A Model-Based Method For Building Reconstruction

Konrad Schindler
Graz University of Technology
Computer Graphics and Vision
Inffeldgasse 16, 8010 Graz, Austria
schindl@icg.tu-graz.ac.at

Joachim Bauer
VRVis Research Center for
Virtual Reality and Visualization
Inffeldgasse 16, 8010 Graz, Austria
bauer@vrvis.at

Abstract

In this paper model-based reconstruction methods are applied to the detailed reconstruction of buildings from close-range images. The 3D points obtained through image matching are segmented into a coarse polyhedral model with a robust regression algorithm, then the geometry of this model is refined with predefined shape templates in order to automatically recover a CAD-like model of the building surface. Reprojection of the 3D shape templates is used to optimally fit their parameters to the image information. Throughout the paper the investigated methods are demonstrated on real datasets.

1. Introduction

Automatic architectural reconstruction is a continuing goal of photogrammetry and computer vision research. More specifically, the building model delivered by a digital reconstruction system should be a structured surface representation similar to the one a human photogrammetric operator or CAD-designer would produce [19]. The problem thus is different from the reconstruction of terrains or other free-form surfaces in that we expect the recovered geometry to consist of simple parametric surfaces (or a CSG-representation deduced from the surfaces). Therefore we do *not* want to optimally fit an 'all-purpose' surface representation such as a a triangle mesh or a NURBS or Bezier surface to the data, but rather model a building as a collection of planes and polygonal indentations and protrusions on these planes. The aim of the paper is to cover the complete process of model refinement, including the *detection* of facade details, the *selection* of the correct templates and their *fitting* to the image data. We will not present major improvements to the theory of model-based object reconstruction, but rather a collection of straight-forward algorithms tuned towards architectural modeling and capable of solving all subtasks of model-based refinement.

Model-based reconstruction is a generic method to introduce prior knowledge, such as the aforementioned expectations about the structure of a building, into the shape reconstruction process [17]. It has been studied extensively in digital photogrammetry for automatic building detection and reconstruction in aerial images, while in computer vision model-based methods are popular for industrial vision and robotics applications.

For model-based reconstruction, generic templates ('models') of the geometric features we expect to find in the data are used as primitives and their parameters are computed, so that they optimally fit the data. This involves a recognition step and an optimization step. First, we have to determine where in the scene to use a template for modeling and which template to use at each such location. The recognition step yields a number of *approximate* locations for each template in the global coordinate system. In the optimization step the parameters of the templates are adjusted to the image data. The overall number of parameters to be estimated depends on the complexity of the used primitive and on the transformations we allow in the fitting process. In our case we use axis-aligned templates with fixed angles (in order to define not only the topology, but also the angular relations). We thus allow for 8 degrees of freedom: 3 for the translation of the template, 2 for its orientation (because of the template being axis-aligned) and 3 scales to allow adjusting width, height and depth to the data.

Our approach can be outlined as follows: First, detect the dominant planes and principal directions of the scene (3 DOF per plane). Then detect axis-aligned features (indentations or protrusions) on the plane (4 DOF). Finally, compute the depth of each feature (1 DOF). The focus of this paper is on the detection and fitting of smaller features on the main scene-planes.

The paper is organized as follows: In section 2 we give an overview of related work. The input data and preprocess-

ing steps for our algorithms are briefly reviewed in Section 3, and the algorithms used for model-based building reconstruction are described in detail in Section 4. Finally a conclusion and outlook are given.

2. Related Work

Model-based reconstruction techniques were first applied in digital photogrammetry for the (semi-)automatic reconstruction of buildings in aerial images with the help of generic building models. Examples include the work at TU Bonn [4] and TU Vienna [15].

Several authors have tried to fully automate the process by using automatically detected lines: Baillard et al. introduced plane-sweeping around a 3D line for the detection of halfplanes [1], which are then grouped to polyhedral models. Noronha and Nevatia use automatically extracted lines and group them first to rectangles and then to more complex roof polygons in a hierarchical grouping procedure [11]. Bignone et al. support the grouping of lines to coplanar polygons for roof modeling with color information [3]. All these methods rely on the assumption that most of the important lines can be reconstructed. Although automatic line extraction and matching have made great progress [16], sufficiently complete line-sets can still not be guaranteed.

An interactive system for reconstructing buildings from close-range imagery was presented by Debevec et al. The system builds a parametric representation of the scene from lines and symmetry constraints given by an operator. The detection part is thus left to the user [6]. Another interactive approach, which lets the user select points and automatically segments the set of points into planar polygons was recently presented by Bartoli [2].

In the method developed by Dick et al. a sampling scheme is used to detect only vertical planes, which are then upgraded by selecting and fitting template primitives with a Bayesian model selection method [7]. The authors have also extended their Bayesian model-fitting framework to global priors, trading off relations such as regularity and symmetry against the visual information in a Markov random-process [8].

Range segmentation algorithms have mainly been investigated for reverse-engineering tasks. They assume a regular $2\frac{1}{2}$-dimensional grid of points (a range image), and use image-based segmentation techniques to detect parametric surface patches and their boundaries. Most algorithms are either based on region growing or on the detection of discontinuities in the range image. Hoover et al. have given an overview and a comparison of range segmentation techniques [10].

The method closest to the one presented here is the work by Werner and Zisserman. They use a coarse modeling and refinement scheme similar to the one in our system [20]. However they use a purely morphological method for feature detection and fit the models to the image data with a homography-based sweeping technique and area-based correlation.

3. Input Data

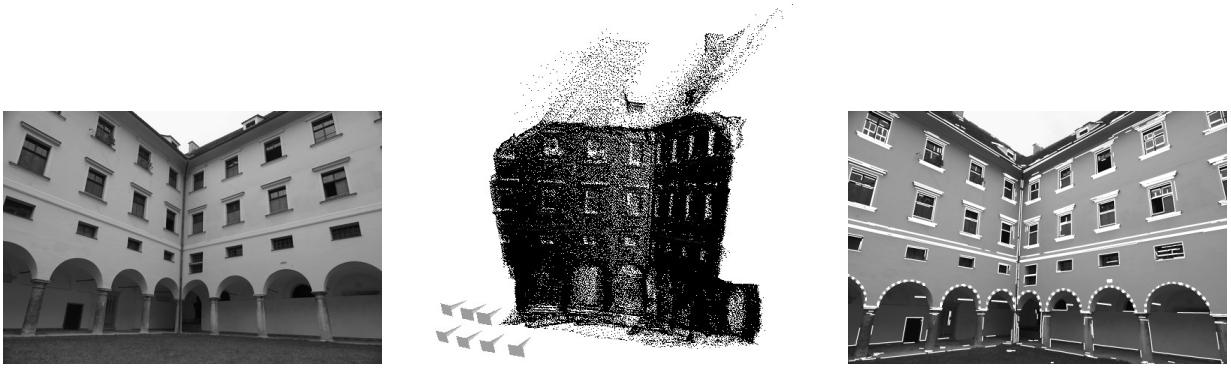
In the following we give a brief overview of the data we use for building reconstruction. We start from a sequence of images taken with a digital consumer camera. In our case the camera is calibrated, however this is only to simplify the preprocessing, which is not the focus of our work. For the methods described in this paper off-line calibration is not essential[1]. The exterior orientations and a dense set of object points are recovered via image matching, orientation, bundle block adjustment and forward intersection of a dense set of automatically matched homologous points. An example of a reconstructed 3D point cloud is given in Figure 1.

Due to effects such as specularities, shadows and occlusions, which impair the matching, the resulting 3D point cloud will not be perfect. For practical purposes this does not seem to be a problem, unless the building has unusual features such as large, curved glass surfaces. In our experiments we have used a standard matching algorithm without any special precautions, still even small features with a depth greater $\sim$10 centimeters are clearly visible and detectable (the smallest window in Figure 1(b) has $\sim$100 points).

NB: Most of the algorithms described in the following are in principle also applicable to 3D point data obtained with a ground-based laser scanner. This would potentially give more accurate detection results because of the higher precision of the 3D data. Note however that the image-based fitting method described in Section 4.3 has to be omitted, unless the laser-scanner is combined with a camera, and the relative position and orientation of the two devices is accurately calibrated. Furthermore the correct treatment of windows is even more difficult when using laser-scans, because the laser beam passes through the glass and recovers the structures in the interior instead of modeling the glass as a surface.

Furthermore line and edge detection is performed on the images. A modified version of the popular Canny edge detector [5] is used to extract oriented edge points (*edgels*) at subpixel accuracy and the edgels are linked to edgel chains. In the set of edgel chains lines and ellipses are detected with a sampling method: in a RANSAC-style procedure straight lines and ellipses are instantiated using minimal in-

1 Note however that in the uncalibrated case relative orientation is underdetermined if all homologous points are incident to a common plane, therefore the reconstruction of facades (such as the 'Steyrergasse' data-set in this paper) from uncalibrated images is not stable.



(a) (b) (c)

Figure 1. Photogrammetric reconstruction and feature extraction. (a) One of seven images of the 'Minorite Monastery' sequence. (b) Cameras and object points after orientation and dense matching. (c) Straight line segments and ellipses (dotted lines) detected in one of the images.

formation, and the rest of the chain is scanned for edgels, which comply with the hypothesis in position and orientation. Hypothesis with high support and a point density above a threshold t are kept and refined with a least-squares fit. For fitting ellipses we use the direct method published by Pilu et al. [12]. An example of detected image lines and ellipses is given in Figure 1.

4. Model-Based Reconstruction

The reconstruction of buildings is split into three steps: in the first step a coarse model is recovered, which consists of the main planes of the building. Then indentations and protrusions on these planes are detected with a sweeping function. In the final step the detected features are refined with predefined shape templates. In the following the three steps are described in detail.

4.1. Coarse Reconstruction

The coarse building model consists of the dominant scene-planes, i.e. the main walls and, if visible, the roof-planes and a ground plane. To find them, the point-cloud is segmented with an orthogonal linear regression algorithm. For the examples in this paper an iterative MAPSAC-scheme [18] has been used, however our experiments show that choosing the optimal algorithm for this task is not critical.

Having found the main planes, we need to determine the principal directions. In the case of rectilinear buildings these are the directions of the intersection lines between the main planes and can be computed directly from the planes' normal vectors, as for example in [21]. A more generic method, which also works if the building is not rectilinear or if only one main wall can be found (as for example on facade images in urban areas) is to compute the principal directions from the extracted image lines: With a random sampling algorithm the dominant vanishing points are detected in the line-set [14]. An example is shown in Figure 2. The vanishing directions are the projections of the principal directions, therefore the projection rays through the vanishing points are the principal directions. The coarse model of the 'Minorite Monastery' building after the initial modeling stage is depicted in Figure 3.

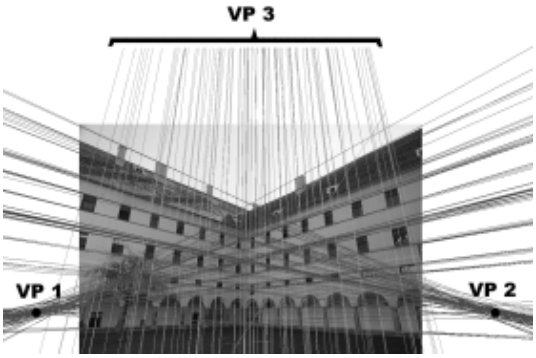


Figure 2. Detection of vanishing points in an architectural scene.

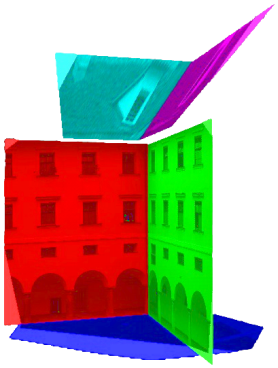


Figure 3. Coarse model of the 'Minorite Monastery' building.

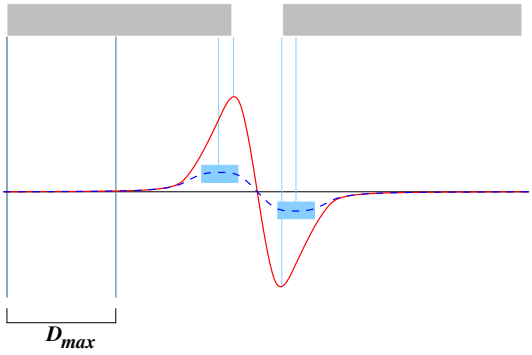


Figure 4. Weighted versus unweighted point count. Two features with regularly distributed points are separated by a small gap (top), the corresponding gradient functions for the weighted (continuous) and unweighted (dashed) case. The weighted version has well-defined extrema close to the correct positions, whereas the unweighted version is severely biased and blurred.

4.2. Feature Detection

The goal of this step is to detect smaller features on the planes of the coarse model, which have to be modeled in more detail. For each recovered plane a fine-grained outlier detection is carried out to find regions with large residuals w.r.t. the plane. The set of outliers is projected to the plane, and a 2D line-sweeping algorithm is applied to find solid regions in the outlier set.

The sweeping algorithm is based on the assumption, that for most buildings all prominent features in the facade plane (e.g. windows, doors etc.) are axis-aligned. We can thus obtain a segmentation of the outliers into rectangular regions by detecting density changes normal to the two axis directions of the wall-plane: inside an indented or protruding facade feature the density of points *not* incident to the plane will be high, whereas the density will be low in regions belonging to the plane (in the absence of noise it would be 0).

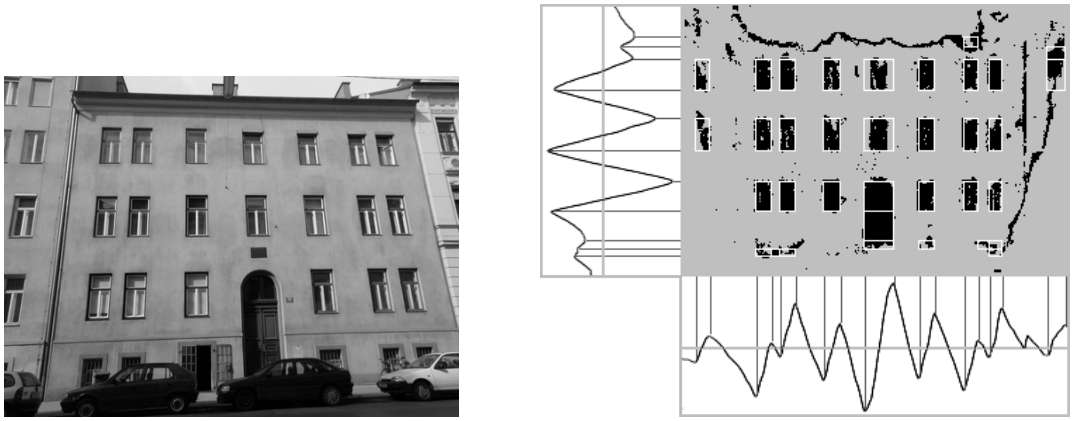
To detect these changes in point density, a parallel sweep line is instantiated for each of the two main axes and swept through the plane containing the projected outliers. Density changes along the sweeping path are recorded by evaluating the gradient of the point count between both sides of the sweep line. The extrema of this function are the positions of the most significant changes in point density (see Figure 5). The finite sampling interval for counting the points inevitably leads to a smoothing, which blurs the position of the extrema (and thus of the detected feature edges). This effect is compensated by using a weighted point count, where a points' contribution is inversely proportional to its distance from the sweep line. The weighted point count greatly reduces the negative effect of sampling, leading to extrema which are steeper and closer to the correct position (see Figure 4). The gradient functions for sweeping a line L along the two principal directions x and y of a plane are

$$
\begin{aligned}
G(x_{\mathrm{L}}) &= \sum_{P_l} \left(1 - \frac{x_{\mathrm{L}} - x_i}{D_{max}}\right) - \sum_{P_r} \left(1 - \frac{x_i - x_{\mathrm{L}}}{D_{max}}\right) \\
G(y_{\mathrm{L}}) &= \sum_{P_b} \left(1 - \frac{y_{\mathrm{L}} - y_i}{D_{max}}\right) - \sum_{P_a} \left(1 - \frac{y_i - y_{\mathrm{L}}}{D_{max}}\right)
\end{aligned}
\tag{1}
$$

where D_{max} is the size of the neighborhood to be considered. The P_l are the points on the left of the line, i.e. all points with an x-coordinate in the interval $[x_{\mathrm{L}} - D_{max}, x_{\mathrm{L}}[$ and the P_r are all points on the right of the line with an x-coordinate in the interval $]x_{\mathrm{L}}, x_{\mathrm{L}} + D_{max}]$. In the same way the P_b and P_a are the points below and above the sweep-line in y-direction (see Figure 5).

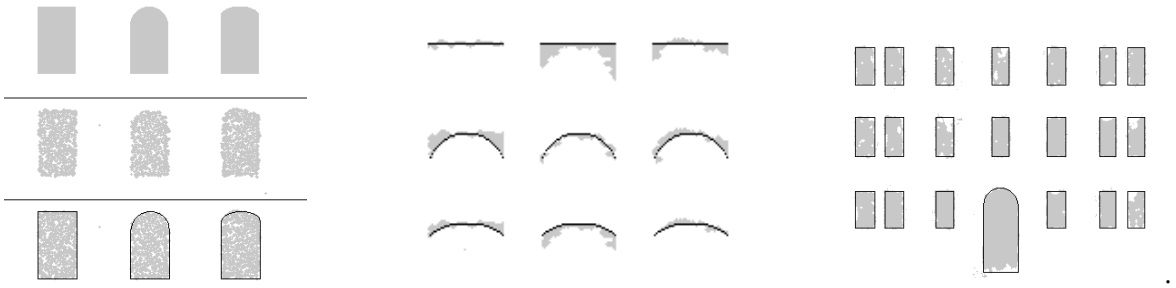
The extrema of the gradient functions (1) correspond to horizontal and vertical lines, which subdivide the wall-plane into a raster of irregular, rectangular tiles. In this tiling the tiles with high point density (found with a simple threshold ρ_{min}) are the regions we are searching for. Neighboring tiles of high point density are merged into one region.

In order to efficiently handle the gradient computation, the projected points are organized in an extended KD-tree data structure optimized for line queries. The standard KD-tree is a hierarchical space-partitioning which allows fast querying of the n nearest neighbors to a given point, and thus also of all points within a given distance [13]. Our variant instead returns all points within a given distance to a line (the sweep line).



(a) (b)

Figure 5. Gradient function for the 'Steyrergasse' dataset. (a) One of four images of the sequence. (b) Rectified wall-plane with projected outlier points (drawn in black) and detected features (white rectangles). Below and left of the wall-plane are the gradient functions in x- and y-direction, respectively. Some false positives occur at the wall borders and on the bottom, where the matching is not correct because of the car windows. However these are discarded later in the fitting process.



(a) (b) (c)

Figure 6. Selection of the correct feature. (a) Top: Synthetic example with three different window features. Center: Same data with noise added to simulate matching errors. Bottom: Outlines selected with the proposed cost function. (b) Unexplained points of the three windows using a rectangular, circular and elliptic template. (c) Outlines selected for the 'Steyrergasse' building.

The parameter which governs the accuracy of the detection result is the step interval s for the sweep. It must be proportional to the average point distance in the point cloud:

$$s = Cd_{mean} \quad , \quad 0.5 < C < 5.0 \tag{2}$$

The average distance, if not known, can be determined statistically by drawing random samples from the point cloud. The factor C is used to tune the algorithm towards different applications: greater C means less steps and thus faster computation at the price of lower accuracy. Useful values are $C \in [0.5, 5]$.

Figure 5 shows the window detection result for the wall of the 'Steyrergasse' building. Note that the density-based sweeping algorithm inherently exploits the raster structure often found among building elements to achieve a better es-

timate: if several features end at the same sweep-step, they jointly contribute to the computation of the gradient and thus give a more accurate and more reliable location of the feature border.

It should be mentioned that the detection method is not limited to axis-aligned features, although it is based on sweeping axis-parallel lines. Also structures, which are not parallel to the sweeping line, such as circular or tapered arches, lead to a maximum of the density gradient, albeit in an approximate position slightly inside the feature. This displacement can be corrected when selecting the correct feature(see next paragraph). The crucial property for feature detection with sweeping is that *features must not overlap* in x- or y-direction.

Once the features have been detected, it has to be decided which type of primitive best describes each feature. A natural decision is to choose the outline which best divides the local neighborhood into a region densely covered by points and a region without points. There are thus two groups of unwanted configurations, namely the M points $\mathbf{p}_i$ outside of the outline and the N holes $\mathbf{h}_j$ inside the outline. In both cases the evidence *against* a potential outline is bigger, if they are far away than if they are close to the outline. Assuming that the relation is linear we can formulate the decision rule as a cost function. The cost for shape S is

$$C_S = \sum_{i=1}^{N} D(\mathbf{h}_i) + W \sum_{i=1}^{M} D(\mathbf{p}_i) \tag{3}$$

where $D(\mathbf{x}_i)$ is the unsigned distance from point $\mathbf{x}_i$ to the outline. The weighting coefficient W can be used to tune the algorithm. Usually minimizing the outliers has a higher precedence than maximizing the coverage, because erroneous 3D points are more likely to occur on complicated structures such as windows with specular reflections, balconies with balustrades etc. than on the near-Lambertian wall plane. Typical values are $W \in [2, 5]$. The cost function (3) is evaluated for every template primitive of the model base and the one with the lowest cost is selected. An example for the selection process is given in Figure 6.

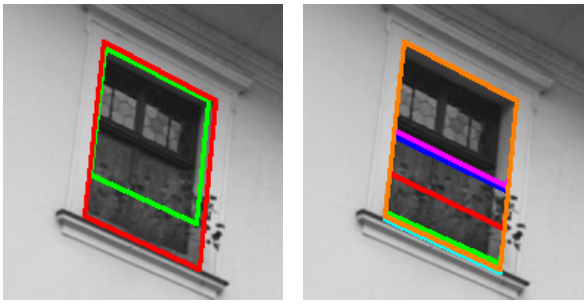
4.3. Fitting

Since the gray-value information has not directly been used until now, we have to refine the parameters of the detected models in order to optimally fit them to the image data. The contours of the detected features are back-projected to the images and a one-parameter search along the normal in the wall-plane (i.e. along the corresponding vanishing line in the image) is performed to snap each segment of the contour to the nearest gray-value edge in the image.

For features, which differ from the facade in gray-value intensity (such as for example windows, which are usually darker), fitting can be improved by adding radiometric properties to the model. Snapping is then done by searching the nearest edge in a 2-dimensional feature space where the discriminating features are the normalized distance to the reprojected edge position and the mean gray-value intensity in the neighborhood of the image edge. For the case of dark windows we found (through statistical analysis of the input images) the following function for the distance w in feature-space:

$$w = \left(\frac{d}{d_{max}}\right)^2 + \left(1 - \frac{\max(g_l, g_r)}{255}\right)^2 \tag{4}$$

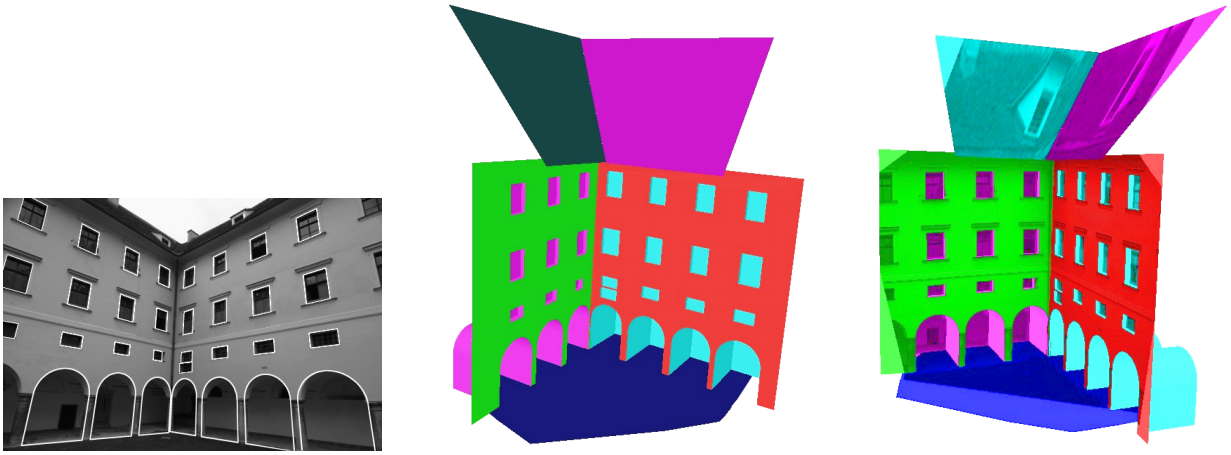
where d is the normal distance between the reprojected edge and an image edge, d_{max} is the maximum allowed normal distance for a candidate and g_l, g_r are the mean gray-values on the left and right side of the image line, respectively. Equation (4) is a promising result, although it was found through explorative statistical analysis and has no theoretical foundation: since the correct match could be found for 94.3% of the edges when using only the geometric distance and one radiometric property (see Table 1), it seems feasible to use a trainable classifier [9] and learn the discrimination function from the data.



(a) (b)

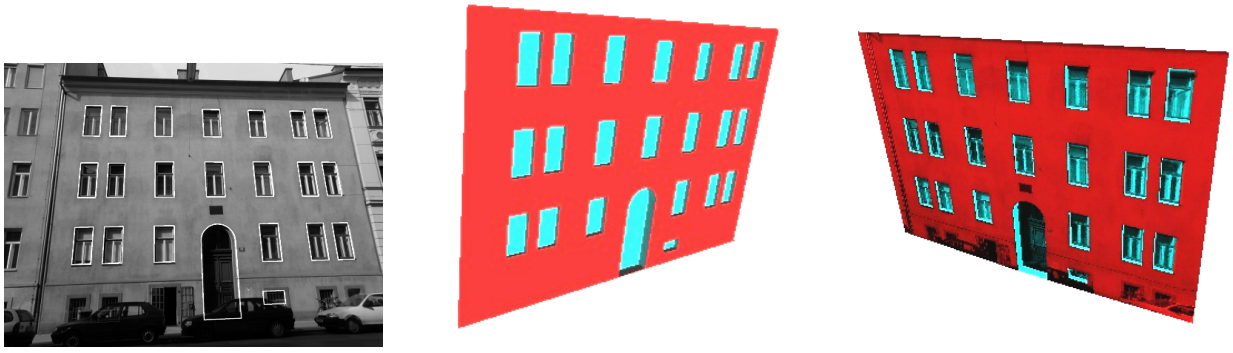
Figure 7. Enhancing fitting results through the use of multiple images. (a) Window positions before refinement (green) and after refinement (red). (b) Snapping results in different images. It can be seen that the consensus over multiple images considerably enhances the result in occluded and cluttered areas.

The robustness of the geometric correction is increased by seeking the consensus over all images of the sequence. The use of multiple images enhances the result especially



(a) (b) (c)

Figure 8. Reconstruction results for the 'Minorite Monastery' building. (a) Reprojected features after fitting. In the center some lines could not be fitted correctly due to weak image contrast. (b) Untextured model to show the recovered geometry. (c) Model with textured walls and windows. Since the columns are not modeled correctly by the used shape primitives the object geometry is not correct behind the columns.



(a) (b) (c)

Figure 9. Reconstruction results for the 'Steyrergasse' building. (a) Reprojected features after fitting. (b) Untextured model to show the recovered geometry. (c) The model with textured wall and windows. Due to the recording geometry (the images were taken from the side-walk in a rather narrow street) the roof and the ground-plane have not been modeled.

in cluttered and occluded parts of the image, as depicted by the example in Figure 7.

After refinement the orthogonal offset from the wall-plane is computed by fitting a parallel plane to the points inside the contour. Two examples for the result of the entire modeling process are shown in Figures 8 and 9.

data set	*edges*	*correct*	*wrong*	*success*[%]
Minorite	92	89	3	97.7
Steyrerg.	82	75	7	91.5
both	174	164	10	94.3

Table 1. Fitting results with 2D feature-space. Window edges, for which no candidate was found, were not considered.

5. Conclusion

We have presented a model-based method for automatically recovering detailed building models from images. The method starts with the reconstruction of a dense point cloud. Then a coarse building model consisting of the main planes is recovered with robust linear regression. Smaller features on these planes, modeled as indentations or protrusions, are detected by examination of the points' residuals and their approximate borders are constructed with a sweeping algorithm which uses the varying density of outliers across the plane. For each detected feature the most suitable template is selected from a model-base and in a final step the borders of the features are refined in the image space with the help of gray-value edges, and their offset is computed from the point cloud.

The method yields geometrically accurate parametric building models. Its main theoretical limitation is that it is not able to reconstruct features which cannot be correctly modeled by one of the predefined shape templates of the model base. Practical problems are the classical difficulties of computer-vision applications such as occlusions and extreme lighting conditions like weak contrast, reflections and shadows. To overcome these problems, it would be desirable to add an image-understanding component to the system in order to allow associative verification and completion of the results, similar to the behavior of a human operator.

Acknowledgments

This work has been supported by the European Commission under contract No. IST-1999-20273. The *VRVis* research center is co-funded by the Austrian government research program *Kplus*. The authors would like to thank Andreas Klaus for providing source code for vanishing-point detection and for orienting and reconstructing the 'Minorite Monastery' sequence.

References

[1] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3d reconstruction of buildings from multiple views. In *Proc. ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, Munich*, pages 69–80, 1999.

[2] A. Bartoli. Piecewise planar segmentation for automatic scene modeling. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii*, 2001.

[3] F. Bignone, P. Henricsson, P. Fua, and M. Strickler. Automatic extraction of generic house roofs from high resolution aerial imagery. In *Proc. 4th European Conference on Computer Vision, Cambridge, UK*, pages 85–96, 1996.

[4] A. Brunn, E. Gülch, F. Lang, and W. Förstner. A multi layer strategy for 3d building acquisition. In *Proc. IAPR TC-7 Workshop: Mapping Buildings, Roads and other Man-Made Structures from Images, Vienna*, pages 11–37. Oldenbourg Verlag, 1996.

[5] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[6] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In H. Rushmeier, editor, *Computer Graphics, Proc. SIGGRAPH 96*, pages 11–20. Addison Wesley, 1996.

[7] A. Dick, P. Torr, and R. Cipolla. Automatic 3d modelling of architecture. In *Proc. 11th British Machine Vision Conference, Bristol, UK*, pages 372–381, 2000.

[8] A. Dick, P. Torr, and R. Cipolla. A bayesian estimation of building shape using MCMC. In *Proc. 7th European Conference on Computer Vision, Copenhagen*, 2002.

[9] R. O. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[10] A. Hoover, J.-B. Gillian, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.

[11] S. Noronha and R. Nevatia. Detection and description of buildings from multiple aerial images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 588–594, 1997.

[12] M. Pilu, A. Fitzgibbon, and R. Fisher. Direct least-squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[13] F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag New York, 1985.

[14] C. Rother. A new approach for vanishing point detection in architectural environments. In *Proc. 11th British Machine Vision Conference, Bristol, UK*, pages 382–391, 2000.

[15] F. Rottensteiner. *Semi-automatic Extraction of Buildings Based on Hybrid Adjustment Using 3D Surface Models and Management of Building Data in a TIS*. PhD thesis, Vienna University of Technology, 2001.

[16] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 666–671, 1997.

[17] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Brooks/Cole Publishing Company, second edition, 1998.

[18] P. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002.

[19] F. van den Heuvel. Trends in CAD-based photogrammetric measurement. *International Archives of Photogrammetry and Remote Sensing*, 33(5/2):852–863, 2000.

[20] T. Werner and A. Zisserman. New techniques for automated architecture reconstruction from photographs. In *Proc. 7th European Conference on Computer Vision, Copenhagen*, 2002.

[21] A. Zisserman, T. Werner, and F. Schaffalitzky. Towards automated reconstruction of architectural scenes from multiple images. In *Proc. 25th Workshop of the Austrian Association for Pattern Recognition, Berchtesgaden, Germany*, pages 9–23, 2001.