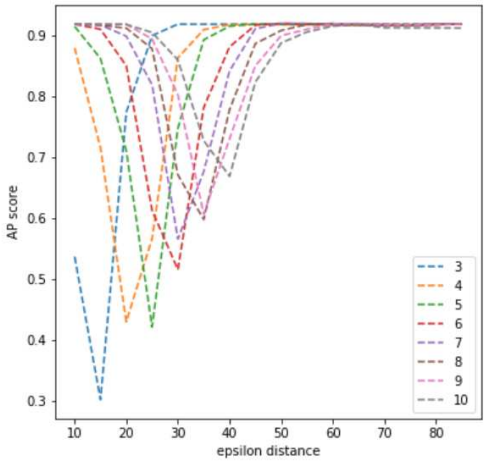

Figure 8. AP score variation with respect to the maximal cluster element distance epsilon (in px) for the three point model, each graph for a cluster with minimum of 3 to 10 window elements according to the label

as seen in table 4 show, that for neither configuration a remarkable improvement was achieved.
The author observed that while improving and removing unnecessary predictions for on image, the same configuration will remove actual windows in other test images. The overall best AP improvement has been made in the configuration of three points, however with a total improvement of less than one percent, the author suggests that this improvement does not proof the general assumption that there is a common clustering methods for all the aerial texture images.

Figure 9. Floor estimation through PCA example; selection of five windows on one line, that lead to a prediction of five floors

5.5. Application: Floor estimation

Using PCA component analysis over the resulting labels, it is possible to determine the maximal number of floors (visible) on the facade. Even though having a quadratic runtime, this is an interesting feature to extend or adjust 3D models level wise.
Given the set of center's c of predictions, look for the line that contains the largest amount of center points with maximal error ϵ. Assuming a correct prediction, this will lead to the maximal number of windows per line, which would be, for a regular facade, either the number of windows per floor or the number of levels, whichever is larger.
Given that it is possible to estimate the facade orientation and the top of the facade, one can constrain the lines on the set of all lines to those that lie between two angles α_1, α_2, where those two angles are the angles of the rectangle of uv coordinates that intersect with the top line.
An application of those estimations is shown in Figure 9.
The correctness of the prediction of the number of levels depends on the correctness of the prediction itself. Under the assumption that there is one or two wrong predictions in the image,it is possible to estimate the number of floors by the largest number of points per line that occurs at least in three lines (instead of one line). However, the estimation of number of floors purely visible on the photo is non trivial, as buildings are covering each other and a lot of those do not allow to see the first floor.

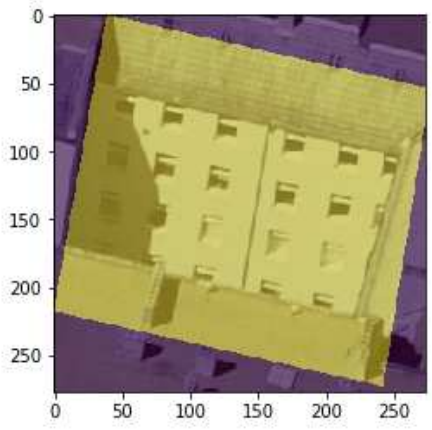

Figure 10. Overlay of texture mapping on a facade texture with roof parts covering the bottom

Figure 11. Detection results before (top) and after applying texture masks (bottom)

Configuration	Recall	Precision	$AP_{@IoU50}$
256 standard	0.5418	0.9181	0.9175
256 w/ texture mask	0.5354	0.9261	0.9293

Table 5. Overview of improvements made through texture masks

5.6. Texture coordinates for enhanced prediction

While expecting a significant improvement of the recall values by using the texture coordinates as masks on the results, the author did not encounter any significant change in the scores, as seen in Table 5.
Looking at sample images, one can see that some facades have a slightly better prediction and the roof windows are left out, see Figure 11, but this is not the case for all of the images. In fact, several facades, that are either covered by trees or other buildings, still contain the building or tree in the texture mask area, see Figure 10, where he bottom part of the building is covered by a roof and that roof is part of the mask.

6. Future Work

Neural Network adjustments As the author has tested results of a significantly shorter backbone and hence an easier to train network and those first results yielded a mere three percent difference in mean average precision and would like to encourage further studies on determining optimal depth and structure of networks for accurate yet minimal graph structure.

Facade element extension There is a huge selection of other objects / facade features that can be extracted from those texture images. This especially concerns objects on the roof. Given that our solution does detect windows on roof texture images, the author can imagine that combining window recognition and chimney recognition will provide optimal outcome for enhancing 3D models.

Application for privacy preserving images This model can be easily applied to automatically detect and replace windows with standard photos, such that the details of curtains, the objects on the window sill and other private property can be automatically removed. For some glass facades, that the interior of those has already been replaced manually and the author would like to propose to test automatic privacy protection.

Detailed 3D modeling from aerial images While most recent work focuses on either constructing basic housing models from aerial images, there is a huge opportunity in going from aerial images over simple models to complex building models. As such, the author's work can be integrated into simple building models.

7. Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.
The author thanks Michael Kasper, Marius Erdt and Henry Johan for their support. The author would also like to thank Rolf Versluis for the additional hardware support, as well as Ariya Priyasantha for his technical guidance.

References

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 5

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 1

[3] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[4] Jürgen Döllner, Thomas H Kolbe, Falko Liecke, Takis Sgouros, and Karin Teichmann. The virtual 3d city model of berlin-managing, integrating, and communicating complex urban information. In *Proceedings of the 25th international symposium on urban data management UDMS 2006 in Aalborg, Denmark, 15-17 May 2006*, 2006. 1

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 4

[6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[7] Gerhard Gröger and Lutz Plümer. Citygml–interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 2012. 1

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 4

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[10] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016. 1

[11] Hantang Liu, Jialiang Zhang, Jianke Zhu, and Steven CH Hoi. Deepfacade: A deep learning approach to facade parsing. 2017. 1

[12] Jingchen Liu and Yanxi Liu. Local regularity-driven city-scale facade detection from aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3778–3785, 2014. 1

[13] Markus Mathias, Andelo Martinović, and Luc Van Gool. Atlas: A three-layered approach to facade parsing. *International Journal of Computer Vision*, 118(1):22–48, 2016. 1

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[15] Mark Wolff, Robert T Collins, and Yanxi Liu. Regularity-driven facade matching between aerial and street views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2016. 1, 4

[16] Xingzi Zhang, Franziska Lippoldt, Kan Chen, Henry Johan, and Marius Erdt. A data-driven approach for adding facade details to textured lod2 citygml models. pages 294–301, 01 2019. 2