AUTOMATIC EXTRACTION OF FACADES AND WINDOWS FROM MLS POINT CLOUDS USING VOXELSPACE AND VISIBILITY ANALYSIS

Ludwig Hoegner [a,b,*] and Georg Gleixner [a]

[a] Photogrammetry and Remote Sensing - TUM School of Engineering and Design - Technical University of Munich
[b] Hochschule München University of Applied Sciences
ludwig.hoegner@hm.edu

Commission II, WG II/4

KEY WORDS: Mobile Mapping, Point Cloud, facade, Windows, 3D Object Extraction.

ABSTRACT:

This contribution presents a method for extracting a 3D model of facades and windows from a point cloud. The point cloud is segmented based on a voxel octree, in which the facades are sought as planes. These can be used to filter out potential window points within the building, which are then analysed on their visibility by checking the occupancy grid of the voxel space. Here, methods of digital image processing are used for analysing both point clusters behind the facade and holes in the estimated facade planes as window candidates. Facades and windows are both simplified as rectangles. The test data set was gathered in a Mobile Laser Scanning campaign. While the segmentation fails in some cases, the extraction of facades and windows shows good results: 25 facades with 702 detected windows yield a detection rate of 86% with a false alarm rate of 13%. The reconstructed sizes of the windows differ from reference measurements in the range of centimetres to a few decimetres. These refined geometries can be used to enrich existing building models or for vehicle navigation without GNSS.

1. STATE OF THE ART

With increasing density and accuracy of mobile laser scanning (MLS) based point clouds possibilities arise for reconstructing 3d building facades according to CityGML level-od.detail 3 (LOD3) of corresponding IFC models. One important task is the detection and extraction of windows. Window extraction from images is an extensively researched area for both images in the visible (Reznik and Mayer, 2008) and infrared spectrum (Michaelsen et al., 2012). In these cases, facade planes are estimated from generated point clouds or given from existing 3d building models. The images are then used as textures on these planes and analysed by their intensity values. In contrast, window detection based on laser scanning point clouds is based on a point cloud segmentation process.

Model-based methods often search for geometric primitives like lines (Widyaningrum et al., 2019), planes (Filin and Pfeifer, 2006), cylinders (Tarsha-Kurdi et al., 2007), or spheres (Rabbani et al., 2006). Radiometric attributes of the points, such as the intensity of reflection (Nobrega and O'Hara, 2006), are also considered here. The points belonging to a primitive and the parameters of the primitive can be determined by a Hough transformation (Rutzinger et al., 2011) or by tensors (Schuster, 2004). Since there are usually significantly more points than necessary to determine the parameters in a segment, the optimal set of parameters is determined in the context of quadratic error minimization (Castillo et al., 2013). A simplification within quadratic error minimization is the Principal Component Analysis- PCA (Lari and Habib, 2014).This also allows to infer from the principal components of each segment the most suitable geometric primitives to describe this segment. Found geometric primitives can be described via Boundary Representation or Constructive Solid Geometries (Brenner, 2005). Thereby complex structures are described by union, intersection or difference of primitives.

Point cloud segmentation is also done by region growing methods. Commonly used for the selection of the local neighbourhood is the k-nearest neighbour (KNN) method. Distance, normal direction or intensities can be used as criteria for adding a point to a segment (Lee and Schenk, 2001). These local point-related parameters can be extended by other geometric parameters such as planarity, curvature, or roughness (Pu and Vosselman, 2009). Region growing methods can also be applied on area patches (Li et al., 2019) or voxels (Xu et al., 2017) instead of points. Unlike region growing methods, clustering requires no starting points. For example, the normal directions (Vo et al., 2015), the distance (Aldoma et al., 2012) and the point density (Aljumaily et al., 2017) in the local neighbourhood can serve as attributes. Clustering can then be performed based on mean-shift method (Yao et al., 2009). Clustering methods can also be averted on voxels (Wu et al., 2013) or surface patches (Vosselman et al., 2017).

In global energy minimization, an energy function is minimized that separates points in segments. One variant is graphical models that derive a set of parameters from the points to estimate similarities (Hong et al., 2019). Normalized cut (Shi and Malik, 2000), min cut (Golovinskiy and Funk, 2009) or graph-based segmentation (Green and Grobler, 2015) can be used to segment this graph. The relationship between points can also be described using a Markov model instead of a weighted graph. They

* Corresponding author

can be described as Markov Random Fields (Hackel et al., 2016) or Conditional Random Fields (Rusu et al., 2009). The actual segmentation is then performed by a graph-cut algorithm (Boykov and Kolmogorov, 2004). In addition to graphical models, energy terms can also be realized using level sets (Kim and Shan, 2011) or as a global energy minimization problem (Dong et al., 2018).

A special case in that field is the detection of windows. It can be seen that laser beams partially penetrate the window glass, so that holes appear in the point cloud in the plane of the facade and, in return, points appear in the building interior in the line of sight of the beam path. In (Tuttas and Stilla, 2013) a method using both the detection of holes in the facade planes and points behind that planes is introduced for coarse airborne laser scanning (ALS) point clouds. Results show a good completeness of the window detection. The limited point density leads to a very coarse geometric accuracy of the extracted windows. In our contribution, this method is adapted for mobile laser scanning point clouds (MLS). As the point density is much higher, the window detection of (Tuttas and Stilla, 2013) is combined with a voxel-space representation as introduced by (Xu et al., 2017). This allows both the search for facade planes and the analysis of the occupancy grid for the window detection.

2. EXTRACTION OF FACADES AND WINDOWS FROM MLS POINT CLOUDS

Our proposed method is split in three parts. First, (Xu et al., 2017) is applied to generate a discrete voxel space. This voxel space will be used in the second step for facade plane detection and the occupancy grid generation. The voxel space is then used for the window detection as described in (Tuttas and Stilla, 2013). We extend the detection by a texture analysis framework inspired by (Schneider and Coors, 2018). The facade plane points are projected onto a 2d facade texture. The same is done for the intersection points of the projection rays of the points behind the facade. In contrast to (Tuttas and Stilla, 2013), not only the intersection points are used for window detection, but also the holes in the facade points. To do so, it is necessary to add a visibility analysis to avoid window detection in occluded or non-recorded parts of facades.

2.1 Partitioning of the point cloud in voxels

The estimation of a facade plane and an occupancy analysis is done by discretizing the space in voxels (Xu et al., 2017). Since most voxels in the space are empty, an octree is constructed to reduce the amount of data. Voxels in the octree are then divided into subspaces if they are not empty until a minimum voxel size is reached, the number of points in a voxel falls below a threshold, or, if the residuals of the plane estimated in the points within a voxel are too large and indicate that there is more than one plane in the voxel.

The standard deviation σ of a voxel indicates the mean deviation of the points from the estimated plane of a voxel (Vo et al., 2015). This plane is determined by the plane equation of the Hessian normal form. As initial voxels for the plane search, voxels aare selected with a standard deviation below an initial threshold. These voxels are assumed to be the best representations for facade planes. All neighbours are now checked to see if the angle between the normal vector of the starting voxel and the normal vector of the neighbour is less than a set threshold. Additionally, it is checked whether the distance of the planes between two voxels is smaller than a threshold value. This prevents closely spaced facades from being merged into one segment. This is repeated iteratively until no more voxel can be added to the segment. Now it is checked whether the segment can be a facade segment. For this purpose, a minimum number of contained points as well as a deviation of the normal vector of max. 30° from the horizontal are assumed. If these rules are met, the segment is saved. Otherwise, the segment is deleted and the points are marked as unsegmented. This procedure is repeated until all voxels have been processed and thus either assigned to a segment or marked as unsegmented.

In phase two, the unsegmented voxels are checked to see if they match an existing segment. For this purpose, the procedure according to (Vo et al., 2015) is applied. If at least 80% of the points in a segment are within a specified distance interval from the assumed plane, the segment is considered planar and so-called Fast Refinement is performed, otherwise General Refinement is performed. Fast Refinement determines for each unsegmented point N_i of a voxel the distance to the estimated plane of the segment R_j. If the distance is less than a threshold, the point is added to the segment. The voxel to which this point belongs is marked as possibly belonging to the segment and stored in a candidate list S_j. This process is repeated for all points of all neighbouring voxels in a segment, and then for all neighbouring voxels of voxels in S_j. If the voxel's normal direction and distance meet the thresholds, the voxel is added to the segment. In General Refinement, the points are directly checked for their distance to the nearest point of the segment R_j and iteratively added one by one. The check of the voxels is omitted.

2.2 Plane estimation

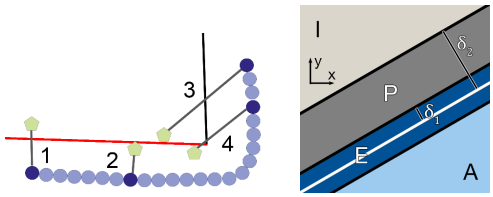
The estimation of the facade planes is now performed for each segment by estimating the parameters of the plane equation in parameter form

$$ax + by + cz + d = 0; a^2 + b^2 + c^2 = 1 \qquad (1)$$

where (a, b, c) are the three components of the normal vector and d is the distance to the origin. It is assumed for the estimation that there are more points on the facade plane than in front of or behind. The MSAC method is used to estimate the parameters (Urbančič et al., 2014). The inliers now make up the facade. All the outliers of the segment are now checked whether they are behind the plane of the facade from the viewpoint of the recording location. For this purpose, the intersection point of the projection ray of a point p with the facade plane is calculated and checked whether it lies within the boundaries of the facade. If so, it is calculated whether the point is behind the facade by transforming of equation 1:

$$l = sign\left(\left\langle p, \begin{matrix} a \\ b \\ c \end{matrix} \right\rangle + d\right) \quad (2)$$

If l is equal to 1, point p lies in the direction in which the normal also points, with -1 in the opposite direction, with 0 the point is in the plane. If the normal direction of the facade is defined away from the line of sight, it means that a point with $l = 1$ lies behind the facade. Figure 1 shows the classification of the points that are not counted to the facade plane. Fig. 1a shows the possible positions and recording directions of the points. A point may be located in front of or behind the facade, or may have been taken from behind when driving around a building corner, for example. Fig. 1b shows the assignment areas for points to the facade. Here E is the range of safe facade points according to MSAC, A is the range of exterior points that lie in front of the facade, I is the interior range of points that lie safely behind the facade, and P is the range of points that were sorted out by MSAC but cannot be safely assigned to the interior segment I due to the noise behaviour of the point cloud. For remaining untested points, we now test whether they intersect another facade within their area along the line of sight.



(a) (b)

Figure 1. Determination of points located behind the facade. (a): top view of possible locations of viewpoint (blue) and measurement point (green) to facade plane (red): (1) outside - inside, (2) outside - outside, (3) inside - inside, (4) inside - outside. (b): Separation of interior and facade points by top view. Dark blue: MSAC inlier in tolerance δ_1 to plane E (white). Light blue area A: Points in front of the facade (exterior points). Light grey area I: Interior points. Dark grey area P between δ_1 and δ_2: Buffer zone whose points are not used further

.

2.3 Window extraction

For window detection, the facade points and the intersection points of the visible rays of the interior points with the facade plane are now projected onto the facade plane and provided with texture coordinates. Feature of a window is a hole in the points of the facade and a segment at intersections with imaging rays. Figure 2 shows the scheme when searching for windows in a facade. First, it is checked if there are enough intersection points (a). In this case, window positions are searched based on the intersection points like in (Tuttas and Stilla, 2013). If positions are not found there, it is additionally checked for gaps in the facade points (b). If this also remains without result, the search is carried out in a finer grid with facade points and intersection points (e) and further window candidates are searched for in the regular structure of the facade (f). For the positions found, a circumscribing rectangle of the windows is now determined in a fine grid with facade points and intersection points (c) and then the windows are checked for completeness (d).

For the case Fig. 2a the extraction strategy of (Tuttas and Stilla, 2013) is used. A binary image with a grid size of 0.5 m is generated from the intersections. The image is then oversampled by a factor of 10. The result is a smoothed, low-noise binary image. A cross-correlation is performed on this image using two templates with a horizontal and vertical bar of length 1.5 m, respectively. Maxima in the correlation indicate the positions of the windows in x (columns) and y (rows).

If not all window positions are found in this way, the intersection points can be supplemented by our extended method. For this purpose, the projection of the facade points is analysed (Fig. 2b). Raster cells with few or no facade points are now also marked in the binary mask and the search is repeated. However, reflective areas where the laser beam was reflected away are now also included, as well as occluded and invisible parts of the facade, e.g. at the edge of the bounding box. The latter is counteracted by the mentioned minimum distance to the edge at least in x-direction. If this search also does not lead to the result, then according to Fig. 2e, both methods are combined where areas with intersection points or holes in the facade points are searched. If the search was successful, the contours of the windows are determined in the next step. Figure 3 shows one example facade where 3a is the mask of the facade point holes, 3b the mask of the intersection points and 3c the combined binary mask. Where barred windows are located, facade points are created on the bars and, at the same time, intersection points are created in the gaps between the bars. In the combined image, the intersection points now overrule the facade points and the barred window is recognized as such. Windows through which few interior points were seen are better represented in the facade point image. This is the case when a window is seen at a very acute angle as with the top floors. All windows are now captured in the added image. There is also the case where, due to the scanning by the scanner, areas have only a few facade points. So that these are not mistakenly considered as holes, in such areas only the intersection points are used for the search for windows.

Due to the high point density of the MLS point cloud compared to the ALS point cloud, a procedure based on (Schneider and Coors, 2018) is used for the contour search instead of the method proposed in (Tuttas and Stilla, 2013). The binary images of facade points and intersection points are sampled in 1 cm. Morphological operators are used to remove disturbances in the windows, e.g. single points of the window intersections. This is to prevent a breakout/leakage effect during later contour tracing. The reconstruction of the contour of the windows (Fig. 2c) is performed for each detected object via the contour tracking and clustering method according to (Gonzalez et al., 2004) in the fused binary image. For the contours, it is specified that they must be the smallest circumscribing rectangle of a segment and that these rectangles must not be wider than 3.5 m and

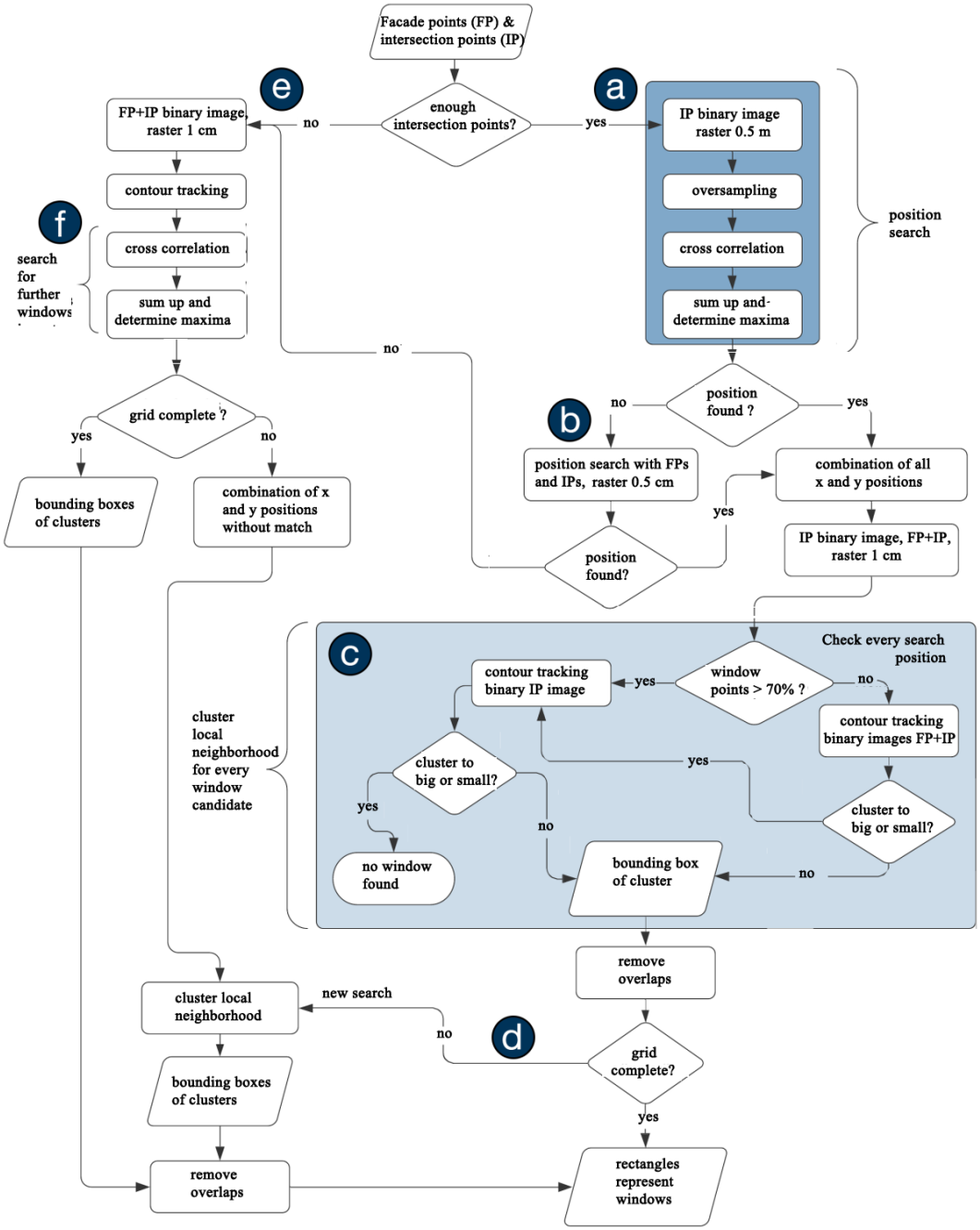


Figure 2. Flowchart for window extraction. (a) Position search with enough intersection points (b) Add facade points to the search. (c) Contour determination. (d) Check for completeness of regular arrangement of windows. (e) Finer search with facades and intersection points. (f) Search for more windows in the grid of the regular grid.

taller than 5.5 m. With these parameters, most windows are covered. In addition, rectangles with a side length of less than 30 cm are excluded. For each rectangle, it is also checked that its centroid is no further than 1.5 m from the search points determined by the cross-correlation. If there are still several matching rectangles around a search point, the one with the smallest distance to the search point is selected. If no rectangle is found for a search point in this way, the search is repeated on the basis of the binary image of the intersection points. If there are still search points without a rectangle, the search criteria are adjusted for these positions. The dimensions of a window are set to 1.5 times the average size of the windows already found in the same row and additional artificial search points are inserted, the distance between which, however, must not be less than 0.4 m. This is to prevent deviations from the grid of the rectangles. Figure 4a shows the contour extraction for a facade piece. The facade points are shown in grey, and the intersection points are shown in orange- The crosses mark the previously found search points for the expected window positions in the grid. The rectangles represent the extracted windows. The two windows on the left were extracted from the merge binary image. The two windows on the right could only be found in the intersect binary image because the point density of the facade

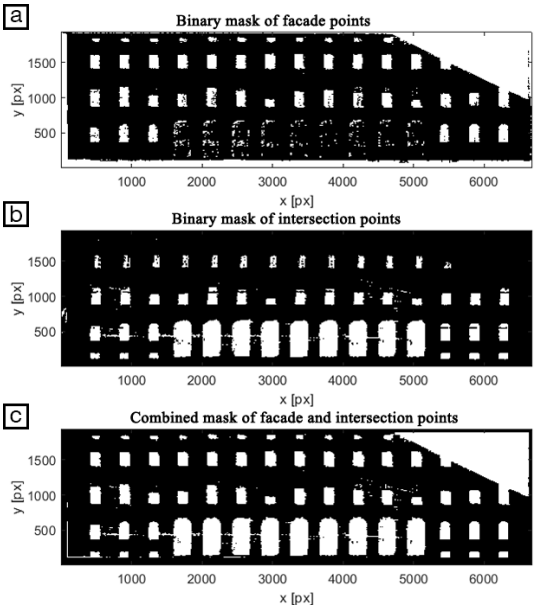
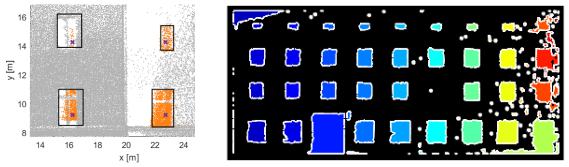


Figure 3. Fusion of the binary images. (a) facade image. Pixels without points are white. (b) Intersection image. Pixels with dots are white (c) Fused image.

points was too low here.



(a) (b)

Figure 4. Results of window extraction. (a) Procedure based on fig. 2c with contour search based on search points. On the right, two windows correctly detected only by intersection points (orange) due to missing facade points (grey). The crosses mark the search positions determined by the cross correlations. On the left, for comparison, the situation in the well detected area. (b) Procedure based on Fig. 2e/f with contour search in the complete binary image. The contours are shown in white, the colours indicate the different clusters.

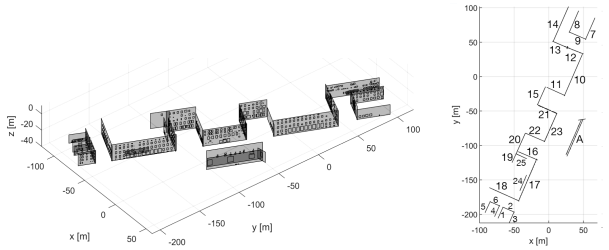
If there are not enough intersection points, no search positions can be determined as a first approximation for window positions. In these cases, as in (Schneider and Coors, 2018), contour tracking is performed over the entire image (Fig. 2e/f). The contour search proceeds as in the case already described, only with a minimum side length of 50 cm. However, there are now significantly more false detections (Fig. 4b). These clusters form the result of the first search. To remove wrong matches, the second step is now to search for a regular grid in the clusters. For this purpose, a new binary image is created that contains only the clusters. Then, as described above, the cross-correlation for horizontal and vertical edges is run over the image and two correlation curves in x- and y-direction are obtained. Based on the search points derived from this, contours are now searched again and the smallest enclosing rectangles are determined.

Finally, the rectangles found are transferred from pixel coordinates of the binary images to texture coordinates of the facades and from there to 3D model coordinates of the building model.

3. RESULTS

The presented method is evaluated using an MLS test dataset (Zhu et al., 2020). The minimum voxel size is assumed to be 0.2 m, the threshold for the standard deviations σ to be labelled as plane is set to 0.3 m, and the threshold for the deviation from the normal direction of the facade is 20°. The minimum distance between two facades is given as 2.5 m. A voxel is selected as a starting voxel if its standard deviation σ is smaller than 0.25 m. The two distances δ_1 and δ_2 for the determination of the interior points are set to 15 cm and 50 cm, respectively. By choosing these values, the accuracy of the 3d coordinates of the points is taken into account accordingly, considering the measurement accuracy and the viewing angles, as well as minor unevenness of the facade.

An overview of the reconstructed facades and windows is shown in Figure 5. All facades could be recognized. Three double facades can be seen. Facade A consists of a facade and the scaffolding in front of it. facades 16 and 24 are the front and the back facade of a balcony. facades 17 and 25 are the front and rear facades of an arcade. facade 14 spans the street and contains a passageway that could not be identified here. The textures and extracted windows of facades 14, 16, and 17 are shown again in Figure 6a. The areas where the recessed facades at 16 and 17 are located are outlined in red, as is the area of facade 14 that is an open thoroughfare.



(a) (b)

Figure 5. Results of facades and windows extraction. (a) View on the whole scene. (b) Top view of the facades. Two facade planes can be seen at A. These are the facade and a scaffolding in front of it. Also visible are facades 24 and 25, which are recessed facades of a balcony (24) and an arcade entrance (25). facade 14 contains a passageway that cannot be detected with the method presented here.

In contrast, Figure 6b shows four textures from different facades and table 1 shows the evaluation of the window detection for these four facades and the overall test scene. Texture 8 has some windows that are only visible in facade points. Texture 4 has quite a few missing windows due to poor coverage in the point cloud. Texture 10 shows a very good reconstruction of the windows except for the unsampled upper right corner. With texture 23 the detection does not work very well. Here the facade consists of several levels, the windows are partly in deep niches and the number and shape of the windows varies

from floor to floor. Over the whole scene, a completeness of 86% and correctness of 87% is reached.

num	GT	TP	FP	FN	COM	COR
4	6	5	3	1	66.7%	83.3%
8	35	34	0	1	100%	97.1%
10	60	60	0	0	100%	100%
23	59	41	7	20	85.7%	69.5%
total	702	605	88	101	86.2%	87.6%

Table 1. Results for the selected facades of figure 4. num: number of the facade, GT: Ground Truth, TP: true positive, FP: false positive, FN: false negative, COM: completeness, COR: correctness.

The correct shape of the windows is compared with a reference data set (tab. 2). The deviations in width and height of the windows are in the range of a few centimetres to a few decimetres. The height estimation for the first to the third floor is heavily underestimated by half a meter whereas it fits for the two upper floors with a few centimetres. The width in contrast shows a good accuracy of a few centimetres for the second to the fifth floor.

floor	height error		width error	
floor	mean [m]	σ [m]	mean [m]	σ [m]
1	-0.48	0.02	-0.20	0.20
2+3	-0.78	0.55	0.05	0.07
4	-0.02	0.07	0.05	0.03
5	-0.02	0.11	-0.08	0.18

Table 2. Evaluation of the window size. Height is underestimated for floor one to 3.

4. DISCUSSION

The most common cause for non-detections is poor coverage of the environment with only very few points. Other typical problem cases are small basement windows close to the ground and barred windows with few to no intersections (due to poor detection). The false alarm rate is just over 10 %. The areas falsely classified as windows are very small in most cases. However, for a few facades there is a particular accumulation of false detections, namely mainly when there are many incorrectly set intersections, which occurs mainly at facade edges or when there are plane jumps in facades.

The shape reconstruction shows a higher accuracy than (Tuttas and Stilla, 2013) which is obvious due to the denser point cloud. The first floor windows show an underestimation in the height. As these windows start directly from the ground level, the foot points are often occluded and thus the windows are not fully seen by the laser scanner. For the second and third floor, the reason for the underestimation is different. Both floor had a couple of windows with shutters partially closed. As these shutter areas are violating our window detection rules, they are seen as part of the facade and the open window area is smaller. The quite constant accuracy in the width indicates the real accuracy in the window extraction. Here, the windows are reconstructed with the correct width in the range of a few centimetres.

There is an outlier at the end of the fourth row, which was reconstructed only by the intersections. Removing this from the row gives a homogeneous result with a standard deviation of the height and width difference of only 7 and 3 cm, respectively. There is also only a very small deviation of 2 or 5 cm from the reference. The situation is similar for row 5, although the standard deviation is somewhat higher here. The reason for this is probably the smaller size of the windows and the more acute viewing angle of the laser scanner. This also shows that the reconstruction is done with a resolution of 1 cm, which is due to the pixel size of the binary image. due to the pixel size of the binary image. Overall, from the non-systematically error-prone values, a weak tendency to slightly too large reconstruction can be seen.

5. OUTLOOK

Overall, the results are satisfactory. Compared to (Tuttas and Stilla, 2013), the detection rate is about 10% better due to the significantly higher point density and the false alarm rate is somewhat lower. The deviations in the sizes of the windows are also lower according to the point density. In (Tuttas and Stilla, 2013) the accuracy of the dimensions of the windows was obtained from the reconstruction. This was taken into account using a probability density function. This could also be considered in continuation of the work presented here to further reduce the inaccuracies. In addition to incorrectly positioned search points, incorrectly set intersection points are a common source of error in window extraction, as mentioned earlier. These usually occur at the vertical edges of the facade or due to facade areas that are offset to the rear. Checking all points for lines of sight with all estimated planes would reduce this problem, but means a much higher computational effort. Also, the model assumption about the shape of windows could be extended from rectangular windows to other shapes. These location and shape parameters could be trained (Schmittwilken and Plumer, 2010) or a library of templates could be created and selected via a Monte Carlo procedure (Nguatem et al., 2014).

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Marcus Hebel, Joachim Gehrung and Björn Borgmann of Fraunhofer IOSB, Ettlingen, Germany, for providing the data of the MLS measurement campaign, Dr. Sebastian Tuttas of 3d Mapping Solutions, Germany, for providing his code for ALS point cloud window extraction, and Dr. Yusheng Xu and Jingwei Zhu of Technical University of Munich, Germany, for preparing and labelling the dataset.

REFERENCES

Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R., Gedikli, S., Vincze, M., 2012. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. IEEE Robotics & Automation Magazine, 19(3), 80–91.

Aljumaily, H., Laefer, D. F., Cuadra, D., 2017. Urban point cloud mining based on density clustering and mapreduce. Journal of Computing in Civil Engineering, 31(5), 04017021.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis & Machine Intelligence, 26(9), 1124–1137.

Brenner, C., 2005. Building reconstruction from images and laser scanning. nternational Journal of Applied Earth Observation and Geoinformation, 6(3-4), 187–198.

Castillo, E., Liang, J., Zhao, H., 2013. Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates. Innovations for Shape Analysis, Springer, 283–299.

Dong, Z., Yang, B., Hu, P., Scherer, S., 2018. An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 137, 112–133.

Filin, S., Pfeifer, N., 2006. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. ISPRS Journal of Photogrammetry and Remote Sensing, 60(2), 71–80.

Golovinskiy, A., Funk, T., 2009. Min-cut based segmentation of point clouds. IEEE International Conference on Computer Vision Workshops, ICCV Workshops, IEEE, IEEE, 39–46.

Gonzalez, R., Woods, R., Eddins, S., 2004. Digital Image Processing Using MATLAB. Pearson Education.

Green, W. R., Grobler, H., 2015. Normal distribution transform graph-based point cloud segmentation. Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), IEEE, IEEE, 54–59.

Hackel, T., Wegner, J. D., Schindler, K., 2016. Contour detection in unstructured 3d point clouds. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 1610–1618.

Hong, D., Yokoya, N., Chanussot, J., Xu, J., Zhu, X. X., 2019. Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction. ISPRS Journal of Photogrammetry and Remote Sensing, 158, 35–49.

Kim, K., Shan, J., 2011. Building roof modeling from airborne laser scanning data based on level set approach. ISPRS Journal of Photogrammetry and Remote Sensing, 66(4), 484–497.

Lari, Z., Habib, A., 2014. An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 93, 192–212.

Lee, I., Schenk, T., 2001. Autonomous extraction of planar surfaces from airborne laser scanning data. Proceedings of ASPRS Annual Conference. Presented at the ASPRS 2001 Annual Conference, St. Louis, Missouri, USA.

Li, M., Rottensteiner, F., Heipke, C., 2019. Modelling of buildings from aerial LiDAR point clouds using TINs and label maps. ISPRS Journal of Photogrammetry and Remote Sensing, 154, 127–138.

Michaelsen, E., Iwaszczuk, D., Sirmacek, B., Hoegner, L., Stilla, U., 2012. Gestalt grouping on façade textures from IR image sequences: Comparing different production systems. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 39(B3), 303-308.

Nguatem, W., Drauschke, M., Mayer, H., 2014. Localization of Windows and Doors in 3d Point Clouds of Facades. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3, 87-94.

Nobrega, R., O'Hara, C., 2006. Segmentation and object extraction from anisotropic diffusion filtered LiDAR intensity data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(4/C42).

Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 64(6), 575–584.

Rabbani, T., Van Den Heuvel, F., Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(5), 248–253.

Reznik, S., Mayer, H., 2008. Implicit Shape Models, Self-Diagnosis, and Model Selection for 3D Facade Interpretation. PFG Photogrammetrie – Fernerkundung – Geoinformation, 2008(3), 187-196.

Rusu, R. B., Blodow, N., Marton, Z. C., Beetz, M., 2009. Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 1–6.

Rutzinger, M., Höfle, B., Oude Elberink, S., Vosselman, G., 2011. Feasibility of facade footprint extraction from mobile laser scanning data. Photogrammetrie - Fernerkundung - Geoinformation, 2011(3), 97–107.

Schmittwilken, J., Plumer, L., 2010. Model-based reconstruction and classification of facade parts in 3D point clouds. The International Archives of the Photogrammetry, Remote Sensing and Geospatial Information Science, 38(3A), 269-274.

Schneider, S., Coors, V., 2018. Automatische extraktion von fenstern in 3d punktwolken mittels einer hierarchischen methode.

Schuster, H.-F., 2004. Segmentation of lidar data using the tensor voting framework. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 35(B3), 1073–1078.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. Departmental Papers (CIS), 107.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P. et al., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. Proceedings of the ISPRS Workshop on Laser Scanning, 36, 407–412.

Tuttas, S., Stilla, U., 2013. Reconstruction of facades in point clouds from multi-aspect oblique als. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3, W3.

Urbančič, T., Fras, M., Stopar, B., Božo, K., 2014. The Influence of the Input Parameters Selection on the RANSAC Results. International Journal of Simulation Modelling, 13, 159-170.

Vo, A.-V., Truong-Hong, L., Laefer, D. F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing, 104, 88–100.

Vosselman, G., Coenen, M., Rottensteiner, F., 2017. Contextual segment-based classification of airborne laser scanner data. ISPRS Journal of Photogrammetry and Remote Sensing, 128, 354–371.

Widyaningrum, E., Gorte, B., Lindenbergh, R., 2019. Automatic building outline extraction from ALS point clouds by ordered points aided hough transform. Remote Sensing, 11(14), 1727.

Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., Liu, H., 2013. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. Remote Sensing, 5(2), 584–611.

Xu, Y., Hoegner, L., Tuttas, S., Stilla, U., 2017. Voxel- and graph-based point cloud segmentation of 3d scenes using perceptual grouping laws. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1, 43–50.

Yao, W., Hinz, S., Stilla, U., 2009. Object extraction based on 3d-segmentation of lidar data by combining mean shift with normalized cuts: two examples from urban areas. Urban Remote Sensing Event, 2009 Joint, IEEE, 1–6.

Zhu, J., Gehrung, J., Huang, R., Borgmann, B., Sun, Z., Hoegner, L., Hebel, M., Xu, Y., Stilla, U., 2020. TUM-MLS-2016: An annotated mobile LiDAR dataset of the tum city campus for semantic point cloud interpretation in urban areas. Remote Sensing, 12(11), 1875.

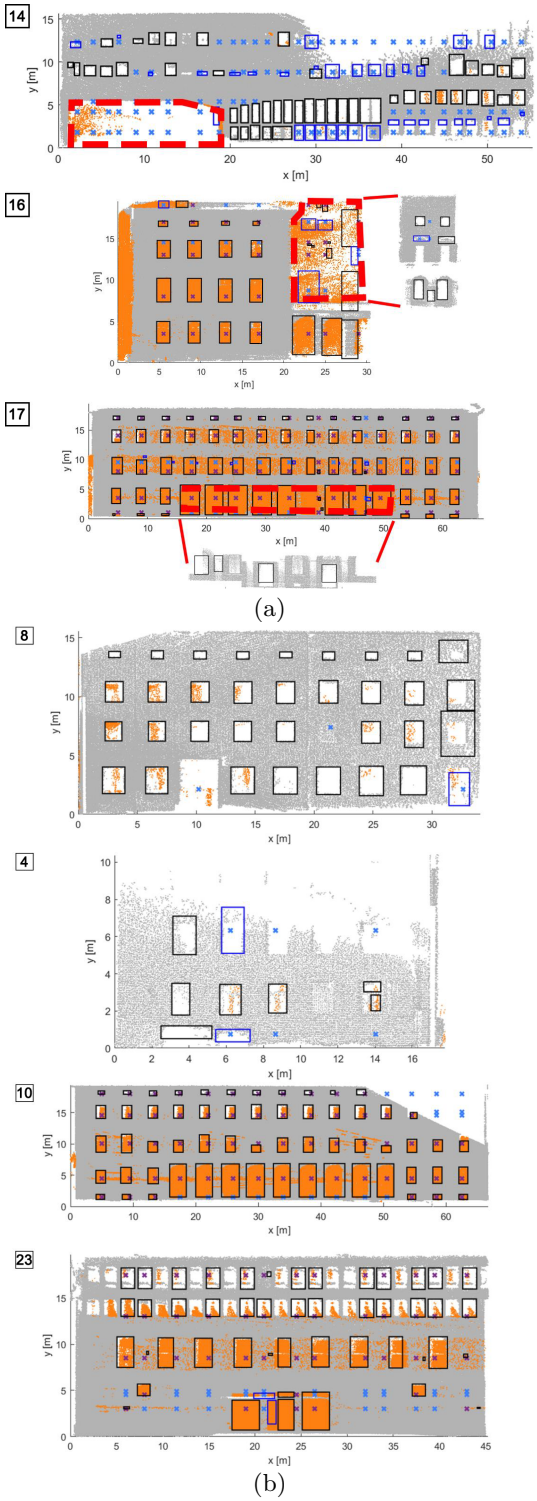


Figure 6. Results of window extraction. Gray dots stand for the facade points, orange ones for intersection points. The purple crosses are the search positions of the first search phase (if with position search), the blue ones mark those of the new detection (or the completion of the grid). The windows found in this process are shown as blue rectangles, those of the first pass in black. (a) Areas to be extracted (outlined in red) in facades 14, 16 and 17. The extracted rectangles each encompass the range of values shown. For 16 and 17, the respective facade behind them (24 and 25, respectively) is also shown. (b) facades 8, 4, 10 and 23.