

FINDING CUBOID-BASED BUILDING MODELS IN POINT CLOUDS

William Nguatem, Martin Drauschke, Helmut Mayer

Institute of Applied Computer Science, Bundeswehr University Munich
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
william.nguatem,martin.drauschke,helmut.mayer@unibw.de

Commission III/2

KEY WORDS: Point Cloud Segmentation, Surface Detection, Building Recognition, Building Reconstruction

ABSTRACT:

In this paper, we present an automatic approach for the derivation of 3D building models of level-of-detail 1 (LOD 1) from point clouds obtained from (dense) image matching or, for comparison only, from LiDAR. Our approach makes use of the predominance of vertical structures and orthogonal intersections in architectural scenes. After robustly determining the scene's vertical direction based on the 3D points we use it as constraint for a RANSAC-based search for vertical planes in the point cloud. The planes are further analyzed to segment reliable outlines for rectangular surfaces within these planes, which are connected to construct cuboid-based building models. We demonstrate that our approach is robust and effective over a range of real-world input data sets with varying point density, amount of noise, and outliers.

1 INTRODUCTION

The derivation of building models from 3D point clouds is a very active research topic in computer graphics, photogrammetry and geoinformatics (Wahl et al., 2008; Meixner and Leberl, 2011; Huang et al., 2011). Research on the derivation of building models is focused on (a) reduction of the amount of data, (b) automatic object recognition and reconstruction from unstructured 3D point clouds, and (c) integration of semantics into building and city models. Most of the work focuses on the analysis of terrestrial and airborne LiDAR point clouds, which are highly accurate and very dense. In contrast to LiDAR point clouds, point clouds reconstructed from multiple images (Snavely et al., 2006; Hiep et al., 2009; Agarwal et al., 2010; Bartelsen and Mayer, 2010; Frahm et al., 2010) are less precise and not so dense, but reliable enough to recognize objects and their parts by visual inspection.

Since point clouds can be generated very cost effectively from images, we see a need for an automated analysis of these point clouds. LOD 1 models representing buildings as cuboids with planar and rectangular surfaces can be used for rectifying images by means of homographies on the facades which can be used for further facade analysis, e.g., the detection of windows (Lee and Nevatia, 2004; Reznik and Mayer, 2008), or the description of repetitive patterns (Wenzel et al., 2007). Furthermore, LOD 1 models will drastically reduce the amount of data. Triangulated meshes of millions of points as obtained, e.g., by Poisson surface reconstruction (Kazhdan et al., 2006) can easily reach several Gigabytes of memory leading to slow or no visualisation possibilities. Contrarily, cuboid-based building models consisting of a few planar, rectangular surfaces only need a few Kilobytes.

The recognition and reconstruction of buildings or their parts is either done based on domain knowledge, e.g., concerning the shape of the objects of interest (Huang et al., 2011), on the relation to other objects of the scene (Schmittwilken et al., 2009), or on the piecewise intersection of planar, cylindrical, toric and conical surfaces (Schnabel et al., 2007). In our work, we want to find cuboid-based buildings, i.e., which either have the shape of a cuboid, or can be decomposed into several, possibly overlapping cuboids. This is the first step towards a hierarchical modeling of detached buildings in rural and suburban areas where build-

ings typically stand alone or in small groups. In urban downtown scenes, our approach will lead to building blocks rather than single buildings.

Our approach is based on the detection of the principal cuboid from 3D point clouds obtained during image orientation. As opposed to LiDAR data, such 3D points are often not very dense. We assume that the four main walls of a building correspond to four faces of a cuboid without top and bottom. By further representing every face by a plane, we obtain a unique pair of parallel orthogonal planes for our principal cuboid.

The cuboid estimation algorithm thus can be reduced to the determination of pairs of unique orthogonal planes in the given data. This is conducted by means of RANSAC (Fischler and Bolles, 1981). Restricting the first stage of modeling 3D buildings to this case of a rectangular groundplan, our approach robustly produces reliable results from which further refinements can be made, progressively increasing the level of detail. This is illustrated by experimental results, using terrestrial data collected at a small village in Southern Germany.

The paper is structured as follows. In Section 2 we give an overview on existing approaches for detecting planes and reconstructing buildings in 3D point clouds. We describe the concept of our algorithm in Section 3 and present algorithmic details in Section 4. After showing and discussing experiments in Section 5, we conclude with future work in Section 6.

2 RELATED WORK

In this section, we summarize related work. We discuss general approaches for the detection of geometric primitives in 3D point clouds w.r.t. the detection of vertical planes. If the vertical direction is known, it can be used as a constraint for plane detection, thus, we also comment on research in this direction. Because we are interested in extending our work towards the detection and modeling of building parts, we also introduce related work on this topic.

There exist several approaches for building recognition and reconstruction from point clouds which depend on precise and very

dense 3D data. E.g., Vosselman (2009) discusses recent methods which rely on LiDAR point clouds with densities of 20 to 50 points per m^2 from airborne scanners or a couple of thousand points per m^2 from terrestrial scanners. For such very dense point clouds typically 3D information of each point's neighborhood can be found by clustering the points or by determining planar surface patches. Yet, it is not suited for noisy and not very dense point clouds derived from image matching.

Vosselman and Klein (2010) review point cloud segmentation, i.e., the detection of subsets of points which form geometric primitives, such as planes, cones, cylinders, or tori. They group the methods for point cloud segmentation into three different types: The first type is based on the Hough transform (Hart, 2009), where the best hypothesis is detected in a derived parameter space. It is often used when searching for planes and other geometric primitives with a low-dimensional parameter space (Vosselman and Klein, 2010). The second type consists of all approaches which reconstruct surfaces based on edge detection or region growing (Vieira and Shimada, 2005). Again, neighborhood information is used to determine normal vectors at each point, which are used to estimate surface curvature values. Also the recent work by Schindler and Förstner (2011) falls into this category. It suffers from gaps in the data and too much noise. Thus, the algorithms often converge showing errors caused by local inconsistencies in the data. The last group contains all approaches based on variants of RANSAC (Fischler and Bolles, 1981), where minimal subsets of points are randomly drawn to generate a primitive. After several iterations, the best model is selected. An efficient approach which successively detects various primitives in large point clouds has been proposed by Schnabel et al. (2007). This approach does not rely on dense data, and, therefore, seems to be adaptable to our problem.

If the vertical direction of a scene is known, the detection of vertical planes can be enforced. The determination of vanishing points from images has been an active topic of research in photogrammetry and computer vision (Rother, 2000; Almans et al., 2003; Schmitt and Priebe, 2009; Förstner, 2010). The main idea is that, on the image plane, line segments, which are projections of parallel lines in object space, intersect in unique points. Yet these above algorithms will probably fail for less regular data, such as old timber-framed buildings (Fig. 1). Additionally, the accuracy of the vertical direction derived in image space is limited when projected in 3D by the accuracy of the camera calibration. Opposed to the work above, Hansen (2007) directly detects the vertical direction on 3D point clouds yet the approach only works on Legoland scenes without any less regular objects such as, e.g., trees, cars, and people. Furthermore, the approach by Hansen (2007) assumes a ground plane, which is often not valid. For our application, the detection of the vertical direction should be invariant to architectural imperfections, such as when doors, or the timber-frame and windows are not perfectly aligned with the vertical direction.

Several approaches make use of LiDAR point clouds to derive detailed facade models for downtown areas (Becker and Haala, 2008; Hernández and Marcotegui, 2009; Hohmann et al., 2009). The LiDAR points are used to detect the principal plane which is interpreted as the facade and facade components such as windows, stairs, or oriels are segmented. All these approaches are not directly applicable to the detection of detached buildings, but will be considered, when we will refine the building models.

3 OVERVIEW OF THE ALGORITHM

The input to our system is an unstructured 3D point cloud \mathcal{P} with points \mathbf{p}_i from image matching (or LiDAR). Its output is a set



Figure 1: Image of a timber-framed building, where the detection of the vertical direction in image space will probably fail.

of connected rectangular surfaces s_k , i.e., the plane parameters, the description of the rectangular outline of each surface, and the adjacency graph between the surfaces. The surfaces are supposed to represent building walls and dominant building parts, such as balconies and oriels.

In our workflow (Fig. 2) we first estimate the surface normal \mathbf{n}_i and the surface curvature value τ_i for every point. For the reliable computation of the vertical direction $\mathbf{v} = (v_x, v_y, v_z)^T$ in \mathbb{R}^3 we suppose a prevalence of vertical walls and orthogonal intersections in the architectural scenes from which \mathcal{P} originates. Thus, \mathbf{v} is derived from the biggest cluster of locally estimated vertical directions, improved by an analysis of points on straight 3D edges in the point cloud, and refined using least-squares adjustment.

The vertical direction is used as a constraint for a multi-step RANSAC-based detection of vertical planes in the point cloud, assuming a known metric scale over \mathcal{P} . In each step, parallel planes are detected by plane sweeping and the surface outlines of these planes are estimated by line sweeping.

Finally, the surfaces are connected to each other, constructing a surface adjacency graph. This is necessary for constructing a consistent geometric polyhedron within the point cloud.

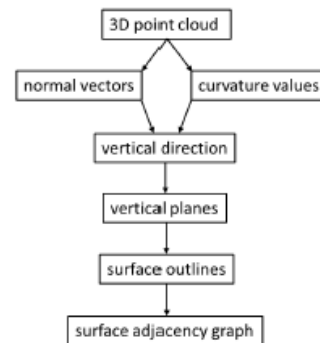


Figure 2: Workflow of proposed algorithm.

4 DETAILS OF THE ALGORITHM

Vertical Direction Computation: For the computation of the vertical direction, we first determine the normal vectors \mathbf{n}_i for each point \mathbf{p}_i by fitting the best plane in least-squares sense to a local neighborhood $\mathcal{P}_{i,r}$ with radius r centered around and including \mathbf{p}_i . The value of r is derived from the input data assuming a known metric scale of the point cloud \mathcal{P} . For points with less

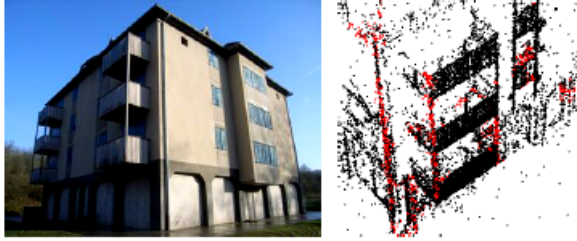


Figure 3: Left: view of a detached building. Right: Part of the derived point cloud with points of high curvature values marked in red

than five neighbors within τ , alternatively a five-nearest neighbor search is performed. Following (Berkmann and Caelli, 1994), an approximation for \mathbf{n}_i is obtained from the eigenvalue analysis of the covariance matrix C_i of $\mathcal{P}_{i\tau}$ defined by

$$C_i = \frac{1}{|\mathcal{P}_{i\tau}|} \sum_{j=1}^{|\mathcal{P}_{i\tau}|} (\mathbf{p}_j - \bar{\mathbf{p}})(\mathbf{p}_j - \bar{\mathbf{p}})^T, \quad (1)$$

with $\mathbf{p}_j \in \mathcal{P}_{i\tau}$ and $\bar{\mathbf{p}}$ its centroid. The eigenvector corresponding to the smallest eigenvalue λ_0 of C_i is a reasonable estimate of \mathbf{n}_i . According to (Pauly et al., 2002), an approximation of the curvature of $\mathcal{P}_{i\tau}$ is given by

$$\tau_i = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \quad (2)$$

where λ_0 , λ_1 , and λ_2 are the non-negative eigenvalues of C_i (Fig. 3).

For the determination of the vertical direction \mathbf{v} , we first perform a RANSAC-based search. Secondly, we evaluate points near straight 3D edges to obtain additional evidence for the vertical direction.

To compute a hypothesis of the vertical direction, we iteratively select randomly two distinct points \mathbf{p}_i and \mathbf{p}_j and compute the cross product of their normals $\mathbf{v}_{ij} = \mathbf{n}_i \times \mathbf{n}_j$. This random point pair selection is repeated many times. Since most buildings have roof areas smaller than the sum of the areas of all walls, we assume to find more point pairs on orthogonal walls than in any other combination. Thus, the biggest cluster of hypotheses corresponds to the vertical direction \mathbf{v}_0 , which is computed by a least-square fit.

We found that for all major surface variations in architectural scenes, i.e. edges in the point clouds, the upper 30%-quantile of points with respect to the curvature value τ_i is usually a good estimate, i.e. we consider all points with $\tau_i > 0.7 * \max(\tau_i)$. In each iteration of our RANSAC-based search for additional evidence of the vertical direction, we randomly select two distinct points of this subset of \mathcal{P} and construct a straight line through both points. If a sufficient number of points lie close to the line and if the line goes towards the vertical direction \mathbf{v}_0 determined above within a tolerance of 20 degrees, we consider the line for an improved determination of the vertical direction \mathbf{v} . The latter is obtained by applying a least-square fit to all accepted lines.

Detection of Vertical Planes: A major goal of our work is to determine planar surfaces s_k with surface normals \mathbf{n}_k approximately perpendicular to \mathbf{v} , because these surfaces can be interpreted as walls of buildings or facade parts. We segment the planes by means of a RANSAC-based search (Schnabel et al., 2007).

1. Randomly select two distinct points \mathbf{p} and \mathbf{q} from \mathcal{P} .

2. Derive plane \mathbf{h}_k from \mathbf{p} and \mathbf{q} taking \mathbf{v} into account by (Polyanin and Manzhurov, 2007)

$$\mathbf{h}_k : \begin{vmatrix} x - p_x & y - p_y & z - p_z \\ q_x - p_x & q_y - p_y & q_z - p_z \\ v_x & v_y & v_z \end{vmatrix} = 0. \quad (3)$$

3. Compute the distance of all points in \mathcal{P} to \mathbf{h}_k .
4. Score \mathbf{h}_k by counting the number of points whose distance to the plane is smaller than a threshold which is derived from the metric scale of \mathcal{P} .

The above steps correspond to a single iteration of the algorithm. They are repeated for a predefined number of times. The best hypothesis is selected and refined by a least-squares fit over its supporting points (Fig. 4). We then look for planes parallel to the fitted plane by sweeping along its normal vector. Afterwards, we repeat the RANSAC-based plane detection, to search for planes with other orientations considering only the remainder of the points of \mathcal{P} . To account for that walls in architectural scenes may not always be perfectly vertically oriented, a final validation step ensures that all the detected planes are perpendicular to \mathbf{v} within a tolerance.

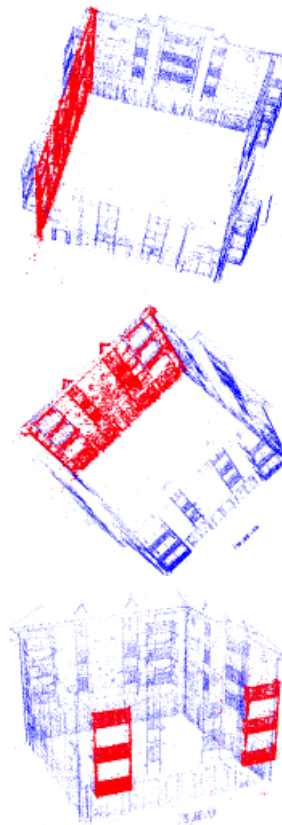


Figure 4: Three views on point cloud (blue). In red, the inlier points of the best plane detected by our RANSAC formulation (top and center), and inlier points of a detected plane with two clusters for balcony fronts (bottom).

Determination of the Surface Outline: While the detected planes are infinite geometric entities, we are looking for surfaces s_k with finite well defined outlines. For this, we analyze the spatial distribution of all points supporting a plane. First, we remove

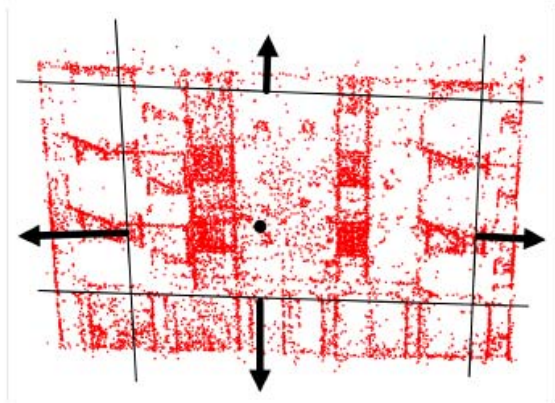


Figure 5: Line sweep for surface outline determination in a perspective view: Points on the surface (red), the center (black dot), the four sweep lines (black lines) and their sweep directions (black arrows).

all isolated points on the plane. Second, we perform a cluster analysis to obtain different surfaces on the same plane. Then for each cluster, we construct the convex hull, as we found that it gives a rough estimation of the surface outline. Finally, to determine a more accurate estimate of the surface outline in the form of a rectangle, we perform four line sweeps in every cluster starting at its center, two along v and the other two orthogonal to v and the surface normal (Fig. 5). Candidates for the borders of the rectangle are taken from the convex hull and are scored by counting the number of inliers supporting these lines. The best line model per sweeping direction is chosen and a least-squares fit is performed for its inliers. In our experiments, we were more successful with this procedure than by analyzing the point densities of inliers at each sweeping step. This is due to the very high point density variations on doors and windows. Finally, we check, if there are close parallel and overlapping surfaces. In this case, we merge the points from those surfaces and determine the outline of the combined surface.

Construction of Surface Adjacency Graph: In this step, we derive topological information for the surfaces and adjust the corresponding parameters. After checking all pairs of distinct surfaces, we recursively connect adjacent surfaces considering only pairs of non-parallel surfaces. If a vertical surface border ends inside another surface, both are connected to each other and the smaller surface is interpreted as building part, such as a balcony or an oriel. If two surfaces have very close borders, we connect the surfaces, interpreting them as walls. This is repeated until no further surfaces can be connected. After these steps gaps may still remain in the boundary of a building. They must be closed, if a closed polyhedral model for the building should be achieved (Fig. 6).

5 EXPERIMENTS

We implemented our algorithms using the point cloud library by Rusu and Cousins (2011) and we tested our approach on several point clouds from image matching. For image orientation and derivation of a sparse point clouds, we used software similar to Snavely et al. (2006); Bartelsen and Mayer (2010). The sparse point clouds often consist of approximately 50 to 100 thousand points. These can be highly unequally distributed and often have huge gaps in the data, especially in the shadow areas. Additionally, we used the semi-global matching by Hirschmüller (2008) to derive significantly denser point clouds containing points in the range of a million and above.

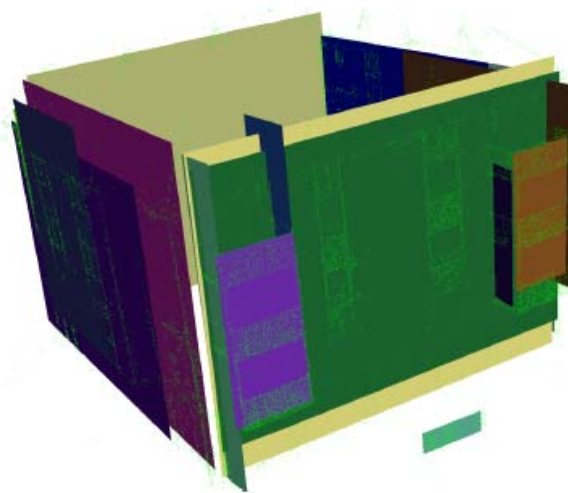


Figure 6: Rectangles derived from the segmented planes to serve as the basis for constructing an adjacency graph.

We have tested our examples on a dozen point clouds derived from images at a small village in Southern Germany, and one additional data set showing the castle of Ettlingen, taken from the benchmark data set of (Strecha et al., 2008). On the one side, we tested our approach on data derived from simply shaped building, and on the other side, we used more difficult objects as L-shaped buildings, or buildings with highly decorated facades or with many other parts, such as balconies, oriels, and stairs. We present some examples on the performance of our algorithm in figs. 7 to 12.

During our experiments, we restricted ourselves during all RANSAC searches by using a unique number of 1000 iterations. However, due to variation in scale from one dataset to another, we adapted the RANSAC threshold according to a known metric scale of each input point cloud. Thus, the absolute value of this threshold is approximately 20 cm.

When inspecting our results, we always find the major four vertical walls of each building. Attached building parts, such as balconies and oriels, are not included in our building models, so far. We are able to detect smaller surfaces corresponding to minor building parts, but we did not consider them when constructing the surface adjacency graph. We will carry on with this refinement step, when modeling building on the next level-of-detail. Then we also consider to close the building models by additional roof and ground surfaces to produce consistent LOD 2 models.

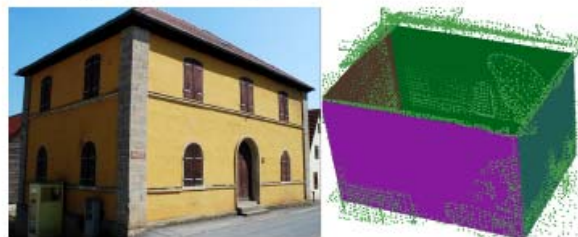


Figure 7: Left: Image of building consisting of four major walls standing on a slope. Right: The derived point cloud consists of 76 000 points, the four major walls were successfully detected, segmented and connected.

6 CONCLUSION

We presented an automatic approach for deriving cuboid-based building models from point clouds reconstructed from multiple



Figure 8: Left: Image of building consisting of four major walls standing on a slope. Right: The derived point cloud consists of 273 000 points, the four major walls were successfully detected, segmented and connected.



Figure 9: Left: Image of building consisting of four major walls having a dominant entrance and stairs at one side. Right: The derived point cloud consists of 271 000 points, the four major walls were successfully detected, segmented and connected.

images. It detects vertical planar surfaces assuming the dominance of orthogonal intersections. The approach was tested with point clouds of buildings from image matching with varying scale, point density and amount of noise and has proven its robustness.

In future, we want to produce a closed building footprint harmonizing the height of each wall and its neighboring walls. Then we are able to estimate the top and the bottom borders of the building, yielding a final closed polyhedral for the building, i.e., its LOD 1 model. Afterwards, we will refine the model to models with more details, LOD 2 and higher. Also, we anticipate as a future work, the integration of many more geometric primitives such as spheres, cones, and cylinders.

References

Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S. M. and Szeliski, R., 2010. Reconstructing Rome. *IEEE Computer* 43(6), pp. 40–47.

Almansi, A., Desolneux, A. and Vamech, S., 2003. Vanishing Point Detection without Any A Priori Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(4), pp. 502–507.

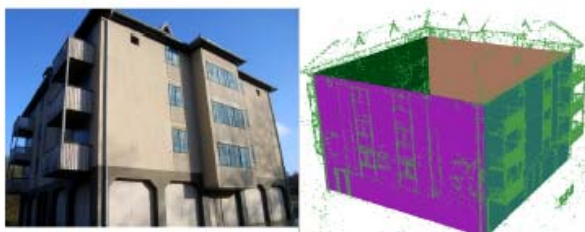


Figure 10: Left: Image of building consisting of four major walls showing balconies or oriels. Right: The derived point cloud consists of 64 000 points, the four major walls were successfully detected, segmented and connected.



Figure 11: Left: Image of building consisting of four major walls showing balconies or oriels. Right: The derived point cloud consists of 1.6 million points, the four major walls were successfully detected, segmented and connected.



Figure 12: Left: Image of the inner yard of a building consisting of four major walls with many ornaments and two cylindrical towers in the yard's corners. Right: The derived point cloud consists of 1 million points, the four major walls were successfully detected, segmented and connected.

Bartelsen, J. and Mayer, H., 2010. Orientation of Image Sequences Acquired from UAVs and with GPS Cameras. *Land and Information Science* 70(3), pp. 151–159.

Becker, S. and Haala, N., 2008. Integrated LIDAR and Image Processing for the Modelling of Building Facades. *Photogrammetrie-Fernerkundung-Geoinformation* 2008(2), pp. 65–81.

Berkmann, J. and Caelli, T., 1994. Computation of Surface Geometry and Segmentation Using Covariance Techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(11), pp. 1114–1116.

Fischler, M. A. and Bolles, R. C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6), pp. 381–395.

Förstner, W., 2010. Optimal Vanishing Point Detection and Rotation Estimation of Single Images from a Legoland Scene. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII (3A)*, pp. 157–162.

Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S. and Pollefeys, M., 2010. Building Rome on a Cloudless Day. In: *11th European Conference on Computer Vision*, pp. 368–381.

Hansen, W., 2007. Automatic Detection of Zenith Direction in 3D Point Clouds of Built-Up Areas. In: *Photogrammetric Image Analysis (PIA07), IAPRS 36 (3/W49A)*, pp. 93–97.

Hart, P. E., 2009. How the Hough Transform was Invented. *IEEE Signal Processing Magazine* 26(6), pp. 18–22.

- Hernández, J. and Marcotegui, B., 2009. Point Cloud Segmentation towards Urban Ground Modeling. In: 2009 Joint Urban Remote Sensing Event, p. on CD.
- Hiep, V. H., Keriven, R., Labatut, P. and Pons, J.-P., 2009. Towards High-Resolution Large-Scale Multi-View Stereo. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1430–1437.
- Hirschmüller, H., 2008. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), pp. 328–341.
- Hohmann, B., Krispel, U., Havemann, S. and Fellner, D., 2009. Cityfit: High-Quality Urban Reconstruction by Fitting Shape Grammars to Images and Derived Textured Point Clouds. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII (5/W1), p. on CD.
- Huang, H., Brenner, C. and 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, pp. 16–24.
- Kazhdan, M., Bolitho, M. and Hoppe, H., 2006. Poisson Surface Reconstruction. In: Eurographics Symposium on Geometry Processing, pp. 61–70.
- Lee, S. C. and Nevatia, R., 2004. Extraction and Integration of Window in a 3D Building Model from Ground View Images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. II: 113–120.
- Meixner, P. and Leberl, F., 2011. 3-Dimensional Building Details from Aerial Photography for Internet Maps. *Remote Sensing* 3, pp. 721–751.
- Pauly, M., Gross, M. and Kobbelt, L. P., 2002. Efficient Simplification of Point-Sampled Surfaces. In: Conference on Visualization, pp. 163–170.
- Polyanin, A. D. and Manzhirov, A. V., 2007. Handbook of Mathematics for Engineers and Scientists. Chapman & Hall, Boca Raton.
- Reznik, S. and Mayer, H., 2008. Implicit Shape Models, Self-Diagnosis, and Model Selection for 3D Facade Interpretation. *Photogrammetrie–Fernerkundung–Geoinformation* 2008(3), pp. 187–196.
- Rother, C., 2000. A New Approach for Vanishing Point Detection in Architectural Environments. In: 11th British Machine Vision Conference, pp. 382–391.
- Rusu, R. B. and Cousins, S., 2011. 3D is Here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation, p. on CD.
- Schindler, F. and Förstner, W., 2011. Fast Marching for Robust Surface Segmentation. In: Photogrammetric Image Analysis, pp. 147–158.
- Schmitt, F. and Priese, L., 2009. Vanishing Point Detection with an Intersection Point Neighborhood. In: 15th IAPR International Conference on Discrete Geometry for Computer Imagery, pp. 132–143.
- Schmittwilken, J., Yang, M. Y., Förstner, W. and Plümer, L., 2009. Integration of Conditional Random Fields and Attribute Grammars for Range Data Interpretation of Man-Made Objects. *Annals of GIS* 15(2), pp. 117–126.
- Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum* 26(2), pp. 214–226.
- Snavely, N., Seitz, S. M. and Szeliski, R., 2006. Photo Tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics* 25(3), pp. 835–846.
- Strecha, C., von Hansen, W., Van Gool, L., Fua, P. and Thoennessen, U., 2008. On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: IEEE Conference on Computer Vision and Pattern Recognition, p. on CD.
- Vieira, M. and Shimada, K., 2005. Surface Mesh Segmentation and Smooth Surface Extraction Through Region Growing. *Computer Aided Geometric Design* 22, pp. 771–792.
- Vosselman, G., 2009. Advanced Point Cloud Processing. In: Photogrammetric Week, Wichmann, pp. 137–146.
- Vosselman, G. and Klein, R., 2010. Visualisation and Structuring of Point Clouds. In: Airborne and Terrestrial Laser Scanning, Whittles, Dunbeath, pp. 45–81.
- Wahl, R., Schnabel, R. and Klein, R., 2008. From Detailed Digital Surface Models to City Models using Constrained Simplification. *Photogrammetrie–Fernerkundung–Geoinformation* 2008(3), pp. 207–215.
- Wenzel, S., Drauschke, M. and Förstner, W., 2007. Detection and Description of Repeated Structures in Rectified Facade Images. *Photogrammetrie–Fernerkundung–Geoinformation* 2007(7), pp. 481–490.