Towards Feature–Based Building Reconstruction From Images

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

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

ABSTRACT

This paper presents a feature-based approach to piecewise planar modeling of architectural scenes from an oriented image sequence. An improved line detection algorithm is presented, which uses vanishing points to detect lines in the images and reconstruct them in 3D. Subsequently an extension of the 3D Hough transform is presented which allows efficient and robust detection of planes in a set of 3D lines. The detected planes and a dense point-set are used to reconstruct a piecewise planar building model. The presented techniques are demonstrated on a real-world dataset.

Keywords
3D building reconstruction, line reconstruction, 3D Hough Transform

1 INTRODUCTION

Automatic building reconstruction from images is a continuing goal of photogrammetry and computer vision. In this paper we investigate algorithms for the special case of piecewise planar reconstruction, which is an important part of the problem, because most architectural scenes are partly or completely bounded by planar patches.

We assume a recording setup in which each part of the building we want to reconstruct is visible in at least three images. This way of recording is common practice in close-range photogrammetry and multi-image vision in order to provide sufficient redundancy for automatic modeling algorithms. Furthermore we assume that a dense matching of the scene can be created. A wide range of dense stereo-matching algorithms exist, e.g. [Koch96], [Sara01].

WSCG SHORT PAPERS proceedings
WSCG'2003, February 3–7, 2003, Plzen, Czech Republic.

We use the algorithm described in [Zach02].

The reconstruction problem can be tackled with two different strategies. The first one, originating in the computer graphics world, is to triangulate the unstructured point cloud obtained from image-matching and apply a mesh-decimation technique, e.g. [Schro92], [Hoppe93]. The main drawback of mesh-based methods is that the problem of topologically correct triangulation in 3D is not yet solved, especially in the presence of noise. The second strategy is to directly detect planes, using either the cloud of 3D features, e.g. [Vosse99], or the gray-values of the images [Schmi97]. Our approach follows the second strategy. We see triangulation as a subsequent step, which should be carried out using all available information about the geometric structure of the data to ensure correct results.

A problem when using automatically matched point clouds is the high noise-level, especially in geometrically complicated regions (such as for example the center wall and the right wall in Figure 5). In these regions matching algorithms tend to either produce noisy results (due to smoothness constraints) or discard many points, leaving holes whithout data. If a sufficiently dense set of lines covers the modeled object (which is usually the case for buildings) it is therefore advantageous to extract and match lines instead of points and use them as base features for plane detection. The set of lines is naturally much sparser than the point set, but it has a higher accuracy (in our experiments the deviations from the wall plane were about 3 times smaller for lines). This is due to the fact that no smoothing occurs and that several edge points (typically at least 20) contribute to the construction of a line with their positions *and* their gradient directions.

The paper is organized as follows: Sections 2 deals with the detection and reconstruction of object lines through line fitting and matching, which is enhanced by using vanishing points. In section 3 we show how the 3D Hough transform can be extended to robustly detect planes in a cloud of 3D lines. Section 4 shows how these planes are delimited using a dense pointcloud, and in section 5 we present results of the algorithm on a real-world data set. Section 6 summarizes the presented strategy and gives an outlook on future work.

2 LINE RECONSTRUCTION

Line extraction starts with a coarse extraction to obtain an initial line-set. Vanishing points are robustly detected using this line-set, and with the help of the vanishing points a more complete set of lines is recovered. The procedure is independently carried out for each image in the sequence. Finally the 2D lines from all images are matched to form a set of 3D lines.

Coarse Line Extraction

The input data for the primitive extraction are chains of 2D edgels, where an edgel is an image point with a large gradient. Edgel chains are extracted at sub-pixel accuracy with the method described by Rothwell [Rothw95]. For all edgel chains collinear sections are searched with a RANSAC approach [Fisch81]: pairs of points on the chain are picked at random to generate hypothetical line segments, and the number of inliers, which support each segment up to a threshold, is counted. The segment with the highest count is considered the best hypothesis. Its parameters are refined with a least-squares fit to the inliers. The extracted lines are globally merged by linking collinear segments originating from different contour chains.

Vanishing Point Detection

When parallel 3D lines are projected with a perspective camera, their images intersect in a vanishing point (VP). Various methods have been proposed for detecting the principal VPs of an image, e.g. [Tuyte98], [van d98] and [Schaf00].

We use the the method presented by Rother in [Rothe00], which can be described as a variant of the Hough transform directly using the image plane as accumulator space. For details see the original paper. For typical architectural scenes one VP for vertical lines and one or two VPs for horizontal lines are extracted. However, the approach can be extended to a higher number of vanishing points, if necessary.

Advanced Line Extraction

Once the locations of the VPs are known, they can be used to extract more line segments pointing towards each VP. An example is shown in Figure 1.

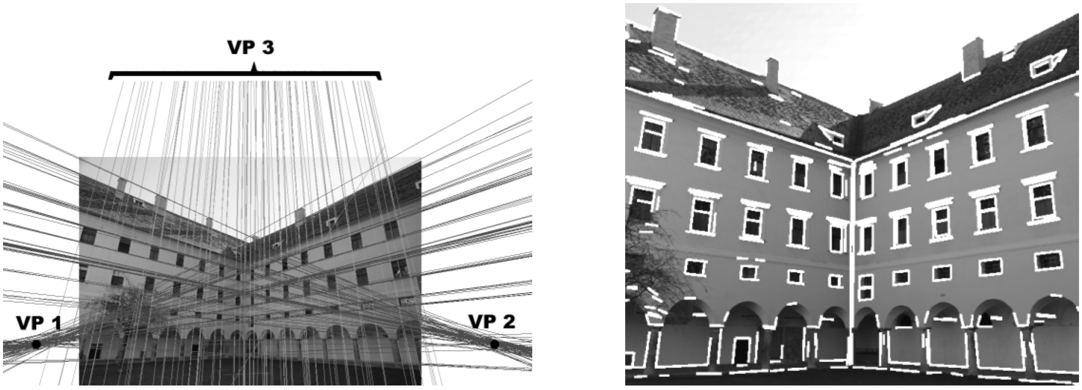
First, another edge extraction is performed using a low gradient threshold in order to detect weaker edgels, too. From the large number of resulting edgels only those are kept, which are oriented towards one of the VPs.

The edgels are then grouped into collinear sets with a line-sweeping algorithm: a line is swept through the image, rotating around the VP, and for each line position the edgels within a perpendicular threshold are selected. The process is illustrated in Figure 2. To efficiently perform the sweep all edgels are sorted according to their polar angle w.r.t. the VP. If in a particular position the line has the polar angle α, the inliers are the edgels with a polar angle of $\alpha \pm \epsilon$. The step angle for sweeping is determined from the distance threshold for inliers.

Sets of collinear edgels without large gaps are connected to line segments, and redundant or overlapping segments are merged, resulting in a new edgel-set with new endpoints for each remaining segment. The final line parameters are found with a least-squares fit to the new edgel-set.

With the known image orientations, the 2D mage line segments can be matched to 3D line segments. For details on this step we refer to the work of Schmid and Zisserman [Schmi97].

The proposed method significantly improves the completeness of the line set: for the example presented in section 5, matching the line segments found with coarse line extraction yielded 134 3D line segments. Matching the results of the refined extraction process yielded 296 line segments, an improvement by 120%.



(a) vanishing points (b) image lines

Figure 1: Line extraction using vanishing points (a) Vanishing points detected with the help of the line segments from coarse line extraction. (b) Line segments found with the sweeping method. The image shows the inner court of the Minorite monastery in the historic center of Graz. The 3D results for the dataset are shown in Figure 4.

3 PLANE DETECTION

In the following we describe in detail how the Hough transform can be used for plane detection in a cloud of lines. The section starts with a brief review of fundamentals of the Hough transform (HT), then introduces the theory of the 3D Hough transform of line segments, and finally gives some remarks about the practical implementation.

Notation in the following sections:

The HT is based on a mapping between object space and a parameter space. Object space is denoted with the letter $\mathcal{R}$ and an index for the number of dimensions, parameter space is denoted by $\mathcal{H}$. The HT thus is a mapping $\mathcal{R}_i \longmapsto \mathcal{H}_j$. Geometric entities in $\mathcal{R}$ are denoted by boldface capital letters, e.g. a point $\mathbf{P}$. The same entities mapped to $\mathcal{H}$ are denoted by the same letters in lowercase, without serifs, e.g. $\mathbf{P} \longmapsto \mathsf{p}$.

Review of the Hough Transform

The 2D Hough transform [Hough62] is a well-known algorithm for robust feature detection in images. The basic idea is to map points, e.g. edgels, to a parameter space. In the simplest case, each point $\mathbf{P}(x_P, y_P)$ is mapped to a curve p in $\mathcal{H}_2(\alpha, r)$ using the line equation

$$\mathsf{p}: \quad r = x\cos\alpha + y\sin\alpha \tag{1}$$

Note that the polar form of the line equation is used, because it avoids the singularity for vertical lines (and the related problem of non-uniform scale), which occurs in the standard line-equation $y = kx + d$.

The parameter pairs (α, r) on p correspond to the lines of the pencil through $\mathbf{P}$. If several points lie on a straight line $\mathbf{S}$, their corresponding curves in $\mathcal{H}_2$ will intersect in a single point which yields the parameters (α_S, r_S) of $\mathbf{S}$.

Each image line gives one such intersection point. The intersection points are found by subdividing $\mathcal{H}_2$ into a regular raster, accumulating the number of curves passing through each raster cell and searching through the raster for local maxima.

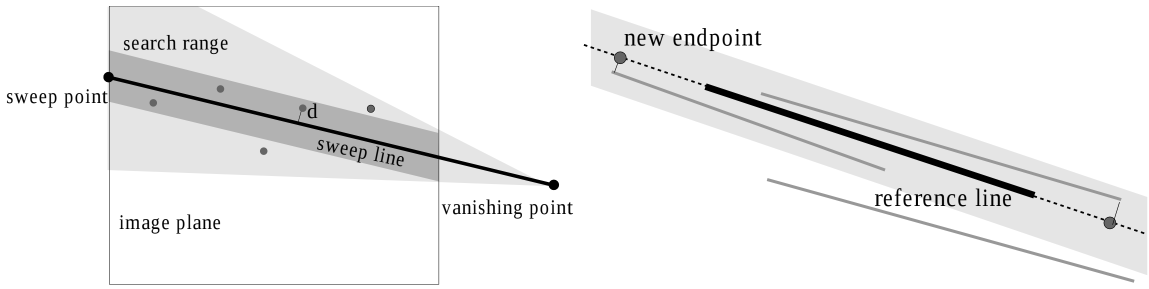
The idea has been extended to 3D-space [Vosse99], [Bian99]. A point $\mathbf{P}(x_P, y_P, z_P)$ then defines a surface p:

$$\mathsf{p}: \quad \begin{array}{l} r = x\cos\alpha + y\cos\beta + z\cos\gamma \quad \text{where} \\ \cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1 \end{array} \tag{2}$$

in $\mathcal{H}_3(\alpha, \beta, r)$, and the voxels with the highest count of passing surfaces correspond to parameter triples of planes through many points.

3D Hough Transform of Lines

Since we want to fit planes to the set of reconstructed 3D lines, the 3D Hough Transform has to be extended to lines. The HT does not seem to have been used with lines as input, perhaps because only recently reliable line-matching strategies have been presented, e.g. [Schmi97], [Heuel01].



(a) line sweeping

(b) grouping of line segments

Figure 2: Line extraction using vanishing points. (a) Line sweeping around the vanishing point. A triangular search space (light gray area) containing the edgels oriented towards the VP is intersected with a rectangular region (dark gray) which satisfies the perpendicular distance criterion. (b) Grouping of detected line segments. All segments inside the valid region (light gray area) are projected onto the reference segment (black) to determine new endpoints for the reference segment.

The idea is straightforward: A line segment $\mathbf{S}$ in $\mathcal{R}_3$ is defined by its two endpoints $\mathbf{P}_i$ and $\mathbf{P}_j$. These endpoints map to surfaces p_i and p_j in $\mathcal{H}_3$ as given by equation 2. Any plane which contains $\mathbf{S}$ must contain both points, thus the valid triples of plane parameters lie on the intersection curve $\mathsf{s} = \mathsf{p}_i \cap \mathsf{p}_j$. The parameter triples (α, β, r) on s correspond to the planes of the pencil through $\mathbf{S}$. The 3D Hough transform of a line segment is illustrated in Figure 3.

If several straight line segments $\mathbf{S}_1 \ldots \mathbf{S}_n$ lie on a common plane $\mathbf{E}$, the corresponding curves $\mathsf{s}_1 \ldots \mathsf{s}_n$ intersect in one point (α_E, β_E, r_E), the parameter triple for $\mathbf{E}$. For each $\mathbf{S}_i$ we thus only have to increment the voxels which lie on s_i.

Implementation Issues

Parameter r is the signed normal distance of the plane from the origin, as can be easily verified using Equation (2). The complete range of possible parameters in $\mathcal{H}_3$ is therefore given by $\alpha, \beta \in [-90°, +90°]$ and $r \in [-d_{max}, d_{max}]$, where d_i is the unsigned distance of a point $\mathbf{P}_i$ from the coordinate origin. To achieve an unbiased angular distribution in $\mathcal{H}$, the origin of the object coordinate system should be set to the center of gravity of the line cloud.

Different strategies have been investigated to find local maxima in $\mathcal{H}$. An overview can be found in [Illin88]. In our implementation three criteria are used. A point is accepted as local maximum if the following conditions are true:

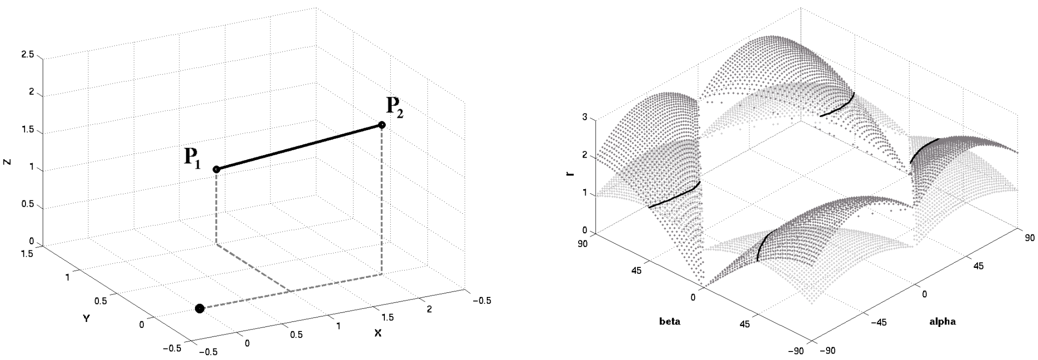
- the accumulator value $N(\alpha, \beta, r)$ is greater than a threshold r_{min} (to filter out peaks due to noise)
- $N(\alpha, \beta, r)$ is a local maximum in a $(u \times u \times v)$ neighborhood, i.e. a peak, and
- the estimated local curvature $N(\alpha, \beta, r)''$ is greater than a threshold k_{min}, i.e. the peak is well-defined.

The HT is computationally efficient. Computation time grows with order $\mathcal{O}(m^2 n)$, where m is the number of steps in α and β (the angular resolution), and n is the number of lines. Time consumption thus depends linearly on the number of features. On the other hand not only computation time, but also memory consumption grows quadratically with the angular resolution. The usual bottleneck is the memory needed to store the parameter voxel space.

We empirically use a quantization of $\mathcal{H}_3$ of 2 degrees for α and β. The number of steps in r is computed using $N_r = \frac{1}{3}\frac{d_{max}}{s_P}$, where s_P is the estimated standard deviation of the lines' endpoints. This reflects that a line supports a plane, if its endpoints lie on the plane within $\pm 3 s_P$. The parameter space thus has $(90 \times 90 \times N_r)$ voxels.

4 PLANE RECONSTRUCTION

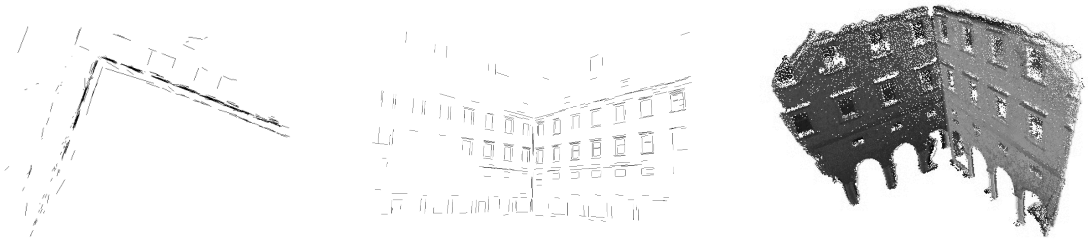
To detect the boundaries of the planar patches we need dense coverage of the building, whereas accuracy is not that important. We therefore go back to the matched point cloud. The inliers to each detected plane are determined, then the plane is subdivided into a regular grid and transformed to a binary image, where cells containing points are set



(a) 3D line segment

(b) Hough-transformed segment

Figure 3: Hough transform of a line segment. (a) line segment in $\mathcal{R}_3$ with the endpoints $\mathbf{P}_1(1,1,1)$ and $\mathbf{P}_2(2,0,2)$. (b) the corresponding surfaces p_1 and p_2 in $\mathcal{H}_3$. The black intersection line is the Hough-transformed segment s. Note that p_1, p_2 and s are not continuous: α and β are the angles between the point-vector $\mathbf{P}$ and the x- and y-axis, respectively, thus they must fulfill $|\alpha| + |\beta| \geq 90°$.



(a) top view

(b) front view

(c) reconstructed planes

Figure 4: Results for the 'Minorite monastery' dataset. (a), (b) Two views of the reconstructed 3D line set. 362 3D-lines were matched, of which 355 are correct. (c) Reconstructed object planes. See also Figure 1.

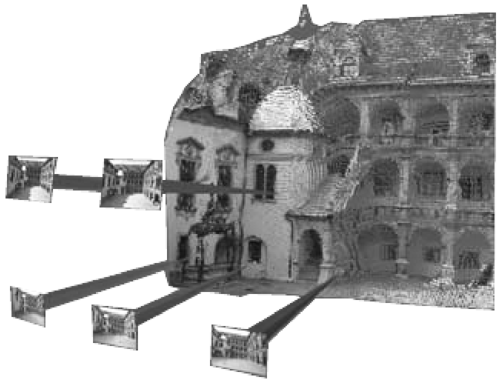
to 1 and empty cells are set to 0. An iterative median filter is applied to determine the solid areas of the image, then an edge-tracing algorithm is used to find the boundaries of planar object patches. An Examples are shown in Figures 4 and 5.

5 EXAMPLES

The described algorithms have been successfully tested with several real-world datasets. Two of these datasets serve as examples to demonstrate the results: the Minorite monastery already shown in Figures 1 and 4, and the inner court of the 'Landhaus' in the historic center of Graz.

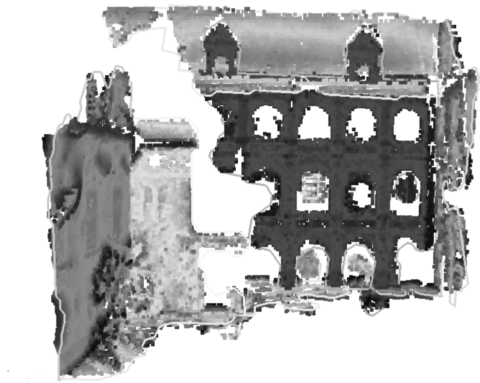
For the 'Landhaus' dataset a sequence of 5 images was used. The images were acquired with a calibrated hand-held camera with a resolution of 2160×1440 Pixels. Lines were extracted from the images and reconstructed as described in section 2. A total of 296 3D-lines were matched, of which 289 are correct. 10 planar patches in 7 planes were recovered with the method described in section 3, all of which are correct. The 2 smallest of the 9 planes visible in the sequence were missed: the heavily foreshortened sidewall of the tower and the roof of the staircase, which is partially occluded by decorations. Several patches of the back wall in the center could not be reconstructed due to the lack of reliable matches. A dense set of 55968 matched points was used for delimiting the patches, of which 39237 are incident to one of the detected planes. The results are summarized in

(a) recorded image with detected lines



(b) dense set of 3D points

(c) original and filtered border of the roof



(d) Reconstructed object planes

Figure 5: Results for example dataset 'Landhaus'. (a) One out of 5 images used for the experiment. (b) The camera positions and a dense set of unstructured object points obtained by image orientation and matching. Points were assigned the gray-value of the central image to give a better visual impression. (c) The binary maps used for delimitation of the roof plane. (d) 3D view of the reconstructed planar patches and the respective inlier points. Gray-value variations *within* one patch encode fitting residuals: darker points have higher residuals.

Figure 5. Note that the wall on the right side could *only* be reconstructed when using lines as base features, because the point cloud is too noisy to give the necessary support for a plane, with the HT as well as with RANSAC fitting.

6 CONCLUSION

Summary

We have described algorithms for automatic feature-based architectural reconstruction from images and demonstrated them on a real-world example. The key ideas of the presented approach are an improved technique for finding line features in images, which is based on line sweeping around automatically detected vanishing points, and the extension of the 3D Hough transform to lines.

Future Work

An important improvement to the algorithm will be to use a weighted combination of lines and points for the plane-fitting process in order to exploit both the precision of lines and the large number of points. Another possible improvement is to use vanishing points not only for line detection in the images, but also to enforce the direction and the parallelity of all lines pointing to a common vanishing point at the stage of line matching. A further idea is to use circular and elliptic arcs in addition to straight lines. A single conic already defines a plane, so that additional support for the detected planes can be expected.

Acknowledgments

We are grateful to Andreas Klaus for providing software for image-matching and orientation. The VRVis research center is co-funded by the Austrian government research program *Kplus*. This work was supported by the European Comission under contract Nr. IST-1999-20273.

References

[Bian99] L. Bian and Z. Yang. Discovering linear and planar patterns from spatial point data using hough transform. Technical report, Geography Department, State University of New York at Buffalo, 1999.

[Fisch81] M.A. Fischler and R.C. Bolles. RANSAC random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 26:381–395, 1981.

[Heuel01] S. Heuel and W. Förstner. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 2, pages 517–524, 2001.

[Hoppe93] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Computer Graphics, Proc. SIGGRAPH '93*, pages 19–26, 1993.

[Hough62] P. Hough. Method and means for recognizing complex patterns. US patent 3,069,654, 1962.

[Illin88] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics and Image Processing*, 44:87–116, 1988.

[Koch96] R. Koch. *Automatische Oberflächenmodellierung starrer, dreidimensionaler Objekte aus stereoskopischen Rundum-Ansichten.* PhD thesis, Universität Hannover, 1996.

[Rothe00] C. Rother. A new approach for vanishing point detection in architectural environments. In *Proc. 11th British Machine Vision Conference*, pages 382–391, 2000.

[Rothw95] C.A. Rothwell, J.L. Mundy, W. Hoffman, and V.D. Nguyen. Driving vision by topology. In *Proc. International Symposium on Computer Vision*, pages 395–400, 1995.

[Sara01] R. Sara. Sigma-delta stable matching for computational stereopsis. Technical report, Center for Machine Perception, Czech Technical University, 2001.

[Schaf00] F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18(9):647–658, June 2000.

[Schmi97] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 666–671, 1997.

[Schro92] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of triangle meshes. In *Computer Graphics. Proc. SIGGRAPH '92*, pages 65–70, 1992.

[Tuyte98] T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons. The cascaded hough transform as an aid in aerial image interpretation. In *Proc. International Conference on Computer Vision (ICCV'98)*, pages 67–72, January 1998.

[van d98] F.A. van den Heuvel. Vanishing point detection for architectural photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, 32(5):652–659, 1998.

[Vosse99] G. Vosselman. Building reconstruction using planar faces in very high density height data. In *Proc. ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, Munich*, pages 87–92, 1999.

[Zach02] C. Zach, A. Klaus, J. Bauer, K. Karner, and M. Grabner. Modeling and visualizing the cultural heritage data set of Graz. In *Proc. VAST2001 Virtual Reality, Archaeology, and Cultural heritage*, 2002.