METROPOGIS: A FEATURE BASED CITY MODELING SYSTEM

Joachim Bauer * **Andreas Klaus*** **Konrad Karner*** **Christopher Zach*** **Konrad Schindler** †

* VRVis Research Center for Virtual Reality and Visualization
email: {bauer,klaus,karner,zach}@vrvis.at

† Institute for Computer Graphics and Vision, Graz University of Technology
email: {schindl}@icg.tu-graz.ac.at

KEY WORDS: feature extraction, vanishing points, line matching, plane detection, 3D reconstruction

ABSTRACT

In this paper we report on a system to extract 3D information from oriented digital facade images using various feature extraction/matching methods. We emphasize on 2D feature extraction from digital images including contour chains, line segments and vanishing points. The extraction of 3D primitives is based on line matching over multiple oriented views and 3D plane detection by a line grouping process. The *MetropoGIS* system is designed for the automatic extraction of 3D data from facade images.

1 INTRODUCTION

This paper describes a work flow for the extraction of 2D features from digital photographic images of building facades and a method for 3D reconstruction using features originating from multiple views.
The process of feature extraction is one of the most important tasks of a modern computer vision system. It reduces the amount of data to be processed, and also increases the robustness and accuracy of measurements in digital images. Most geometric features are detected by extracted edge information, thus they rely heavily on the results of the previous edge extraction methods. In order to correctly group edgels, the images must be free of lens distortion. To achieve this, image distortion is corrected in a preprocessing step by resampling all images with the parameters of a previous offline camera calibration. The paper deals with two types of geometric features: low level image features and derived high level image features and 3D primitives. The low level image features are:

- edgels,
- contour chains, and
- line segments.

High level image features and 3D primitives are:

- vanishing points
- 3D lines, and
- 3D planes.

This ordering also reflects the complexity of the geometric primitives. In our work we concentrate on 2D lines since lines are the most prominent features detected in facade images. The process of line extraction is done in several steps and with several different approaches. Methods used in this context are line fitting to contour chains, advanced line extraction using the information of previously detected vanishing points and merging of the results from different methods. In order to estimate the 3D coordinates of the 2D line sets we apply a method proposed by Schmid and Zisserman [11] which allows a reliable matching of 2D line sets from multiple, oriented views of a 3D scene. The 3D lines together with a dense point set obtained from an image matching algorithm are then used to detect planar object regions.

2 RELATED WORK

Our work can be related to similar approaches that also cover the whole data acquisition process, from the raw digital facade images to a piecewise planar description of the depicted objects. The feature based methods are comparable to the ones described by Zisserman et al. [17], Schmid and Zisserman [12] and Schaffalitzky and Zisserman [10] where an emphasis is put on the automatic extraction of planes from architectural images. The detection of vanishing points using previously extracted lines is similar to the approach of van den Heuvel [14] where the location of the detected vanishing point is also used to adjust the orientation of the line segments.

3 OVERVIEW OF OUR WORK FLOW

The work flow of our proposed system can be split into three sections:

1. **Image Preprocessing:** In this step a resampling is done to compensate for any lens distortion. The lens distortion and intrinsic parameters are determined offline with an automatic calibration process described by Heikkil"a [3].
2. **2D Feature Extraction:** This step includes edge and contour extraction, 2D line fitting on the extracted contours, detection of vanishing points using the previously extracted 2D lines, advanced line extraction

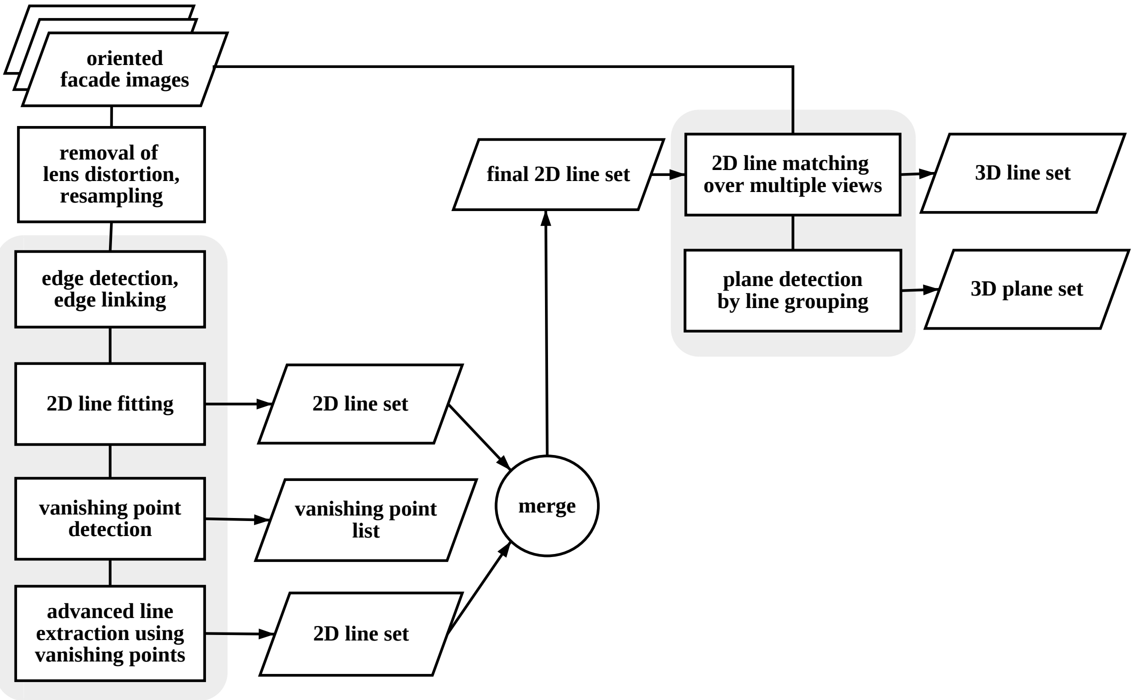

oriented facade images
removal of lens distortion, resampling
edge detection, edge linking
2D line fitting
vanishing point detection
advanced line extraction using vanishing points
2D line set
vanishing point list
2D line set
merge
final 2D line set
2D line matching over multiple views
plane detection by line grouping
3D line set
3D plane set

Figure 1: Sketch of the work flow: On the left side the feature extraction work flow is visualized, on the right side the 3D modeling pipeline is depicted. The grey shaded areas are discussed in this paper.

using the vanishing points, and a final merging of all extracted line segments.

3. **3D Reconstruction:** In this process we match the 2D line segments from multiple views to 3D line segments, apply a dense matching algorithm to reconstruct an unstructured 3D point set and fit planes to the points and lines.

Figure 1 shows the proposed work flow: the feature extraction process is depicted on the left side, the 3D modeling work flow on the right side. The input data are digital facade images, which have been corrected for lens distortion, and the orientations of all images.

The following section gives a detailed description of the different steps of the work flow.

3.1 Line Extraction

The extraction of 2D line segments is based on a RANSAC approach followed by a least-squares fitting. The input data for the primitive extraction are contour chains extracted with the method described by Rothwell [8]. This algorithm yields contour chains at sub-pixel accuracy.

For all contour chains collinear sections are searched with a RANSAC approach by Fischler and Bolles [2]: Pairs of points on the contour are picked at random and each distinctive pair forms a hypothetical 2D line. The number of inliers, i.e. points that support the hypothetical line up to a small perpendicular threshold, is counted for all point pairs. The point pair with the highest number of inliers is considered the best hypothesis for a line segment and the inliers are used for a least-squares fitting in order to optimize the line parameters.

The lines extracted by this approach are then fed into a global merge process in order to link collinear lines originating from different contour chains.

3.2 Vanishing Point Detection

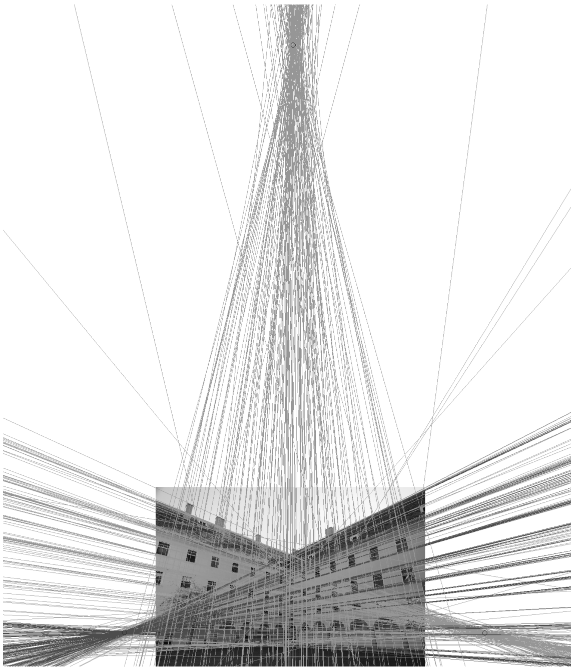


Figure 2: Lines converging to three vanishing points for a typical facade image.

If lines which are parallel in 3D object space are projected with a perspective camera, they form a vanishing point (VP). The VPs are points in the (infinite) image plane where the projected lines intersect, for an illustration see Figure 2.

The location of the VPs depends only on the relative rotation between the camera and the direction of the 3D lines. The location of vanishing points in image sequences can be used to compute the relative rotation between image pairs adapting a photogrammetric standard method described by Kraus [6].

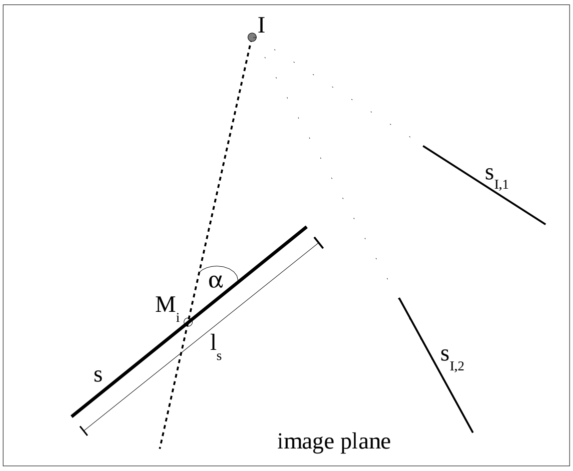


Figure 3: Orientation of a line segment $\mathbf{s}$ vs. the direction towards the potential vanishing point $\mathbf{I}$ (the intersection point of the segments $\mathbf{s}_{I,1}$ and $\mathbf{s}_{I,2}$). The angular difference α and the length l determine the contribution of the line segment $\mathbf{s}$ to the weight of the potential vanishing point $\mathbf{I}$.

VPs are detected using either edgels or previously extracted 2D lines. Among the various methods proposed for detecting VPs the Hough transform approaches by Tuytelaars et al. [13] and Rother [7], a statistical approach described by van den Heuvel [14] and a RANSAC approach described by Schaffalitzky and Zissermann [10] have been investigated.

In our system we use Rothers method because of its simplicity and robustness. The VPs are detected by applying an algorithm similar to the Hough transform [4], where the image plane is directly used as accumulator space. The input data for the algorithm are line segments that are extracted in a preprocessing step. For all extracted line segments in the image, the mutual intersection points are computed. These points are then used as accumulator cells. Each intersection is treated as a *potential vanishing point* and the votes for each cell are determined by testing against all other line segments. Figure 3 shows a line segment $\mathbf{s}$ of length l_s with the midpoint $\mathbf{M}_i$. The support $w(s)$ of $\mathbf{s}$ for the intersection point $\mathbf{I}$ of the line segments $\mathbf{s}_{I,1}$ and $\mathbf{s}_{I,2}$ is calculated as follows:

$$w(s) = k_1 \left(1 - \frac{\alpha_s}{\alpha_{max}}\right) + k_2 \frac{l_s}{l_{max}} \tag{1}$$

where α_{max} is the threshold for the enclosed angle between $\mathbf{s}$ and the direction $\overline{\mathbf{M}_i\mathbf{I}}$. If a line segment exceeds this maximal angle it does not vote for the respective intersection. l_{max} is the length of the longest line segment and k_1 and k_2 are set to 0.3 and 0.7, respectively (following Rothers suggestion). The total vote for intersection point $\mathbf{I}$ in a set $\mathcal{S}$ of line segments is given by:

$$W(i) = \sum_{\forall \mathbf{s} \in \mathcal{S}:\ \alpha_s \leq \alpha_{max}} w(s) \tag{2}$$

This voting process is applied to all intersections, the intersection point with the highest weight is selected and the inliers w.r.t. this point are determined. In practice the extracted lines are split up into an approximately vertical and an approximately horizontal line-set in order to speed up the computation. We assume that in typical architectural scenes one VP for vertical segments and one or two VPs for horizontal lines exist. Figure 2 shows a facade image where the extracted lines are elongated to illustrate the clustering of intersections in the vicinity of vanishing points.

3.3 Advanced Line Extraction

The location of the VPs can be used to extract more line segments pointing to them.

The approach for our advanced line extraction is based on four steps:

1. edge extraction with coarse edge selection
2. line segment detection by sweeping over the edgels (using the known VPs)
3. elimination of redundant line segments by grouping
4. least-squares estimation of line parameters

In the first step an edge extraction is performed using a low gradient threshold in order to detect weaker edgels, too. The usually huge amount of extracted edgels is reduced by taking advantage of the restriction of the search space: only the subset oriented towards one of the VPs is kept for the advanced line extraction approach.

In the following grouping step sets of collinear edgels are detected by a sweeping process. A sweep line is formed by the VP and a sweep point. The sweep point is moving along the image borders in order to cover the whole edgel set. The sweep-range is therefore determined by the outermost edgels in the image plane, as seen from the VP. All edgels are sorted according to their polar angle w.r.t. the VP. This allows to easily determine the a set of candidate edgels for a discrete sweep line: if the sweep line has the polar angle α_l, only the edgels with a polar angle of $\alpha_l \pm \epsilon$ have to be tested.

All edgels within a perpendicular threshold from the sweep line form the set of inliers and dense subsets are connected to chains and stored as potential line segments.

Figure 4 shows the scenario: all edgels within the search range (light gray area) are tested for their perpendicular distance d to the sweep line. Edgels within the distance

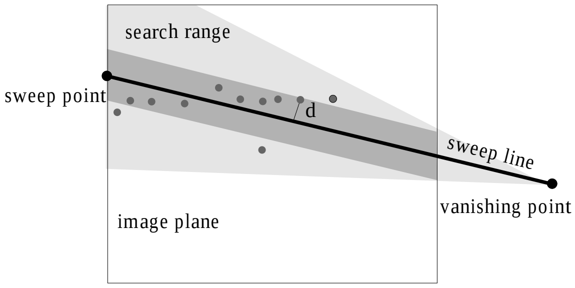


Figure 4: Illustration of line sweeping around the sweep line (line from vanishing point to the sweep point). A triangular search space (light gray area) containing the edgels roughly oriented towards the VP is intersected with a rectangular region (dark gray), which satisfies the perpendicular distance criterion.

threshold (dark gray area) are considered inliers and used for the line detection process.

The discrete step angle of the sweep line is determined from the distance threshold for inliers (see also Figure 4).

All valid edgels are sorted according to their distance from the VP and line segments are formed from those subsets, which contain a minimum number of collinear points and also fulfill a density criterion. The density criterion simply enforces that only small gaps are allowed between the edgels forming a line segment. During the sweeping process many line segments are detected, but most of them are redundant. We thus take the strongest lines, i.e. those with the highest inlier count, and assign edgels of neighboring and overlapping segments to them, resulting in new endpoints for the segment (see illustration in Fig. 5). The remaining segments are refined by a least-squares fitting to the new set of inliers.

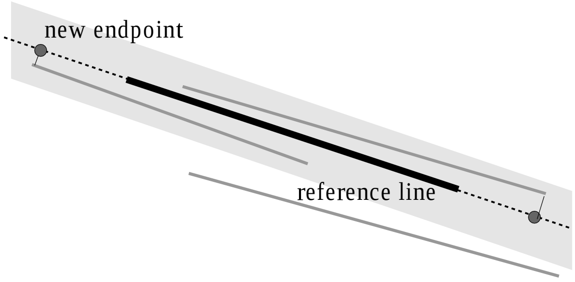


Figure 5: Illustration of the grouping process: all line segments inside the valid region (light gray area) are projected onto the reference line segment (black) and result in new endpoints for the reference segment.

The performance of the algorithm depends on the number of vanishing points that are detected in the images. If two or three vanishing points are available, the number of extracted lines proved to be significantly higher than the one delivered by a standard algorithm. In the presented example the advanced line extraction method yielded 145 percent more lines than the standard approach described in 3.1. Figure 6 shows the result of the advanced line extraction process.

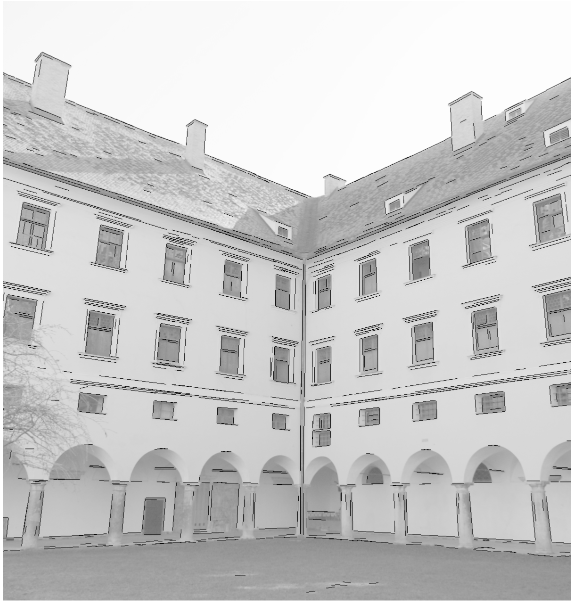


Figure 6: Detected line segments by the advanced extraction approach.

3.4 Line Matching

The set of line segments per image together with the known orientation of the image sequence are the input for line matching. Our approach closely follows the one described by Schmid and Zisserman [11]. The result of the line matching process is a set of 3D lines in object space.

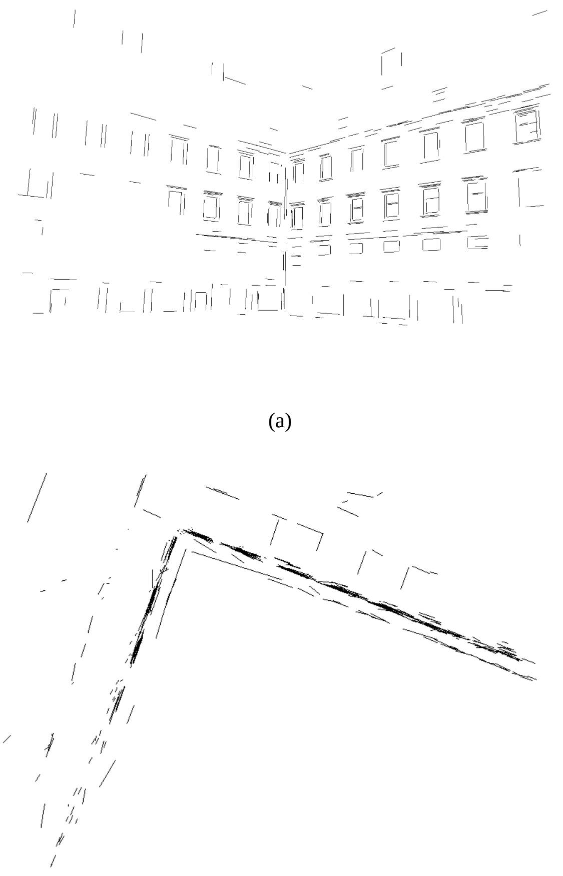
Basically the algorithm works as follows: For a reference line segment in one image of the sequence potential line matches in the other images are found by taking all lines that intersect with the epipolar lines induced by the endpoints of the reference line segment. Also lines lying completely within the beam formed by the two epipolar lines are taken into account.

Each of these potentially corresponding line pairs gives a 3D line segment (except for those, which are parallel to the epipolar line, since in this case no intersection between the epipolar line and the image line can be computed).

The potential 3D lines are then projected into all remaining images. If image lines are found which are close to the reprojection, the candidate is confirmed, else it is discarded. Finally a correlation based similarity criterion is applied to select the correct line. The method yields a set of robust 3D lines. Figure 7 shows two views of the extracted 3D line set. Obviously, due to the small vertical baseline the geometric accuracy of the horizontal line segments is limited.

3.5 Plane Detection

Once a set of 3D lines has been reconstructed, the previously detected line segments can be used to detect object planes. Two strategies are being used and will be discussed in this section. Note that in both cases we test support by lines rather than points, because our experiments show that the accuracy of the reconstructed lines is about 3 times higher than the accuracy of the points.



(a)

(b)

Figure 7: Two views of the extracted 3D line set of the facade in Figure 6. (a) front view (b) top view

3.5.1 Feature-based Plane Sweeping The principal direction of a building are known from vanishing point detection. We can thus construct a plane orthogonal to each direction and sweep it along the its normal vector. Plane positions which get high support by the object lines are candidates for object planes. The support function for a plane $\mathbf{P}$ is defined as

$$S_p = \sum_{\forall\, \mathbf{l}_i :\, d_i \leq d_{max}} \left(1 - \frac{d_i}{d_{max}}\right) \tag{3}$$

where d_i is the deviation of line $\mathbf{l}_i$ from the plane and d_{max} is a threshold which separates inliers w.r.t. the plane from outliers.

The plane parameters for the detected planes are then computed with a least-squares fit to the line segments' endpoints.

3.5.2 3D Hough Transform Obviously the sweeping method will miss any plane, which is not normal to a principal building direction. For such cases a more general plane-detection algorithm is needed. The 3D Hough transform has been proposed for similar applications, e.g. by Bian and Yang [1] and Vosselman [15]. We have extended this method to mapping lines to a 3D parameter space: each straight line in 3D-space is mapped to a line in a 3D parameter space. The intersection point of several such lines then gives the parameters of the common plane.

The plane parameters which are given by the Hough transform are not yet very accurate due to the discretization of the parameter space. Therefore we have to search for the supporting line segments of each plane. This has to be done with a larger threshold than in section 3.5.1 in order to be sure to find all inliers in spite of the inaccurate plane position. The final plane parameters are then computed with a robust least-squares fit to the line segments' endpoints.

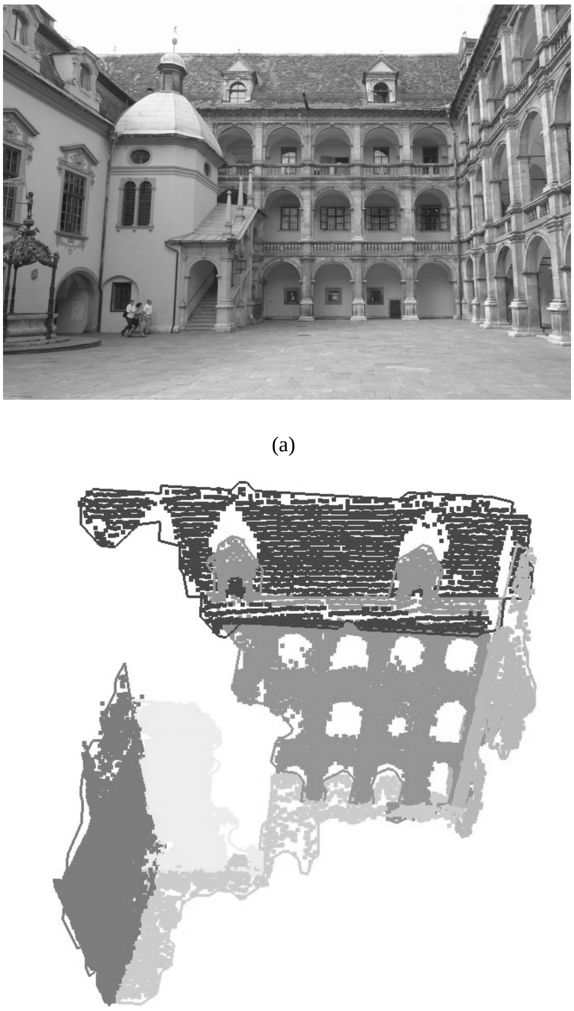
3.5.3 Plane Verification The planar patches are verified and their borders constructed using a dense set of matched object points. Many algorithms for dense image matching are available to achieve such an unstructured point-set, e.g. Koch [5], Sara [9], Zach et al.[16]. We use the approach of Zach et al. A distance threshold is applied to the point-set to find the points belonging to each plane and only planes with a minimum number of supporting points are accepted. Up to now we prefer not to repeat the least-squares fitting with the points, because of their lower accuracy. Jointly fitting the planes to points, lines and arcs with different weights for each class of primitive may be the best solution. Further research is needed to clarify this.

Using the dense set of matches for plane verification implicitly exploits the gray-value correlation between the images which have been matched to reconstruct the object points. In this sense the procedure is related to the plane-sweeping approach proposed by Zisserman et al. in [17]. However their strategy is entirely based on gray-value information whereas we explicitly compute low-level features in object space from the gray values and use their coordinates for further computations.

To detect the boundaries of the planar patches the point are projected onto the plane. The plane is then sampled to a binary image which is 1 for pixels containing points and 0 for pixels not containing points. After applying an iterative median filter to remove sampling artefacts the binary image is segmented and the borders of regions containing points are mapped back to object space. An example is given in Figure 8.

4 CONCLUSIONS AND FUTURE WORK

We have presented a framework for robust feature extraction from facade images and used 2D line sets from multiple views for 3D reconstruction of facades. The line segment extraction from contour chains and the advanced line segment extraction using previously extracted vanishing points result in dense line sets for each image in the sequence. The new line segments can also be used to adjust the position of the vanishing point, especially if the vanishing points are used to determine the relative orientation between two images.



(a)

(b)

Figure 8: Piecewise planar reconstruction of the 'Landhaus' court in the historic center of Graz. (a) image from the recording sequence. (b) detected planar patches.

ACKNOWLEDGMENTS

Parts of this work have been done in the VRVis research center, Graz and Vienna/Austria (`http://www.vrvis.at`), which is partly funded by the Austrian government research program *Kplus*.

REFERENCES

[1] 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.

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

[3] J. Heikkilä. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, October 2000.

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

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

[6] K. Kraus. *Photogrammetrie II*, pages 58–61. Duemmler, 1996. ISBN 3-427-78653-6.

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

[8] C.A. Rothwell, J.L. Mundy, W. Hoffman, and V.D. Nguyen. Driving vision by topology. In *In Proceedings IEEE Symposium on Computer Vision SCV95*, pages 395–400, 1995.

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

[10] 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.

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

[12] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *IJCV*, 40(3):199–233, December 2000.

[13] T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons. The cascaded hough transform as an aid in aerial image interpretation. In *Proc. ICCV*, pages 67–72, January 1998.

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

[15] 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.

[16] 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*, page to be published, 2001.

[17] 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*, pages 9–23, 2001.