BUILDING ROOF VECTORIZATION WITH PPGNET

Simon Hensel[1]*, Steffen Goebbels[1], Martin Kada[2]

[1] Niederrhein University of Applied Sciences, Institute for Pattern Recognition, Krefeld, Germany
(simon.hensel, steffen.goebbels)@hs-niederrhein.de
[2] Technische Universität Berlin, Institute of Geodesy and Geoinformation Science Berlin, Germany
martin.kada@tu-berlin.de

KEY WORDS: Roof Reconstruction, Vectorization, Deep Learning, PPGNet, Orthoimages.

ABSTRACT:

A challenge in data-based 3D building reconstruction is to find the exact edges of roof facet polygons. Although these edges are visible in orthoimages, convolution-based edge detectors also find many other edges due to shadows and textures. In this feasibility study, we apply machine learning to solve this problem. Recently, neural networks have been introduced that not only detect edges in images, but also assemble the edges into a graph. When applied to roof reconstruction, the vertices of the dual graph represent the roof facets. In this study, we apply the Point-Pair Graph Network (PPGNet) to orthoimages of buildings and evaluate the quality of the detected edge graphs. The initial results are promising, and adjusting the training parameters further improved the results. However, in some cases, additional work, such as post-processing, is required to reliably find all vertices.

1. INTRODUCTION

Typically, 3D city models have been reconstructed from airborne laser scanning point clouds. Since most points belong to roof facets, the algorithms generate a roof topology and then add walls as simple planar surfaces that connect the roof with the ground. Two different approaches are often combined to find a graph that represents vertices and edges of roof polygons, see (Tarsha-Kurdi et al., 2007). In a model-based approach, standard parameterized roof tops are fitted to often rectangular segments of the cadastral footprint by optimizing their parameters. In a data-based approach, geometric primitives are detected in the point cloud (for example with a RANSAC algorithm that considers point normals) and are then combined. An at least partially data-based approach is necessary to model small structures like dormers and non-standard roofs, for example of churches. The biggest challenge of a data-based approach is to find bounding polygons of detected roof facets. Whereas ridge lines can be computed rather precisely by intersecting planes, step edges are difficult to model. Step edges are jump discontinuities that occur at the sides of dormers and chimneys but also with building parts of different height. For example, the footprint of a penthouse consists of step edges. Straight lines running through step edges can be estimated only with low precision on sparse point clouds with classical algorithms like RANSAC or the Hough transform. The edges are also visible in orthoimages. But convolution-based edge detectors find many wrong line segments due to shadows and textures. For this reason, we investigate whether the graph consisting of corners (denoted as junctions) and roof edges can in principle be better recognized by machine learning. Since recently, several deep neural network architectures have been available for edge graph detection on images. The given paper is a feasibility study in which the deep neural network Point-Pair Graph Network (PPGNet) (Zhang et al., 2019) is trained on hand-labeled orthoimages. The goal is to show that the quality of the recognized graph is sufficient to be used in a roof reconstruction pipeline. To this end, we focus on rural areas with clearly separated building roofs.

Deep learning is a powerful tool in applications in which it is difficult to find processing rules. Such a task is to distinguish between edges of roof facets and edges originating from shadows or textures.

The next section deals with current neural network-based approaches to 3D city modeling. Then, a brief overview of the PPGNet architecture is given in Section 3. Section 4 briefly discusses the training and test data, followed by a description of our results, which show that the method is feasible in principle.

2. RELATED WORK

An alternative network architecture to the PPGNet is the convolutional message passing network (Conv-MPN) (Zhang et al., 2020), which uses a message passing network for generating an adjacency matrix for a previously computed set of junctions (vertices of the graph). Training the Conv-MPN is a two-step process. The first step is to train a network to detect junctions and then, in the second step, the results of the first step are used to generate the adjacency matrix. To the contrary, the PPGNet is trained end-to-end. This is the reason why we chose the PPGNet for this feasibility study.

Classification of roofs is another domain in which neural networks are used. For example, in (Castagno and Atkins, 2018) an approach is proposed that fuses image and LiDAR data and uses a convolutional neural network (CNN) in combination with a random forest classifier to label roofs by roof class. Another example for roof classification in satellite images is described in (Partovi et al., 2017). Two strategies are presented here, one is classification using VGGNet and the other is a combination of CNN features and a support vector classifier.

There are many approaches to vectorizing building outlines in aerial imagery, including the application of deep neural networks to footprint segmentation. However, segmented foot-

* Corresponding author

prints often need to be vectorized using classical contour detection and simplification algorithms. An example is given in (Cheng et al., 2019), where a Deep Active Ray Network (DARNet) is proposed for the segmentation of building footprints. DARNet is an evolution of the active contour approach. In this algorithm, an initial polygon-based contour is placed in the center of the building in the image. Then, in an iterative process, an energy function is maximized using a CNN-generated feature map. Recently, PolygonCNN, an end-to-end trained neural network for shape modeling, was presented, see (Chen et al., 2020). It uses a modified PointNet to optimize the shapes of the detected footprint contours and outputs a ready-to-use vectorization. However, unlike the PPGNet, it cannot handle individual roof facets.

Based on a digital surface model (DSM) and a panchromatic satellite image, the algorithm in (Wang et al., 2021) reconstructs LoD 2 building models. To do this, an improved conditional Generative Adversarial Network (cGAN) is used to refine the DSM in order to filter out non-building objects and sharpen building shapes. Then, a semantic segmentation network similar to the generator part of the cGAN detects building corners and edges. It labels pixels. These data are used in a subsequent, separate pipeline to reconstruct a roof graph and generate a 3D model. However, the pipeline is not built on machine learning.

Deep neural networks have been also applied in model based roof reconstruction, see (Partovi et al., 2019). The described workflow applies a fine-tuned pre-trained ResNet-152 to gain roof-type classifications, which are later used as parameters for a model-based reconstruction.

Within the algorithm in (Alidoost et al., 2020), a Y-shaped CNN is applied to detect eave, ridge, and hip lines in a single aerial image. In a post-processing step outside the neural network, eave lines are used to subdivide building footprints into individual roof areas. Then ridge lines yield parametric models of these areas for which height information are estimated.

Neural networks can be applied even to point clouds, which are difficult to handle due to their unstructured nature. Studies such as (Zhang and Zhang, 2018) used a combination of multiple networks in a larger framework that includes a 3D CNN and a recurrent neural network (RNN). Another example can be found in (Pohle-Fröhlich. et al., 2019), where the PointNet++ (Qi et al., 2017) was used to segment point clouds of roofs and to determine gradient directions.

Other applications of neural networks with regard to city models are more domain specific, such as detecting suitable roof surfaces for solar panel installations (House et al., 2018), determining roof damage (Hezaveh et al., 2017), or disaster mitigation for earthquakes (Li et al., 2015).

3. PPGNET

In this chapter, we give a brief description of the PPGNet. For details of the PPGNet see (Zhang et al., 2019). The PPGNet is a CNN that takes an image as input and outputs detected line segments (edges) on that image as a graph. The neural network can be divided into three main parts,

1. Junction Detection Module (JDM):
   The module generates a list of detected junctions using a feature map generated by a neural network backbone. ResNet or UNet are supported by default, but can be replaced by other network architectures. The final output of the JDM is a heat map. Local maxima of this map define the junctions.

2. Line Segment Alignment Module (LSAM):
   The module returns line segment candidates, i.e., edge candidates, given by the two endpoints and a feature vector. The feature vector results from the feature map values between the two endpoints.

3. Adjacency Matrix Inference Module (AMIM):
   The module takes line segment candidates with their feature vectors and transforms them into an adjacency matrix. To this end, a binary classification problem is solved with additional network layers. The derived adjacency matrix contains the scores of the given pairs of junctions, which indicate the determined probability that a pair of junctions is an edge.

All three parts are trained simultaneously.

4. TRAINING AND TEST DATA

Training and test data were created from orthoimages of the city of Detmold. These images are freely available from Geobasis NRW[1]. Each image covers an area of one km² with a resolution of 10000×10000 pixels. This results in an accuracy of 10 cm² per pixel, which is well suited for recognition tasks on roof images. To create a ground truth, these images had to be annotated with a graph whose edges are line segments of roof structures. For this purpose, the software QGIS (QGIS Development Team, 2009) was used, which supports the format multilinestring. The annotations cover only boundary edges of roof facets at and inside the building footprint. The orthoimages were too large to become PPGNet input. Therefore, we cut out smaller images of individual buildings so that there was a distance of at least five pixels between the building footprints and the borders of the images. Thus, for each building, both an image file and an edge graph file (ground truth) were created. This building data was split into a training and a test data set. The size of the training data was artificially increased by color changes and image mirroring so that 7501 training samples were generated, 139 samples were put aside and left unmodified for testing[2].

5. EXPERIMENTS AND DISCUSSION

As recommended by the original developers, we used a pretrained backbone for transfer learning. Following the default setup, 30 epochs of training were divided into three phases. While the weight decay parameter was set uniformly to 0.0005 for all phases, the learning rate and the composition of the loss function were defined differently in each phase. The learning rates of the three phases were 0.2, 0.02 and 0.002, respectively.

The total loss function is a linear combination of the loss function for JDM weighted by the factor λ_{junc}, and the loss function for LSAM and AMIM weighted by the factor λ_{adj}. These factors were found to have a large impact on the detection results. Therefore, we varied these factors while keeping the other parameters constant, see Table 1.

[1] GEOportal.NRW `www.tim-online.nrw.de/tim-online2`, accessed: 29th August 2021

[2] The data set is openly available at `www.github.com/SimonHensel/Vectorization-Roof-Data-Set`.

Figure 1. Results of case E6, see Table 1. The lines are colored according to the confidence value determined.

Experiment	Phase 1		Phase 2		Phase 3	
	λ_{junc}	λ_{adj}	λ_{junc}	λ_{adj}	λ_{junc}	λ_{adj}
original	1.0	1.0	1.0	1.0	1.0	1.0
E1	5.0	1.0	10.0	1.0	10.0	1.0
E2	5.0	1.0	10.0	1.0	1.0	10.0
E3	5.0	1.0	20.0	1.0	20.0	1.0
E4	5.0	1.0	15.0	1.0	15.0	1.0
E5	6.0	1.0	12.0	1.0	12.0	1.0
E6	8.0	1.0	17.0	1.0	17.0	1.0

Table 1. Original (Zhang et al., 2019) and tested loss-function factors used in numbered experiments. Cases $\lambda_{\text{junc}} < \lambda_{\text{adj}}$ resulted in zero precision and recall and thus are not listed.

The examples in (Zhang et al., 2019) consist of graphs with up to 75 times more junctions and edges than are present in our data. Moreover, the goal of this study was to detect every edge present in the image. For the reconstruction of roofs, however, the few edges of significant roof facet polygons must be distinguished from other edges belonging to shadows, for example. This greatly reduces the number of relevant junctions and edges to be detected. Therefore, the correct detection of each of these junctions is important to achieve sufficient detection quality. For example, if the center junction of a pyramid-shaped roof is not detected, only one roof facet can be found instead of four.

Experiment	AJP	AJR	AEP	AER
E1	0.93596	0.88832	0.88056	0.79090
E2	0.90513	0.72202	0.82674	0.55194
E3	0.93550	**0.92589**	0.88489	**0.85916**
E4	0.94793	0.87773	0.90820	0.78559
E5	0.92592	0.87689	0.87421	0.78049
E6	**0.95408**	0.89186	**0.91670**	0.81011

Table 2. Precision and recall of experiments with different loss factor setups, see Table 1. Cases E3 and E6 performed best, where we used a higher factor λ_{junc} in the training phases.

Experiment	Junction F1	Edge F1
E1	0.91151	0.83332
E2	0.80327	0.66195
E3	**0.93067**	**0.87183**
E4	0.91148	0.84245
E5	0.90073	0.82469
E6	0.92192	0.86011

Table 3. F1 score of experiments with different loss factor setups, see Table 1. Case E3 performed best, where we used a higher factor λ_{junc} in the training phases.

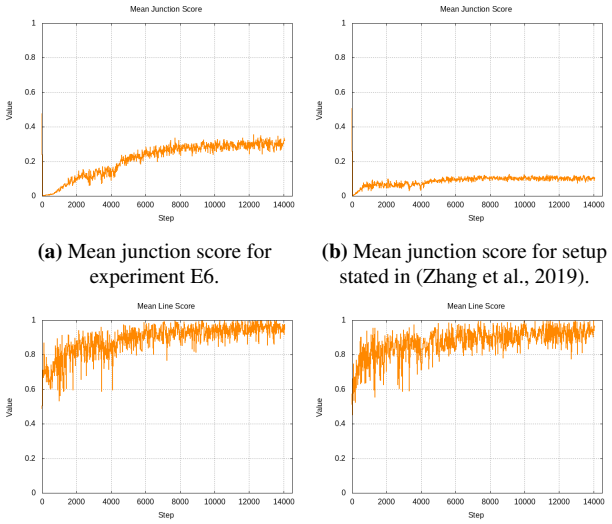
To evaluate our experiments, we calculated the average precision AJP and average recall AJR for junction detection, and the average precision AEP and average recall AER for the detection of edges represented by the adjacency matrix, see Table 2. As usual, precision and recall are defined by

$$\begin{aligned} \text{precision} &= \frac{\text{tp}}{\text{tp}+\text{fp}}, \\ \text{recall} &= \frac{\text{tp}}{\text{tp}+\text{fn}}, \end{aligned}$$

where tp, fp and fn are the true positives, false positives and false negatives, respectively. Detected junctions were classified as true positives if they were within 80 cm of a ground truth junction. Detected edges were compared with ground truth edges by matching their junctions with ground truth junctions. We computed the values for precision and recall for each individual building and then used the arithmetic mean over all buildings to obtain the values for AJP, AJR, AEP, and AER. As expected, precision and recall increased for both junctions and edges when missing junctions were avoided by choosing λ_{junc} greater than λ_{adj}. Experiments E3 with the highest recall and E6 with the highest precision were the most successful. This is also reflected in the F1 score, which combines precision and recall, see Table 3 :

$$\text{F1} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}.$$

In contrast, when we chose $\lambda_{\text{junc}} \leq \lambda_{\text{adj}}$, the network did not learn to detect junctions, so all average precision and recall values for junctions and edges became zero.



(a) Mean junction score for experiment E6.

(b) Mean junction score for setup stated in (Zhang et al., 2019).

(c) Mean line score for experiment E6.

(d) Mean line score for setup stated in (Zhang et al., 2019).

Figure 2. Graphs showing the mean line and junction score for case E6 (see Table 1) and the original setup during the training process. An exponential moving average with a smoothing factor of 0.9 was chosen to smooth the graph.

The importance of choosing λ_{junc} larger than λ_{adj} can also be seen in Figure 2. Here, experiment E6 is compared with the default setting $\lambda_{\text{junc}} = \lambda_{\text{adj}} = 1$ of (Zhang et al., 2019). The plots show mean junction and mean line scores for training iterations as implemented in the PPGNet code. The mean junction score is the arithmetic mean of the values of the JDM output junction heat map, restricted to elements belonging to ground truth junctions. The entries of the AMIM output adjacency matrix are confidence values for the existence of edges. An undirected edge is counted as one detected edge, if both directed opposite edges have a confidence value above 0.3. Of the confidence values of detected edges the arithmetic mean is calculated to obtain the mean line score. Note that the mean junction score is far below the mean line score. To some extent, this can be explained by the different definitions of the scores. However, we also suspect that the network does not obtain higher confidence scores for junctions because it must distinguish between the relevant roof junctions and other junction candidates.

In comparison, the mean junction score of E6 is up to three

times higher than with $\lambda_{\text{junc}} = \lambda_{\text{adj}} = 1$. It reaches values above the threshold $\tau = 0.3$. Thus in experiment E6, confidence values for junctions exceeded τ. This threshold value had to be reached in order to classify a junction candidate as a detected junction. With the original setup in (Zhang et al., 2019), all confidence values were significantly below τ on the test data such that precision and recall became zero.

6. CONCLUSION

Compared to other related approaches, the PPGNet creates a vectorization that is not limited to the footprint of the building. Advantages of the PPGNet are that it can be trained end-to-end and that it can, in principle, compute vectorizations without the need for additional post-processing algorithms.

First results are promising. The examples in Figure 1 show that the detected edges have sufficient accuracy if appropriate loss factors are chosen. We were able to achieve an average F1 score of 0.93 for junction detection and 0.87 for edge detection.

However, even when λ_{junc} is chosen significantly larger than λ_{adj}, there were still examples where an important junction could not be found. As a consequence, all adjacent edges were also missing. This is currently the main problem that could prevent a complete reconstruction of roofs based on the detected roof graphs.

The vectorization of a roof is a hit-or-miss situation. Either the vectorization is completely perfect or it is incomplete and thus not directly usable. But even at this stage, the network output helps to support the roof reconstruction process. For example, the framework in (Goebbels and Pohle-Fröhlich, 2016) snaps estimated roof edges to support lines, which can now be taken from the network output.

Future work should investigate how to make junction detection more reliable so that the average junction score increases. Improvements could also be made through post-processing. It would be interesting to integrate the PPGNet into a roof reconstruction workflow.

7. ACKNOWLEDGMENTS

The authors are grateful to Regina Pohle-Fröhlich, Kai Peter Kamphausen, Frederik Terstappen, and Lucas Weissbeck for sharing their experiences with PPGNet, and the authors also thank Aaron Heyde, Armin Kick, Aida Nejadsalim, Mahboba Rezai, Anastasia Schlegel, and Kriti Shrestha for manually creating ground truth data.

REFERENCES

Alidoost, F., Arefi, H., Hahn, M., 2020. Y-shaped Convolutional Neural Network for 3D Roof Elements Extraction to Reconstruct Building Models from a Single Aerial Image. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020, 321–328.

Castagno, J., Atkins, E., 2018. Roof Shape Classification from LiDAR and Satellite Image Data Fusion Using Supervised Learning. *Sensors*, 18(11), 1–23. https://www.mdpi.com/1424-8220/18/11/3960.

Chen, Q., Wang, L., Waslander, S. L., Liu, X., 2020. An End-to-End Shape Modeling Framework for Vectorized Building Outline Generation from Aerial Images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 170, 114–126.

Cheng, D., Liao, R., Fidler, S., Urtasun, R., 2019. DARNet: Deep active ray network for building segmentation. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7423–7431.

Goebbels, S., Pohle-Fröhlich, R., 2016. Roof Reconstruction From Airborne Laser Scanning Data Based on Image Processing Methods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, 407–414.

Hezaveh, M. M., Kanan, C., Salvaggio, C., 2017. Roof damage assessment using deep learning. *Proc. IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 6403–6408.

House, D., Lech, M., Stolar, M., 2018. Using deep learning to identify potential roof spaces for solar panels. *Proc. 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 1–6.

Li, S., Tang, H., He, S., Shu, Y., Mao, T., Li, J., Xu, Z., 2015. Unsupervised Detection of Earthquake-Triggered Roof-Holes from UAV Images Using Joint Color and Shape Features. *IEEE Geoscience and Remote Sensing Letters*, 12(9), 1823–1827.

Partovi, T., Fraundorfer, F., Azimi, S., Marmanis, D., Reinartz, P., 2017. Roof Type Selection based on Patch-Based Classification Using Deep Learning for High Resolution Satellite Imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-1/W1, 653–657.

Partovi, T., Fraundorfer, F., Bahmanyar, R., Huang, H., Reinartz, P., 2019. Automatic 3-D Building Model Reconstruction from Very High Resolution Stereo Satellite Imagery. *Remote Sensing*, 11, 1–37.

Pohle-Fröhlich., R., Bohm., A., Ueberholz., P., Korb., M., Goebbels., S., 2019. Roof segmentation based on deep neural networks. *Proc. 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*, 326–333.

QGIS Development Team, 2009. QGIS Geographic Information System. Open Source Geospatial Foundation.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Proc. 31st International Conference on Neural Information Processing Systems*, 5105—5114.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., Koehl, M., 2007. Model-driven and Data-driven Approaches Using LIDAR Data: Analysis and Comparison. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W49A), 87–92.

Wang, Y., Zorzi, S., Bittner, K., 2021. Machine-learned 3D building vectorization from satellite imagery. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 1072–1081.

Zhang, F., Nauata, N., Furukawa, Y., 2020. Conv-MPN: Convolutional Message Passing Neural Network for Structured Outdoor Architecture Reconstruction. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2798–2807.

Zhang, L., Zhang, L., 2018. Deep Learning-Based Classification and Reconstruction of Residential Scenes From Large-Scale Point Clouds. *Proc. IEEE Transactions on Geoscience and Remote Sensing*, 56(4), 1887–1897.

Zhang, Z., Li, Z., Bi, N., Zheng, J., Wang, J., Huang, K., Luo, W., Xu, Y., Gao, S., 2019. PPGNet: Learning Point-Pair Graph for Line Segment Detection. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7098–7107.