Extraction and Refinement of Building Faces in 3D Point Clouds

Melanie Pohl, Jochen Meidow and Dimitri Bulatov

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB),
Gutleuthausstr. 1, 76275 Ettlingen, Germany

ABSTRACT

In this paper, we present an approach to generate a 3D model of an urban scene out of sensor data. The first milestone on that way is to classify the sensor data into the main parts of a scene, such as ground, vegetation, buildings and their outlines. This has already been accomplished within our previous work. Now, we propose a four-step algorithm to model the building structure, which is assumed to consist of several dominant planes. First, we extract small elevated objects, like chimneys, using a hot-spot detector and handle the detected regions separately. In order to model the variety of roof structures precisely, we split up complex building blocks into parts. Two different approaches are used: To act on the assumption of underlying 2D ground polygons, we use geometric methods to divide them into sub-polygons. Without polygons, we use morphological operations and segmentation methods. In the third step, extraction of dominant planes takes place, by using either RANSAC or J-linkage algorithm. They operate on point clouds of sufficient confidence within the previously separated building parts and give robust results even with noisy, outlier-rich data. Last, we refine the previously determined plane parameters using geometric relations of the building faces. Due to noise, these expected properties of roofs and walls are not fulfilled. Hence, we enforce them as hard constraints and use the previously extracted plane parameters as initial values for an optimization method. To test the proposed workflow, we use both several data sets, including noisy data from depth maps and data computed by laser scanning.

Keywords: building detection, building reconstruction, refinement, roof details, depth map

1. INTRODUCTION

1.1 Motivation

Due to the fast progress in sensor technology in the recent decades, a huge amount of high resolution sensor data of urban terrain, such as aerial photos and videos, laser scans etc. has been captured and is available for a wide use. This is certainly also driven by numerous applications, such as automated navigation in 3D urban terrain, urban planning, disaster management, and reconnaissance. Semantic representations of urban terrain are needed to compress sensor data and to create virtual models allowing interoperability and interpretation. An example of such interoperable models is given by our previous work.[1] It is shown how the data containing millions of points or several high resolution images can be compressed to up to several dozens of surface polygons, enriched with context information, and exported into a simulation system. Of course, while being modeled manually, these models usually look more appealing and are more accurate than the data processed automatically because the level of detail can be arbitrarily high if sufficient man-power and/or time for modeling is available. However, especially for time- and cost-critical applications, it is more valuable to perform a fully-automated procedure for reconstruction in a short time accepting possible abstractions and short-comings of quality. For a building roof, for example, one can model every chimney and every roof tile. Alternatively, one may assume that it consists of one or several dominant planes. In a yet lesser level of detail, one is just interested about the building height and so a roof is simplified to a plane. Which kind of representation is more appropriate, depends on the application.

Generally, any procedure of modeling, or abstraction, is a balance act between the quality, accuracy etc. of the data and the assumptions about the model. If model assumptions are neglected and the result of a modeling procedure essentially trusts in the data, several artifacts that are not compatible with a model can be expected. Hence, the larger the percentage of outliers is (i.e. data severely inconsistent with the model), the harder it is – for any method – to eliminate the negative effects caused by these outliers. Contrary, if the model assumptions are given too much weight and the data is rather unimportant, the danger grows to end up in a fantasy world. In case of urban terrain modeling, some reasonable assumptions consider the kinds of objects to be modeled (the two classes *building* and *tree*[1] were extended, by the classes *meadow*, *street* and

{melanie.pohl, jochen.meidow, dimitri.bulatov}@iosb.fraunhofer.de

Image and Signal Processing for Remote Sensing XIX, edited by Lorenzo Bruzzone,
Proc. of SPIE Vol. 8892, 88920V · © 2013 SPIE · CCC code: 0277-786X/13/$18
doi: 10.1117/12.2028625

forest area in our current work), piecewise planarity/smoothness of objects, and their metric properties (size, area, elevation etc.). Especially in case of building walls and roof details, we have and wish to exploit properties that usually characterize man-made objects, e.g. symmetry, parallelism and orthogonality.

1.2 Contribution

This paper is dedicated to building reconstruction from a segmented point cloud. On this, we consider regular structures of man-made objects and formulate these as a hard-constrained optimization problem. The process of building detection, i.e. segmentation of the whole point cloud, is not a focus of our contribution, so only a brief overview will be given in Sec. 2. The procedure of building wall/roof detail analysis is subdivided into four sub-modules because before the mentioned constraints can be applied, plenty of preliminary work has to be done. The crucial step of this preliminary work requires to compute dominant planes, for which we may need to subdivide too complex buildings and to identify small roof details that degrade computation of dominant planes. Starting with the extracted dominant planes and their intersection points, we cannot expect the models to fulfill the requirements of geometric properties. On the contrary, the consistency between geometry and topology must be enforced by geometric reasoning.

2. SOLUTION STATEMENTS

In this paper, we introduce an approach to obtain boundary representations of buildings that fulfill the conditions of orthogonality and parallelism. Two situations will be regarded: On the one hand, for every position (x, y), one height value is available. In our previous publications, we worked with 2.5D digital elevation models because points observed from considerable altitudes can be resampled into an elevation function $z(x, y)$ without significant loss of information. This elevation function can be interpreted as an image. On the other hand, in some cases, more than one height value can be assigned to one position, namely at building walls or trees. In this case, we need to decide whether this 3D data becomes rasterized into an elevation image or is treated as it is – a 3D point cloud.

In Sec. 3, we will introduce tools for processing both 3D data from airborne laser scans (ALS) and 2.5D digital elevation models (DEM) from multi-view configurations of 2D images taken from different positions. A more detailed view to the data is given in Sec. 5.

2.1 Detection of buildings in DEMs

The process of building detection starts by setting elevation values for a few uncovered pixels. To do it, we solve a partial differential equation. Next, we obtain the digital terrain model (DTM) by detecting first some ground points. These are local minima of the DEM that are found by a filtering procedure. Then, a robust 2.5D surface[2] that approximates all ground points is calculated.[3]

A thresholding of the difference image (DEM – DTM) according to its heights delivers elevated regions. We assume that the larger elevated regions of this difference either correspond to buildings or to vegetation. We also assume that a digital orthophoto is available. In order to filter out vegetation, several regions that correspond to isolated trees are used as training data to extract their characteristic colors (or their combination, compare Normalized Difference Vegetation Index NDVI) and to denote all pixels with similar colors to belong to vegetation.[3] Isolated trees can be extracted interactively or by using model assumptions, such as gradient distribution within neighbored pixels (rather constant for buildings while varying for tree-like regions) or by computation of straight line segments in the (smoothed) DEM and/or orthophoto. Since long straight lines are mostly characteristic for man-made objects, a small amount of lines within an elevated region allows identification of the isolated trees.[4]

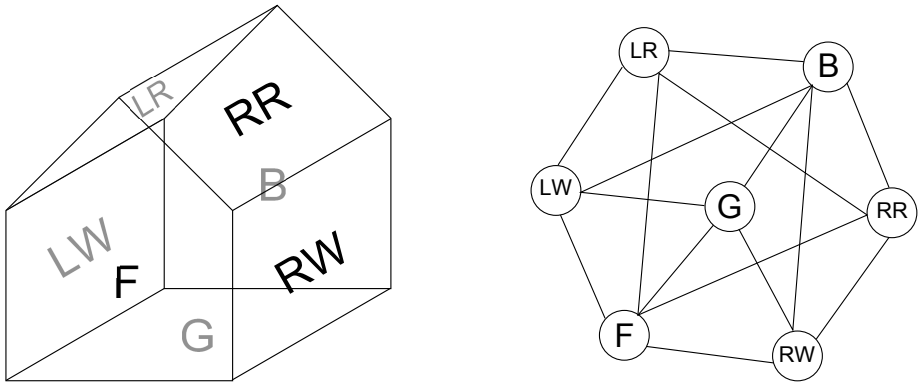
Finally, the task is to perform labeling of buildings. At first, a coherent elevated area is labeled as one building. For a better operability of different steps of the algorithm, it is useful to separate larger building complexes into parts and accordingly to label each part of the building complex for its own. This is done either according to exclude pixels from consideration around jumps in elevation (step-edges[5]) or/and to create a temporary image by a strong morphologic erosion followed by a new labeling and assigning a new label to the missed pixels of the original image. We refer to Sec. 3.2 for a more detailed insight into the default method while Ref. 6 presents also several alternatives. The elevated regions are filtered by their altitude, area and eccentricity. This is done in order to suppress false alarms. The remaining regions are our (labeled) building hypotheses.

The process of building outlining in case of 2.5D DEMs works in the same way as described in Ref. 1. Starting with the minimal bounding rectangle of the binary mask of a building, the contours are refined for each blob by recursive adding and removing rectangular subparts. If a polygon is orthogonal, the axes of the rectangle are given by the dominant directions of the building orientation. Otherwise, a pixel-wise polygonization is performed while straightness constraints and orthogonality constraints are imposed to improve the positions of vertices. The only difference is an assessment of orthogonality for a building: By computing straight lines in the DEM within it, their directions modulo $\pi/2$ are stored in a histogram. The entries of the histogram are weighted by the lengths of the line segments. The peaks of the histogram indicate possible dominant directions. The assessment of orthogonality is carried out by calculating the quotient between the second-best and best value of the histogram. If this value is below 0.5, the dominant direction of the building ground plan is recalculated from the line orientations corresponding to this peak and *all* edges of the ground plan are set to be either parallel or orthogonal to this direction; otherwise, there is no dominant direction. Buildings with a dominant orientation will be considered in Sec. 4.

2.2 Detection of buildings in airborne laser scan data

In case of the data obtained from a fusion of several airborne laser scans, the 3D data is already available. In previous works,[7,8] the process of labeling has taken place not on the per-building but rather on the per-segment level, using a region growing method.[7] For every point, the normal vector and the curvature value are obtained by considering a small neighborhood around this point. Then a local minimum of the curvature is determined and all its neighbors with approximately the same normal vector (up to sign) are added to a cluster. After forming such a cluster, the set of inliers is deleted temporarily and the process begins again. In a consequence, one can differentiate between planar structures of low curvature like building faces and other structures that are assumed to be either vegetation or remaining surface between the buildings. By regarding the normal vector and curvature of each point, it is possible to separate the building faces from the ground as well as the vegetation and to cluster them with the region growing process. For building detection, one can use the results from the region growing for segmentation instead of J-linkage clustering (see Sec. 3.3). Both techniques lead to similar results. In the ideal case, every point cluster corresponds to a building face afterwards.

For building outlining, it is necessary to determine all point clusters that belong to the same building. This can be done by regarding adjacent point clusters. Ideally, every group of adjacent and cohesive point clusters represents one building. First, we determine all points that correspond to each specified cluster. The next step is to derive adjacency relations between these clusters. The necessary condition – that allows to reduce drastically the computation time – for two clusters being adjacent is to assert whether their bounding boxes overlap within a margin of tolerance and the shortest point distance between the classes is smaller than a predefined threshold. To verify if these clusters are indeed adjacent, the well-known Approximate Nearest Neighbor method[9] is applied. Since we focus on the adjacency of the point clusters, we transform the conception of a building into a graph: The vertices (also called 1-cliques in a graph) represent the point clusters and the edges (2-cliques) connect adjacent clusters as shown in Fig. 1.



(a) building faces (b) building graph

Figure 1. Two representations of the building topology. Left: By connecting neighbored corners with edges, the building is represented by its faces. Right: Each face depicts a vertex in a region adjacency graph. Adjacent faces are connected by edges.

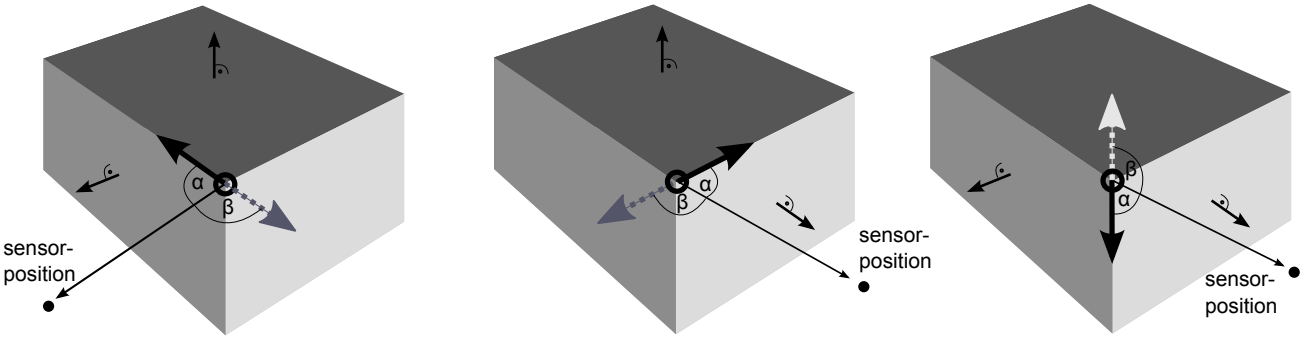


Figure 2. Considering at a time one part of the trihedron, resulting from the intersection of the top and left (left), the top and right (middle) and the left and right (right) face. The direction of the trihedron results from the cross product of the two normal vectors. The angle α is below 90°, β exceeds 90°. Thus, it is possible to orient the arm of the trihedron in the right direction. The sensor position is the sensor position of the data point next to the intersection point.

For a first view of the building just from these adjacent point clusters, we now search for 3-cliques (three vertices connected pairwise by edges). Each 3-clique corresponds to three regression planes (see Sec. 3.3) that are expected to intersect in one "corner" of the building, if it is barred that these three planes form a prism adjustment. By implying neighbored 3-cliques (that means 3-cliques that differ in exactly one of its vertices and do not form a 4-clique), we can determine neighbored corners of the building and therewith edges between two building faces.

For example in Fig. 1 (left), the top left corner in the front results by intersecting the faces F, LW and LR. This corresponds to the 3-cliques in the graph representation F – LW – LR. It has a neighbored 3-clique, namely B – LW – LR that results from replacing F by B. This has the consequence that the top left corner in the front and the top left corner in the back of the building must be connected by an edge.

Using corners and edges, we are able to trace the point clusters that belong to one building with polygons. The polygons form a first skeleton of the buildings. The goal is now to complete the building skeleton by closing the received polygon parts. For that, it is necessary to know about the orientation of the faces, namely inside and outside. We determine a trihedron for each intersection point. If an intersection point does not have enough neighbors to close the polygon, the corresponding trihedron gives us a hint which is the predominant search direction along the building surface. By intersecting always two normal vectors of the triplet, we determine the direction of one arm of the trihedron. We need to ensure the orientation of the arm to be close fitting to the building. This is important to decide about the orientation of the face, where is inside and outside. To fit each arm of the trihedron close to the faces, we use the sensor position as a reference point outside the building. As sensor position we consider the one of the data point next to the intersection point. If the angle between the vector from the intersection point to the sensor position and the corresponding part of the trihedron does not exceed 90°, the trihedron is close-fitting (see Fig. 2).

By regarding these adjacency relations and the graph analysis problem, we can extract point clusters that belong to single building faces as well as clusters that belong to one building. In the view of the building as an adjacency graph, a building consists of one component. By picking one of these components and all points belonging to that component, we can use the method for building outlining introduced in Sec 2.1.

3. EXTRACTION OF BUILDING STRUCTURES

For the extraction of the building structure and its components, we consider three substeps that are explained in the three following subsections. For geometric reasoning, it is useful to have only a small number of planes. Hence, we first exclude small segments, such as chimneys, from consideration and model them in a proper way. Also, we perform a decomposition of the building in the case that it has a too complex structure. Every building is bounded by the outline of its footprint and split this up into its components. For these modules, we will need the elevation image created from the input sensor data. In the case of DEM available, we simply use the DEM as input; in case of laser scans we have to raster the 3D information to determine an image representation. Finally, the single parts of the building are appropriate for plane extraction and the refinement.

3.1 Small elevated objects

The procedure of roof detail analysis starts by extracting small elevated roof objects, for example, chimneys and dormers. This is done first of all in order to reduce the number of plane hypotheses: From the contents of Sec. 4, it will become clear that the desired input should consist of few big, dominant plane segments. Additionally, the computation of dominant planes usually converges faster and yields better results if the point sets are cleaned from these outliers with respect to the dominant planes. Last, the model benefits if small structures are detected in advance and modeled in a proper way.

The algorithm is described in a detailed way in Ref. 10. First, small elevated regions of the DEM are found by applying a hot-spot detector, such as the MSER-operator.[11] Since the input of the MSER-method must be an 8-bit image, those parts of the DEM that were labeled as buildings are discretized in a suitable way. The hot-spot detector then searches for fair regions on a dark background that are our chimney hypotheses. However, since (especially) the DEMs obtained from images are noisy and produce spurious hypotheses, several filters have to be applied on the hypotheses. These filters consider the distance to the building border (because there are many occluded areas with spurious elevation values near building borders) and the variation of texture in the regions around hypotheses (if color images are available), but also the height, area, and form of a chimney.

3.2 Building partitioning

As already mentioned, for various reasons (for example, to avoid ghost planes in the plane determination step; ghost planes are planes that have inliers in different parts of a building without coherent building face) it has been advantageous to work on single building parts, if the building itself is large and of complex structure. So we used different approaches to split up a large and complex building.

Two different methods were tested. Splitting of a building at short diagonals is a very fast method but it is required that the building outline is already available. In addition, the procedure has problems with very crooked buildings or complexes that contains "holes" in their ground polygon, like atriums. An alternative approach is performed by the separation with Random Walks.[12] This method is more flexible but needs more computation time.

Partitioning with polygons. The idea is to subdivide the building along the diagonals of its ground plan polygon. Among all diagonals that lie completely inside the building, the shortest one is selected. If it is shorter than a threshold and both resulting polygons have an area larger than a second threshold, the polygonal chain is cut into two parts by inserting a new diagonal. The procedure is then applied on both remaining subunits. It terminates as soon as no diagonals with mentioned properties can be found anymore.

Partitioning with Random Walks. The alternative approach is performed on the binary level. We assume that a complex building is connected by narrow structures. By a strong morphological erosion, the building is subdivided into subunits. Then we consider the difference of the initial building mask and the remaining mask of the eroded image. The "lost" pixels have now to be assigned to one of the remaining subunits. We decided to use the Random Walks method[12] because it is very robust against noise, especially in DEMs extracted with photogrammetric methods, and works accurate with even weak boundaries. At least one pixel of each subunit (and one of the background) is labeled as seed. The seeds of the subunits are labeled with $k \in \{1, ..., n \in \mathbb{N}\}$ and the seeds of the background are labeled with 0. Next, for each unseeded pixel, a tuple $(p_0, p_1, ..., p_n)$ of probabilities that the pixel belongs to one group of seeds is computed. Afterwards, the regarded pixel is assigned to the label $k = \text{argmax}_{j=\{1,...,n\}}(p_j)$. To compute the probabilities, the image is regarded as a graph where the pixels represent the vertices and the weighted edges result from a 4-neighborhood. The weights indicate a change in elevations and can be computed by a Gaussian weighting function with the pixel intensity difference as input. A random walk along the weighted edges (e.g. weighted gradients) starts in an unseeded pixel and ends in a seeded pixel of the DEM. The probabilities can be computed by solving the combinatorial Dirichlet problem.[12] Because this method is image-based and works with a regular neighborhood, we have to raster the 3D data of the ALS to apply the algorithm.

3.3 Extraction of building faces

After the process of simplification described above, we strive to extract the single faces as well as their parameter vectors. For that, one can use RANSAC methods as introduced in Ref. 13. A reasonable alternative is to use the J-linkage method[14] that partly compensates insufficiences of the multi-model RANSAC. For example, by preferable choice of neighboring points to form a hypothesis and by clustering all the hypotheses afterwards, the presence of ghost planes can be significantly reduced though not completely excluded. Since the idea of J-linkage method is similar to global RANSAC approaches, the

best way to exclude ghost planes lies in a reasonable decomposition of buildings as described in the previous section. The more a building can be decomposed into its core parts, the less ghost planes occur and the results produced with RANSAC are similar to the results of J-linkage.

In the case of the DEM data, we use the J-linkage method on the one hand to cluster the points in one building (part) to building faces and on the other hand to compute a regression plane for each cluster of points. In the ALS data, we can decide whether we want to dispose again the points belonging to one building into clusters or to use the initial segmentation of the region growing and use J-linkage just for the regression part. In our case, we generally used the available results from the region grwoing process but completed them - if necessary - by J-linkage clustering. In each case, the final regression plane is estimated and optimized for the inliers of the results of the J-linkage clustering.

Now, a polyhedral is available and can be used as a first result. But because the obtained planes have an unlimited extension, we need an additional semantic on superposition of several planes. Furthermore, because of numerical insufficiencies, the estimated planes do not necessarily fulfill the usual attitudes of a building, namely parallelism and orthogonality or uniqueness of a cutting point of four planes that corresponds with a corner of the building. Hence, the final step of the workflow is to refine the received plane parameters to enforce orthogonality and parallelism and furthermore use the aforementioned view of a building skeleton.

4. GEOMETRIC REASONING WITH PLANES

To take the uncertainty of the noisy observations into account, we exploit uncertain projective geometry[15] for the reasoning process. This refers to both, the generation and testing of hypotheses for geometric relations as well as the subsequent optimization process. For efficiency, we apply a two-step procedure: After extracting bounding planes, we perform the geometric reasoning with the estimated plane parameters only. Alternatively, a parameter estimation with constraints for observations and parameters can be applied.[16] To enforce the required consistency, we formulate and apply hard constraints only. The following explications are focused on the formulation of the constraints and the optimization process.

4.1 Geometric Constraints

Empirical investigations show the dominance of orthogonality and parallelism in buildings.[17] Since parallelism of planes can be enforced by multiple orthogonality constraints, we restrict ourselves to orthogonality. Additionally, we add the intersection of four planes in one point as a constraint to close gaps in boundary representations. Examples for such a constructions are the corners of a hip roof or a saddle-back roof with T-junctions.

More elaborated, we used the flowing constraints:

- To make the homogeneous representation $\mathbf{A}$ of a plane $\mathcal{A}$ unique, a *spherical normalization* is introduced by requiring $\mathbf{A}_i^\top \mathbf{A}_i = 1$ for all planes $\mathcal{A}_i$.
- The *orthogonality* $\mathcal{A}_i \perp \mathcal{A}_j$ of two planes $\mathcal{A}_i$ and $\mathcal{A}_j$ is enforced by $\boldsymbol{A}_{ih}^\top \boldsymbol{A}_{jh} = 0$ with the homogeneous vectors $\mathbf{A} = (\boldsymbol{A}_h^\top, A_0)^\top$ split into a homogeneous part $\boldsymbol{A}_h$ and Euclidean part A_0.
- *Intersection of four planes in one point* $\mathcal{X} = \mathcal{A}_i \cap \mathcal{A}_j \cap \mathcal{A}_k \cap \mathcal{A}_l$: Bearing in mind that points and planes are dual entities, the formulation of the corresponding constraint is straightforward. Four 3D points are collinear if the determinant of their corresponding homogeneous vectors is zero. Consequently, four planes intersect in one point if $\det([\mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_k, \mathbf{A}_l]) = 0$ holds,[15] a relation which is exploited for the formulation of trifocal constraints for multi-view geometry, too.

4.2 Optimization

For the optimization, we consider P extracted and spherically normalized planes $\boldsymbol{p} = (\mathbf{A}_1^\top, \mathbf{A}_2^\top, \ldots, \mathbf{A}_P^\top)^\top$, their estimated block-diagonal covariance matrix $\boldsymbol{\Sigma}_{pp}$, and the constraints $\boldsymbol{h}(\widehat{\boldsymbol{p}}) = \mathbf{0}$ for the adjusted parameters $\widehat{\boldsymbol{p}}$ derived by hypothesis generation and verification. A solution of the problem is given by an adaption of the Sequential Quadratic Programming.[18] Alternatively, by considering the extracted planes as given observations, the solution is equivalent to the adjustment with constraints for observations only.[19]

Linearization of the constraints yields the first order expansion $\boldsymbol{h}(\widehat{\boldsymbol{p}}) = \boldsymbol{h}(\boldsymbol{p}_0) + \boldsymbol{H}\widehat{\Delta\boldsymbol{p}}$ with the Jacobian $\boldsymbol{H} = (\partial \boldsymbol{h}(\widehat{\boldsymbol{p}}) / \partial \widehat{\Delta\boldsymbol{p}}^\top)$. The adjusted parameters $\widehat{\boldsymbol{p}}$ can be found by correcting given approximate values $\boldsymbol{p}_0$ by $\widehat{\Delta\boldsymbol{p}}$, or by adding

a correction $\widehat{\boldsymbol{v}}$ of the unconstrained parameters $\boldsymbol{p}$, thus $\widehat{\boldsymbol{p}} = \boldsymbol{p} + \widehat{\boldsymbol{v}} = \boldsymbol{p}_0 + \widehat{\Delta \boldsymbol{p}}$ holds and the linearized constraints are $\boldsymbol{h}(\widehat{\boldsymbol{p}}) = \boldsymbol{p}_0 + \boldsymbol{H}\widehat{\boldsymbol{v}} = \boldsymbol{0}$ with $\boldsymbol{h}_0 = \boldsymbol{h}(\boldsymbol{p}_0) + \boldsymbol{H}(\boldsymbol{p} - \boldsymbol{p}_0)$. The Lagrangian is then

$$L = \frac{1}{2}\widehat{\boldsymbol{v}}^\top \boldsymbol{\Sigma}_{pp}^{+}\widehat{\boldsymbol{v}} + \boldsymbol{\mu}^\top(\boldsymbol{h}_0 + \boldsymbol{H}\widehat{\boldsymbol{v}}) \tag{1}$$

with the Lagrangian multipliers $\boldsymbol{\mu}$. Because of the spherical normalization of the unconstrained parameters, the null space $\boldsymbol{K}$ of the covariance matrix $\boldsymbol{\Sigma}_{pp}$ is given by the estimated, unconstrained plane parameters. Thus $\boldsymbol{\Sigma}_{pp}\boldsymbol{K} = \boldsymbol{0}$ holds and the pseudo inverse in (1) can easily be computed by

$$\boldsymbol{\Sigma}_{pp}^{+} = \left(\boldsymbol{\Sigma}_{pp} + \boldsymbol{K}\boldsymbol{K}^\top\right)^{-1} - \boldsymbol{K}\boldsymbol{K}^\top \tag{2}$$

taking $\boldsymbol{K}^\top\boldsymbol{K} = \boldsymbol{I}$ into account. Setting the partial derivatives of (1) equal to zero yields the normal equation system

$$\begin{bmatrix} \boldsymbol{\Sigma}_{pp}^{+} & \boldsymbol{H}^\top \\ \boldsymbol{H} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{v}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{h}_0 \end{bmatrix} \tag{3}$$

to be solved. Within the iterative estimation procedure, the update of the parameters is simply $\widehat{\boldsymbol{p}} = \boldsymbol{p} + \widehat{\boldsymbol{v}}$.

The constraints derived by hypothesis generation and verification can be linearly dependent or even contradicting due to the noise in the observations. However, finding a set of linearly independent and non-contradicting constraints is a prerequisite for the application of the adjustment presented above.[20] A simple greedy algorithm can be used to compile such sets by selecting one constraint after the other and putting it into the minimum set if it is independent on the previous. To make decisions, we keep track on the rank and condition number of the normal equation matrix at hand.

5. EXPERIMENTS

To expose the functionality of the introduced workflow, we will show the results for the 2.5D data set from depth maps obtained from areal surveys and a 3D data set from a laser scan. For the DEM data, we regard a little town in Southern Germany, named Vaihingen. For the ALS data, fusion of airborne laser scans of the village Abenberg (another small town in Germany) is considered.

5.1 2.5D data

We consider the point cloud obtained from the multi-view configuration of seven images showing the village of Vaihingen. This is ISPRS WG III/4* benchmark on urban area reconstruction and we will prove here the principle of our algorithm by showing just a few buildings. The particular challenge are many shadowy areas that make dense reconstruction from images more complicated. Also, because of a high building density, is it hard to separate buildings and roof details from each other and preserve topology.

Some results of the extraction of small elevated objects can exemplarily be seen in Fig. 3. Up to 75% of the small elevated objects can be extracted in the whole scene. More detailed results are given in the original paper.[10]

We consider only a small number of airborne images in which most of the walls are hidden by other buildings and trees or just have an insufficient resolution because of the Nadir view. So, we only consider the rooftops of the buildings for reconstruction and refinement. In the case one wants the building to be complete, there is a possibility to include synthetic walls along the outline of the building ground plan. While just focusing on the roof planes, the number of possible constraints is limited.

The image shows that for the building the heights of the roof ridges are identical, cf. Fig. 4 (second from the left). Therefore, we can apply the constraint $\det([\mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_k, \mathbf{A}_l]) = 0$ leading to the result depicted in a cutout of the intersection point in Fig. 4 on the right.

*ISPRS test project on 3D building reconstruction `http://www.commission3.isprs.org/wg4/`

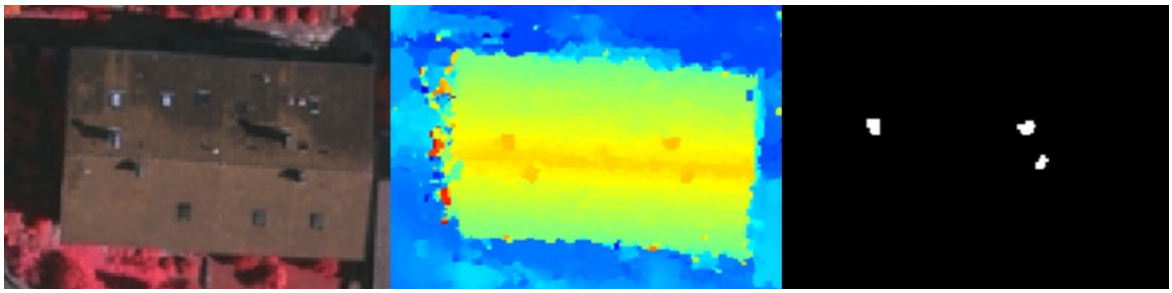


Figure 3. Detection of chimneys. Left: Ortho view, middle: DEM, right: Binary mask of detected chimneys.[10]

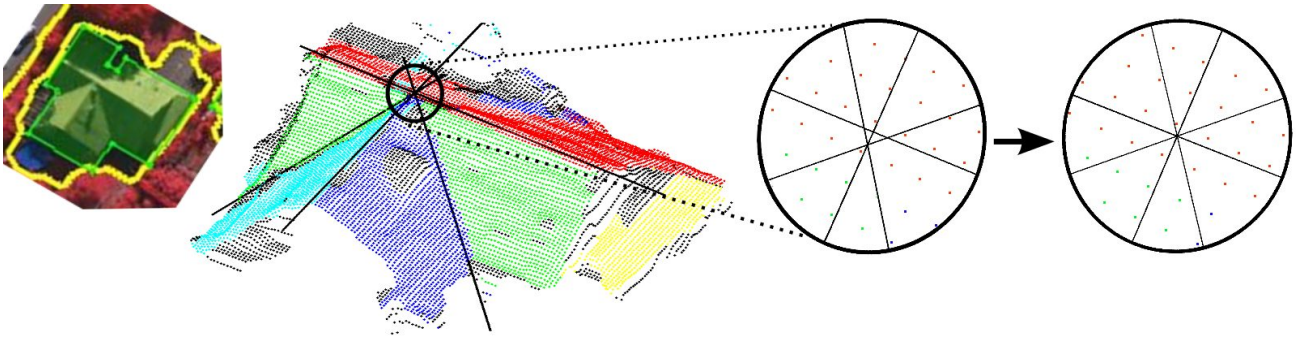


Figure 4. Reasoning for a data set obtained by photogrammetric surface reconstruction. From left to right: Rotated orthoview with vegetation boundary (yellow) and building outlining (green); Point clusters of inliers concerning the generated roof planes and unconstrained cutlines; Close-up of the area around the coincidence of the two ridge parts for the unconstrained cutlines; Close-up of this area after constrained optimization.

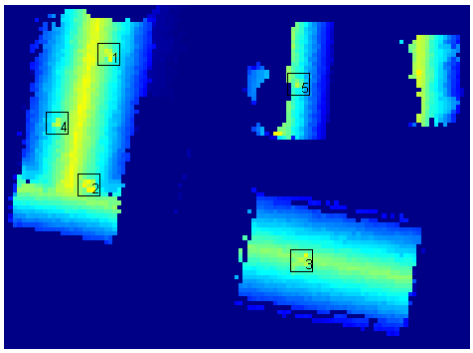


Figure 5. Detection of chimneys in the 3D dataset. All chimneys (denoted by rectangles) in that cutout could be detected.

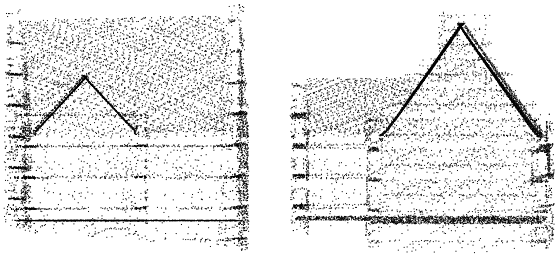


Figure 6. Two views of the extracted point cloud representing a single building with two saddleback roofs, arranged in a L-shape.

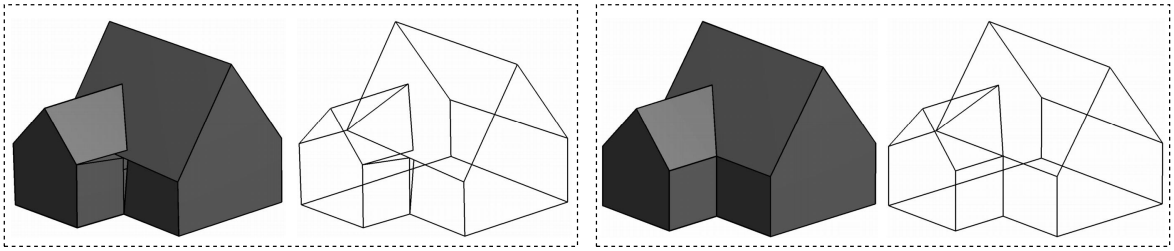


Figure 7. Unconstrained (left) and constrained (right) planes corresponding to the data set depicted in Fig. 6. One clearly sees the gaps and unbalances in the unconstrained model.

5.2 3D data

The "Abenberg 2009" data set is an accumulated point cloud captured by airborne laser scanning.[8] The test site has been approached by a helicopter in a cross pattern, which results in overlapping point clouds. The co-registration and accumulation of the stripes led to a reference data set which features building facades and an average point density of 21 pts/m^2.

For the detection of chimneys, it is necessary to raster the data points into an equally spaced grid, because the presented method is image-based. The resolution has to correspond to the ground sample distance. In that case the input is not the original DEM but a conditioned DEM with several scaling parameters. We chose a little cutout of the whole scene to demonstrate the results (see Fig. 5). All chimneys in this part of the scene could be detected without false alarms.

From the data set, a second subset representing an L-shaped building with a saddle back roof has been extracted, see Fig. 6. In the present example, this leads to subsets representing the eleven planes being the bounding surfaces of the building. Subsequently, the corresponding plane parameters and their corresponding covariance matrices have been estimated. Figure 7 (left pair) shows the unconstrained surface representation of the building with gaps where four planes should intersect.

For the geometric reasoning according to Sec. 4.2, we set up 15 independent constraints: One constraint for the four planes intersecting in one point ($\det([\mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_k, \mathbf{A}_l]) = 0$), and 14 constraints for pairs of planes that are supposed to be orthogonal ($\boldsymbol{A}_{ih}^{\top}\boldsymbol{A}_{jh} = 0$). The result of the geometric reasoning is depicted in Fig. 7 (right pair). The gaps are closed, the eaves are parallel to the bottom plane and the roof ridges orthogonal and parallel respectively to the walls.

6. CONCLUSION AND OUTLOOK

The presented approach allows determining a skeleton or face model of a building out of point clouds collected by various sensor data. We are able to decompose complex buildings into more local parts and therewith enhance the accuracy of the estimated planes that form the faces of the building. Small elevated objects are treated separately in order not to oversmooth them as well as not to falsify the plane estimation. With the terminal refinement of the initial planes, the model fulfills the well known aspects of building faces to be either orthogonal or parallel to each other. The method is suitable both for 2.5D DEMs and for 3D data gained by airborne laser scans and the modular composition of the whole procedure allows to switch on and off the single parts as well as to replace them by other methods, if wished. The results of the substeps of the introduced procedure can be used to develop a view from a rough model (by concerning just the building outlining) via a detailed skeleton (with chimneys and other small elevated objects) to a model that fulfills also the requirements of orthogonality and parallelism of the building edges. This is not only an advantage with regard to optical viewing but also helps to close holes caused by four planes that otherwise would not intersect in one point and is therewith a benefit when it comes to texturing the received surfaces.

For the future work, also the aspect of symmetry shall be included. Furthermore, we will need to automate the determination of the constraints that are necessary for the optimization in the refinement step. Additionally, the optimization of the generated plane hypotheses is dependent from the initialization at the moment. We need to assure that the optimized model still fits to the input data by statistical tests.

In the performed clustering step for the 3D data, not all walls and roofs were detected. Some point clusters exhibit too little planarity and were not identified. Probably, these faces are covered with vegetation or are spuriously filtered out because of distortions in the co-registration. At the moment, every wall that has not sufficient planarity is assigned to the class *vegetation* or *other surface*. So, the clustering methods need to be adapted to deal with insufficient planarity in order to reduce the number of undetected building faces.

One further issue of our future work includes considering characteristic edges in images near the cut-lines of the planes representing building faces. Each of the constraints according to Sec. 4 is formulated in 3D and reduces the number of degrees of freedom for all planes by one. By introducing constraints on plane parameters such that these pairs of lines coincide in 2D, we move into the direction of a mixed 2D/3D approach and provide constraints that reduce the number of degrees of freedom by two.

Acknowledgement. We wish to thank Marcus Hebel for providing the Abenberg data and the segmentation of the scene by the region growing algorithm.

REFERENCES

[1] Bulatov, D., Solbrig, P., and Wernerus, P., "Ad-hoc model acquisition for combat simulation in urban terrain," in [*SPIE Remote Sensing*], 85380G, 10 p (2012).

[2] Bulatov, D. and Lavery, J. E., "Reconstruction and texturing of 3D urban terrain from uncalibrated monocular images using L1 splines," *Photogrammetric engineering and remote sensing* **76**(4), 439–449 (2010).

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

[4] Bulatov, D., Solbrig, P., Gross, H., Wernerus, P., Repasi, E., and Heipke, C., "Context-based urban terrain reconstruction from UAV-videos for geoinformation applications," in [*Proc. of Conference on Unmanned Aerial Vehicle in Geomatics, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*], **38**, 75–80 (2011).

[5] Gross, H., Thönnessen, U., and v. Hansen, W., "3D-Modeling of urban structures," in [*Proc. of Joint Workshop of ISPRS/DAGM Object Extraction for 3D City Models, Road Databases, and Traffic Monitoring CMRT05, International Archives of Photogrammetry and Remote Sensing*], **36, Part 3W24**, 137–142 (2005).

[6] Geibel, R. and Stilla, U., "Segmentation of laser-altimeter data for building reconstruction: Comparison of different procedures," *Int. Arch. of Photogrammetry and Remote Sensing* **33 part B3**, 326–334 (2000).

[7] Hebel, M. and Stilla, U., "Simultaneous calibration of als systems and alignment of multiview LiDAR scans of urban areas," *Geoscience and Remote Sensing, IEEE Transactions on* **50**(6), 2364–2379 (2012).

[8] Hebel, M., Arens, M., and Stilla, U., "Change detection in urban areas by direct comparison of multi-view and multi-temporal ALS data," in [*Photogrammetric Image Analysis 2011*], Stilla, U., Rottensteiner, F., Mayer, H., Jutzi, B., and Butenuth, M., eds., *LNCS* **6952**, 185–196, Springer, Heidelberg (2011).

[9] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y., "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)* **45**(6), 891–923 (1998).

[10] Bulatov, D. and Pohl, M., "Detection of small roof details in image sequences," in [*Image Analysis*], 601–610, Springer (2013).

[11] Matas, J., Chum, O., Urban, M., and Pajdla, T., "Robust wide baseline stereo from maximally stable extremal regions," *Image and Vision Computing* **22**(10), 761–767 (2002).

[12] Grady, L., "Random walks for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(11), 1768–1783 (2006).

[13] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**(6), 381–395 (1981).

[14] Toldo, R. and Fusiello, A., "Robust multiple structures estimation with J-linkage," in [*Computer Vision–ECCV 2008*], 537–547, Springer (2008).

[15] Heuel, S., [*Uncertain projective geometry. Statistical reasoning in polyhedral object reconstruction*], vol. 3008 of *Lecture Notes in Computer Science*, Springer (2004).

[16] Meidow, J., Beder, C., and Förstner, W., "Reasoning with uncertain points, straight lines, and straight line segments in 2d," *ISPRS Journal of Photogrammetry and Remote Sensing* **64**(2), 125–139 (2009).

[17] Loch-Dehbi, S. and Plümer, L., "Automatic reasoning for geometric constraints in 3D city models with uncertain observations," *ISPRS Journal of Photogrammetry and Remote Sensing* **66**(2), 177–187 (2011).

[18] Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A., "Bundle adjustment – a modern synthesis," in [*Vision algorithms: theory and practice*], Triggs, B., Zisserman, A., and Szeliski, R., eds., *Lecture Notes in Computer Science* **1883**, 298–372, Springer-Verlag (2000).

[19] Koch, K.-R., [*Parameter estimation and hypothesis testing in linear models*], Springer, Berlin, 2 ed. (1999).

[20] Förstner, W., "Mid-level vision processes for automatic building extraction," in [*Automatic extraction of man-made objects from aerial and space images*], Gruen, A., Kuebler, O., and Agouris, P., eds., 179–188 (1995).