Statistical Reconstruction of Indoor Scenes

Philipp Jenke, Benjamin Huhle, Wolfgang Straßer
University of Tuebingen
Sand 14, 72076 Tuebingen, Germany
{jenke|huhle|strasser}@gris.uni-tuebingen.de

ABSTRACT

In this paper we consider the problem of processing scanned datasets of man-made scenes such as building interiors and office environments. Such datasets are produced in huge quantity and often share a simple structure with sharp crease lines. However, their usual acquisition with mobile devices often leads to poor data quality and established reconstruction methods fail – at least at reconstructing sharp features. We propose to overcome the lack of reliable information by using a strong shape prior in the reconstruction method: we assume that the scene can be represented as a collection of cuboid shapes, each covering a subset of the data. The optimal configuration of cuboids is found by formulating the reconstruction problem as a discrete *maximum a posteriori* (MAP) optimization in a statistical sense. We propose a greedy algorithm which iteratively extracts new shape candidates and optimizes over the shape of the cuboids. A new candidate is selected by scoring its ability to reconstruct previously uncovered data points. The iteration converges at the first significant drop in the score of new candidates. Our method is fast and extremely robust to noisy and incomplete data which we show by applying it to scanned datasets acquired with different devices.

Keywords

Surface Reconstruction, Statistical Methods, Bayesian Methods

1 INTRODUCTION

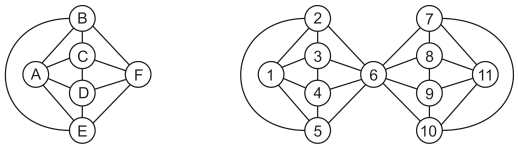
An ever-increasing number of more powerful 3D scanning devices are being constructed (e.g. [BFW+05]), leading to a huge number of datasets of many different environments. One class of scenes, which has gained a specifically great attention, especially in the computer graphics and the robotics communities, is the class of building interiors and office environments. However, since most scanning devices focus on fast and easy acquisition, the resulting data quality is often limited. Many systems have in common, that they capture the general structure of a scene very well, but fail to scan the fine details in a sufficient resolution. On the other hand, many applications, including surveillance systems, cultural heritage projects, path planning for autonomous robots or emergency and evacuation simulations require high-quality models of the general structure of a scene.

Recently, some authors proposed to use statistical surface reconstruction methods building upon Bayes' theorem which allow to include external knowledge about the scene into the reconstruction process (e.g. [DTB06, JWB+06, HAW07]). However, since these methods are usually applied on a local scale, they pose strong requirements on the quality of the data which the previously mentioned datasets rarely meet (non-uniform sampling, low signal-to-noise ratio, holes). In such cases, the methods tend to fall back to a smooth reconstruction such as Moving Least Squares (MLS, [ABCO+03]). For the reconstruction of man-made scenes, however, this is not desirable, since the sharp creases, which are very descriptive for the structure of the scene, are then lost. We preserve this information by putting more knowledge about the scene into the reconstruction. In the Bayesian setting this means that a stronger prior has to be used. Jenke et al. [JWB+06], for instance, only provide priors on the local smoothness and density of their reconstruction.

When selecting a strong prior for a reconstruction system, one therefore has to balance it between two poles. On the one hand, the statistical model has to be general enough to fit to all the scenes of the type one addresses. On the other hand, it has to encode as much external knowledge about the scene structure as possible in order to overcome the limitations of the data. For the scans of building interiors, the general structure of the scenes can often be assembled from a combination of cuboids. We therefore describe our reconstruction model as a set of cuboids of unknown size, orientation

and scale: each cuboid is parameterized via a rotation $R \in \mathbb{R}^{3\times3}$, an anisotropic scaling $S \in \mathbb{R}^{3\times3}$ and a translation $t \in \mathbb{R}^3$, leading to the 9-dimensional parameter vector $(\alpha, \beta, \gamma, s_x, s_y, s_z, t_x, t_y, t_z)$. These parameters in combination with the unknown number of cuboids are the free variables of our reconstruction model.

We formulate the reconstruction problem as an iterative optimization process. At each iteration, we add a new cuboid candidate which covers a previously uncovered subset of the data points to the model and optimize for the parameters of all cuboids. For each candidate, we compute a score, based on its power to explain data points which were previously not covered by other cuboids. We run the iteration until the score drops significantly compared to scores of cuboids which are already in the model. In order to find good initializations for new cuboid candidates, we detect planes in the uncovered data points by using a RANSAC (*random sample consensus*) approach. Then, we establish a topology graph over the planes and connect spatially close planes with perpendicular normals. From all the subgraphs which match the graph generated from a cuboid (Figure 1), we choose the one that best fits into the uncovered data points.



(a) template graph of a cuboid (6 nodes)

(b) primitive graph $\mathscr{P}$ created from adjacent planes of a fictive dataset

Figure 1: Graph matching.

The remainder of the paper is structured as follows: Section 2 describes work related to our reconstruction method, Section 3 presents the reconstruction pipeline in detail and Section 4 discusses results we were able to produce. Finally, Section 5 concludes the paper.

2 RELATED WORK

Due to its comparatively long history, the literature in the field of surface reconstruction is very rich. It includes implicit methods [HDD+92], Moving Least Squares [ABCO+03] and Multilevel Partition of Unity [OBA+03] approaches, Poisson Surface Reconstruction [KBH06] methods as well as statistical techniques [DTB06]. However, most of these traditional approaches fail to handle datasets of poor quality such as the ones addressed here. They make implicit assumptions about the sampling quality which datasets of real environments rarely meet. Especially feature-preserving methods (e.g. [FCOS05, JWB+06, DHOS07]) are only applicable to datasets which meet stringent sampling spacing requirements and have a high signal-to-noise ratio.

Debevec and colleagues addressed the problem of reconstructing building exteriors in [DTM96]. They merge geometric components computed from photogrammetric modeling from images with an image-based analysis-by-synthesis approach. The main reason for the astonishingly good reconstructions is that the general geometry is created in a manual modeling process. Overcoming the limitation of poor depth data has been addressed by some authors by focusing on the type and properties of specific datasets – especially for the reconstruction of buildings. Most of them approach the problem from an outside perspective since aerial images are broadly available. The group of Vosselman (see e.g. [SV02]), use a parametric model to describe buildings as a set of basic building blocks in a *Constructive Solid Geometry* (CSG) representation, which they try to fit into the aerial images. Similarly, Hu et al. [HYNP04] represent building facades as cuboids, which they fit into edges detected in aerial images combined with depth values from registered airborne $2\frac{1}{2}$D LiDAR height field data. Another approach for building reconstruction has been proposed by Lafarge et al. [LDZD06]: from stereo satellite images of cities they extract candidate positions for buildings. Then, they fit a simple model of buildings consisting of rectangular ground shapes with a roof to the height field data in a Bayesian sense. Compared to our method, these systems operate on 2D input (height field data or images), while we use 3D point clouds from scanners within the scene as input.

Some authors tried to infer semantic knowledge about a scene from connected components in a graph structure over extracted primitives. Nüchter et al. [NSH03] suggest a model for building interiors consisting of *floors*, *ceilings*, *walls* and *doors*. In order to classify the different entities, they encode the relationship between the classes (e.g. 'a ceiling is always above a wall'). The additional information gained from this process is used to improve the quality of the input data. Schnabel and colleagues extend their previous system [SWK07] to detect primitive shapes in point clouds in [SWWK08]. They enrich the primitives with topological connectivity. Upon the resulting topology graph they build a query-interface to allow for the recognition of user-specified patterns (e.g. of windows or roofs) in the data. We use a similar query method for our template matching (see Section 3.3).

An alternative is to approach building reconstruction from a 2D perspective. Schindler and Bauer [SB03] detect planes and principle directions in scans of building facades and afterwards employ the specific characteristics of facades such as 2D features and 2D primitives to reconstruct the fine details. Jenke and colleagues [JKS08] assemble a reconstruction directly from primitive structures which they detect using the RANSAC approach. For datasets of low quality, our method is

more stable, since the lack of information in poorly sampled facets can be compensated for by the information in adjacent facets in a cuboid shape. The approach of Chen and Chen [CC08] works rather similar: they extract planar regions in range images by normal clustering and afterwards extract simple polyhedrons by computing the intersections between adjacent planes. In cases, where this approach fails, they propose a user-guided process.

Bahmutov et al. presented a combined acquisition and reconstruction system in [BPM06] in an operator-guided system. Similar to our system, they employ external knowledge about the scene: they assume that each room can be represented as a box and additionally provide a set of construction blocks for all other parts of the building (e.g. hallways). However, this limits the shape space of their system, e.g. they would not be able to reconstruct the cuboids of different height as in Figures 2 and 3. The impressive results they are able to get come at the cost that their method is limited to their self-made acquisition device and requires for some user-intervention.

3 ALGORITHM

We formulate our surface reconstruction method as an optimization problem on the free parameters of the cuboid shapes. However, due to the discrete nature of the problem (unknown number of cuboids) we run the optimization in an iterative manner. At each iteration, we add a new cuboid to the model. A new candidate is selected by scoring its ability to reconstruct previously uncovered data points. The iteration converges at the first significant drop in the score of new candidates. The pipeline of our method is shown in Figure 2.

Following the ideas of Jenke et al. [JKS08], we compute a sampling estimate ε_j and a noise standard deviation $\sigma_{noise,j}$ for each data point $d_j \in D$ (set of all input data points) in a preprocessing phase: we employ an extended definition of *sampling* which goes beyond the average spacing of a point towards the nearest neighbors. Instead, with the term *sampling* we denote the radius of the minimal influence sphere required for a stable normal estimation. This also takes the local signal-to-noise ratio and sampling anisotropy into consideration. The size of this *sampling* is determined by iteratively growing the *sampling* radius and comparing the eigenvalues and eigenvectors of the covariance matrices of the data points in the environments. The process stops when the eigenvalues are sufficiently anisotropic and the direction of the eigenvector corresponding to the smallest eigenvalue (normal direction) does not change any more. Then, we fit a second-order polynomial to the points within *sampling*-radius and infer the local noise level from the offsets of the data points to the polynomial surface. We use this estimate later to determine if a data point d_j fits into a primitive (distance to the primitive is smaller than $3\sigma_{noise,j}$, corresponding to a 99% probability in an assumed Gaussian noise distribution).

3.1 Model Description

Our scene model assumes that the general structure of all (indoor building) scenes can be assembled from a combination of a finite and small number of cuboid shapes. The number of parameters in the reconstruction is $9|M|$, where M is the set of cuboids. For each cuboid i we maintain a list of assigned data points $D_i \subset D$. A point x in each cuboid's local coordinate system can be transformed into world coordinates via

$$x \rightarrow R_i S_i x + t_i,$$

with rotation R_i, scaling S_i and translation t_i. For the optimization we use its inverse to transform a point x into the corresponding cuboid coordinate system:

$$x \rightarrow S_i^{-1} R_i^{-1} (x - t_i).$$

Cuboids with parallel facets, which are close to each other, most likely result from a surface structure which consists of a combination of several cuboids – e.g. an L-shaped object. In order to enforce consistency constraints in such an arrangement, we need to track the connectivity between the cuboids. We organize this connectivity in a *shape graph* structure $\mathscr{S}$. Parallel facets of cuboids which are close to each other (distance $< 3\sigma_i$; σ_i is the average noise standard deviation of the data points in cuboid i) are connected in the graph. Two facets are considered parallel, if the absolute dot product of the normals is larger than 0.9 (≈ 25 degrees). For such connected facets, we check if either corner points of the connected cuboids are close or if a corner point of one cuboid is close to a facet of the other cuboid. We denote the shape graph storing the corner to corner connections by $\mathscr{S}_{CC}$ and the graph storing the connections between corner points and facets $\mathscr{S}_{CF}$. Figure 3 shows this connectivity between facets and corner points (corner to corner connections are visualized as red cubes, corner to facet connections as blue spheres).

3.2 Bayesian Problem Formulation

We use a statistically motivated formulation similar to [JWB+06, DTB06] based on Bayes rule:

$$p(M|D) = \frac{p(D|M)p(M)}{p(D)}, \tag{1}$$

where our model M is the set of cuboids represented via their free parameters (rotation, translation and scaling). The term $p(M|D)$ is the posterior, $p(D|M)$ the likelihood and $p(M)$ the prior. In the optimization process, we maximize the right-hand side of Equation 1, leading

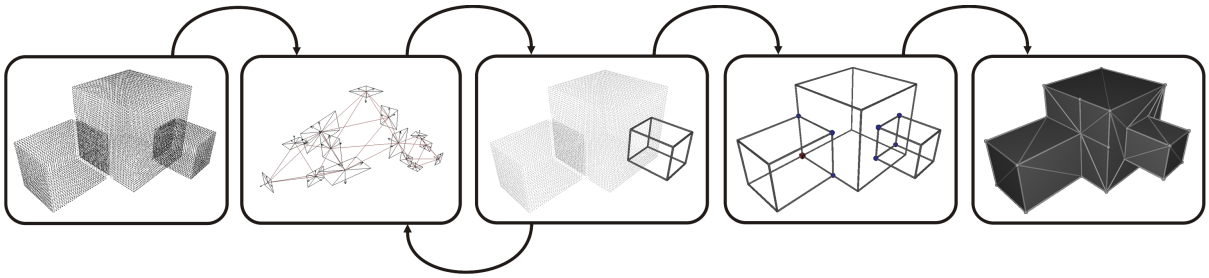


Figure 2: Reconstruction pipeline: data preprocessing, plane detection (extent of planes is downscaled), cuboid candidate extraction, optimization, meshing.

to the maximum a posteriori (MAP) optimization problem. For the sake of stability and simplicity, we discard the constant evidence term $p(D)$ and transform the formulation into negative log-space. Therefore, from now on, we will denote the resulting potentials with $\phi = -\log p$:

$$\phi_{map}(M|D) = \arg\max_M(\phi(D|M) + \phi(M)). \quad (2)$$

The likelihood potential $\phi(D|M)$ models the attraction between each cuboid i and its associated data points D_i:

$$\phi(D|M) = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{1}{|D_i|}\sum_{j\in D_i}\frac{1}{\sigma^2_{noise,j}}\tau_i^2(d_j),$$

where $\tau_i(x)$ is a function which computes the distance to the closest projection of the point x onto the facets of cuboid i.

The prior potential $\phi(M)$ ensures consistency between connected cuboids. It attracts corners connected in the shape graph $\mathscr{S}_{CC}$ and corners connected to facets in $\mathscr{S}_{CF}$ towards each other:

$$\begin{aligned}\phi(M) \quad = \quad & \frac{1}{|\mathscr{E}(\mathscr{S}_{CC})|}\sum_{(k,l)\in\mathscr{E}(\mathscr{S}_{CC})}\frac{1}{\sigma^2_{noise,k,l}}||c_k - c_l||_2^2 \\ & + \frac{1}{|\mathscr{E}(\mathscr{S}_{CF})|}\sum_{(k,l)\in\mathscr{E}(\mathscr{S}_{CF})}\frac{1}{\sigma^2_{noise,k}}d_l^2(c_k),\end{aligned}$$

where $\sigma_{noise,k,l} = \frac{\sigma_{noise,k}+\sigma_{noise,l}}{2}$ is the average noise of the cuboids and $d_l(x)$ is the distance of point x from facet l. The set of edges in $\mathscr{S}_{CC|CF}$ is denoted by $\mathscr{E}(\mathscr{S}_{CC|CF})$.

3.3 Detection of Cuboids

Equation 2 can not directly be solved, because it is neither known in advance, how many cuboids are optimal to cover the input data points, nor which data point is represented by which cuboid. We propose an iterative greedy algorithm to determine this information. In each iteration, we use the RANSAC principle as presented in [SWK07] to automatically detect plane primitives in the data (Figure 2, second step). From this forwards, by the term *plane* we will mean a bounded planar surface. The extent of a plane is given by the lengths of the tangents t_u and t_v:

$$x = p_{plane} + \lambda_u t_{u,plane} + \lambda_v t_{v,plane},$$

with $\lambda_u, \lambda_v \in [-1,1]$. The scaling of the planes is required to determine distances between planes. We align the tangent vectors of each plane to the principle axes of the corresponding data points and scale them such that the projections of the associated data points fit into the span of the tangent vectors. The planes are then organized in a *primitive graph* structure $\mathscr{P}$: we insert an edge for each pair of perpendicular planes (absolute value of the dot product is smaller than 0.1, which corresponds to approx. 85 degrees). In order to avoid connections between nodes which are not reasonable, we prune connections between planes k and l if their shortest distance d is larger than the smaller diagonal of the area spanned by the tangent vectors of the planes:

$$d > \min(\sqrt{||t_{u,k}||_2^2 + ||t_{v,k}||_2^2}, \sqrt{||t_{u,l}||_2^2 + ||t_{v,l}||_2^2}).$$

In the assembled graph structure, we search for the graph pattern of a cube (Figure 1a). For most datasets, this pattern can be found several times in $\mathscr{P}$. However, not all of these combinations lead to cuboid shapes existing in the data. In order to only extract the candidate with the highest probability, we rerun the pattern extraction process several times (50 in all our examples) and compute a score δ for each candidate:

$$\delta = \frac{\sum_{j\in D_{cand}} exp(-\frac{\tau^2_{cand}(d_j)}{\sigma^2_{noise,j}})}{A_{cand}},$$

where $\tau_{cand}(x)$ is the distance function of the candidate shape, D_{cand} denotes the set of data points represented by the planes in the current pattern. A_{cand} is the surface area of the candidate cuboid which can easily be computed from its scaling parameters. Deprecated candidate shapes with small volume due to close parallel planes (distance between opposite planes smaller than noise level) are automatically pruned. In order to correctly evaluate the score for a candidate, it has to be (at least roughly) fitted to its final shape. This is done by

initializing the rotation parameters from the perpendicular normals of the planes in the pattern and adjusting the scaling and translation to best fit the cuboid's facets to the planes. For the subgraph-matching we use a simple randomized approach which randomly chooses a seed node and then grows along the primitive graph $\mathscr{P}$ – if possible – until all nodes and edges of the template are matched.

The cuboid with the highest score is then added to the set of cuboids M. After each iteration, we rerun the optimization of Equation 2 and update the data point assignment. Therefore, we compute the distance of each data point d_j to all cuboids in M and assign it to the cuboid with the smallest distance, if this distance is smaller than $2\sigma_{noise,j}$. The RANSAC plane detection is only applied to data points which have not previously been assigned to a cuboid. We also insert a plane node for each facet of already detected cuboids into the primitive graph $\mathscr{P}$. This is especially required for structures where cuboids share a facet. In order to be able to also handle surfaces with open ends, we additionally use a template of a cube with a missing facet which consists of 5 nodes only.

Finding an automatic iteration stopping criterion is generally hard and very much depends on the expectations one has of the reconstruction. Some might only be interested in the general interior structure of a building, while others might want to extract each cuboid-shaped small box in a room. A good compromise in our experience is to stop the process when a significant drop of the score of a newly inserted cuboid appears (we set this threshold to 0.25 times the average score of the shapes which are already in the model M). In all our tests, this criterion found all the cuboids expected in a dataset.

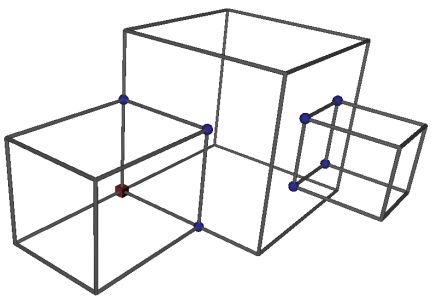


Figure 3: Connectivity between cuboids: corner-corner (red cubes), corner-facet (blue spheres).

3.4 Meshing

The final step in the reconstruction process is the extraction of a triangle mesh (Figure 2, right). For a single cuboid, this is rather simple: two triangles are created for each facet. However, for cuboids, which are connected in the shape graph $\mathscr{S}$, parts of some facet area need to be left open. In such cases, we use the *Geometric Tools*[1] library, to intersect the polygons of adjacent facets and triangulate the resulting polygons with holes. However, the intersection routine runs into stability problems for polygons with parallel edges. We solve this problem by slightly extruding the subtrahend polygon along such edges. Cuboid facets which account for less than 1% of the data points assigned to a cuboid are completely left open, since they most likely result from open ends in the data.

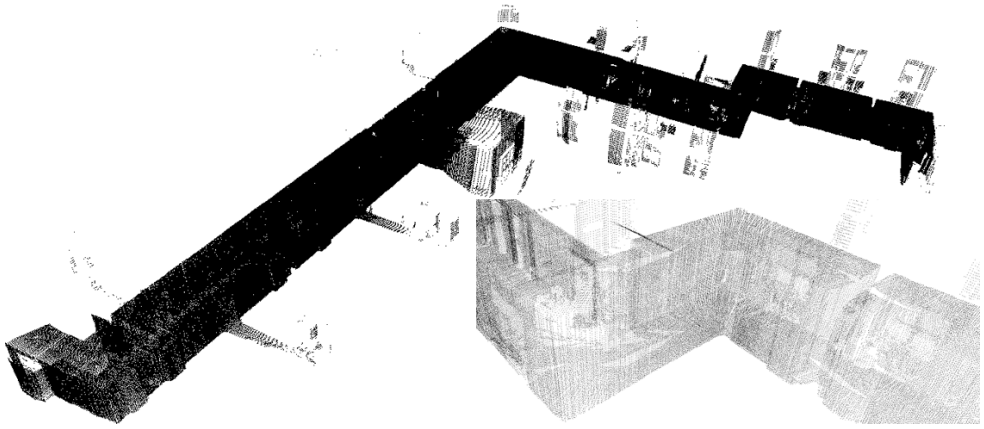
For the textured meshes in Section 4, we created a texture image for each facet from the color information given at the data points. Please note, that incorrect colors (Figure 4 d) result from calibration errors in the data and cannot be corrected by our reconstruction method.

4 RESULTS

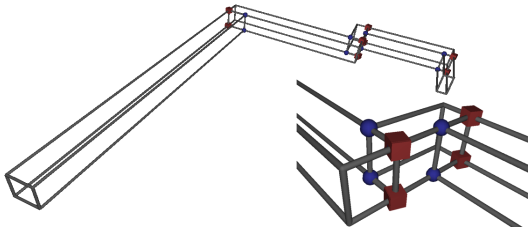
In this section, we present reconstruction results produced with the described method. Our prototype implementation was written in C++ using Visual Studio 2008 on the Windows XP platform. The timings in Table 1 were performed on an Intel Core 2, 2.4 GHz system with 4 GBs of RAM. We optimize the energy function with the Polak-Ribiere conjugate gradient optimization routine described in the Numerical Recipes [PTVF07]. The threshold on the minimal number of points required to accept a plane candidate in the RANSAC detection phase n_{RANSAC} is a user parameter which we set to 3000 in the examples in the paper. A data point j is considered to fit into a plane, if its distance is smaller than $3\sigma_{noise,j}$.

The dataset in Figure 4 was acquired with a mobile device based on a laser-scanner setup mounted on a cart which is dragged through the scene. 2D scans are continuously acquired and registered into a global coordinate system solving the *self-localization and mapping* (SLAM) problem with the data from a second, horizontally oriented 2D laser scanner [BFW+05]. This setup on the one hand allows for the reconstruction of large scenes, but on the other hand produces data of poor quality. Especially, frequently occurring 2D scans for which the registration into the world coordinate system was incorrect, pose a great challenge to any reconstruction system (Figure 4a). Figure 4b shows the reconstructed cuboid shapes and their connectivity between adjacent facets (red and blue spheres). Especially in the detail view one can see, that the consistency constraints between adjacent facets from different cuboids cannot perfectly be met. This results from the slightly incorrect global registration during the data acquisition. However, our reconstruction method is still able to find an optimal alignment of the cuboid shapes. Figures 4c and d show renderings of the extracted meshes. Please note, that some facets seem to be tessellated with too

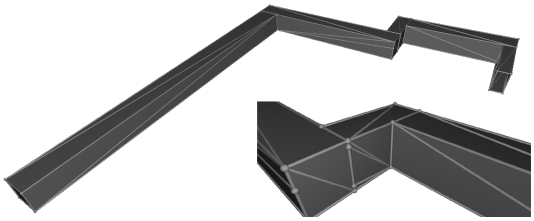
[1] http://www.geometrictools.com



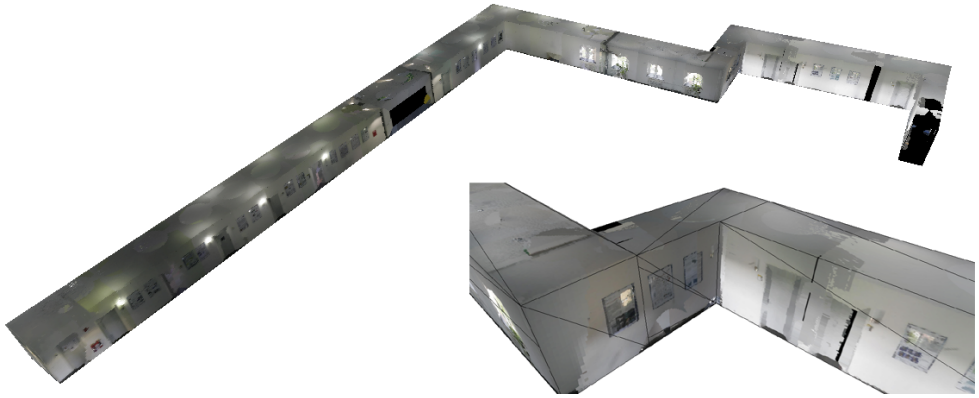
(a) input data



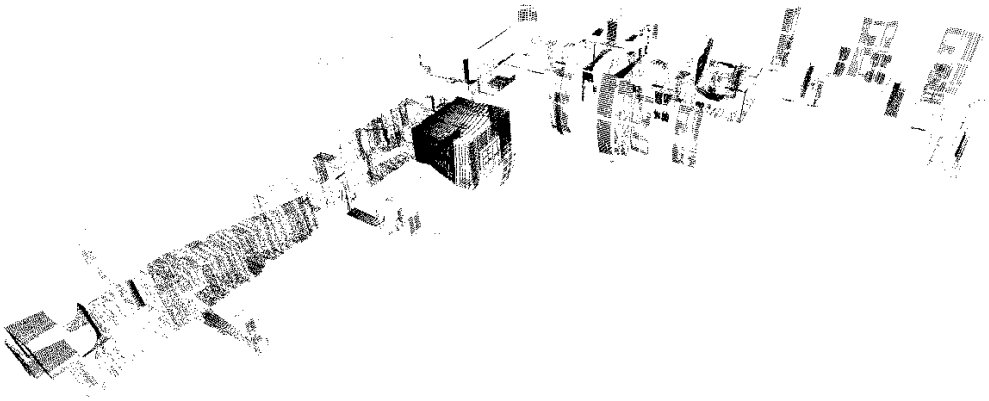
(b) extracted shapes with shape graph $\mathscr{S}$



(c) created mesh



(d) textured mesh



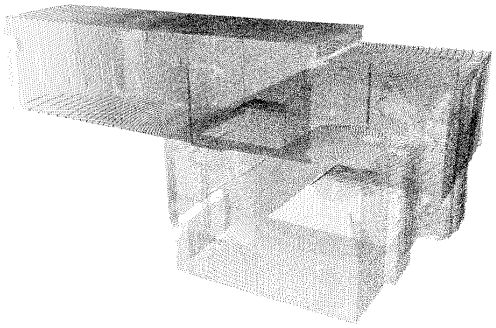
(e) remaining uncovered data points

Figure 4: Dataset *floor* (detail views as sub-images).

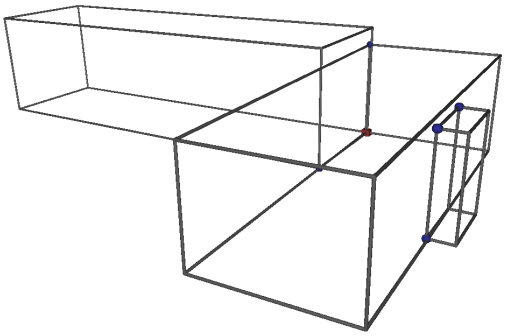
many triangles which is due to the polygon intersection routine, which sometimes produces unnecessary additional polygon points. Finally, Figure 4e shows the points remaining unassigned during the reconstruction process. Most of them belong to small objects in the scene which are not scanned with a sufficient resolution or are outlier points created outside of windows. The remaining points either result from incorrectly registered frames during the acquisition or from parts of the building interior which were only partly acquired (center part of the left wing). There, no primitive planes could be detected by the RANSAC method which would allow for the extraction of cuboid shape candidates.

The scanning system used to assemble the *corner* dataset in Figure 5 consists of a 2D laser scanner and a color camera mounted on a pan-tilt unit. Compared to the *floor* dataset, its data quality is significantly better allowing the system to correctly reconstruct the room and a doorway. In order to improve the performance of the method, we down-sampled the original datasets to the sizes provided in this section and used the original point cloud for the texture-creation only.

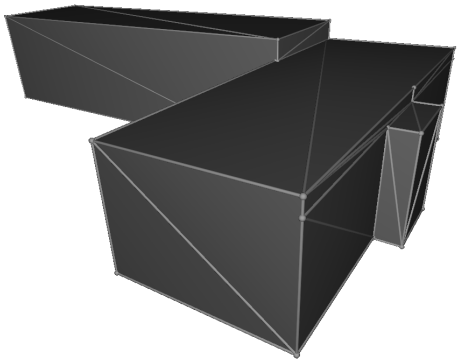
Considering the huge number of data points, our reconstruction method is comparatively fast. However, it is very hard to compare either the timings or the reconstruction quality with previous work. To our knowledge, no previous method is able to automatically reconstruct sharp crease lines in datasets of the quality presented here. The reconstruction times for the preprocessing phase are roughly linear in the number of data points, however, the absolute timings strongly depend on the local characteristics of the data (stronger noise requires for larger influence radii which makes the point neighborhood queries more expensive). The overall time consumptions are quite acceptable since we use the efficient method of [SWK07] to improve a naive RANSAC plane detection. The randomized approach to detect the template cuboid shapes in the plane primitive graph $\mathscr{P}$ is rather fast, however, the iteratively computed score for new cuboid shape candidates makes the candidate selection process the most time-consuming part. The time required to extract the mesh for the final reconstruction is negligible and the time required to create the textures is linear in the resolution of the textures created. For all the renderings in the paper, we extracted textures of the size 512×512 pixels. Table 1 lists the timings required for the datasets in this section.



(a) input data



(b) extracted shapes with shape graph $\mathscr{S}$



(c) extracted mesh

Figure 5: Dataset *corner*.

4.1 Limitations

Our method is not capable of analyzing holes in the data: while the intersection of cuboids is correctly handled, missing data resulting from scanning errors (e.g. due to occlusions) as well as holes in the scene (e.g. windows or doors in a room) will completely be filled by the algorithm. Also, large rectangular furniture pieces will be reconstructed as cuboids, which can be considered incorrect if one only looks for building geometry. We believe that distinguishing between such cases can be done within the same model by more carefully analyzing the scene semantics. Another limitation results from the fact that we only reconstruct cuboids if at least 5 facets have been detected as planes. Consequently, some doors in Figures 4 and 5, where the scanner acquired too few points at the door frames for the detection of a planar structure, cannot be reconstructed.

	#data points	preprocessing	plane/cand. extraction	optimization
floor (5)	509k	461.8s	96.7s/108.3s	9.4s
corner (3)	106k	28.9s	11.8s/8.6s	1.2s

Table 1: Timing results in seconds: name (number of extracted shapes), input data points, preprocessing time, plane detection (RANSAC) and the candidate extraction time, parameter optimization time.

5 CONCLUSIONS

We have presented a robust and fast method to extract the general structure of indoor-scans of many man-made scene environments such as office or department buildings. We cast the surface reconstruction problem to a discrete optimization problem which we solve in an iterative manner. The underlying energy function is motivated in a statistical Bayesian sense and consists of a data fitting and a consistency potential. At each iteration, we add one more cuboid shape until the reconstruction does not improve any more. From the reconstructions, we extract a textured triangle mesh which can be used for many further processing applications including efficient rendering of the scene and compression of the point-based datasets to a representation using only few triangles.

6 REFERENCES

[ABCO+03] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9:3–15, 2003.

[BFW+05] P. Biber, S. Fleck, M. Wand, D. Staneker, and W. Straßer. First experiences with a mobile platform for flexible 3d model acquisition in indoor and outdoor environments – the wägele. In *3D-ARCH '05*, 2005.

[BPM06] G. Bahmutov, V. Popescu, and M. Mudure. Efficient large scale acquisition of building interiors. *Computer Graphics Forum*, 25/3, 2006.

[CC08] Jie Chen and Baoquan Chen. Architectural modeling from sparsely scanned range data. *Int. J. Comput. Vision*, 78(2-3):223–236, 2008.

[DHOS07] J. Daniels, L. K. Ha, T. Ochotta, and C. T. Silva. Robust smooth feature extraction from point clouds. In *Proceedings Shape Modelling International (SMI '07)*, 2007.

[DTB06] J. R. Diebel, S. Thrun, and M. Bruenig. A bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics*, 25:39–59, 2006.

[DTM96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings ACM Siggraph '96*, 1996.

[FCOS05] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. In *Proceedings SIGGRAPH '05*, 2005.

[HAW07] Q. Huang, B. Adams, and M. Wand. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Proceedings Symposium on Geometry Processing (SGP '07)*, 2007.

[HDD+92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings SIGGRAPH '92*, 1992.

[HYNP04] J. Hu, S. You, U. Neumann, and K. K. Park. Building modeling from lidar and aerial imagery. In *Proceedings ASPRS '04*, 2004.

[JKS08] P. Jenke, B. Krückeberg, and W. Straßer. Surface reconstruction from fitted shape primitives. In *Proceedings Vision, Modeling and Visualization (VMV '08)*, 2008.

[JWB+06] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer. Bayesian point cloud reconstruction. *Computer Graphics Forum (Proceedings EG '06)*, 25(3):379–388, 2006.

[KBH06] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings Symposium on Geometry Processing (SGP '06)*, 2006.

[LDZD06] F. Lafarge, X. Descombes, J. Zerubia, and M.-P. Deseilligny. An automatic 3d city model : A bayesian approach using satellite images. In *Proceedings IEEE Acoustics, Speech and Signal Processing (ICASSP '06)*, 2006.

[NSH03] A. Nuechter, H. Surmann, and J. Hertzberg. Automatic model refinement for 3d reconstruction with mobile robots. In *Proceedings 3DIM '03*, 2003.

[OBA+03] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *Proceedings SIGGRAPH '03*, 2003.

[PTVF07] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, August 2007.

[SB03] K. Schindler and J. Bauer. A model-based method for building reconstruction. In *Proceedings HLK '03*, 2003.

[SV02] I. Suveg and G. Vosselman. Automatic 3d building reconstruction. *Photonics West 2002: Electronic Imaging*, 4657 - 4677:59–69, 2002.

[SWK07] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26:214–226, 2007.

[SWWK08] R. Schnabel, R. Wessel, R. Wahl, and R. Klein. Shape recognition in 3d point-clouds. In *Prodeedings WSCG '08*, 2008.