Automatic Reconstruction of Roof Overhangs for 3D City Models

Steffen Goebbels[a] and Regina Pohle-Fröhlich
Institute for Pattern Recognition, Faculty of Electrical Engineering and Computer Science, Niederrhein University of Applied Sciences, Reinarzstr. 49, 47805 Krefeld, Germany

Keywords: Building Reconstruction, CityGML, Oblique Aerial Images, Airborne Laser Scanning Point Clouds.

Abstract: Most current 3D city models, created automatically from cadastral and remote sensing data and represented in CityGML, do not include roof overhangs, although these overhangs are very characteristic for the appearance of buildings. This paper describes an algorithm that procedurally adds such overhangs. When a CityGML model is textured, the size of the overhangs is determined by recognizing overhangs in facade textures. In this case, the method only needs an already existing model in CityGML representation. Alternatively, if an additional point cloud (e.g., from airborne laser scanning) is available, this cloud can be utilized to calculate the overhang sizes. We compare the results of both methods.

1 INTRODUCTION

Most automatically generated semantic 3D city models have a level of detail (LoD) that provides planar roof and wall surfaces that form a 3D solid, but do not include facade details such as windows, doors, or roof overhangs. The models, which are usually available in the CityGML description language (Gröger et al., 2012; Kutzner et al., 2020) or in the CityJSON[1] format in LoD 1 (flat roofs only) or LoD 2 (more realistic roofs, mostly without details like dormers), are truncated at the cadastral footprint (that is included in CityGML models as the ground surface) so that overhanging roof surfaces are cut off.

For example, roof overhangs expand the roof area but also lead to shadowing. Both effects can be considered as input for solar potential analysis of roofs and facades (see (Biljecki et al., 2015) for applications of city models). Switzerland already uses a nationwide 3D model with roof overhangs[2], but manual interaction was required to obtain this model.

In this paper we describe an algorithm[3] that automatically adds overhangs to building and building part objects in CityGML models. Thus, LoD 3 elements are added to LoD 2 building representations.

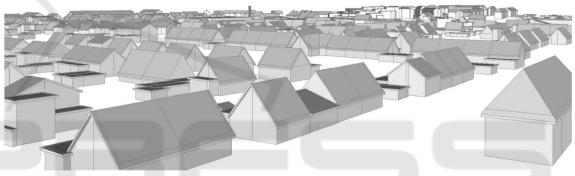


Figure 1: Overhangs were added to a model provided by Geobasis NRW (Oestereich, 2014).

For use in energy modeling of buildings, a similar tool is presented in (Malhotra et al., 2021) that can be used to extend a CityGML model from LoD 1 to LoD 2 by replacing flat roofs with predefined roof shapes that correspond to cadastral information. However, this tool does not deal with roof overhangs.

We differentiate between overhangs inside and outside the cadastral footprint. Overhangs in the interior of the footprint occur if a roof has step edges. But then the use of airborne laser scanning data in creating the models (cf. (Wang et al., 2018)) leads to a situation where no distinction is made between overhangs and roof surfaces of the closed building hull. Thus, walls representing step edges may be erroneously placed at the end of overhangs. The position of walls is not corrected by our algorithm. Instead, it generates small overhangs inside the cadastral footprint for appearance and larger, realistic overhangs along the outside of the footprint, see Figure 1. If the model has textured walls, the size of the latter overhangs is determined using image processing methods on facade images showing the overhangs. Then, the new structures are calculated using only a given city model without any additional data.

[a] https://orcid.org/0000-0003-4313-9101

[1] https://cityjson.org/specs/ (all websites accessed: September 11, 2022)

[2] https://www.swisstopo.admin.ch/de/geodata/landscape/buildings3d2.html

[3] C++ source code is available at https://github.com/SteffenGoebbels/citygml_overhangs

Goebbels, S. and Pohle-Fröhlich, R.
Automatic Reconstruction of Roof Overhangs for 3D City Models.
DOI: 10.5220/0011604200003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 1: GRAPP*, pages 145-152
ISBN: 978-989-758-634-7; ISSN: 2184-4321

With additional information, one could find other ways to estimate the size of overhangs.

A true orthophoto can be used to determine building footprints, e.g., by applying deep neural networks as in (Cheng et al., 2019) or (Chen et al., 2020). These footprints can be compared with cadastral footprints. Differences indicate roof overhangs.

If a sufficiently dense point cloud is available (e.g., from airborne laser scanning or photogrammetry) the size can be derived from building footprints (e.g. taken from fitted wall planes) and a digital surface model based on this point cloud, see (Dahlke et al., 2015; Frommholz et al., 2017). A zero crossing of the second difference of surface points in the direction of a roof plane gradient indicates a discontinuity and thus the end of the plane including its overhang. In our case, roof planes are given by the model. Therefore, a straightforward approach is to look for inlier points of these planes that are outside the roof facet, cf. (Yan et al., 2012; Chen et al., 2014; Goebbels and Pohle-Fröhlich, 2020) for RANSAC-based roof reconstruction. We use this method in addition or as an alternative to texture-based sizing for overhangs, see Section 5.2.

All these approaches combine data from different sources, which usually do not match perfectly and lead to inevitable errors.

Our algorithm consists of following steps:

- The first task is to select the roof edges where there may be roof overhangs. This step is described in the next section.
- Then, for each edge candidate, the size of a potentially adjacent overhang is either calculated or set as a default value. To calculate the size, two methods based on CityGML textures and one method based on additional airborne laser scanning point clouds are used, see Section 5. One texture based method utilizes dominant colors, the other approach evaluates an edge image.
- Now, the overhangs can be constructed in a generic way, see Section 3.
- When dealing with textured city models, overhangs need to be colored consistently with the textures. We apply a median cut algorithm. Since the colors are also used in one of the methods for calculating the overhang size, this is described in Section 4 prior to the size estimate approaches in Section 5.1.

Finally, the results are discussed in Section 6.

2 CANDIDATE EDGES WHERE OVERHANGS MAY BE PLACED

For each building or building part, we analyze a directed graph (V,E) with edges $E \subset V \times V$ and vertices $v = (v.x, v.y) \in V \subset \mathbb{R}^2$ which are obtained from the 3D vertices of all roof polygons by omitting their height coordinate. So we project the boundary edges of each roof facet onto a horizontal plane. If a roof contains step edges, multiple 3D vertices could be mapped to one vertex in V. The CityGML representation ensures that each roof polygon is oriented counter-clockwise when viewed from the outside of the building. Thus, the edges of the 3D polygons are oriented. We connect u and $v \in V$ with an arc (u,v) if and only if there exist corresponding 3D vertices $\tilde{u}$ and $\tilde{v} \in \mathbb{R}^3$ such that a polygon edge connects $\tilde{u}$ with $\tilde{v}$. A consistent 3D solid given, each arc (u,v) belongs to a unique polygon edge denoted by $(L(u,v),R(u,v)) \in \mathbb{R}^3 \times \mathbb{R}^3$. We use the notion $L(u,v) = (L(u,v).x, L(u,v).y, L(u,v).z)$ to access coordinates.

Let $(u,v) \in E$ be such that also $(v,u) \in E$. Then the arc (u,v) is not a candidate for attaching a roof overhang if $L(u,v).z \leq R(v,u).z$ or $R(u,v).z \leq L(v,u).z$, because then a roof segment of (partially) equal or greater height is attached. In particular, this excludes all ridge lines.

We also check whether the arcs are adjacent to walls of other buildings. In this case, we also do not attach overhangs, as they could intersect with neighboring buildings. Thus, for each candidate arc (u,v), we look for parallel 2D footprint edges with the same orientation such that the straight lines through the arc and footprint edge are closer than a threshold (0.4 m). The footprint polygons are defined as GroundSurface objects in CityGML and have normals pointing down. Therefore, the selected edges belong to different buildings or building parts and could be adjacent to the arc. To check this, we project arc (u,v) onto the line through the footprint edge. If this edge intersects with the projected arc in its interior, (u,v) may not qualify for attaching a roof overhang. To make a final decision, we also consider wall polygons belonging to the footprint edge and determine a maximum z-coordinate. If $L(u,v).z$ or $R(u,v).z$ does not exceed this maximum coordinate, a roof overhang for (u,v) will intersect with the neighboring building and is therefore omitted. We only deal with parallel footprint edges because this is the typical case of terraced house development.

3 CONSTRUCTION OF OVERHANGS

3.1 Procedural Extension of Roof Facets

The algorithm iterates through all arcs $(u,v) \in E$ of positive length that are candidates for attaching an overhang. The overhang shape is determined by incoming arcs (w,u) of u and outgoing arcs (v,w) of v.

If there exists exactly one incoming arc (w,u) such that $R(w,u).z = L(u,v).z$, and the arc is selected for attaching an overhang, then the two overhangs are connected as described in Section 3.2, creating the vertices of the left side of the overhang in 2D. This is also done at the second vertex: If there exists exactly one outgoing arc (v,w) with $L(v,w).z = R(u,v).z$ that also is selected to attach an overhang, then these two overhangs are also connected as described in Section 3.2, creating vertices of the right side of the overhang in 2D.

If there is no need to connect adjacent roof overhangs on the left or right or both sides, we try to extend the roof orthogonally to (u,v) on this side or these sides. The normal $\vec{n} := (v.y - u.y, -v.x + u.x)/|v-u|$ points in the direction in which the overhang must extend. In a straightforward situation, the 2D-points u, $u + d\vec{n}$ and/or $v + d\vec{n}$, v can be used to calculate the vertices of the left and/or right side of the overhang. The parameter $d > 0$ describes the size of the overhang. Section 5 explains how to estimate this size. However, the use of an orthogonal extension may result in overlaps with neighboring structures belonging to the incoming arcs of u or the outgoing arcs of v, see Figure 2. To avoid this, we compute adapted shift directions for both u and v. Let E_{in} be the set of incoming arcs (w,u) of u for which $R(w,u).z \geq L(u,v).z$ and for which both angles between (u,w) and (u,v) as well as (u,w) and $\vec{n}$ are less than 90°. If the set is non-empty and if we do not connect roof overhangs at u, we use $\vec{n}_u := -\vec{d}_u/|\vec{d}_u|$ instead of $\vec{n}$ to obtain the overhang vertex $u + \vec{n}_u \cdot d/\sqrt{1-\cos^2(\alpha)}$, where $\vec{d}_u \in E_{\text{in}}$ such that the angle α between $-\vec{d}_u$ and (u,v) is smallest. By considering outgoing arcs of v, we proceed similarly to replace $v + d\vec{n}$ by $v + \vec{n}_v \cdot d/\sqrt{1-\cos^2(\beta)}$ for a minimum angle β, if necessary.

The construction for connecting two overhangs in the following Section 3.2 also involves a projection in the direction $\vec{n}$, which may need to be replaced by a projection in the direction of the vectors $\vec{d}_u$ or $\vec{d}_v$.

So far, the 2D coordinates of the vertices of the overhang have been determined, either by the algorithm in Section 3.2 or by the above considerations.

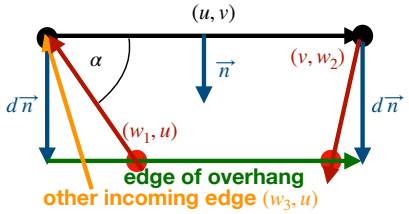


Figure 2: An orthogonal attachment of an overhang to the edge (u,v) might intersect with walls belonging to incoming edge (w_1,u) and outgoing edge (v,w_2). In that case, the red vertices have to be used in the construction of the overhang.

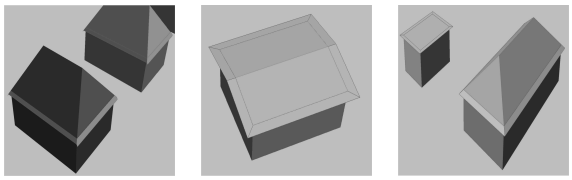


Figure 3: Equations (1)–(5) define construction points P_1, P_2, and P_3. Whereas in the left image projection points $P_3 \neq P_2$ are used, $P_3 = P_2$ holds at the corners of the other buildings.

Finally, we need to compute z-coordinates (height-values) for the new vertices by considering the normal $\vec{n}_P$ of the roof polygon belonging to (u,v): Using the inner product, the z-coordinate of vertex (x,y) can be computed by solving the equation $L(u,v) \cdot \vec{n}_P = (x,y,z) \cdot \vec{n}_P$.

We merge overhang polygons, belonging to the same roof polygon and having a common edge. However, we avoid inner polygons by using edges that are traversed in both directions. Such inner polygons describe openings that can occur if a roof consists of a single polygon and overhangs are attached to all edges. The merged polygons are extruded into 3D by adding "wall" polygons with a small default height and a ceiling polygon. These 3D solids are added to the CityGML model as building installation objects.

3.2 Connecting Roof Overhangs

In this section we discuss the scenario that overhangs have to be attached to two roof edges that are incident to the same vertex u in 3D, see examples in Figure 3. For simplicity, we describe the calculation for u replaced by $(0,0)$ so that the point u has to be added to the resulting 2D points, see Figure 4.

We shift the 2D roof edge indicated by $\vec{a}$ in Figure 4 belonging to a roof polygon with 3D normal $\vec{n}_1$ by adding a 2D vector $\vec{d}_1$ that is orthogonal to $\vec{a}$ and pointing outwards. Thus, $\vec{d}_1 = s(\vec{a}.y, -\vec{a}.x)$ for some $s > 0$. In the same way, we deal with the edge indicated by $\vec{b}$ of the second polygon and shift it with $\vec{d}_2$ orthogonal to $\vec{b}$ and pointing outward.

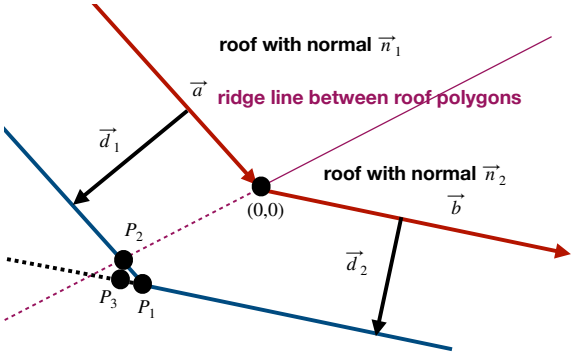


Figure 4: Connecting two overhangs.

We obtain the intersection point P_1 of the boundary of the overhang polygon via

$$P_1 = (0,0) + \vec{d}_1 + s_1\vec{a} = (0,0) + \vec{d}_2 + s_2\vec{b}, \quad (1)$$

i.e., by solving the two corresponding linear equations with Cramer's rule. If there is no unique intersection point ($\vec{a}$ and $\vec{b}$ are parallel), then the two overhang segments are treated separately at this vertex if $\vec{d}_1 \neq \vec{d}_2$. We do not connect them. If the two vectors are equal, we set $s_1 = s_2 = 0$ and can also use point P_1. If $|P_1| > |\vec{d}_1| + |\vec{d}_2|$ then the angle between $\vec{a}$ and $\vec{b}$ is close to 180°, and the segments are also treated separately.

If $\vec{n}_1 = \vec{n}_2$, the edges belong to the same roof facet. In this case, the overhangs meet at the edge between P_1 and $(0,0)$. Otherwise, there exists an intersection line between the two roof facets belonging to the two roof edges. Then we also need to find the intersection P_2 with one of the two overhang edges. Using the cross product, the direction of the ridge line can be described with 2D vector

$$\vec{r} = ((\vec{n}_1 \times \vec{n}_1).x, (\vec{n}_1 \times \vec{n}_1).y). \quad (2)$$

Then either

$$P_2 = s_3\vec{r} = P_1 + s_4\vec{a} \quad (3)$$

or

$$P_2 = s_5\vec{r} = P_1 + s_6\vec{b}. \quad (4)$$

Both equations can be solved with Cramer's rule. If a solution does not exist, we only consider the other equation. If both solutions exist, we select one by comparing s_4 and s_6. If $s_6 \geq 0$ (or equivalently, $s_4 \geq 0$), we compute P_2 with (4). Otherwise, P_2 is given by (3) as shown in Figure 4.

To obtain the usual appearance of overhangs, we add an additional point P_3:

If P_2 results from (3), we project P_2 orthogonally onto the line $P_1 + s\vec{b}$:

$$P_3 = P_2 + s_7\vec{d}_2 = P_1 + s_8\vec{b}. \quad (5)$$

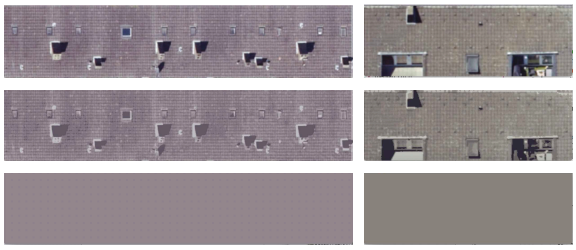


Figure 5: From top to bottom: original roof texture, texture reduced to eight colors, and dominant color.

Then the points $(0,0)$, P_2, and P_3 are vertices of the left side of the overhang for edge $\vec{b}$, and the points P_2 and $(0,0)$ are vertices of the right side of the overhang for edge $\vec{a}$.

If P_2 is calculated with (4), we project P_2 onto the line $P_1 + s\vec{a}$:

$$P_3 = P_2 + s_7\vec{d}_1 = P_1 + s_8\vec{a}. \quad (6)$$

Then the points P_3, P_2 and $(0,0)$ are vertices of the right side of the overhang for the edge $\vec{a}$, and the points $(0,0)$ and P_2 are vertices of the left side of the overhang for the edge $\vec{b}$.

The orthogonal projection leading to P_3 could cause an intersection with neighboring structures. We replace $\vec{d}_1$ or $\vec{d}_2$ similar to $\vec{n}$ in Section 3.1 if necessary.

4 COLOR OF OVERHANGS

For textured city models, the overhangs may not be fully visible in the textures provided. Therefore, we do not equip overhangs with textures from the model, but with the dominant color of the corresponding roof facet. The dominant color is determined using the median cut algorithm (Heckbert, 1982). This algorithm clusters color values iteratively by comparing values of the channel with the largest range to their median. Finally, the channel-wise arithmetic mean of all colors in a cluster is used to represent their replacement color. Applied to a texture image, the algorithm can reduce the number of colors to very few, say eight. Then the color with the largest number of pixels is chosen, see Figures 5 and 6.

To obtain homogeneous looking roof planes, one could also remove all roof textures and use the corresponding colors instead. This also removes perspective-distorted images of structures such as dormers, which are not represented in LoD 2 models.

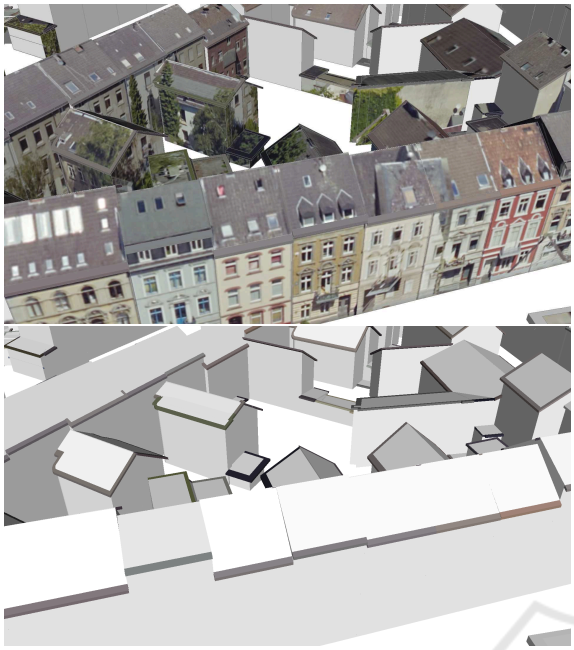


Figure 6: Overhangs painted with the main color of the corresponding roof facet.

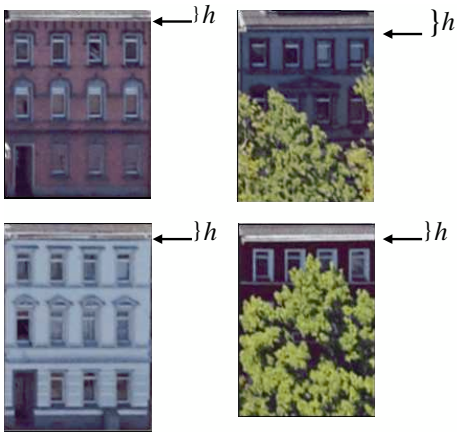


Figure 7: Overhangs with height h are visible in facade textures.

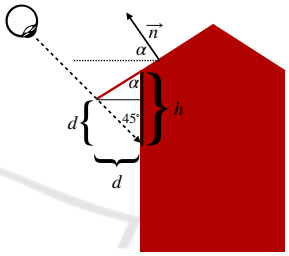


Figure 8: Estimating the size d of an overhang with normal $\vec{n}$ from a texture in which the roof section has height h, see (7).

5 SIZE OF OVERHANGS

5.1 Size Derived from Textures

For each roof edge for which we have to construct an overhang, we need to estimate the overhang size d, as used in Section 3.1, or the sizes $|\vec{d}_1|$ and $|\vec{d}_2|$, as used in Section 3.2. To this end, we utilize textures from CityGML models. Many communities maintain such textured models. While the camera position and parameters are precisely known for oblique aerial images, these data are not available for the textures. We therefore assume – and this is not far-fetched – that facade images are generated from oblique aerial images that are taken at a 45° angle. Then roof overhangs are visible in the top of facade textures[4], see Figure 7. Let $h > 0$ be the vertical extent of this texture region, measured in meters. We derive the size of the overhang using h. If $h > 2$, i.e, the size exceeds typical values, we assume that the size results from artifacts and that the roof facet is not clearly visible in the facade image. Then we replace h by 2 for computing the box-plots but change the value to zero for reconstructing overhangs. If the roof edge belongs to a flat roof, then the size is $d = h$. Otherwise, let $\vec{n}$ be the normal of the corresponding roof polygon, then, see Figure 8, $\tan(\alpha) = d/(h-d)$ and

$$d = h \cdot \frac{\tan(\alpha)}{1+\tan(\alpha)}, \quad \tan(\alpha) = \frac{\vec{n}.z}{|(\vec{n}.x, \vec{n}.y)|}. \qquad (7)$$

To obtain h, we create an image based on texture coordinates simulating a UV-mapping by applying either an affine transformation based on three polygon vertices or by a perspective transform based on four polygon vertices with the OpenCV library. Then we shift, rotate, and cut off the image such that the roof edge corresponds with the top boundary of the image, see facades in Figure 7 and 9.

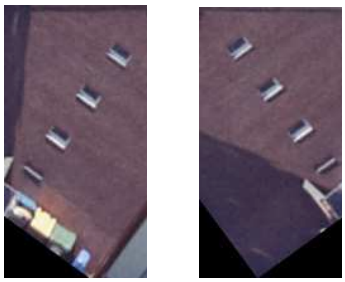


Figure 9: Rotated wall texture: The upper boundary of each image coincides with the roof edge to be examined.

[4]We textured a city model with oblique areal images that were provided by the cadastral office of the city of Krefeld.

Then, h can be determined using image processing methods. We compare two approaches, one based on an edge image and one based on homogeneous color regions.

After creating an edge image, the edge pixels per line are summed to get a histogram function. Then, the function is normalized so that the smallest value is 0 and the largest value is one. We seek for the first value that exceeds the experimentally determined threshold of 0.75, see Figure 11. Considering the image resolution, the corresponding line index defines h. Since it has been found that dominant edges often appear at the bottom of facades, we restrict the computation to the upper half of the texture.

The second approach is based on color regions. With the median cut algorithm, cf. Section 4, we determine both the dominant color of the facade and the dominant color of the corresponding roof facet. Based on the transformed texture image with n rows, a histogram function is computed. This function maps the line index to the number of those pixels in the image line for which the l_2-distance of the R-G-B color to the roof color is smaller than a quarter of the distance to the facade color. Then the histogram function is scaled so that its minimum is zero and its maximum is one. We denote this function by f. In an ideal world, where the pixels can be uniquely assigned to either the roof or the facade such that the corresponding regions are divided horizontally by a line, f is a function of type g_l with

$$g_l(k) := \begin{cases} 1 & : \quad k \in \{0,1,\ldots,l\} \\ 0 & : \quad k \in \{l+1,\ldots,n-1\}. \end{cases} \tag{8}$$

Therefore, we compute the smallest line index l for which $\sum_{k=0}^{\lfloor n/2 \rfloor - 1} (f(k) - g_l(k))^2$ is minimal. Again, we focus on the upper half of the image and obtain h from this smallest line index as well by considering the image resolution. Figure 12 demonstrates the concept but also shows that shadows of overhangs can extend the roof area.

For the automatic reconstruction of overhangs, the minimum size of both approaches can be used. Roof and wall textures can come from different aerial or oblique aerial images with different cameras. Therefore, it is not surprising that both approaches lead to differences that are shown in Figure 10 for the square kilometers specified by zone 32U UTM interval $[330,000;\ 331,000] \times [5,687,000;\ 5,688,000]$ (8,261 roof edges with available roof and facade textures) and the next interval $[330,000;\ 331,000] \times [5,688,000;\ 5,689,000]$ (7,733 roof edges). These square kilometers are used in other figures, too.

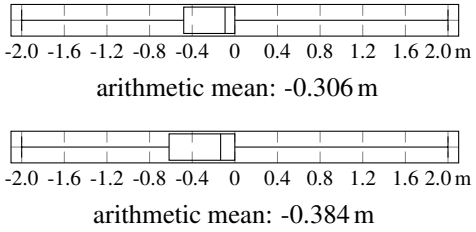


Figure 10: Differences between overhang sizes estimated based on dominant colors and overhang sizes estimated with an edge image; the box plots belong to different square kilometers.

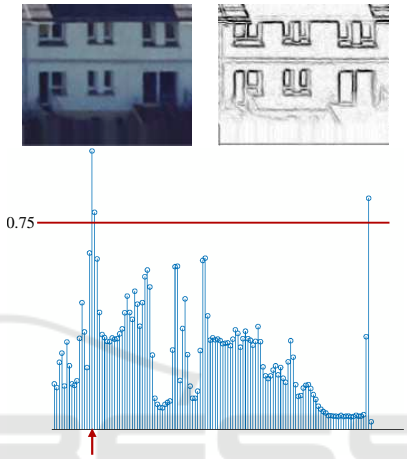


Figure 11: Detection of roof regions with edge histograms.

5.2 Size Derived from a Point Cloud

Figures 14 and 15 demonstrate that overhangs are indeed visible even in sparse airborne laser scanning point cloud data[5]. When such a point cloud is available, we also use it to compute sizes of overhangs. To this end, we organize the point clouds in a quad-tree for faster access. For each relevant roof edge, we consider a rectangular area in the x-y ground plane. One side of this rectangle is given by the middle part of the roof edge (with half the length of the edge to avoid interference with other structures at the corners). The rectangle extends two meters beyond the roof facet.

Figure 12: The blue lines separate the roof and facade based on an edge image, the red lines separate the dominant colors of the roof and facade.

[5]Available from Geobasis NRW, https://www.bezreg-koeln.nrw.de/brk_internet/geobasis/hoehenmodelle/3d-messdaten/index.html

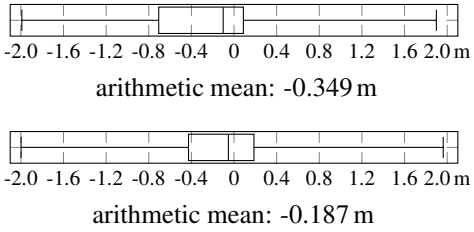


Figure 13: Quality of overhang sizes estimated from sparse airborne laser scanning point clouds: The box plots show the distribution of differences between σ and γ (see Section 5.2) for the same two square kilometers as in Figure 10.

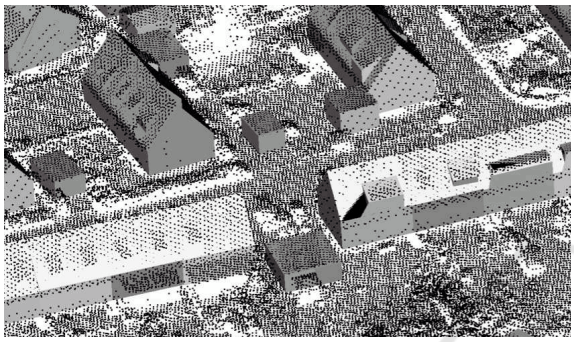


Figure 14: Matching of an airborne laser scanning point cloud with an LoD 2 city model: Visible black points of the cloud are not occluded by the model.

Larger overhangs occur only very rarely and pose the risk of overlapping with other buildings. Then all laser scanning points within this rectangle are determined and their distance to the boundless roof plane belonging to the edge is calculated by using the Hesse normal form with the given plane normal. If a distance is below one meter, the point is considered as a plane inlier. Let γ be the largest distance of all inliers to the edge under consideration. If there are no inliers, γ is set to zero. We have an outlier point, if its distance to the plane exceeds one meter. This threshold is chosen to ignore noise and because almost all facades are higher than two meters. Outliers are ignored if they are above the plane. This is necessary to avoid false outliers, e.g., due to dormers that are

Figure 15: Roof regions shown in facade textures of an LoD 2 city model indeed correlate with point cloud points belonging to roof overhangs.

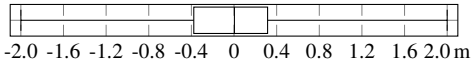
missing in the given building model. Outliers below the plane indicate the end of the roof facet. Let σ be the shortest distance of all outliers to the edge. If there are no such outliers, σ is set to the maximum overhang size of 2 m. Finally, we estimate the size d of the overhang:

$$d := \begin{cases} \frac{\gamma+\sigma}{2} & : \text{ outliers exist} \\ \gamma & : \text{ otherwise.} \end{cases} \quad (9)$$

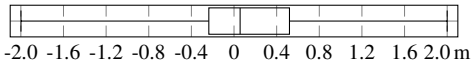
Figure 13 shows differences between γ and σ for sparse airborne laser scanning point clouds in conjunction with roof facets of a city model computed on these data and on cadastral footprints using RANSAC. One would expect $\sigma > \gamma$ with the difference being small. Inliers should be near the roof edge, and outliers should be farther away. However, the LoD 2 models have simplified roof topologies. For example, a roof facet may be separated by a dormer so that there is no overhang in front of the dormer. But the dormer may be absent from the model, leading to outliers close to the roof edge under consideration. Also, vegetation leads to false inliers.

6 RESULTS

Since we are not aware of any ground truth, we compare sizes of overhangs obtained from facade textures (included in a CityGML model and previously computed from oblique areal images with a pixel size of about $0.1\,\text{m} \times 0.1\,\text{m}$) on the one hand and from airborne laser scanning point clouds (provided by Geobasis NRW with a resolution between 5 and 10 points per square meter) on the other hand. The differences between these size estimates are shown in Figures 16 and 17. Compared to the size of the overhangs, the differences between the size estimates are substantial. However, apart from the implicit inaccuracies in the estimation methods, the reasons for this also lie in the data. Roof planes in 3D city models usually simplify real roof structures. In reality, walls have a certain thickness that they do not have in the model. Cadastral footprints do not exactly match oblique aerial image data. These inaccuracies result in texture shifts. Textures can show objects or other buildings that occlude the wall in question. We also use a point cloud from airborne laser scanning, whose relative accuracy within the cloud is very good, but the entire cloud may be slightly shifted compared to cadastral data. The point cloud is sparse, so small structures such as overhangs are barely represented.

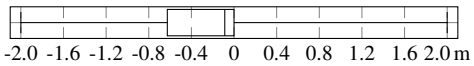


arithmetic mean: 0.006 m

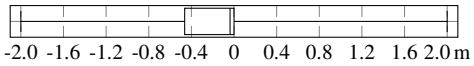


arithmetic mean: 0.148 m

Figure 16: Size estimates based on texture edges minus estimates from aerial laser scanning point clouds for two square kilometers.



arithmetic mean: -0.302 m



arithmetic mean: -0.238 m

Figure 17: Size estimates from texture colors minus estimates from aerial laser scanning point clouds for two square kilometers.

7 CONCLUSIONS

With the developed tool, roof overhangs can be added to existing CityGML models with or without using additional data, giving the models a much more realistic appearance. The tested heuristics for estimating the size of the overhangs work in principle, but with the poor-resolution image and point cloud data available to us, the estimates are comparatively inaccurate. Better quality can be expected, for example, when using higher resolution point clouds, but they are not available throughout a widespread area.

Flat roofs often do not have overhangs but a small elevated frame along their perimeter. Such frames can be added to city models in a similar way as overhangs, by extending edges into direction $-\vec{n}$, see Section 3.1, and without considering ridge lines, cf. Section 3.2. Future work may also explore estimating the size of roof overhangs using machine learning.

ACKNOWLEDGMENTS

The authors are grateful to Udo Hannok from the cadastral office of the City of Krefeld for providing oblique aerial images. The authors are also grateful for valuable comments by the anonymous reviewers.

REFERENCES

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4:2842–2889.

Chen, Q., Wang, L., Waslander, S. L., and 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.

Chen, Y., Cheng, L., Li, M., Wang, J., Tong, L., and Yang, K. (2014). Multiscale grid method for detection and reconstruction of building roofs from airborne LiDAR data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 7(10):4081–4094.

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

Dahlke, D., Linkiewicz, M., and Meissner, H. (2015). True 3D building reconstruction: Façade, roof and overhang modelling from oblique and vertical aerial imagery. *International Journal of Image and Data Fusion*, 6(4):314–329.

Frommholz, D., Linkiewicz, M., Meißner, H., and Dahlke, D. (2017). Reconstructing buildings with discontinuities and roof overhangs from oblique aerial imagery. *International Archives of Photogrammetry and Remote Sensing*, XLII-1 (W1):465–471.

Goebbels, S. and Pohle-Fröhlich, R. (2020). RANSAC for aligned planes with application to roof plane detection in point clouds. In *Proc. GRAPP*, pages 193–200.

Gröger, G., Kolbe, T. H., Nagel, C., and Häfele, K. H. (2012). *OpenGIS City Geography Markup Language (CityGML) Encoding Standard. Version 2.0.0*. Open Geospatial Consortium.

Heckbert, P. (1982). Color image quantization for frame buffer display. *Computer Graphics*, 16(3).

Kutzner, T., Chaturvedi, K., and Kolbe, T. H. (2020). CITYGML 3.0: New functions open up new applications. *PFG*, 88:43–61.

Malhotra, A., Raming, S., Frisch, J., and van Treeck, C. (2021). Open-source tool for transforming CityGML levels of detail. *Energies*, 14(24).

Oestereich, M. (2014). Das 3D-Gebäudemodell im Level of Detail 2 des Landes NRW. *Nachrichten aus dem öffentlichen Vermessungswesen Nordrhein-Westfalen*, 47(1):7–13.

Wang, R., Peethambaran, J., and Chen, D. (2018). LIDAR point clouds to 3-D urban models: A review. *IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing*, 11(2):606–627.

Yan, J., Jiang, W., and Shan, J. (2012). Quality analysis on RANSAC-based roof facets extraction from airborne LIDAR data. *International Archives of Photogrammetry and Remote Sensing*, XXXIX(B3):367–372.