Comparison Study of Three Building Regularization Algorithms

Dimitri Bulatov[1,3], Yousif A. Mousa[2,3], Petra Helmholz[3]

[1] Fraunhofer IOSB Ettlingen, Gutleuthausstr., 1, Ettlingen 76275, Germany – dimitri.bulatov@iosb.fraunhofer.de
[2] Civil Engineering Department, Al-Muthanna University, Al-Muthanna, Iraq – yousif.mousa@mu.edu.iq
[3] School of Earth and Planetary Sciences, Curtin University, Perth, WA 6845, Australia – petra.helmholz@curtin.edu.au

Keywords: Building outlining, Building outline delineation, 2D regularization, footprint extraction

Abstract

Building outline generation and regularization is an ongoing topic in remote sensing applications. The success of the methods used for building outline detection impacts studies that depend on the accuracy of the methods applied, such as urban planning, geospatial analysis, and 3D city modeling. The results of the building outline detection methods can vary due to several factors, such as the area to which they are applied and/or the parameters used for the methods. Since there are well-established deep-learning based software and plugins for building outlining, this paper compares the results of such a method, namely an open-source AI implementation (Mapflow), with the standard non-deep-learning based tools introduced by (Mousa et al., 2019) and (Bulatov et al., 2014). We present a comparative analysis of these two methods for regularizing building outlines in terms of accuracy, efficiency, and robustness in dealing with different levels of buildings’ complexity and structures. While the results of (Mousa et al., 2019) and (Bulatov et al., 2014) are comparable and outperform the AI method, (Mousa et al., 2019) is the method least impacted by the different parameters, however, has a higher computing time.

1. Introduction

Automating the process of building detection and regularization from sensor data is an ongoing challenge in photogrammetry, remote sensing, and geographic information science-related applications, such as topographic mapping, urban planning, and 3D city modelling. Usually, the output of building detection methods is in raster format, which needs to be converted into vector format (e.g., building polygons in shapefile format) to be usable for further applications. However, extracting accurate building polygons that meet the standards of mapping generalization is still a challenging task, especially in large-scale datasets. Manual digitization using remote sensing images consumes significant time and resources, especially in large-scale scenarios; it is, consequently, not an option applied. Hence, maintaining the inherent properties of buildings poses a challenge in their automatic conversion to vector graphics, even with supplementary digital surface model data available.

Before implementing building regularization methods, it is necessary to identify buildings at the raster level, which is, in general, a challenging task. Firstly, buildings come in varying appearances, e.g. depending on their location (country and continent, Medieval European historical town center vs new development in the surrounding residential areas vs high-density residential buildings in cities, etc.). Secondly, buildings create shadow areas, which negatively impact the accuracy of machine vision algorithms that utilize only RGB (Hossain and Chen, 2022) or multi-spectral images (Ok, 2013). Therefore, additional information such as elevation, has been proven to be a very useful source to improve the result of building detection. In aerial data, Digital Surface Models (DSM) can be retrieved and filtered to create the Digital Terrain Model (DTM). The difference between DSM and DTM (a normalized DSM, nDSM) then makes it possible to determine the building heights. Some approaches have utilized multiple available data, such as RGB, NDVI, NIR, and elevation data in the form of nDSM, as proposed by (Qiu et al., 2022).

After building candidates are detected, the regularization of building polygons is usually the next step. For this step, some algorithms use optical remote sensing imagery (Kong et al., 2023) and some require supplementary data resources (e.g., airborne LiDAR scanning) to increase accuracy. Despite there are still challenges in generating accurate edges and handling occlusions due to dependencies on curve initialization or limitations of parametric curves (Zhao et al., 2021), the CNN-based methods have achieved tremendous progress in recent years and are considered state of the art, so that they are already implemented as plugins in well-known and widely-used platforms, such as QGIS. Examples of such methods are (Mapflow, 2024) or Polymapper, developed by (Li et al., 2019). However, it remains to investigate how suitable they are in comparison with the conventional, bullet-proved approaches, such as (Gross et al., 2005, Bulatov et al., 2014, Mousa et al., 2019), or a contour-based outlining method (Wei et al., 2023).

The contribution of this paper is to compare one deep-learning-based method (Mapflow, 2024), freely available online since only 2024, with two very successful conventional methods for building outlining, introduced in (Mousa et al., 2019) and (Bulatov et al., 2014) and occasionally denoted, for the sake of brevity, as Al-Muthanna and IOSB method, respectively. For all methods, classification results have to be available to serve as the common ground. Finally, a detailed analysis of the dependencies of parameters will be provided.

The paper is structured as follows: Related works will be introduced in Section 2, followed by the proposed methodology in Section 3. The dataset and the obtained results are presented in Section 4. The conclusion is provided in Section 5.

2. Related Works

As previously mentioned, building footprint extraction and regularization from remote sensing data is usually performed in

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-2024-59-2024

two stages. Extracting the building locations (mostly in raster format) followed by the extraction of building outlines including their regularization (usually in vector format).

Building regularization required a pre-processing step to extract initial boundary points for further simplification and refinement. Despite there are some remarkable exceptions like alpha shapes (Edelsbrunner et al., 1983) or concave hulls (Pohl and Feldmann, 2016), making use of unorganized 2D or 3D point clouds, the boundary is mostly retrieved by identifying only the pixels located on the border and indexing their coordinates. Image processing toolbox can be utilized for this task (e.g., Moore-Neighbor Tracing algorithm) (Gonzales et al., 2004). The extracted border points define a vector sequence that represents the structure of buildings, which often appear as irregular shapes. Therefore, further polygonal simplification or regularization procedures may not be avoided.

Many algorithms have been developed to generate building outlining or regularization from a given list of boundary points. Generally, these algorithms can be categorized into two groups, namely, data-driven and model-driven (Mousa et al., 2019). Data-driven algorithms, such as (Douglas and Peucker, 1973, Pohl et al., 2017), are defined as a polygonal simplification procedure that is free from predefined criteria or knowledge regarding the building structures. Thus, such methods could be applied to extract regular and irregular shapes. In contrast, model-driven methods yield building polygons according to pre-defined rules related to building shapes (e.g., Recursive Minimum Bounding Rectangle) (Kwak and Habib, 2014).

Most recently, AI has been used for the extraction of buildings. For example, (Kong et al., 2022) used a generative adversarial network (GAN) to extract building outlines from gridded binary images with default resolution, whereas no other input parameters are required. Furthermore, (Yang et al., 2018) compared four CNN architectures, namely: Branch-out CNN, fully-convolutional neural network (FCN), conditional random field as recurrent neural network (CRFasRNN), and SegNet, support semantic pixel-wise labeling and focus on capturing textural information on various scales. The CNNs were applied on 1-meter resolution aerial images and achieved satisfying results. Authors of (Wei et al., 2021) used the U2-net semantic segmentation model. The extraction results showed that the U2-net model provides the building outlines with a higher accuracy than other models based on comparisons with semantic segmentation models (Segnet, U-Net, and FCN) and edge detection models (RCF, HED, and DexiNed). Finally, there are a few instant outlining approaches that do not perform classification explicitly, but work directly on images (Marcos et al., 2018, Mahmud et al., 2020). Since we consider the classification result as given, these approaches are not very relevant. Indeed, there are good reasons to use classification as an intermediate output, for example, if the superordinate task is of urban terrain reconstruction, while the building class is simply a by-product of the more general land-cover classification task (Bulatov et al., 2019).

3. Methodology

3.1 Freely available AI-based method – Mapflow

Mapflow (Mapflow, 2024) is used in research to extract, among others, buildings and roads and determine the extent of changes before and after the liquefaction disaster (Giussani et al., 2024) and extraction of photovoltaic data within the urban morphology (Purwanto, 2023). Mapflow can be operated using a web interface or as a QGIS plugin (QGIS, 2024). For this work, the QGIS version 3.36.0-Maidenhead, and Mapflow version 2.5.0 are used. While Mapflow can use web imagery providers (Maxar, Skywatch, and other premium imagery) for the extraction of buildings, agricultural fields, forests (optionally, with height), roads, and construction sites, in this manuscript, we utilize the same ortho-image input as the other methods for comparability and focus on buildings only.

The building extraction consists of the following steps. Firstly, the building roof contours are segmented from high-resolution satellite or airborne imagery. Building candidates with an area of less than 25 sqm. are removed. The AI model used is adopted based on the imagery and location. For instance, there are models for satellite imagery, airborne imagery (GSD of around 10cm), and the so-called high-density housing, designed for areas with terraced or otherwise densely built buildings, common in the Middle East, parts of Africa, etc.

Then, buildings are classified into categories *apartment buildings* (red); *single-household dwellings* (orange); *industrial*; *commercial* (purple), and *other non-residential* (green). An example is shown in 1. The option that, for each building, its height can be estimated using its walls and shadow’s lengths, was not used for this manuscript.

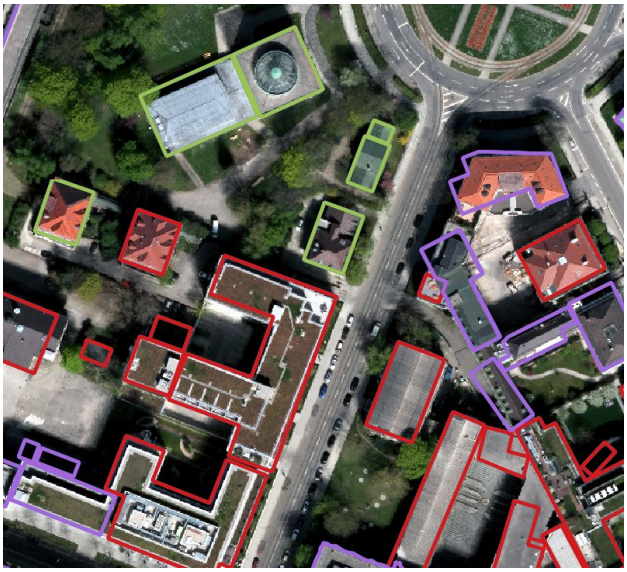

Figure 1. Presentation of building classes in Mapflow

Next, building outlines are simplified to correct the irregularities of the building contours. The irregular geometries are replaced with rectangles, circles, or arbitrary polygons with 90-degree angles, which fit better the original shape. Also, the corrected buildings are rotated to align with the nearest roads.

As some areas have a sufficiently large coverage of OpenStreetMap (OSM) data, there is the option that for each building, it is checked whether it has a corresponding object in OSM (Jaccard index exceeds 0.7). If there is one, Mapflow replaces the result with OSM contour. This makes the result not based on the image, so the buildings can be shifted from actual positions so that some changes that have occurred after OSM mapping may be lost. For this manuscript, we did not utilize the option of using OSM polygons.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-2024-59-2024 | © Author(s) 2024. CC BY 4.0 License.

Details on which approaches are used for the different steps within Mapflow could not be revealed. According to the website information, Mapflow is an artificial intelligence (AI) mapping platform that uses machine learning models.

3.2 Al-Muthanna method

The procedure was firstly introduced in (Avbelj, 2015) and further developed in (Mousa et al., 2019) and (Mousa, 2020) to create building polygons. Assuming that a list of boundary points representing the building border is given, the simplification procedure aims to reduce these boundary points to a minimum while maintaining the original building shape. In other words, the method aims at identifying only the boundary points that are located at building corners (called vertices) and eliminating the remaining ones. To achieve this task, the so-called Likelihood or cornerness function is applied to every polygon vertex. The method is designed in a mathematical form that involves building characteristics (e.g., area, right angles or corners, and edges) to be considered in the following equation:

$$L_i = A_i + \alpha_1 \cdot D_i \cdot \sin(\theta_i) - \alpha_2 \cdot E_i^2, \quad (1)$$

The equation above calculates the values called L_i sequentially, starting from the first point given in the list to the rest of the boundary points in order. In other words, a boundary point in which L_i is calculated is called the boundary point under evaluation while A_i represents the area of simplified polygons that are estimated from the starting point to the point under evaluation, coming back to the first starting point either clockwise or anticlockwise. The distance between the starting point and the point under evaluation is indicated by D_i. The sine of θ_i is calculated at each boundary point under evaluation while E_i refers to mean squares orthogonal distances from the boundary points, located between the starting point and the point under evaluation. Lastly, α_1 and α_2 represent fixed coefficients used to adjust the weighting of various terms, and they are established experimentally at 20.0 and 2.0, respectively.

The extracted building polygons from the applied polygonal simplification method are usually in irregular format; therefore, a refinement procedure seems required. The first job for refinement is creating right angles at building corners (e.g., rectilinear buildings). Second, solving the best-fit problem between the input boundary points and the extracted polygon edges. Both jobs are solved by implementing the Gauss-Markov model and the Gauss-Helmert model through the least squares adjustments procedure as introduced in (Mousa, 2020)

There are two main thresholds in the applied procedure: the-root-mean-square-error (RMSE) threshold δ_e and the minimum distance threshold for angle detection δ_θ. The RMSE threshold refers to the distances of the input boundary points to the predicted polygon edges. In general, a small value of δ_e threshold provides a high number of edges or vertices and, therefore, requires a longer processing time. In contrast, high values of δ_e result in a high degree of simplification, which means fewer edges or vertices. Thus, this setting requires less processing time. However, a very high value of δ_e (over-simplification) may also result in insignificant more processing time due to repeating more adjustment attempts. The second threshold, δ_θ is defined as the minimum length of the sides of the expected corners. A small or high value of δ_θ threshold would not significantly affect the processing time but could produce a change in the positions of the detected corners. This threshold δ_θ should be set to the shortest building edges.

3.3 IOSB method

The method was first published in (Gross et al., 2005) and further developed in (Bulatov et al., 2012, Bulatov et al., 2014). To determine the dominant building direction, straight lines (Burns et al., 1986) are obtained in the slightly smoothed DSM. If an orthophoto is available, straight lines can be computed from a gray image, as well. Their slopes are stored in a histogram modulo 90°. The histogram entries are weighted by the lengths of the line segments and by the data source. Per default, the weights for DSM and orthophoto are 3/4 and 1/4, respectively, chosen to emphasize the importance of 3D data for building detection. However, if the elevation map is rather noisy, a higher weight can be conceded to the orthophoto. An assessment of orthogonality is performed for each building. If the histogram has a strong peak, resulting from the ratio second-best to best falls below a threshold η, the dominant direction corresponds to this peak. Otherwise, there is no dominant direction.

To obtain the ground-plan polygon of a rectilinear building, the axes of the minimum bounding rectangle are given by the dominant directions of the building orientation. Using the binary mask, the contours are refined for each blob by recursively adding and removing rectangular subparts until the area of the remaining blob lies under a threshold μ_A. The subparts correspond to the small convexities or indentations in the building contour with respect to the minimum bounding rectangle while μ_A, similar to the other thresholds for the building reconstruction, reflects the maximal deviation of areas that one is ready to accept. All thresholds depend on resolution, quality of the data, etc. To generalize the orthogonal polygon, we first check collinear points and, finally, make sure that no rectangular subpart of an area below a threshold and whose one side is the currently shortest building edge can be added.

For non-orthogonal polygons, pixel-wise polygonization relying on a standard contour-tracing algorithm (Gonzales et al., 2004) is performed followed by a three-module generalization routine. The first step is the exploitation of line constraints to reduce the number of vertices. To do this, similarly to the previous section, two steps, namely computing vertices’ usefulness and deleting low-usefulness vertices, are performed alternatingly. In the second step, this number is further reduced by checking whether too short edges can be replaced by the intersection point of the adjacent edges. The key parameter here is whether the intersection point lies far away from the edge in question. The third and last step presupposes enforcing rectangularity, wherever possible, and thus improving the positions of vertices. Differently to Section 3.2, this is not an optimization process. That is, previously formed orthogonalities can be destroyed – again – by the new ones, leaving to the theoretically infinite process. However, using iterative calculations, we could find some sensible heuristics.

A special case for outlining orthogonal and non-orthogonal polygons is if a polygon has one or a few holes of non-negligible size. Such holes are easy to detect. Every hole is declared as unvisited at the beginning of the hole-outlining algorithm, and the outline is just the exterior polygon. The shortest diagonal between one vertex belonging to an unvisited hole and the other one belonging to the outline is traced and added to the outline polygon. The vertices of the hole are added in between, after which the hole is declared visited. Then the process starts from anew, and it terminates if there are no unvisited holes anymore.

There are a few important parameters in the method, and we will concentrate on two: the rectangularity threshold η defines

the ratio between the second-best and best edge direction. The lower it is, the fewer buildings will be outlined as rectangular ones. The area tolerance threshold μ_A can be defined in absolute and relative in terms of total building size terms. The software may consider both, but our focus lies on the latter choice.

4. Evaluation

4.1 Dataset and classification algorithm

The dataset considered in this paper is from the German city of Munich; since this dataset covers a large part of the inner city, it enables us to evaluate our method on a quite large area with many nested buildings. The data was provided by the Institute of Photogrammetry of Stuttgart (IfP) as a result of photogrammetric reconstruction, that is, a DSM and an ortho-photo were available Additional features, such as nDSM and planarity (that is, how likely a neighborhood of single pixel can be approximated by plane), can be computed as in (Bulatov et al., 2014) and, respectively, (Gross and Thoennessen, 2006) from the DSM.

To obtain a land cover classification of this dataset, the approach of (Qiu et al., 2022) was employed. This approach was developed to process combined multi-source data, such as optical and elevation data. In a nutshell, the original architecture of the state-of-the-art DeeplabV3+ approach (Chen et al., 2018) is modified to have two input branches. The first is the RGB orthophoto, and the second is the synthesized image with channels made up of truncated nDSM, NDVI, and planarity map. Given that the backbone of the DeeplabV3+ approach is ResNet-encoder (He et al., 2016), both images are passed through their first residual block, after which the corresponding features are averaged and processed by the remaining ResNet blocks together. The standard cross-entropy function was our choice for the loss function.

Since the DeeplabV3+ is a supervised method, we need some reference data. For the Munich dataset, (Bulatov et al., 2019, Häufel et al., 2018) showed that it is possible to obtain training data for buildings and some other larger classes using heuristics, such as freely available (OSM) data, segmentation results, and simple features such as the aforementioned ones. However, to obtain more accurate outlines (and also to be able to extract some less frequent classes), we annotated a few training patches and fine-tuned the previously computed Potsdam model (Qiu et al., 2022). The Postdam dataset is an openly available ISPRS benchmark with an abundant amount of labeled data (Rottensteiner et al., 2014).

4.2 Qualitative Analysis

We refer to Figures 2 to 6 for the qualitative results. We denote by cyan, blue, yellow and green color: the Mapflow result, Al-Muthanna result, IOSB result with a too high value for η, and a regular, maybe slightly too low value of η in Figures 3 and 6.

Figure 2 shows a relatively simple example of building outlining using all methods. The Mapflow result looks artificial and reminds more manual than automatic tracing of the outline. In the Al-Muthanna result, one part seems to be overgeneralized while in IOSB result, the non-rectangular building seems to be reliably reconstructed.

In another, relatively easy example in Figure 3, we can see in the left-most image that Mapflow produces the by far cleanest outlines for the church. Contrarily, the building with two holes has not been reconstructed in a uniform format by Mapflow. Furthermore, the method developed by Al-Muthanna produces an incorrect outline of the small building between the church and the large building (left image). The outlines obtained by the IOSB look relatively similar. Here, the yellow results look, intuitively, slightly more sensible, e.g., the long narrow passage to the South of the church and the church itself. However, we see a few mis-detections that were not suppressed and will, therefore, negatively impact the performance on the vector level (see Section 4.3).

In Figure 4, we see that, despite the results of Mapflow are meaningful, they exhibit subdivision of connected buildings as if the OSM-Data were used. Al-Muthanna and IOSB methods perform almost equally, only that the structure surrounded by the dark-red circle in the IOSB result seems to be an outlier. One can see that the numerous holes are being handled well by these methods.

Figure 5 shows two parameter settings for the Al-Muthanna method, whereby white means a larger threshold d_e, leading to more artifacts. We can see from this figure that only a few buildings are fully rectangular and were reconstructed by GHM procedure. We specified them by dark-red dashed circles. Finally, Figure 6 shows a building in which how too many rectilinear walls are fitted in both building complexes because they were spuriously assigned to the rectilinear building category. Due to over-generalization, one edge does not satisfy this rectilinearity constraint. By a more adjusted choice of η, the green outlines in Figure 6, bottom, intuitively better fit the actual building mask, however, there are some very adventurously looking inner courts. The surrounding, partly visible in Figure 6 buildings, have a simpler shape and, hence, both green and yellow outlines practically coincide and look both fairly correct.

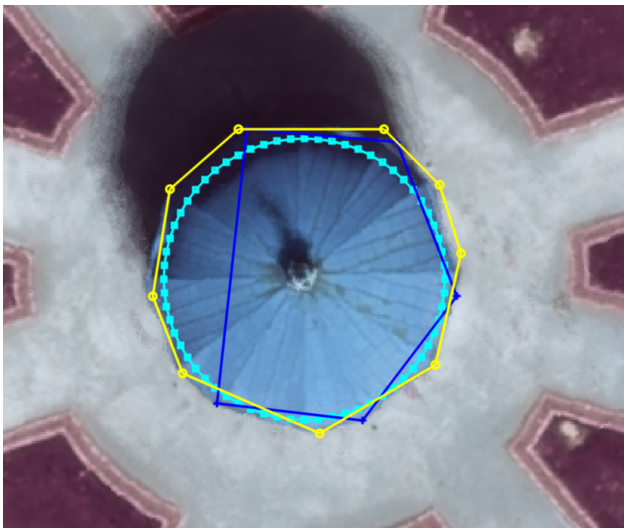

Figure 2. Qualitative comparison of methods: Easy example.

4.3 Quantitative Analysis

For quantitative results on the raster level, we use the typical in Remote Sensing measures for overall accuracy (OA) and Cohen’s kappa. On the vector level, we upsample the building edges of the prediction p and the reference r to a desired tolerance (here 1 pixel), and from every detected vertex v_p, we determine the closest point v_r belonging to the reference building mask. From the distance $d_p = \min_r(d(v_p, v_r))$ between

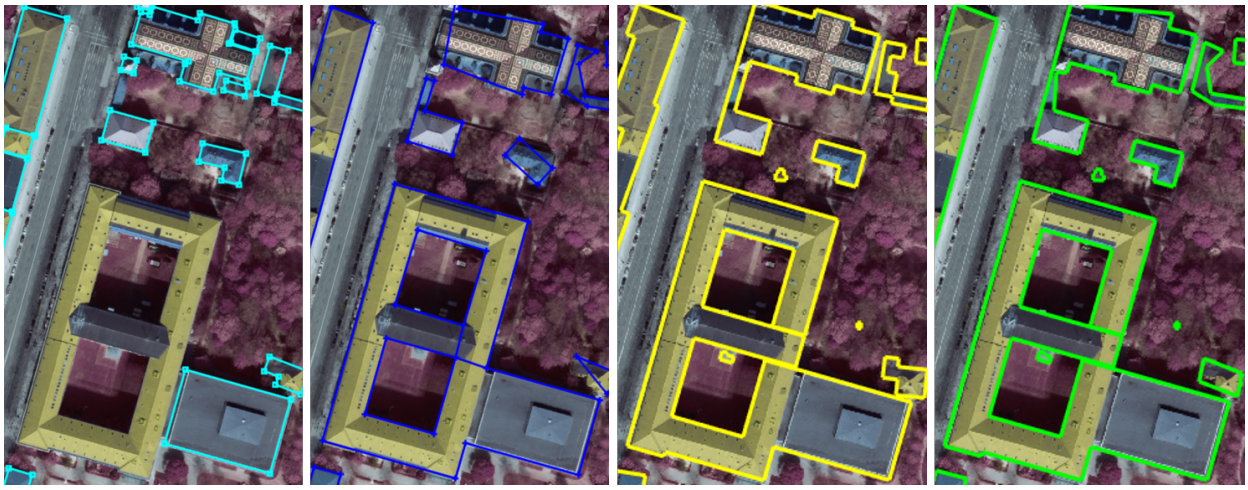

Figure 3. Extended building with inner courtyard. Cyan, blue, yellow, and green color: Mapflow, Al-Muthanna, IOSB results (too high value of η, and a regular η).

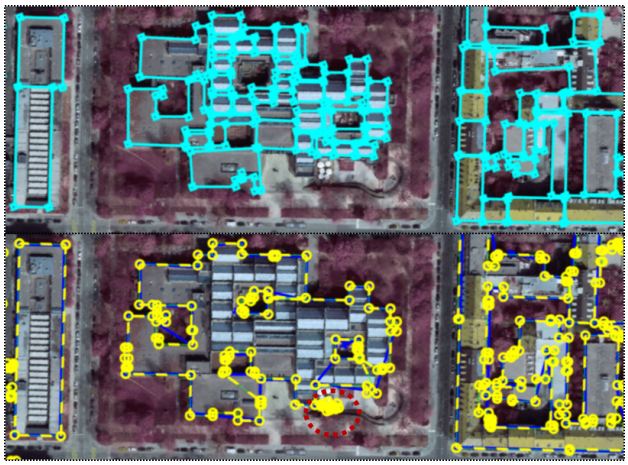

Figure 4. Difficult example on building outlining using the considered methods, colors as in Figure 3.

Figure 5. Difficult example on building outlining using the Al-Muthanna method, with two different parameter settings.

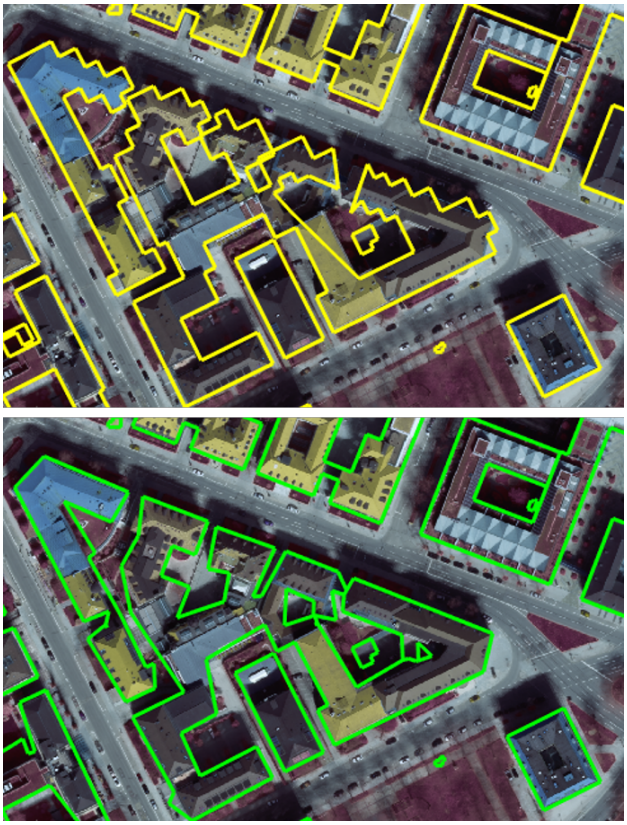

Figure 6. Example on building outlining from Figure 5 using the IOSB method, with too high (top) and regular, slightly too low (bottom) value for η.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-2024-59-2024 | © Author(s) 2024. CC BY 4.0 License.

the corresponding pairs v_p and v_r, we derive several measures: The average distance (L1), the root-mean-square distance (L2) but also the median of maximum per-building values of d_p.

$$d_m = \text{median}_{B_i} \left(\max_{p \in B_i} d_p \right), \tag{2}$$

where B_i denotes the building index under evaluation. Switching the roles of r and p and taking the maximum, we obtain the symmetric distances for the aforementioned measures.

We note that the deviations are computed between the reconstructed outlines and the reference data, not the classification result. That is, all methods are accurate up to classification. However, the state-of-the-art result of (Qiu et al., 2022) achieved more than 95% on overall accuracy for this dataset, as stated in (Böge et al., 2024). Moreover, all methods had the same basis for comparison.

The results of the Al-Muthanna method are comparable to those of the IOSB method (see Table 1). Both achieve OA of over 90% and peak values exceeding 95%. The highest kappa values are also comparable, with around 92%. In contrast, OA and kappa are considerably less for the Mapflow method, which is caused due to the missing buildings in the detection. A similar trend is visible for the L1 value. The Al-Muthanna method outperforms the IOSB method significantly for $\mu_A = 0.9$ and slightly for $\mu_A = 0.99$. In contrast, for the L2 comparison, the results for μ_A are quite similar, which leads to the conjecture that the IOSB method could be slightly more robust with respect to outliers handling, as seems to be confirmed by qualitative results.

Looking more closely at the dependencies of the results on the achieved accuracies, for the Al-Muthanna method, it can be summarized that all results are little impacted by the change of the parameters. As for the IOSB method, the parameter μ_A seems to be decisive for the quantitative evaluation. With $\mu_A = 0.99$, the absolute change between the initial and the reconstructed outline can only be relevant for very big buildings. The rectangularity threshold η only impacts qualitative evaluation and on computation time; the quantitative results are, unsurprisingly, stable. The procedure for generalization of the non-rectilinear building outlines is time-consuming since every point is checked in an iterative procedure, resulting in roughly quadratic processing time.

5. Conclusion and Future Works

Accurate building outlines are crucial for urban planning, geospatial analysis, and 3D city modeling. If decision-makers are dependent on this data, then the data should be not only accurate but also relatively insensitive to the input parameters.

This paper analyzed the parameter dependencies on results of the methods introduced by (Mousa et al., 2019) (Al-Muthanna) and (Bulatov et al., 2014) (IOSB). Both methods can achieve very high overall accuracies with over 90% and kappa values of around 93%. These results outperform an AI implementation (Mapflow) which is freely available. The main reason is the completeness of the extracted buildings. Mapflow missed a few buildings due to an unknown yet reason. Finding out this reason and trying out possible alternatives to Mapflow will comprise the future works.

Investigations showed that Al-Muthanna method is the least impacted by the different parameters. However, it seems that the method requires longer processing times compared to the IOSB method, especially if the RMSE threshold d_e is small.

Acknowledgement

The authors thank Prof. Norbert Haala and Dr. Mathias Rothermel from the Institute of Photogrammetry of Stuttgart (IfP) for providing image and elevation data of the Munich dataset.

References

Avbelj, J., 2015. Fusion of hyperspectral images and digital surface models for urban object extraction. PhD thesis, Technische Universität München.

Böge, M., Mousa, Y., Bulatov, D., Qiu, K., 2024. Building detection and outlining in multi-modal remote sensor data: A stratified approach. *Article under Review*, None, None.

Bulatov, D., Häufel, G., Lucks, L., Pohl, M., 2019. Land cover classification in combined elevation and optical images supported by OSM data, mixed-level features, and non-local optimization algorithms. *PE&RS, Photogrammetric Engineering & Remote Sensing*, 85(3), 179–195.

Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P., Wernerus, P., 2014. Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 157–170.

Bulatov, D., Wernerus, P., Gross, H., 2012. On applications of sequential multi-view dense reconstruction from aerial images. *Proc. of International Conference on Pattern Recognition Applications and Methods*, 2, 275–280.

Burns, J., Hanson, A., Riseman, E., 1986. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4), 425–455.

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-Decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611. http://arxiv.org/abs/1802.02611.

Douglas, D. H., Peucker, T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: International Journal for Geographic Information and Geovisualization*, 10(2), 112–122.

Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 29(4), 551–559.

Giussani, F., Wilczynski, E., Zandonella Callegher, C., Dalle Nogare, G., Pozza, C., Novelli, A., Pezzutto, S., 2024. Use of machine learning techniques on aerial imagery for the extraction of photovoltaic data within the urban morphology. *Sustainability*, 16(5), 2020.

Gonzales, R. C., Woods, R. E., Eddins, S. L., 2004. *Digital image processing using MATLAB*. Pearson Prentice Hall.

Method	parameter		OA	kappa	L1	L2	d_m	Time (s)
Al-Muthanna 0	2	1.2	95.65	90.87	8.23	31.05	26.88	2456.00
Al-Muthanna 1	2	1.8	96.10	91.81	7.83	30.86	24.18	2818.00
IOSB 0	50	90	92.18	83.73	13.30	36.38	26.93	715.93
IOSB 1	50	99	96.23	92.09	9.66	32.95	22.96	767.68
IOSB 2	80	90	91.43	82.21	14.06	36.75	26.93	73.57
IOSB 3	80	99	96.14	91.92	9.66	32.63	23.25	74.48
Mapflow	withoutOSM		89.65	77.61	19.97	34.80	66.03	420.00

Table 1. Quantitative assessment: The parameters for the Al-Muthanna method are δ_θ and δ_e, given in meters, while the parameters for the IOSB methods are η and μ_A, both given in percent points. L1 and L2 denote the absolute and RMSE errors, respectively, while d_m is the only per-building measure explained in (2). These measures are given in pixels while the raster-based OA and kappa are in percent points.

Gross, H., Thoennessen, U., 2006. Extraction of lines from laser point clouds. *Symposium of ISPRS Commission III: Photogrammetric Computer Vision PCV06. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36number Part 3, 86–91.

Gross, H., Thoennessen, U., Hansen, W. v., 2005. 3D modeling of urban structures. *International Archives of Photogrammetry and Remote Sensing*, 36(Part 3), W24.

Häufel, G., Lucks, L., Pohl, M., Bulatov, D., Schilling, H., 2018. Evaluation of CNNs for land cover classification in high-resolution airborne images. *Earth Resources and Environmental Remote Sensing/GIS Applications IX*, 10790, SPIE, 12–22.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hossain, M. D., Chen, D., 2022. A hybrid image segmentation method for building extraction from high-resolution RGB images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 192, 299–314.

Kong, G., Fan, H., Lobaccaro, G., 2022. Automatic building outline extraction from ALS point cloud data using generative adversarial network. *Geocarto International*, 37(27), 15964–15981.

Kong, L., Qian, H., Xie, L., Huang, Z., Qiu, Y., Bian, C., 2023. Multilevel regularization method for building outlines extracted from high-resolution remote sensing images. *Applied Sciences*, 13(23), 12599.

Kwak, E., Habib, A., 2014. Automatic representation and reconstruction of DBM from LiDAR data using Recursive Minimum Bounding Rectangle. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 171–191.

Li, Z., Wegner, J. D., Lucchi, A., 2019. Topological map extraction from overhead images. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Mahmud, J., Price, T., Bapat, A., Frahm, J.-M., 2020. Boundary-aware 3D building reconstruction from a single overhead image. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 441–451.

Mapflow, 2024. Mapflow.ai - ai mapping and imagery analysis platform, version 2.5.0. Mapflow.AI.

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., Urtasun, R., 2018. Learning deep structured active contours end-to-end. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8877–8885.

Mousa, Y. A., Helmholz, P., Belton, D., Bulatov, D., 2019. Building detection and regularisation using DSM and imagery information. *The Photogrammetric Record*, 34(165), 85–107.

Mousa, Y. A., 2020. Building Footprint Extraction from LiDAR Data and Imagery Information. PhD thesis, Curtin University.

Ok, A. O., 2013. Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts. *ISPRS journal of photogrammetry and remote sensing*, 86, 21–40.

Pohl, M., Feldmann, D., 2016. Generating straight outlines of 2D point sets and holes using dominant directions or orthogonal projections. *VISIGRAPP (1: GRAPP)*, 59–71.

Pohl, M., Meidow, J., Bulatov, D., 2017. Simplification of polygonal chains by enforcing few distinctive edge directions. *Proceedings of 20th Scandinavian Conference on Image Analysis: Part II 20*, Springer, 3–14.

Purwanto, Ajun, P., 2023. Utilization of Deep Learning for Mapping Land Use Change Base on Geographic Information System: A Case Study of Liquefaction. *Journal Penelitian Pendidikan IPA*, 9(10), 8059-8064.

QGIS, 2024. Qgis, a free and open source geographic information system. https://qgis.org.

Qiu, K., Budde, L., Bulatov, D., Iwaszczuk, D., 2022. Exploring fusion techniques in U-Net and DeepLab V3 architectures for multi-modal land cover classification. *Earth Resources and Environmental Remote Sensing/GIS Applications XIII*, 12268, SPIE, 190–200.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J., Breitkopf, U., Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 256-271. https://www.isprs.org/education/benchmarks/UrbanSemLab/results/potsdam-2d-semantic-labeling.aspx.

Wei, S., Zhang, T., Ji, S., Luo, M., Gong, J., 2023. BuildMapper: A fully learnable framework for vectorized building contour extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197, 87–104.

Wei, X., Li, X., Liu, W., Zhang, L., Cheng, D., Ji, H., Zhang, W., Yuan, K., 2021. Building outline extraction directly using the u2-net semantic segmentation model from high-resolution aerial images and a comparison study. *Remote Sensing*, 13(16), 3187.

Yang, H. L., Yuan, J., Lunga, D., Laverdiere, M., Rose, A., Bhaduri, B., 2018. Building extraction at scale using convolutional neural network: Mapping of the United States. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(8), 2600–2614.

Zhao, W., Persello, C., Stein, A., 2021. Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework. *ISPRS Journal of Photogrammetry and Remote sensing*, 175, 119–131.