

A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data

Hai Huang^{*}, Claus Brenner, Monika Sester

Institute of Cartography and Geoinformatics, Leibniz University Hannover, Appelstr. 9A, D-30167 Hannover, Germany

ARTICLE INFO

Article history:

Received 3 May 2012

Received in revised form 1 February 2013

Accepted 5 February 2013

Available online 8 March 2013

Keywords:

Building

LIDAR

Point cloud

Urban

Extraction

Reconstruction

Three-dimensional

ABSTRACT

This paper presents a generative statistical approach to automatic 3D building roof reconstruction from airborne laser scanning point clouds. In previous works, bottom-up methods, e.g., points clustering, plane detection, and contour extraction, are widely used. Due to the data artefacts caused by tree clutter, reflection from windows, water features, etc., the bottom-up reconstruction in urban areas may suffer from a number of incomplete or irregular roof parts. Manually given geometric constraints are usually needed to ensure plausible results. In this work we propose an automatic process with emphasis on top-down approaches. The input point cloud is firstly pre-segmented into subzones containing a limited number of buildings to reduce the computational complexity for large urban scenes. For the building extraction and reconstruction in the subzones we propose a pure top-down statistical scheme, in which the bottom-up efforts or additional data like building footprints are no more required. Based on a predefined primitive library we conduct a generative modeling to reconstruct roof models that fit the data. Primitives are assembled into an entire roof with given rules of combination and merging. Overlaps of primitives are allowed in the assembly. The selection of roof primitives, as well as the sampling of their parameters, is driven by a variant of Markov Chain Monte Carlo technique with specified jump mechanism. Experiments are performed on data-sets of different building types (from simple houses, high-rise buildings to combined building groups) and resolutions. The results show robustness despite the data artefacts mentioned above and plausibility in reconstruction.

© 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.

1. Introduction

Many approaches for the reconstruction of 3D city models from measurement data have been reported in the past decades. The introduction of laser scanning makes the acquisition of 3D data easier and more accurate. Overviews are given by Brenner (2005), Schnabel et al. (2008) and Vosselman (2009).

Current bottom-up approaches include (Rottensteiner et al., 2008), in which a roof plane delineation from LIDAR data is presented. Statistical tests and robust estimation are employed for stable edge detection against the clutter. Using manually generated geometric constraints, topological correction is ensured without additional 2D data. Sampath and Shan (2010) segment and reconstruct more complicated buildings from airborne LIDAR point clouds based on polyhedral models. First, non-planar points are detected by means of the eigenanalysis making the roof planar segmentation more robust. The latter is implemented through an extended fuzzy *k*-means clustering. An adjacency matrix is derived

after the segmentation. For reconstruction, the roof vertices, ridges, and edges are determined by intersecting the corresponding planes, which include roof segments and possibly vertical walls or roof boundaries as the imposed constraints. Also starting from planar roof segments, Zhou and Neumann (2012) try to organize them and roof boundary segments with “global regularities” considering orientation and placement constraints. Matei et al. (2008) and Poullis and You (2009) present the fast processes to generate simplified building models of large-scale urban areas, in which the input LIDAR data is segmented presenting regularized buildings or building parts and simple polygon models are used for an efficient reconstruction. Meng et al. (2009) introduce a method to identify individual buildings from airborne laser data based on the morphology processing. Algorithms are developed to separate ground points and then filter out the other non-building parts (mostly the vegetation).

Approaches that employ top-down methods have been increasingly reported in the last several years. In Verma et al. (2006), the parametric modeling is employed for detection and reconstruction of 3D building models from airborne laser data. Relatively complex buildings can be represented by combining simple parametric roof

^{*} Corresponding author.

E-mail address: hai.huang@ikg.uni-hannover.de (H. Huang).

shapes. Lafarge et al. (2010a) present building reconstruction from a Digital Surface Model (DSM) combining generic and parametric methods. Buildings are considered as an assemblage of 3D parametric blocks. 2D-supports (approximate building footprints) are firstly extracted manually or automatically (Ortner et al., 2007). 3D blocks are then assembled based on 2D-support and optimized within a Bayesian framework. Dealing with more sophisticated buildings, basic geometric primitives, e.g., planes, cylinders and cones are extracted and combined with mesh-patches to present irregular roof forms (Lafarge and Mallet, 2012). The approach is extended to model urban environment including buildings, trees and ground surface with a semantic scene description. A method for the quality assessment of 3D building models extracted and reconstructed from point clouds is presented by Oude Elberink and Vosselman (2011).

As summarized in (Oude Elberink and Vosselman, 2011), airborne laser scanning data of urban areas often has the following quality issues: (1) systematic and stochastic errors in the measurement, (2) variable and relatively low point cloud densities, and (3) data gaps/flaws due to the occlusion by neighboring objects (e.g., clutter of trees), the absorption of the laser pulse by water features and the reflection from windows on the roof. The segmentation of relatively small roof structures and an accurate determination of roof edges are always hard. Results of the bottom-up reconstruction may thus be limited to a number of incomplete and irregular roof facets or building parts. A regularization with given constraints is always needed during the extraction or afterwards. In many cases it is not easy to be conducted even manually. For a regularized plane detection a probability-driven edge sweeping method is proposed by Huang and Brenner (2011). Although it works robustly in spite of clutter and data flaws, it encounters difficulties by processing complex roofs.

This paper extends the work described by Huang and Sester (2011) and Huang et al. (2011). We present a generative statistical reconstruction of building roofs with emphasis on top-down processes. A library of roof primitives is predefined and a building roof is considered as a variant of one primitive or a combination of a set of primitives. In comparison with the approaches that share the “LEGO” scheme (Kada and McKinley, 2009; Lafarge et al., 2010a), i.e., the building is first cut down into building parts and primitives are found to fit the parts, we design new combination rules and merging process to allow primitive overlaps. A more plausible as well as stable result is thereby gained because all the primitives can maintain complete during the combination and deviations caused by random sampling can be compensated (cf. Section 3). A variant Markov Chain Monte Carlo (MCMC) sampler with specified jump mechanism is employed for the sampling of the model parameters because of its adaptation to the change of types and numbers of building parts during the search process. Unlike most

of the related researches, the bottom-up effort, e.g., points clustering, plane detection (Sampath and Shan, 2010), or 2D building footprint data (Kada and McKinley, 2009; Lafarge et al., 2010a), are not required in the proposed work. To compensate the absence of the initial information (which is normally provided by the bottom-up analysis) we conduct explicit model selection methods in (1) the estimation of the 2D building (footprint) size (cf. Section 4.3) and (2) the change of the number and type of building parts (“jump” mechanism, cf. Section 5.1) to perceptively guide the reconstruction.

The paper is organized as follows. In Section 2 a pre-segmentation is proposed for large urban scenes to identify potential buildings and thus limit the search space. Section 3 introduces the definition of roof primitives and the combination rules. The overall scheme of the stochastic modeling of roofs is given in Section 4. Section 5 describes the jump mechanism between different types and numbers of primitives in the search. The optimal reconstruction is achieved by Maximum A Posteriori (MAP) estimation. In Section 6 we show the reconstruction results of varied building types and data resolutions. Section 7 shows an extension of our approach for building model generalization. The paper ends up with conclusions.

2. Pre-segmentation

In (Huang et al., 2011) we explored and demonstrated the potential of a pure top-down approach. It shows that in a relatively simple scene (32,000 m², nine buildings) the proposed pure top-down algorithm can find the target buildings with only generic prior information. Note, however, that in practical applications a complete replacement of the bottom-up process is hard. For large scenes, e.g., a whole city district, the complexity of the parameter distributions as well as the number of disturbances is so high that the search cannot guarantee appropriate results. The practical way is to balance the bottom-up and top-down partitions to achieve robustness as well as efficiency.

We propose a pre-segmentation method based on “blob” detection. As shown in Fig. 1, the input point cloud (left) is rasterized with the simple “natural neighbor” method and converted into an image (middle) with the gray values indicating the heights. In this image a “blob” (right) is a potential building, i.e., an area of a certain size which sticks out from the ground with a certain height. The image is binarized using a z-value threshold, which is gained automatically by finding the most concentrated (range of) z-value in its histogram. This step works stably for flat and slightly wavy terrain because the difference between roof and ground is relatively large and a few meters deviation will not affect the result. Simple slope or curved ground can be compensated by detecting its form and converted to flat terrain for processing. Based on

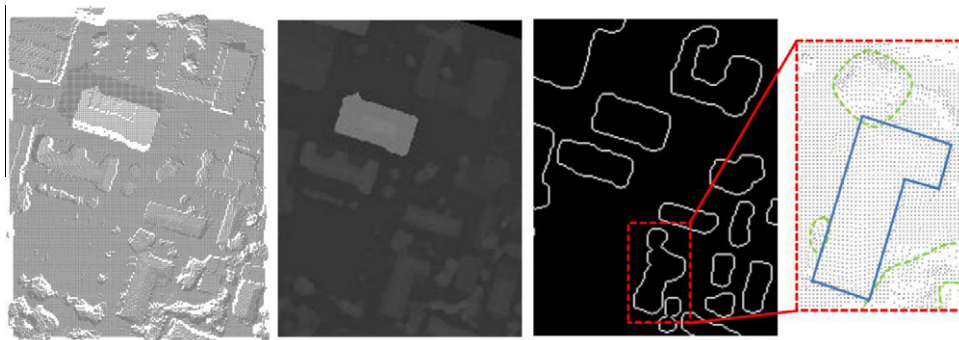


Fig. 1. Pre-segmentation: the given point cloud (left) is converted into a raster image (middle). After morphology processing blobs (right) containing individual buildings are labeled and segmented (red dashed rectangle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the detected “blobs” the input point cloud can be segmented into small subzones focusing on buildings.

In the urban scenes, buildings, as well as a number of other objects (mostly trees) are all presented as bright areas with similar elevation. We employ mathematical morphology (Serra, 1983; Köthe, 1996) to reduce small “spots”, which are supposed to be non-building objects. Let I be the input image and s the structuring element, which is defined as a disk with a radius of 5 m, an “Opening” operation

$$I \circ s = (I \ominus s) \oplus s$$

is conducted with “Erosion” and subsequent “Dilation”. Essential effects of “Opening” are to remove the trivial spots and restore the outline. The relatively large blobs, which are supposed to contain individual buildings or building groups, are thereby clearly separated in spite of the numerous adjacent noises (Fig. 1, right).

Please note that in this case we prefer relatively small structuring element rather than losing objects. As a result, one blob containing more than one buildings or/and clutter objects, i.e., adjacent trees (Fig. 1, right, green dashed lines) may happen. Even though, the proposed algorithm can actually deal with a scene with multiple buildings as well (cf. also Huang et al., 2011) so that the pre-segmentation must not be perfect and a fixed radius of the structuring element is used for simplification.

Given detected “blobs”, the subzones are defined as the minimum circumscribed rectangles (or “bounding boxes”, cf. Fig. 1, right, red dashed rectangle) of the “blobs” and a tolerance that equal to the radius of the structuring element is given.

As results the pre-segmentation provides the following advantages:

1. A number of non-building objects (e.g., trees) can be ignored because of their relatively small size.
2. The efficiency and the stability of the further statistical reconstruction are improved. The search areas are well limited in the segments so that the computational complexity is significantly reduced. The search of optimal model is conducted locally instead of travel the whole scene overcoming many local minima, which is the most unstable and time-consuming part in the search. (A comparison of runtimes can be found in Section 6.)

We consider the city blocks (cf. Section 6, Fig. 12), as the largest objects to be tackled after pre-segmentation as they cannot be further divided into separate houses with “blob” detection. The pure top-down approach described later will demonstrate its ability to deal with them.

3. Roof primitives

Parametrized roof primitives are the basis of the stochastic modeling. We present a library of primitives and rules for their combinations. An important idea to gain more flexible and stable reconstruction is to allow the overlapping of primitives in the reconstruction.

Most related works treat building roofs as a collection of facets as plane detection has been well studied and works stable (Rottensteiner et al., 2008; Vosselman, 2009; Sampath and Shan, 2010). In comparison with the facet-based reconstruction, the primitive-based modeling has the following advantages:

- There are no more irregular and incomplete roof facets or building parts caused by flaws or conflicts in the plane detection. In roof primitives the constraints of member facets are predefined and ensure regularized reconstructions.

- The combination of a few primitive is much simpler than the organization of a bunch of facets. The derived rules can be more legible because of fewer participants and simpler (in comparison with networked facets) relationships (Huang et al., 2011).

Allowing overlaps, furthermore, keeps primitives complete during the reconstruction and assembling (Huang et al., 2011) instead of being cropped to fit the adjacent building parts. In the final roof model, the redundant parts are always hidden inside the assembly and they could be easily removed afterwards, e.g., by intersecting with CAD tools.

Beside the horizontal intersection we allow the vertical one as well. By this means some combined roofs, e.g., platform roofs (multi-level flat roofs, cf. Section 6, Fig. 11, buildings 1 and 8) and complex buildings (buildings 3 and 5) are possible to be reconstructed without adding any more particular models in the library.

3.1. Library of primitives

The library, as shown in Fig. 2, provides three groups including 11 types of roof primitives. Please note that this library contains only a limited number of entries with planar shapes and rectangular footprints. In this work we prefer simpler and less number of primitives for more efficient reconstruction. The idea is to present complex roof shapes with the combination of basic primitives. Planar roofs and rectangular footprints are chosen not only because they are simple (with less shape parameters), they are actually also the basic form that most buildings follow, i.e., roofs derived from the primitives or their combinations will cover the majority of the buildings in urban areas. Furthermore, the proposed primitive merging process is also able to represent buildings with non-rectangular footprints (cf. Section 3.2).

Their parameters θ are defined as:

$$\theta \in \Theta; \Theta = \{\mathcal{P}, \mathcal{C}, \mathcal{S}\} \quad (1)$$

where the parameter space Θ (Fig. 2, top right) consists of position parameters $\mathcal{P} = \{x, y, azimuth\}$ and contour parameters $\mathcal{C} = \{length, width\}$ as all primitives are defined to have a rectangle footprint. \mathcal{P} and \mathcal{C} have fixed members. \mathcal{S} contains shape parameters, e.g., the ridge/eave height and the depths of hips, and is varied for different primitives.

- Group F: generalized flat roofs including flat roof (F1) and shed roof (F2).

All possible shape parameters: $\mathcal{S}_{F,max} = \{z_1, z_2\}$ with $z_1 = z_2$ the roof height for flat roof or z_1 and z_2 the lower and higher eave heights for shed roof.

- Group H: generalized hipped roofs including all variants of hipped roofs. Gable roof (H4) and mansard roof (H6) are considered as special cases of hipped roofs.

$\mathcal{S}_{H,max} = \{z_1, z_2, hip_{l1}, hip_{l2}, hip_{d1}, hip_{d2}\}$ with z_1 the eave height, z_2 the ridge height, hip_{l1} and hip_{l2} the longitudinal hips, and hip_{d1} and hip_{d2} the lateral hips.

- Group G: Gambrel roof (G1) and others. The common character of this group is every roof has three different height levels. (Please note, same as Groups H, z_1 and z_2 are kept for the eave and ridge heights. z_3 is the additional roof level in the middle.) E.g., although the salt-box roof (G3) is very similar to the asymmetric gable roof (H5), it cannot be represented as an H-type because the parameter number of “heights” is limited to two in Group H. In this library we do not define elliptic roofs, which can be roughly approximated by gambrel roofs.

$$\mathcal{S}_{G,max} = \{z_1, z_2, z_3, hip_{l1}, hip_{l2}, hip_{d1}, hip_{d2}\}$$

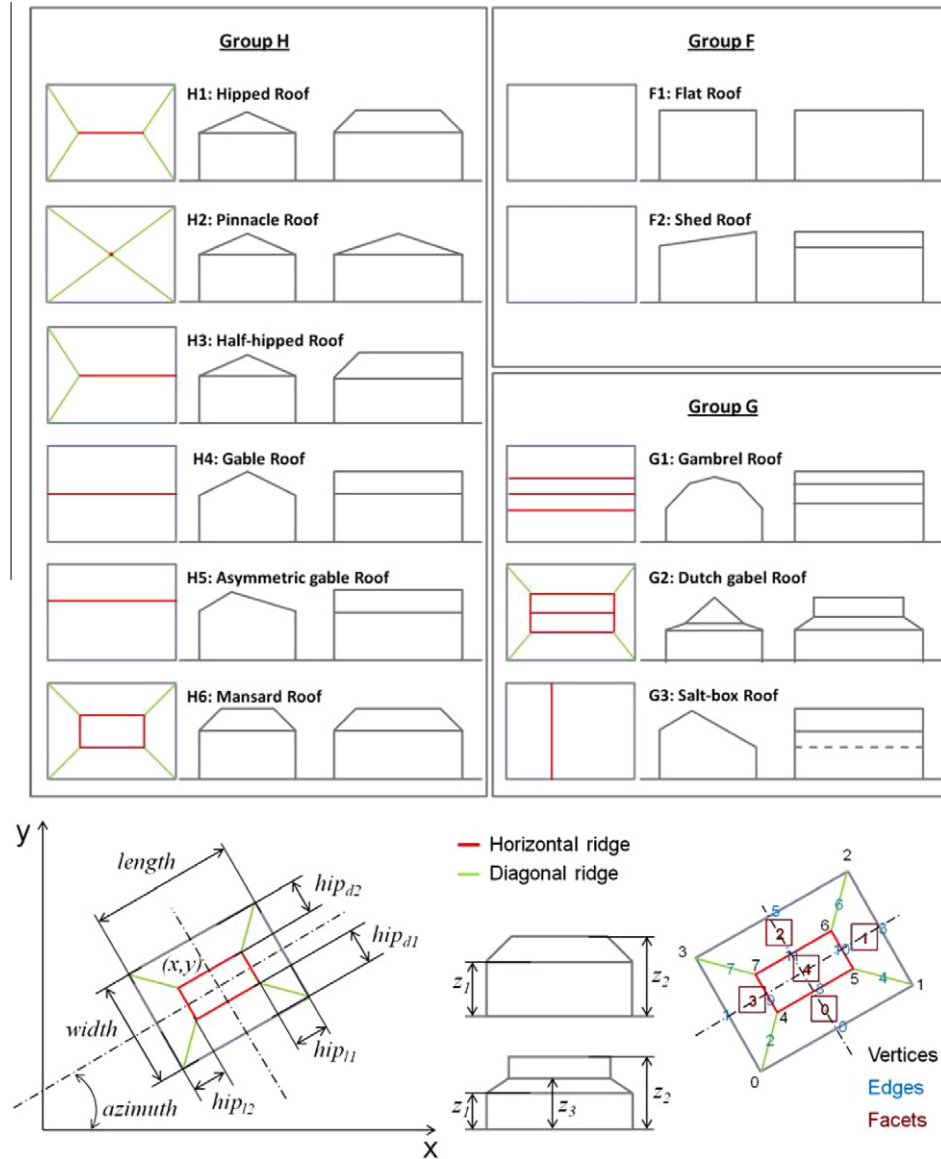


Fig. 2. Library of roof primitives.

Geometrical features of different levels, i.e., vertices, edges, and facets, and their relationships are all predefined and encapsulated in the primitives (Fig. 2, bottom right). They can be called as members of the primitives for different purposes like primitive merging, calculating reconstruction errors, drawing building footprints, etc.

3.2. Primitive combination and merging

By combining primitives we propose a context-sensitive regularization with the following rules:

Rule 1: Intersection angle of the primitives are conditionally regularized (cf. also below). Let i and j indicate the two intersected primitives, there are:

if $\exists |azimuth_i - azimuth_j| < 5^\circ$, then $azimuth_i = azimuth_j$;
 if $\exists |azimuth_i - azimuth_j| \in (90 \pm 5)^\circ$, then $|azimuth_i - azimuth_j| = 90^\circ$.

Rule 2: Heights of flat roofs or ridge heights of hipped roofs are harmonized if they are close to each other:

if $\exists |z_{2,i} - z_{2,j}| < 0.25$ m, then $z_{2,i} = z_{2,j}$.

Rule 3: Eave heights of all non-flat roofs are harmonized if they are close to each other:

if $\exists |z_{1,i} - z_{1,j}| < 0.25$ m, then $z_{1,i} = z_{1,j}$.

Please note that in principle the thresholds for the azimuth/height differences should be linked to the point densities because they somehow reflect the reconstruction accuracy. Here for the primitive merging, however, we simply use general thresholds for all datasets with the empirical values derived from the sparsest data-sets (1 point/m²). Rule 1 helps us to solve another general problem in flat roof extraction. As the data points (especially in low density data) on the roof corners are very likely missing and that on the edges are not perfect either, a slight deviation in azimuth (rotation in the roof plane) may lead to similar or even better evaluation for the candidate model and therefore cannot be detected. It is especially true for the flat roofs with less length to width ratio and hard to be solved without any prior information. Fig. 3 shows two examples. For flat roofs the azimuth cannot be determined as reliably as for shed or hipped roofs, which are composed by non-horizontal plane(s). Therefore, we implement additional rules for flat roofs as follows:

Rule 1.1: If a flat roof is in the same “blob” with shed/hipped roofs, then align to the nearest one (cf. Fig. 3, left).

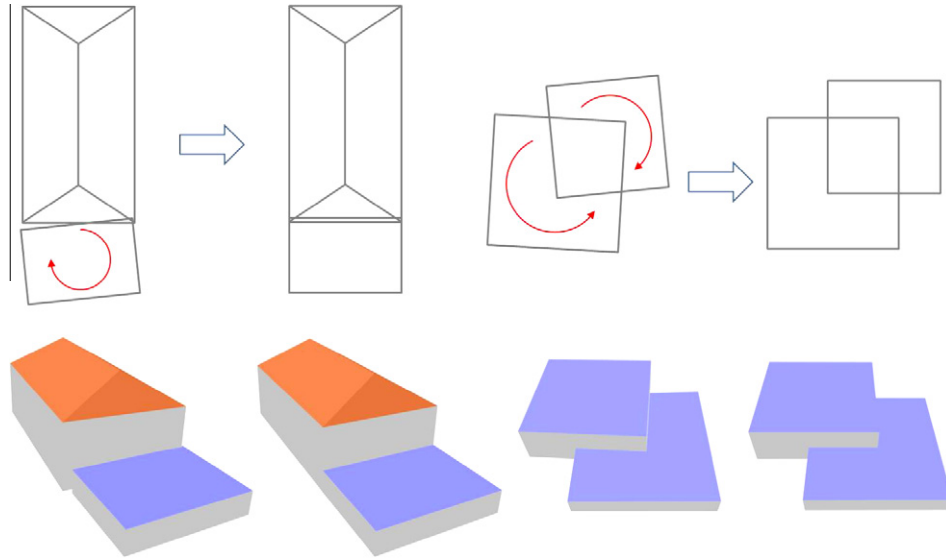


Fig. 3. Azimuth deviation of flat roofs can be adjusted jointly with the adjacent roofs.

Rule 1.2: If two flat roofs are in the same “blob” (connected or not), their azimuths are jointly adjusted weighted according to their areas (cf. Fig. 3, right).

Although the primitives are regularized with the combination rules, deviations still often exist because of the occlusions from clutter objects/other building parts and generally the possible low point density. Most of them do not jeopardize the reconstruction results as they are hidden inside the intersected domain. However, for the primitives that share multiple planes the deviation can be crucial (cf. also Fig. 5). A geometrical adjustment is needed to “merge” the primitives into a plausible model. We conduct the following rules for the primitive merging:

Rule 4: Geometric merging by primitive intersection

$$\forall |azimuth_1 - azimuth_2| > 5^\circ$$

- Unify the ridge heights (z_2) and eave heights (z_1) of the two primitives weighted to their footprint sizes $A = length \times width$:

$$z_1 = (A_i \cdot z_{1,i} + A_j \cdot z_{1,j}) / (A_i + A_j); \quad z_2 = (A_i \cdot z_{2,i} + A_j \cdot z_{2,j}) / (A_i + A_j);$$
- Calculate the cross point of the ridges (or their extension lines) and align the end points of the two ridges to it (Fig. 4, red);

- Calculate the cross points of the two pairs of longitudinal (along the ridge direction) eave lines and align each end point of the eave line to the second cross point (Fig. 4, green) to avoid geometric conflicts.

Two examples with different intersection angles can be found in Fig. 5 (left and middle).

Rule 5: Geometric merging by primitive extension

$$\forall |azimuth_1 - azimuth_2| < 5^\circ \text{ (actually } 0^\circ \text{ here after the Rule1)}$$

- If the lower (z_2) roof has higher eave (z_1), its eave has to be unified to another primitive to avoid geometric conflicts:
- if $\exists z_{1,i} > z_{1,j}$ and $z_{2,i} < z_{2,j}$, then $z_{1,i} = z_{1,j}$.

An example can be found in Fig. 5 (right).

4. Stochastic modeling

Generative roof models employ full probability information for all parameters. Multiple hypothetic models are generated via stochastic sampling of the parameters and evaluated by comparing them with the given measurement data (the point cloud). The goal is to find the optimum combination of parameters, i.e., the best estimation of the underlying roof model. The question to be answered in this top-down extraction is if the target roof can still be stably found without any initial information, which is conventionally provided by bottom-up analyses. Besides the heavier search task, there are also two difficulties:

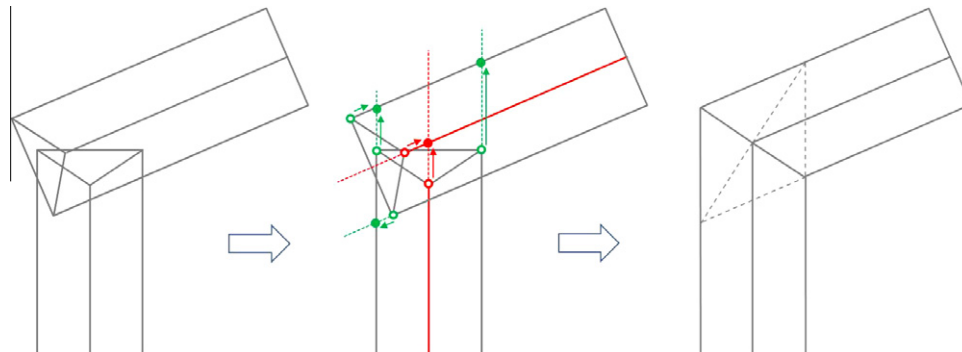


Fig. 4. Primitive merging: (left) two building parts generated by sampling; (middle) shifting of ridge (red) and eave (green) lines; (right) the merged building parts. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

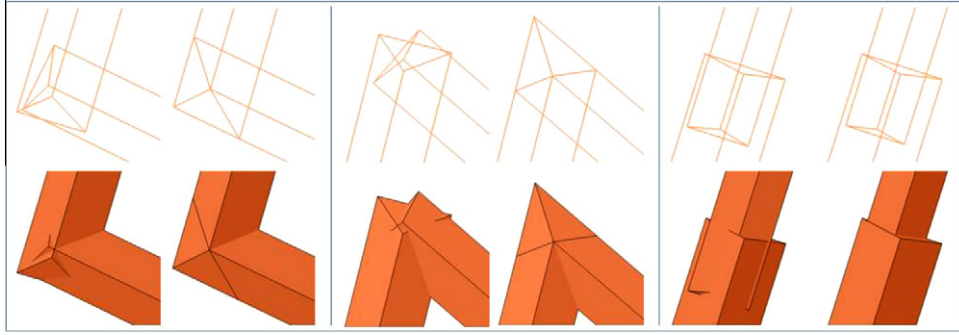


Fig. 5. Examples of the primitive merging with intersection angle larger (left) and smaller (middle) than 90° and longitudinal extension (right).

1. local extrema: in urban scenes clutter objects, e.g., trees, are very often adjacent to the buildings with similar height and size;
2. model-minimizing tendency: using the deviation to data points to evaluate the hypothetic models, the finally found “best” model tends to shrink to a very small piece as smaller comparison area (corresponding to fewer data points) means less error. In this case the model size A should be reasonably estimated (cf. Section 4.3).

We, therefore, use a sampling procedure derived from the MCMC algorithm for efficiently exploring the high-dimensional solution space. Information criterion is integrated to give instructive values for A , which guides the search of the parameter-set \mathcal{C} .

The proposed sampling procedure can be seen as closely related to the Reversible Jump Markov Chain Monte Carlo (RJMCMC) technique while the jump mechanism is re-designed to break the Markovian property for search efficiency. In this procedure conventional MCMC sampler is still employed when the number of parameters is fixed. A jump mechanism guided by model selection and specific routine is conducted for the change of primitive numbers and types during the search.

4.1. Search scheme

The search scheme consists of the following four steps and starts from the center of the point cloud (with no prior for the position) with generic parameter priors.

4.1.1. Finding the first primitive

First, a coarse search is conducted to find a primitive arbitrarily. The gable roof (cf. Fig. 6a) is chosen as the initial primitive instead of the simpler flat roof because the former is more sensitive to the azimuth and can actually represent the flat roof as well by harmonizing the ridge and eave heights. Please note that generally the initial state is not very critical for the MCMC sampler, but in this case the jumps are not proposed in every MCMC step (cf. Section 5.1) so that the initial primitive will have more influence at the beginning of the search until the jump happens. This is particularly advantageous to the preliminary determination of the roof location and orientation.

The model parameters are then refined locally (cf. Fig. 6b) and the model has the possibility to switch to more appropriate primitives with jumps mechanism (kernel “Switch”, cf. Section 5.1) in the MCMC sampler (cf. Fig. 6c and d). Once the primitive is accepted, the parameter priors are refined with this new “evidence” in a Bayesian framework.

Please note that after a primitive has been found, we do not as usual delete the corresponding points from the source data as these points may be shared with other parts of this building. This

is more meaningful for low density data. Instead, we record its parameters and update the prior distributions to avoid this combination of parameters being sampled twice, i.e., the same primitive will not be proposed again in the next rounds of search.

4.1.2. Finishing the whole roof

To find other possible roof parts the movements “birth” and “death” (cf. Section 5.1) are activated. New roof parts are proposed by “birthing” new primitives near the already found building part (cf. Fig. 6e, red) and the worse proposals will be removed by the “death” jumps later. The sampling stops till no more primitive can be found with an acceptable reconstruction error. The found roof parts are then assembled (cf. Section 3.2) into a complete roof model (cf. Fig. 6g and h).

4.1.3. Reconstructing superstructures

In dense point clouds (e.g., with a resolution of about 0.18 m, cf. Fig. 19) superstructures, e.g., chimneys and dormers, will be searched in the area above the base roof. Two simple primitives, flat and gable roofs, are used as primitives for superstructures with modified priors.

4.1.4. Updating priors

The refined priors of parameters, especially that of the azimuth and heights, can be (optionally) applied as references for their neighbor subzones.

4.2. Likelihood function

During the sampling, we use “z-error” (Δ_z), the average absolute deviation in z-direction from the proposed model (\mathcal{M}) to the data points (\mathcal{D}), to evaluate the proposed candidate models.

$$\Delta_z = \frac{\sum_{f \in \mathcal{F}} \left(\sum_{i \in \Omega_f} |z_{\mathcal{M}} - z_{\mathcal{D}}|_i \right)}{K} \quad (2)$$

with f an individual facet from the facet-set (\mathcal{F}) of the primitive, i the data points in the domain of f : Ω_f , and K the number of the involved data points.

Please note that for the evaluation we consider the roof surface as a horizontal mesh rather than a set of 3D planes, so that we just calculate the deviations in z-direction instead of the normal distances. By the calculation we, thereby, just observe the points inside the 2D (x - y) model contour, any effort for finding and labeling points that belong to specific planes is not required. z-error is easier to implement and, most important, also more feasible in this approach: during the stochastic search process the model and the roof points could have large offset or even far away from each other and z-error is more stable than other measures, e.g., the absolute normal distance. The absolute deviation (L1-Norm) is employed instead of the square deviation (L2-Norm) because

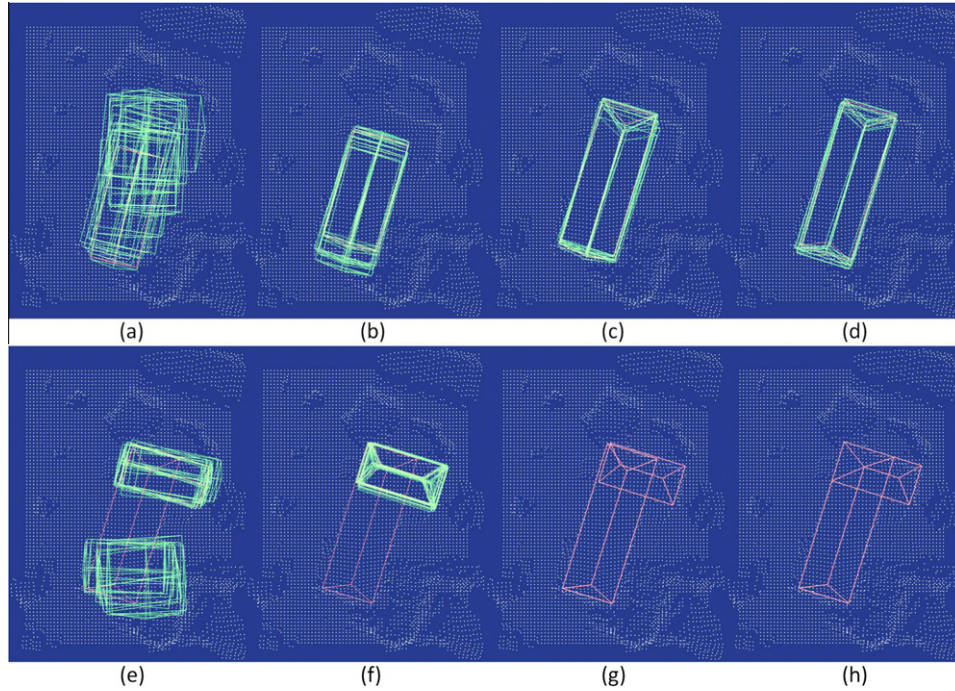


Fig. 6. An example of roof reconstruction: (a) rough search with simple gable roof, (b) local search refining the model, (c) jump to the half-hipped roof, (d) jump to the hipped roof, (e) “birth” of a new primitive near the primitive one (red), (f) refining the new primitive with constraints derived from primitive one, (g) the two found primitives, and (h) roof model after primitive merging. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

we do not search for the “best fit” to all the (involved) points but a “consensus” against outliers. Z-error is, therefore, more appropriate for this case with less sensitivity to such data points caused by the clutter objects and building part occlusions.

The likelihood function $L(\mathcal{D})$ can be expressed as:

$$\Theta \mapsto L(\mathcal{D}) = L(X|\Theta) \propto \exp(-\Delta_z) \quad (3)$$

with Θ the parameters of the model and X the observations.

4.3. Model size estimation

To overcome the model-minimizing tendency mentioned above, an instructive constraint for the model size A (corresponding to the parameter-set \mathcal{C}) is needed. We conduct a perceptive estimation employing information criterion to balance the goodness of fit and the size of the model.

Fig. 7 (left) shows the average deviation (z-error, blue) from the proposed model to the data points while the model size (indicated by K) increases. Please note that this function is neither linear nor monotonic increasing under the influences of data flaws and clutter objects.

We follow the basic idea of Akaike Information Criterion (AIC) (Akaike, 1973)

$$AIC = 2k - 2\ln(L) \quad (4)$$

to build our goal function. In Eq. (4), k indicates the number of parameters, which implies the complexity of the model, and L the maximum likelihood while the employed parameters have been optimized. In our case, however, what we want to prevent is not the model being too complicated but the size of the model being too small. We use the evaluation of model for the L (cf. Section 4.2) and $-K^\alpha$ to represent the influence of the model size. K , again, is the number of data points in the domain of the proposed model, which implies the model size (linear proportion in raster data). The actual number of involved points is more sensitive than A to the influence as the likelihood is just calculated with these points. α is the influence factor for K , which has relatively large impact because of the usually large number of K . To balance the influences of both the model size and the goodness of fit, it is empirically determined $\alpha = 0.1$. The total information entropy of the proposed model (\mathcal{M}) can then be expressed as:

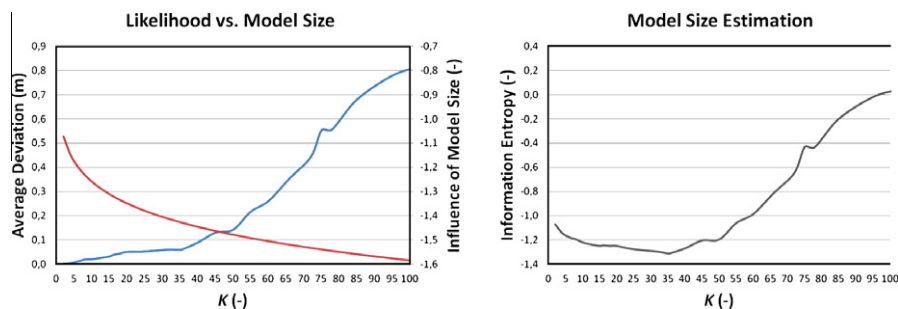


Fig. 7. Plots of the average deviation (blue), the influence of model size $-K^\alpha$ (red) and the information entropy (black) over K . The minimum entropy indicates the optimal balance of goodness of fit and the model size. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$H_{\mathcal{M}} = -K^2 - 2\ln(L(\mathcal{D}|\mathcal{M})) \quad (5)$$

By these means a better fit is rewarded while size decrease gaining trivial improvement is discouraged. The conducted information criterion, as shown in Fig. 7 (right), is employed to guide the search of the parameter-set \mathcal{C} . Please note, unlike AIC, this is not a trade-off between reconstruction accuracy and model complexity because less error does not mean better reconstruction either in this case. Besides, there is no general threshold for acceptable errors. The goal of the entropy function is to find the lower limit of the model size as large as possible while perceiving its influence on the tendency of error change.

5. Sampling procedure and optimization

5.1. Specified jump mechanism

Reversible Jump Markov Chain Monte Carlo (RJMCMC) is an extension of the MCMC algorithm to handle solution spaces of variable dimensions. As introduced by Green (1995), the Markov Chain sampler is allowed to switch (“jump”) between subspaces with variable dimensions in the search. Inspired by this concept, we conduct a modified MCMC technique with specified “jump mechanism” in this work. The jumps, i.e., modeling with varying numbers of parameters (dimensions), are employed to simulate the change of configurations (number and type of roof primitives).

We define the possible jumps in a mixed transition kernel:

$$\mathcal{K} = \{Sw_1, Sw_2, Bi, De\} \quad (6)$$

with

- Sw_1 : “switch case 1” to more complex primitive (with more parameters);
- Sw_2 : “switch case 2” to simpler primitive (with less parameters);
- Bi : “birth” of a new primitive adjacent to the original one;
- De : “death” of the last born primitive.

The “switch” is the jump between different types of primitives. Studying the primitives, we narrow down all the possible movements into a specific “jump routine”, as shown in Fig. 8 (left). It ensures that each jump step only changes a limited number and more sensible parameters.

Let \mathcal{M}_i and \mathcal{M}_j be two models in $\{\mathcal{M}_n; n = 1, \dots, N\}$ with N the number of possible states, in this case the number of primitive types. The jump from i to j will be accepted according to the probability:

$$\mathcal{A}(\mathcal{M}_i, \mathcal{M}_j) = \min \left\{ 1, \frac{p(\mathcal{D}|\mathcal{M}_j)p(\mathcal{M}_j)}{p(\mathcal{D}|\mathcal{M}_i)p(\mathcal{M}_i)} \cdot \frac{\mathcal{J}_{i \rightarrow j}}{\mathcal{J}_{j \rightarrow i}} \right\} \quad (7)$$

with $\mathcal{J}_{i \rightarrow j}$ the Jacobian matrix, in which the proposal density, i.e., the probabilities for all the possible jumps from i to j , are coded. As mentioned in (Lafarge et al., 2010b), the computation of the transition matrix can be greatly simplified in the practical implementation. In our case, we use a fixed transition matrix τ , as shown in Fig. 8 (right), instead of the Jacobian \mathcal{J} . Note that τ is sparse and asymmetric because the number of possible movements is significantly reduced and we encourage jumps to more complex models rather than the other way around.

The “birth” and “death” are jumps between different numbers of primitives. Including all kinds of jumps defined in \mathcal{K} , the \mathcal{M}_i and \mathcal{M}_j can be extended to represent the states (i.e., the numbers as well as the types of primitives) before and after the jump. The “detailed balance” condition in the Markov Chain can be expressed as:

$$p(\mathcal{M}_i|\mathcal{D}) \cdot \mathcal{T}(\mathcal{M}_i, \mathcal{M}_j) = p(\mathcal{M}_j|\mathcal{D}) \cdot \mathcal{T}(\mathcal{M}_j, \mathcal{M}_i) \quad \forall \mathcal{M}_i \neq \mathcal{M}_j \quad (8)$$

with the posterior:

$$p(\mathcal{M}_i|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_i) \cdot p(\mathcal{M}_i)}{p(\mathcal{D})} \quad (9)$$

where $P(\mathcal{D})$ is the marginal probability and a constant. $\mathcal{T}(\mathcal{M}_i, \mathcal{M}_j)$ is the transition density from state \mathcal{M}_i to state \mathcal{M}_j . The Markov Chain is constructed with Metropolis–Hastings algorithm as:

$$\mathcal{T}(\mathcal{M}_i, \mathcal{M}_j) = q(\mathcal{M}_i, \mathcal{M}_j) \cdot \mathcal{A}(\mathcal{M}_i, \mathcal{M}_j) \quad (10)$$

where $q(\mathcal{M}_i, \mathcal{M}_j)$ indicates the proposal density for the jump between states and $\mathcal{A}(\mathcal{M}_i, \mathcal{M}_j)$ again the acceptance probability.

The sampling procedure can then be summarized as follows:

1. Initialization: $(\mathcal{M}^{(i=0)}, \mathcal{O}^{(i=0)})$.
2. Proposing new state \mathcal{M}' .
 - 2.1 Sampling configuration from $\mathcal{K} = \{Sw_1, Sw_2, Bi, De\}$.
 - 2.2 Sampling parameters \mathcal{O}' .
3. Accepting new proposal with the probability.

$$\mathcal{A}(\mathcal{M}^{(i)}, \mathcal{M}') = \min \left\{ 1, \frac{p(\mathcal{M}'|\mathcal{D})}{p(\mathcal{M}^{(i)}|\mathcal{D})} \cdot \frac{q(\mathcal{M}^{(i)}, \mathcal{M}')}{q(\mathcal{M}', \mathcal{M}^{(i)})} \right\} \quad (11)$$
4. $\mathcal{M}^{(i+1)} = \mathcal{M}'$ if accepted, otherwise $\mathcal{M}^{(i+1)} = \mathcal{M}^{(i)}$.

As long as the jump kernels keep the balance condition, i.e., being reversible – possible to return to the previous state, the conducted sampler is able to explore in a great wider variety of

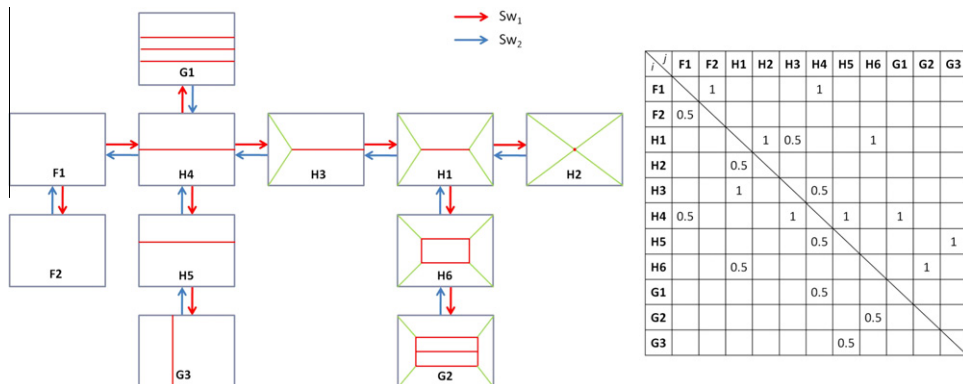


Fig. 8. Possible “switch”-jumps between primitives are limited by a “jump routine”(left) and the transition matrix (right) becomes therefore sparse.

hypothesis models. Although this also implies that such sampling procedure has itself model selection effect, it is usually time-consuming. We, therefore, employ explicit model selection mechanism in the transition kernel. The information entropy of the model $H_{\mathcal{M}}$ is calculated as:

$$H_{\mathcal{M}} = k^{\beta} - 2\ln(L(\mathcal{D}|\mathcal{M})) \quad (12)$$

where β is given an empirical value of 1/12. β is set small to reduce the sensitivity to the parameter number (model complexity) as in this case we prefer better reconstructions than simpler models.

The acceptance probability in the kernel can thus be expressed as:

$$\mathcal{A}(\mathcal{M}_i, \mathcal{M}_j) = \min \left\{ 1, \frac{H_{\mathcal{M}_j}^{-1}}{H_{\mathcal{M}_i}^{-1}} \cdot q(\mathcal{M}_i, \mathcal{M}_j) \right\} \quad (13)$$

Please note that in the sampling process, jumps are not proposed in every MCMC move but only when the maximum likelihood for the current configuration (with determined number and types of the primitives) has been reached. The specified schedule (cf. Section 4.1) and the jump routine keep the search from reaching any state in the solution space, i.e., the irreducibility is not guaranteed. Although the jumps break the stationary (irreducible and aperiodic) rules required by Markovian process (which makes the whole scheme no more a MCMC process), this allow our process to be more efficient and stable with the scheduled “cooling down” of the search entropy.

5.2. Optimization

For optimization we employ the posterior instead of the likelihood of the proposed model as the goal function, which integrates the priors of the roof parameters:

$$p(\theta_i) = p(\mathcal{M}_{found}|\theta_{i,0}) \cdot p(\theta_{i,0}); \theta_i \in \Theta; \theta_{i,0} \in \Theta_0 \quad (14)$$

with Θ_0 the initial parameters and \mathcal{M}_{found} the already found model(s), i.e., new evidence(s) to refine the $p(\Theta_0)$ in a Bayesian framework. The priors are supposed to provide the following information:

- generic value ranges of parameters, which sort out implausible candidates, e.g., roofs with the height near ground;
- recorded parameters of the already found primitives: thus the joint distribution is simplified by labeling particular areas. Possible further (often smaller) primitives are thereby easier to be found.

The Maximum A Posteriori (MAP) estimate of Θ can then be expressed as:

$$\hat{\Theta}_{MAP} = \underset{\Theta}{\operatorname{argmax}} \left\{ \frac{L(\mathcal{D}|\Theta)p(\Theta)}{P(\mathcal{D})} \right\} = \underset{\Theta}{\operatorname{argmax}} \{L(\mathcal{D}|\Theta)p(\Theta)\} \quad (15)$$

where $p(\Theta)$ is the synthesis prior with Θ indicating all the parameters. For the computation the priors of individual parameters are combined by multiplication with the assumption that they are basically independent to each other. The multiplication is chosen to combine the priors because it is more sensitive to unreasonable parameters. $P(\mathcal{D})$ is again the marginal probability, which can be seen as a constant in the optimization as it does not depend on Θ .

In practice, the z-errors of individual roof reconstructions may be varied even in the same scene. E.g., in the example scene shown in Fig. 11 the error range of individual accepted primitives is between 0.05 and 1.10 m. The reconstruction is conducted to maintain the plausibility of the model in spite of the outliers. So that some qualified primitives have relatively large deviation to the data points due to the occlusion of adjacent trees and/or intersected primitives. Therefore, the z-error should be seen as a mea-

surement of “consensus” instead of “reconstruction error”. For the same reason a fixed threshold of z-error for acceptance is no more feasible. We therefore set the stop criteria as: (1) the z-error becomes stable or (2) a predefined maximum number of iterations is reached. Fig. 9 presents the convergence of z-errors in a two-primitive search process (cf. also Fig. 6).

6. Experiment results and assessment

Experiments are performed on data-sets of different building types, i.e., from simple houses, high-rise buildings to combined building groups, and resolutions, i.e., from 1 m (cf. Fig. 10a), 0.5 m to ~0.18 m (cf. Fig. 19, left), using a laptop with a 2×2.8 GHz processor.

6.1. Experiments

An experiment with 1 m raster data (approximate 89,000 m², 21 buildings and 1 city block) of Hannover, Germany, in shown in Fig. 10. The input point cloud (a) is firstly segmented into subzones according to the detected blobs (b). Primitive models are extracted (c) and assembled into entire buildings in the form of Virtual Reality Modeling Language (VRML) models (d). The building roofs in the test scene have been reconstructed with correct positions and plausible shapes. The runtime for this scene is about 24 min in total.

In Fig. 11 we compare one section (left) of the reconstruction result with the reference image (right). Some small and narrow building parts, e.g., that of the east wing of building 2, the fire escape shaft of building 1, the small structures in the north of building 4, and the terrace of building 7 (pointed by the red¹ arrows), have not been extracted as the point cloud is not dense enough to represent them meaningfully. It is the same reason why a number of small structures on the roofs have not been reconstructed. Even relatively bigger dormers (e.g., that on building 5) have only less than 10 data points on it. It is hard to tell if they belong to an individual structure or just data flaw or clutter. Structures of unusually low height (e.g., the two storages of building 3, blue arrows) are also missing because they are out of the search range (which is limited by the given priors).

We also compare the runtimes of this section (nine buildings including 20 primitives) with and without the pre-segmentation (cf. Section 2), which are about 8 and 15 min respectively. The pre-segmentation improves the efficiency significantly by avoiding the most expensive “global” search.

A city block, as mentioned before, is usually the largest “blob” and the most complicated subzone after the pre-segmentation. Fig. 12 shows an example of free-form city blocks which are typical in European cities. In this case the building group is reconstructed with a larger number of connected primitives and the primitive merging plays an more important role to ensure a plausible result. Under the limit of errors we prefer primitives as large as possible to simplify the model. The runtime of this city block with 11 primitives is about 7 min. Compared with the section shown above, which takes 8 min for 20 primitives, the search of each primitive in the larger “blob” is more time-consuming.

6.2. Assessments

In this section we present assessments of the reconstruction results without 3D ground truth data.

¹ For interpretation of color in Fig. 11, the reader is referred to the web version of this article.

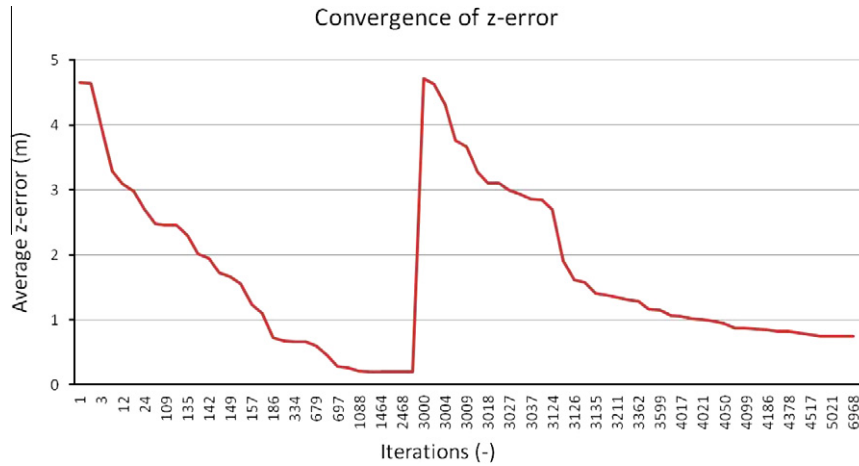


Fig. 9. The convergence of z-error during the search of a two-primitive model: after 1500 iterations the z-error of the first primitive converged at 20 cm. The second primitive was then “born” and brought an additional error to the model. The z-error of the whole model became stable under 80 cm after about 4900 iterations in total.

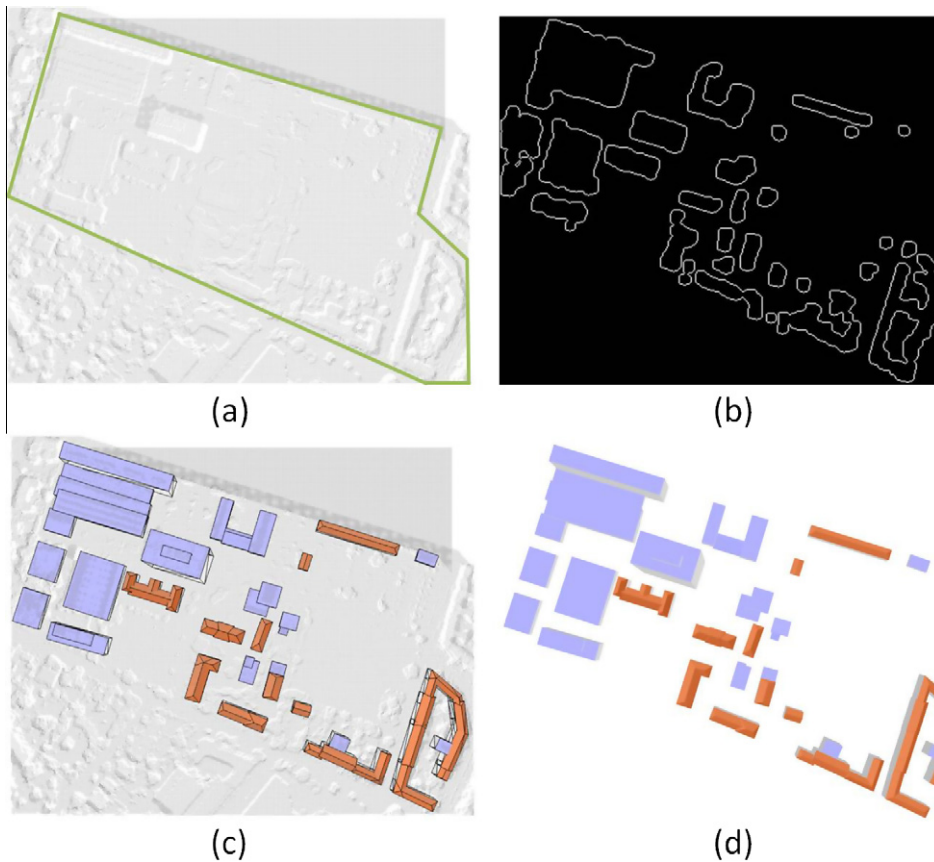


Fig. 10. Building reconstruction: (a) input point cloud; (b) pre-segmentation; (c) primitive (blue/orange: flat/non-flat roofs) extraction and combination; and (d) building models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.2.1. Qualitative assessment

A qualitative assessment, as shown in Fig. 13, is manually conducted by comparing the reconstructed models with the reference image (e.g., Fig. 11e).

We use the following two measures:

1. “Completeness”: the percentage of the reconstructed building parts.
Assuming one building consists of n essential parts and m auxiliary parts, fail in finding one of the essential part will cost

$(1 - m \times 0.1)/n$ of the completeness while each auxiliary part costs 0.1. For instance, the building 7 in Fig. 11, which has two essential parts (the two wings) and one auxiliary part (the terrace), is given the reconstruction completeness of 0.9 because of the missing terrace.

2. “Plausibility”: the score given to the reconstructed shape of the roofs.
Totally false roof type, e.g., flat roof (model) for hipped roof (ground truth), will lose all its partition of score while “inaccurate” reconstruction, e.g., flat roof for saw-tooth roof (buildings

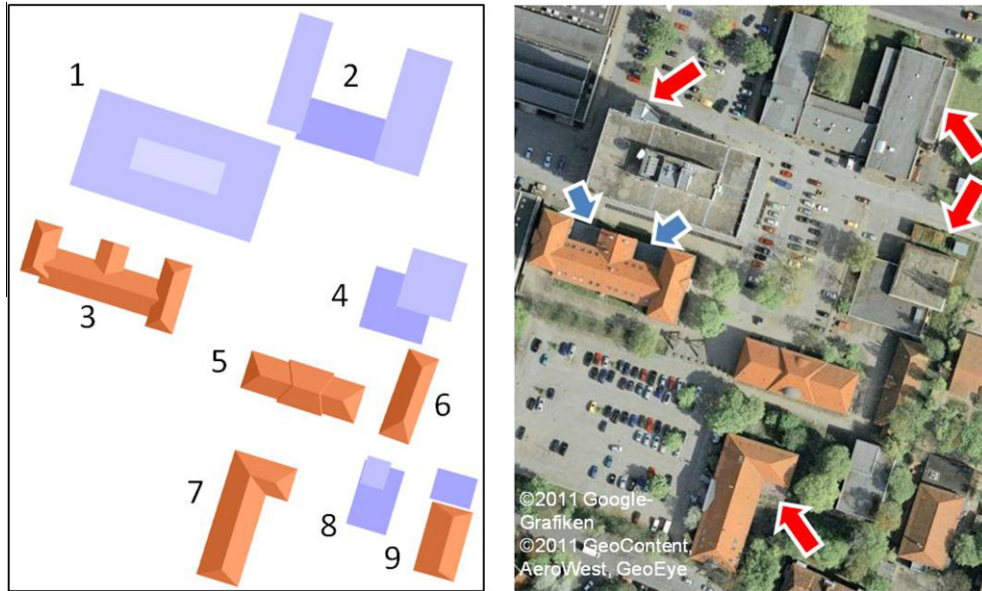


Fig. 11. The comparison of the reconstructed models (left) with the reference image (right).

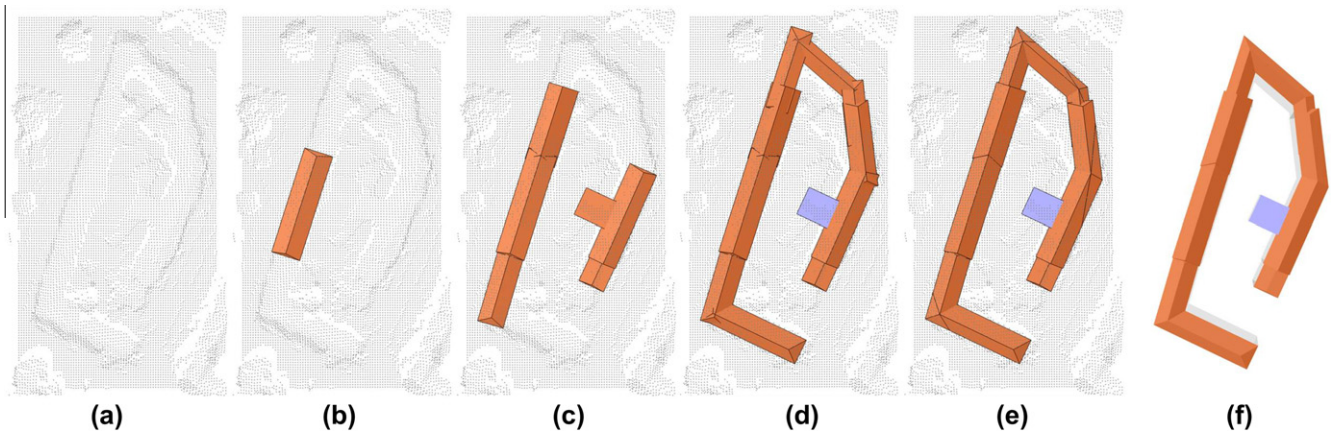


Fig. 12. Reconstruction of a city block: (a) the point cloud; (b) the first primitive; (c) further primitives; (d) all extracted primitives; (e) primitive merging; and (f) the reconstructed building model.

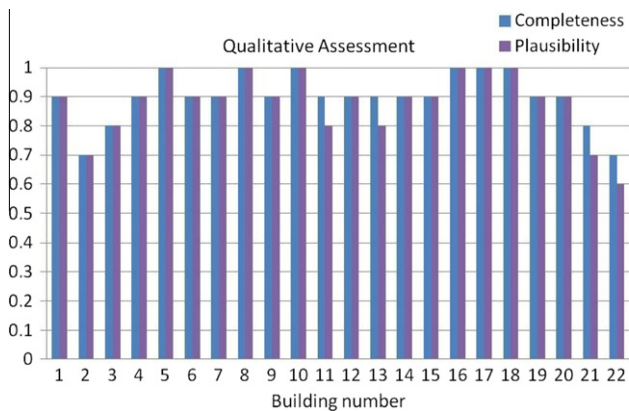


Fig. 13. Qualitative assessment: completeness and plausibility. Building 2 has 0.7 as three auxiliary parts of the east wing are missing. Building 11, 13, 21 and 22 have the penalty of 0.1 in “plausibility” because of the inaccurate roof type of one of the primitives.

11 and 13) or slightly hipped roof for gable roof (buildings 21 and 22), will have a penalty of 0.1. The same weight is given to the building parts as in the “completeness”. The “plausibility” is therefore no larger than “completeness”. Fig. 13 shows the results (for building numbers please refer to Fig. 14).

6.2.2. Quantitative assessment

For a quantitative measurement of the reconstruction errors we, as shown in Fig. 14, derive footprints of the building models and compare them with the manually surveyed cadastral map.

The precision and accuracy of the reconstruction are calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}; \quad \text{Accuracy} = \frac{TP}{TP + FP + FN} \quad (16)$$

with

TP: True Positive, regions of ground plan that have been reconstructed;

FP: False Positive, incorrect reconstructed regions;

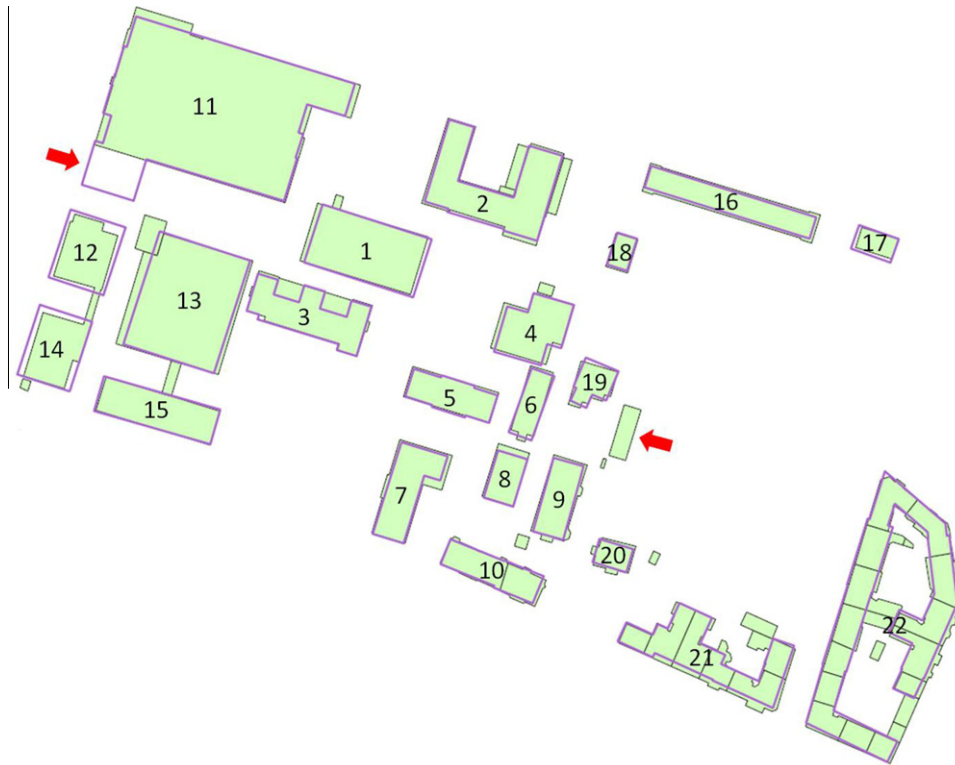


Fig. 14. The footprints (lilac) of the reconstructed models is compared with the cadastral map (green) for a quantitative assessment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

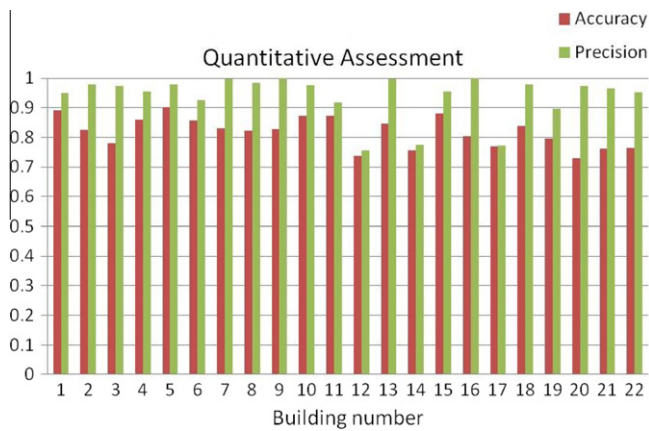


Fig. 15. Quantitative assessment: accuracy and precision.

FN: False Negative, regions of ground plan that have not been covered.

The results are shown in Fig. 15. Referring to the qualitative measures (cf. Fig. 13), the “completeness” of reconstruction may directly influence the quantitative assessments, particularly the “accuracy” (e.g., buildings 2, 3, 21 and 22), which is more sensitive by taking the False Negative regions (the missing building parts) into account. “Plausibility” evaluates only the roof shape and has less impact to the footprints coverage. The roof types themselves, however, do influence the performance. Flat roofs (e.g., buildings 12, 14 and 17) generally show more errors than hipped roofs. The latter may have more accurately extracted azimuths (because of ridges, cf. Section 3.2) and embedded constraints like symmetry to ensure better reconstruction of the footprints.

The assessments show that the essential parts of all the buildings, have been reconstructed with the average values of 82% and 94% for the precision and accuracy. Furthermore, we notice that a newly demolished building as well as a constructed building part (Fig. 14, red arrows) are also correctly presented in the result, which gives a potential possibility of using laser scanning data to update the existing cadastral maps.

6.3. Stability in the results

The generative modeling is driven by the random sampling, which implies we will not obtain the exactly same result every time we rerun the program. Furthermore, we prefer more efficient search and the sampling will stop when the z-error is stable. To evaluate the stability of the sampling we repeat the experiment with the same data points and search settings (priors and iteration limits) for multiple times and observe the results. Six most different results are selected from 20 independent runs and shown in Fig. 16. First, the deviations of the output models are quite slight in comparison with the roof size. Second, all the trials present robustness against the tree clutter and occlusion shown in Fig. 1 (right). We compare the individual parameters as well as the z-errors (red) for one of the primitive (the upper one) in Fig. 16 (right). The deviation of most of the parameters are under 2% and the z-errors show less difference either, i.e., anyone of the results can be identically qualified.

The relatively large deviation (approximate 20% of 6 m) of the hip-depth, however, demonstrates in certain extent the possible instability. The hipped end of the roof has much fewer scanning points and therefore higher data uncertainty. These two different models (trials 3 and 4) are considered fitting the data points equally well because of the similar z-errors.

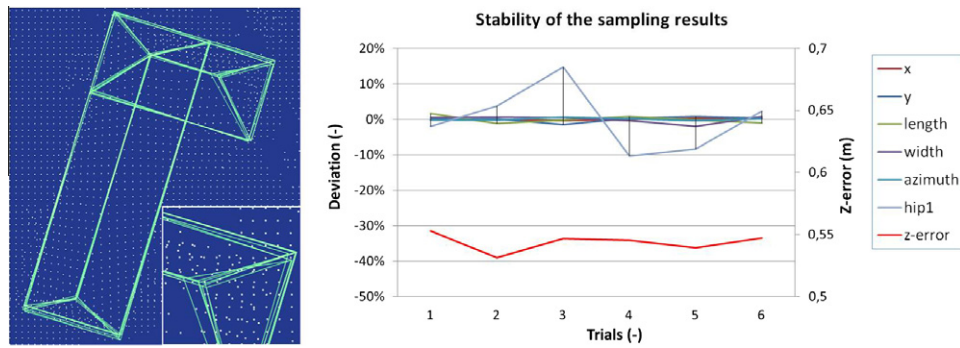


Fig. 16. Stability of the reconstruction: the roof models generated by rerunning the program multiple times (left) and the deviations of roof parameters as well as the z-error (right).

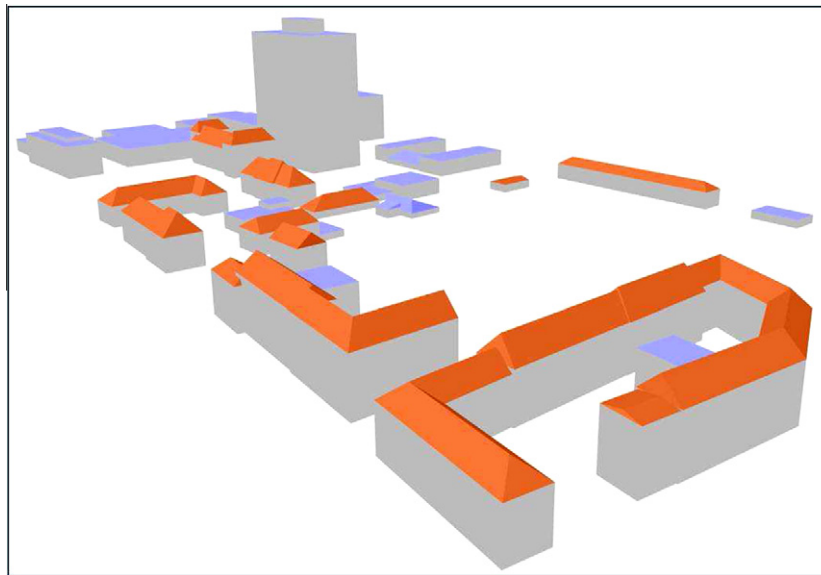


Fig. 17. Building models derived from the extracted roofs.

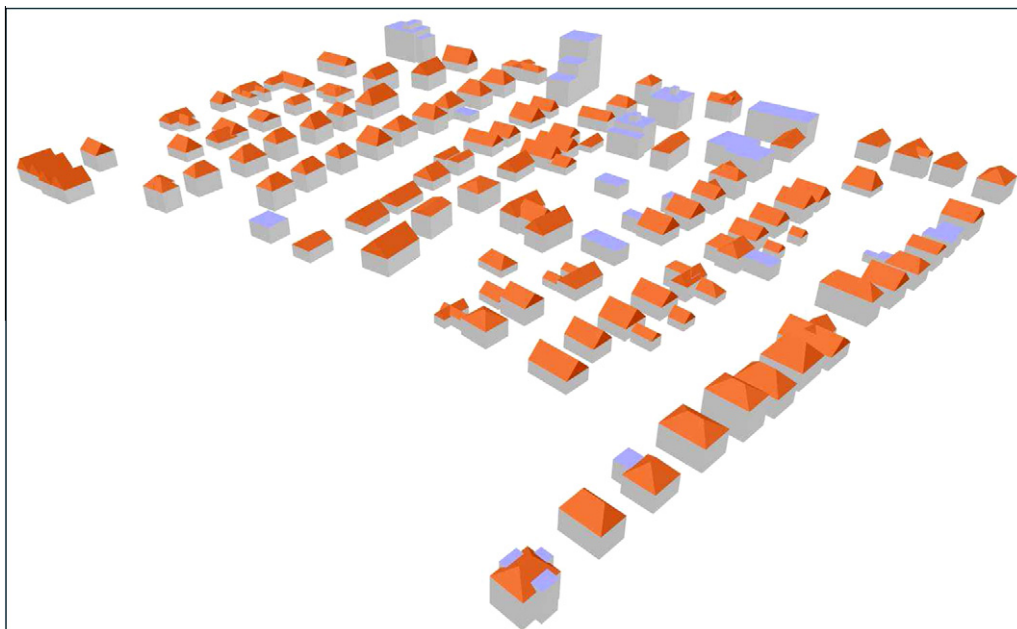


Fig. 18. Reconstruction of an urban area of 0.4 km².

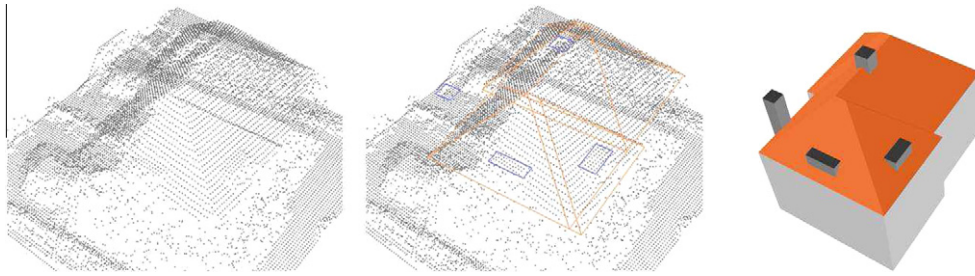


Fig. 19. High-density data reconstruction: the point cloud (left), the extracted primitives (middle) for roof (orange) and superstructures (blue), and the reconstructed building model (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

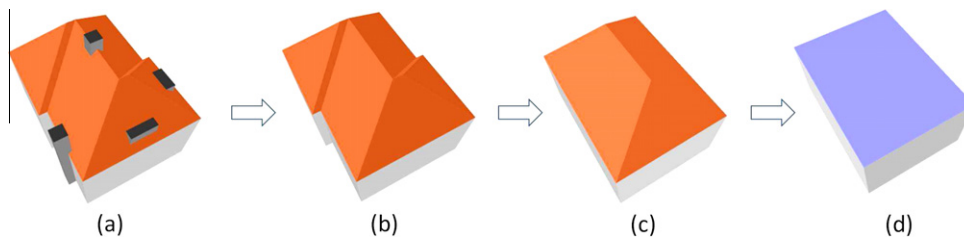


Fig. 20. An example of building model generalization.

6.4. Building models and other datasets

3D building models, as shown in Fig. 17, can be derived from the roof models by extruding the eave contours to the ground.

Another reconstruction result of a larger urban area of Oldenburg, Germany, is presented in Fig. 18. The total runtime of this data set (0.5 m raster, approximate 0.4 km², 96 buildings) is about 60 min.

High resolution data provides denser points representing smaller objects on the roof. As shown in Fig. 19, superstructures, e.g., chimney and dormers, have been reconstructed from a point cloud with the average density of about 0.18 m. Simple flat roof (F1) and garble roof (H4) are used as primitives for the superstructures. The search is limited just in the area above the base roof and more constraints for the azimuth and size can also be derived accordingly.

7. Extension of the approach for building generalization

The presented approach for building reconstruction can also be adapted to create building models in different levels of detail (LoDs). The proposed primitive library and the modeling rules (cf. Section 3) provide a good basis. Instead of geometrically modifying existing high resolution models, which is the usual way for determining different LoDs, we re-generate new models with corresponding descriptions. We modify the employed primitives and even the statistical search scheme for different LoDs. For the selection of primitives it is easy to derive a multi-stage simplification routine, e.g., G2 → H1 → H4 → F1 (cf. Fig. 2). The primitive library as well as the jump routine is designed considering the number of parameters, which also implies their complexity and geometric inheritance (cf. Fig. 8). For complex buildings that contain multiple parts, the combinations of primitives can be replaced with fewer and simpler primitives for simplification. An example can be found in Fig. 20.

As mentioned before, in our approach, the simplified models are not obtained using a geometrical transformation of a higher LoD but are newly generated by statistical sampling with reduced complexity and numbers of primitives. As shown in Fig. 20, in the case of lower resolution the model (b) is achieved by just skipping the

search of superstructures (cf. Section 4.1); (c) is generated by re-sampling in the roof points with only one primitive instead of a flexible number of them; LoD1 model (d) can be seen as using the simple flat roof model to fit the scanning data. By these means a flexible multi-stage simplification can be conducted and the models of every level maintain integrity and regularity. This approach corresponds to the “star”-approach in generalization, where the different representations are generated from one original representation (in this case the raw data in terms of point clouds). It generates homogeneous representations in different LoDs, as it is based on the same set of primitives, which are selected based on geometric criteria to differentiate varied resolutions.

8. Conclusions and outlook

In this paper we have proposed a hybrid generative statistical framework to automatic extraction and reconstruction of building roofs from airborne laser scanning point clouds. The main contributions of this work can be summarized as follows:

- A novel primitive-based modeling scheme with combination and merging rules designed to allow primitive overlapping.
- Generative statistical reconstruction driven by reversible jump MCMC with explicit model selection methods.
- A new extension of generative models for building generalization/multi-scale representations.

Additionally, a pre-segmentation based on blob detection is conducted for the efficient processing of large urban scenes. By all these means the reconstruction becomes robust against data flaws and clutter objects and a plausible result is guaranteed.

Although we have shown the power and flexibility of the top-down method, it has its own issues of uncertainty and instability. Also the completeness of the reconstruction, as shown in Section 6.1, is influenced by the prior knowledge (e.g., building 3 in Fig. 11) and the scene complexity (e.g., buildings 21 and 22 in Fig. 14). Concerning possible future work we, therefore, first consider integrating additional bottom-up information. Cadastral

maps, for instance, will provide reliable initial values for the model parameters and help to segment the point cloud more reasonably concerning individual buildings. The reconstruction results could become more complete and stable while the computational complexity is reduced.

In this work our primitive library contains only planar roofs with at most three differing height levels. New entries, e.g., flat roofs in the forms of triangle and ellipse, domes, cones, and other curved shapes, can be added to represent the more sophisticated constructions like churches, exhibition centers and stadia.

Acknowledgement

We thank the anonymous reviewers for their helpful comments. The research described in this paper was funded by the Deutsche Forschungsgemeinschaft (DFG) under grant BR2970/2-1. The Oldenburg data used in the experiment were kindly provided by the Landesvermessung und Geobasisinformation Niedersachsen (LGLN).

References

- Akaike, H., 1973. Information theory and an extension of the maximum likelihood principle. In: *Second International Symposium on Information Theory*. Akademiai Kiado, Budapest, Hungary, pp. 267–281.
- Brenner, C., 2005. Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation*, Theme Issue on Data Quality in Earth Observation Techniques 6 (3–4), 187–198.
- Green, P., 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 711–732.
- Huang, H., Brenner, C., 2011. Rule-based roof plane detection and segmentation from laser point clouds. In: *Joint Urban Remote Sensing Event (JURSE)*, 11–13 April. IEEE, Munich, Germany, pp. 293–296.
- Huang, H., Brenner, C., Sester, M., 2011. 3D building roof reconstruction from point clouds via generative models. In: *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, 1–4 November. ACM Press, Chicago, IL, USA, pp. 16–24.
- Huang, H., Sester, M., 2011. A hybrid approach to extraction and refinement of building footprints from airborne lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (Part 4/ W25), 153–158.
- Kada, M., McKinley, L., 2009. 3d building reconstruction from lidar based on a cell decomposition approach. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (Part 3/W4), 47–52.
- Köthe, U., 1996. Local appropriate scale in morphological scale-space. *Fourth European Conference on Computer Vision (ECCV)*, 15–18 April, vol. I. Springer, Cambridge, UK, pp. 219–228.
- Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M., 2010a. Structural approach for building reconstruction from a single DSM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (1), 135–147.
- Lafarge, F., Gimel'farb, G., Descombes, X., 2010b. Geometric feature extraction by a multi-marked point process. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (9), 1597–1609.
- Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision* 99 (1), 69–85.
- Matei, B., Sawhney, H., Samarasekera, S., Kim, J., Kumar, R., 2008. Building segmentation for densely built urban regions using aerial lidar data. In: *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 24–26 June. IEEE Computer Society, Anchorage, AK, USA, pp. 1–8.
- Meng, X., Wang, L., Currit, N., 2009. Morphology-based building detection from airborne lidar data. *Photogrammetric Engineering & Remote Sensing* 75 (4), 437–442.
- Ortner, M., Descombes, X., Zerubia, J., 2007. Building outline extraction from digital elevation models using marked point processes. *International Journal of Computer Vision* 72 (2), 107–132.
- Oude Elberink, S.J., Vosselman, G., 2011. Quality analysis on 3d building models reconstructed from airborne laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (2), 157–165.
- Poullis, C., You, S., 2009. Automatic reconstruction of cities from remote sensing data. In: *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 20–25 June. IEEE Computer Society, Miami, FL, USA, pp. 2775–2782.
- Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2008. Automated delineation of roof planes in lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (Part 3/W19), 221–226.
- Sampath, A., Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing* 48 (3), 1554–1567.
- Schnabel, R., Wessel, R., Wahl, R., Klein, R., 2008. Shape recognition in 3d point-clouds. In: *The 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 4–7 February. Plzen-Bory, Czech Republic.
- Serra, J., 1983. *Image Analysis and Mathematical Morphology*. Academic Press, New York, NY, USA.
- Verma, V., Kumar, R., Hsu, S., 2006. 3d building detection and modeling from aerial lidar data. *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 17–22 June, vol. 2. IEEE Computer Society, New York, NY, USA, pp. 2213–2220.
- Vosselman, G., 2009. Advanced point cloud processing. In: *Fritsch, D. (Ed.), Photogrammetric Week '09*. Heidelberg, Germany, pp. 137–146.
- Zhou, Q.-Y., Neumann, U., 2012. 2.5D building modeling by discovering global regularities. In: *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 16–21 June. IEEE Computer Society, Providence, RI, USA, pp. 326–333.