

# KNOWLEDGE-BASED AUTOMATIC 3D LINE EXTRACTION FROM CLOSE RANGE IMAGES

S. Zlatanova and F. A. van den Heuvel

Delft University of Technology, Department of Geodesy  
Thijssseweg 11, 2629JA Delft, The Netherlands  
Email: {S.Zlatanova, F.A.vandenHeuvel}@geo.tudelft.nl

Commission V, WG V/3

**KEY WORDS:** Object reconstruction, Edge detection, Feature-based matching, Topology, 3D Databases, Augmented reality

## ABSTRACT:

The research on 3D data collection concentrates on automatic and semi-automatic methods for 3D reconstruction of man-made objects. Due to the complexity of the problem, details as windows, doors, and ornaments on the facades are often excluded from the reconstructing procedure. However, some applications (e.g. augmented reality) require acquisition and maintenance of rather detailed 3D models.

In this paper, we present an automatic method for extracting details of facades in terms of 3D line features from close range imagery. The procedure for 3D line extraction consists of four basic steps namely edge detection, edge projection on one or more sequential images, edge matching between projected and detected ones and computation of the 3D co-ordinates of the best-matched candidates. To reduce the number of candidates for matching, we use the rough representation of facades (i.e. simple rectangles) obtained from 3D reconstruction procedures completed prior to the 3D line extraction. The paper presents the method, discusses achieved results and proposes solutions to some of the problematic cases.

## 1. INTRODUCTION

3D data is becoming of a critical importance for many applications in the last several years. Urban planning, telecommunication, utility management, tourism, vehicle navigation are some of the most appealing ones. The huge amount of data to be processed, significant human efforts and the high cost of 3D data production demand automatic and semi-automatic approaches for reconstruction. The research on 3D reconstruction focuses mainly on the man-made objects and more particularly the buildings. The attempts are towards fully automatic procedures utilising aerial or close range imagery. A lot of work has been already completed on this subject and the progress is apparent. However, the efforts of most of the researchers are concentrated on reconstructing the rough shape of the buildings neglecting details on the facades such as windows, doors, ornaments, etc. Depending on the application, such details may play a critical role. A typical example is an augmented reality application utilising a vision system for orientation and positioning require both accurate outlines of the building and many well visible elements on the facades. Here, we present our approach for collecting 3D details on facades. The research is a part of the interdisciplinary project UbiCom carried out at the Delft University of Technology, The Netherlands (UbiCom project, 2002).

Within this project, an augmented system is to be developed that relies on a vision system for positioning the mobile user with centimetre accuracy and latency of 2 ms (Pasman & Jansen, 2001). The initial idea, i.e. utilising only an inertial tracker, failed due to the rather large drift observed during the experiments. The current equipment (assembled within the project) is capable of positioning the user in the real world with an accuracy of 5 m (Persa & Jonker, 2001). This accuracy however does not suffice the requirements of the application

and therefore is used only for obtaining the rough location. The accurate positioning is going to be completed by the vision system, i.e. tracking features. Among the variety of tracking approaches reported in the literature, we have concentrated on tracking line features (Pasman et al., 2001). This is to say, the accurate positioning is to be achieved by a line matching algorithm between line features extracted in real time from a video camera (mounted on the mobile unit), and lines available in an a priori reconstructed 3D model (rough and detailed). The approximate positioning (obtained by the inertial tracker and GPS) provides input information to the DBMS searching engine in order to obtain the 3D line features in the current field of view. Figure 1 shows an example of such a vision system.

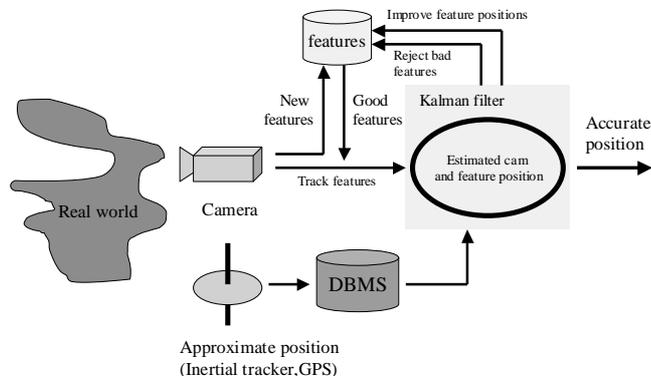


Figure 1: Typical setup of camera tracking system

The accuracy of the 3D model (rough and detailed) is the most critical requirement. The extracted 3D line features need to ensure decimetre accuracy to be able to suffice the rendering requirements. Furthermore, the tracking system has to be able to

work at different times of the day and under different weather conditions. Therefore only well visible elements have to be available in the 3D model. This is to say that influence of shadows and occluding objects (e.g. cars, trees, lamp posts) has to be reduced, if not completely eliminated.

The issues discussed in this paper refer to the process of extracting 3D line features to complete the required 3D model. Details on the 3D reconstruction procedure to obtain the rough (topologically structured model) can be found in (Zlatanova & van den Heuvel, 2001) and (Vermeij & Zlatanova, 2001). Details on the topological organisation of the 3D model in relational DBMS are given in (Zlatanova, 2001).

## 2. THE APPROACH

Since the UbiCom system aims at serving a walking person, the concentration is basically on 3D features visible from a street level, i.e. details on facades and on the terrain surface. This paper focuses on the reconstruction of details on facades. Our approach to extract 3D line features is based on two assumptions: 1) 3D rough geometry of the buildings of interest is available (e.g. Figure 1) and 2) the orientation parameters of the images are known. The 3D rough model can be obtained following different approaches: 3D automatic (Suveg & Vosselman, 2002) or semi-automatic (Vermeij & Zlatanova, 2001) reconstructing procedures or by extruding footprints of buildings from topographic maps (e.g. in ArcView, ESRI). In order to achieve the requirements for decimetre accuracy of the UbiCom project, we have reconstructed manually the 3D facades within the test area by using the commercial software PhotoModeller (Zlatanova & van den Heuvel, 2001). The facades through which knowledge on the “depth” of the 3D line features is introduced, support the 3D line feature extraction.

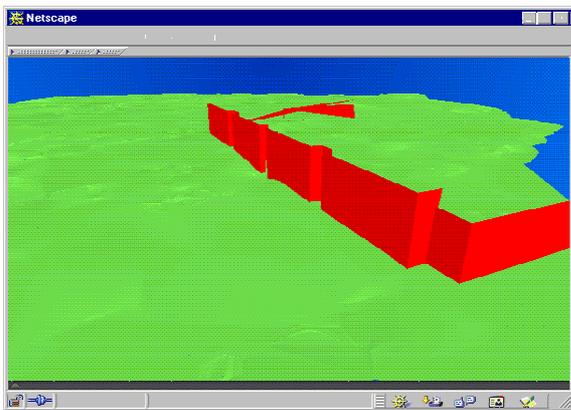


Figure 2: Rough 3D model, i.e. walls represented as rectangles

The interior and exterior orientation parameters of the images have to be available as well. In our case, we use the parameters obtained in the process of manual reconstruction, i.e. obtained by an integrated least-squares adjustment of all photogrammetric measurements in close-range and aerial images (Zlatanova & van den Heuvel, 2001).

The procedure for 3D line extraction can be separated into the following general steps: edge detection, projection of edges on the rough 3D model and back projection on the next image, edge matching, and computation of the end points of the matched 3D edges.

**Edge detection:** The edge detection utilises the line-growing algorithm proposed in (Foerstner, 1994), i.e. edges (straight lines) are extracted by grouping adjacent pixels with similar gradient directions and fitting a line through them. After calculating the gradients, the line-growing algorithm selects the pixel with the strongest gradient as a starting pixel (the normal of the edge through this pixel is determined by the grey value gradient). Then, if a pixel is eight-connected to a pixel of already classified ones and has a gradient that is perpendicular to the edge, it is added to the area that describes the line. The direction and the position of the edge are re-computed using the first and the second moments of the pixels in the edge area. The process continues until no more pixels can be added to the edge. This algorithm is performed on all the images that contain the façade of interest. The outlines of the façade (available from the 3D rough model) are used to restrict the search area to only those edges that represent features on the facades. Only edges that fall within the area enclosed by the borders of the façade are considered for further processing.

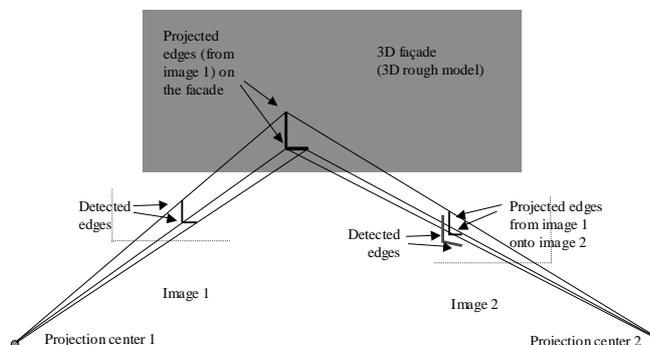


Figure 3: Knowledge-based edge projection

**Edge projection on sequential images:** Next, all the selected edges from the first image are projected onto the second image by applying intermediate projection onto the façade in 3D space. This is to say that the rays passing through the end-points of an edge (of images 1) and the projection centre 1 intersect the 3D plane of the façade into two 3D points that give the position of the edge in 3D space. This edge is back projected onto the second images, i.e. the rays passing through the 3D end-points of the edge and projection centre 2 are intersected with the image plane 2 (see Figure 3). Thus, image 2 contains already two sets of edges, i.e. projected and detected ones. Indeed, the two sets contain a different number of edges with slightly different position and a length that can vary considerably. The systematic shift in the position is influenced by the accuracy of the façade and the quality of the exterior orientation of the images, while the length of the detected edges depends on the parameters set for the edge detection

**Edge matching:** To match the projected and detected edges, we apply four constraints. The first one is related to the distance between projected and detected edges. A search algorithm looks for matching candidates within an area of interest (buffer) defined as a rectangle around the projected edge. The second constraint takes into account the number of endpoints (one or two) of a detected edge that are located within the buffer. The detected edges from the second image that have at least one endpoint falling in the buffer are considered as candidates. The third criterion filters the candidates with respect to the angle between detected and projected edges. The fourth and last constraint refers to the length of the two matched edges, i.e. the

difference between the two lengths should not be greater than a reasonable threshold. Among all the candidates, the edge that matches best is selected. Note, that an edge from image 1 may be matched with more than one edge from image 2.

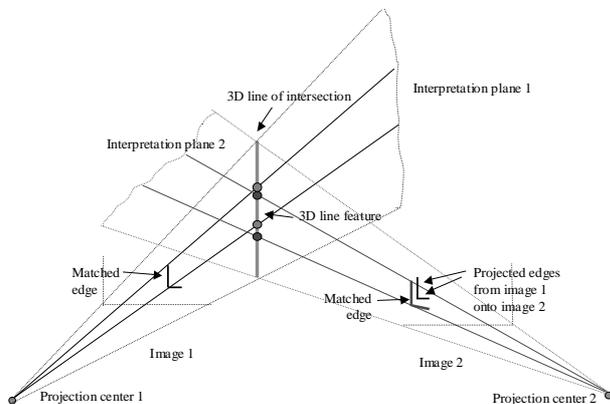


Figure 4: Forward intersection to obtain the 3D line feature

**Computations of 3D line features:** Last, the parameters of the 3D line feature are computed by a forward intersection. The algorithm considers again ray intersections between the end-points of the matched edges. Since the rays may not intersect in 3D space, a two-step procedure is applied. First, the 3D line of intersection is computed by intersecting the two interpretation planes, each defined by the projection centre and the edge in the image (van den Heuvel, 1998). Second, the rays passing through projection centres and the end-points of the edges are intersected with the 3D line of intersection. In the common case, the intersection results in four points (see Figure 4). The two points with the largest distance between them are selected as end-points of the constructed 3D line feature.



Figure 5: Aerial image of TU Delft, The Netherlands

### 3. EXPERIMENTS

The algorithms are tested on images taken with a handheld camera Kodak DCS420 (black and white) with 1524x1012 pixels of 9  $\mu$ m and a focal length of 20 mm. The images are used for both 3D reconstruction of the rough 3D model and 3D line extraction. More than 300 images are taken but actually less than 100 are considered appropriate for 3D reconstructing of the facades. For the 3D line feature extraction, we have concentrated on the building denoted with number 2 (see Figure 5) because it exhibits a very regular pattern of vertical and horizontal line features that usually cause the greatest problems in line matching. Two of the images (called here image 1 and image 2) are used to illustrate the results.

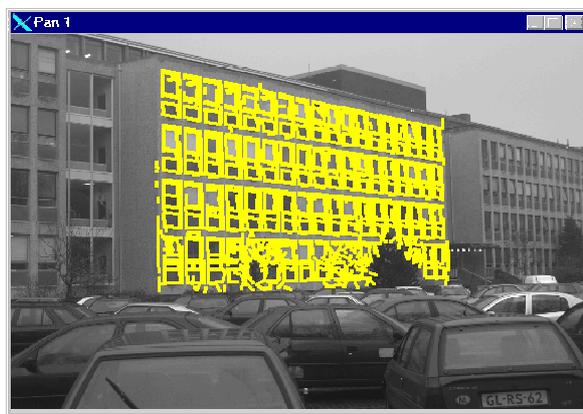


Figure 6: Detected edges on image 1 within the facade of interest

The edge detection algorithm is performed on both images with a) gradient threshold set to 1000, b) minimal length of the edge 10 pixels and maximal width 3 pixels. These settings resulted in 2363 and 2009 edges detected respectively on image 1 and image 2. The first constraint (i.e. the edges should be within the area delineated by the façade) reduced the number of edges to 631 and 217 (see Figure 6 and Figure 7). Furthermore, many "fake" edges (e.g. from cars, stairs) were eliminated.

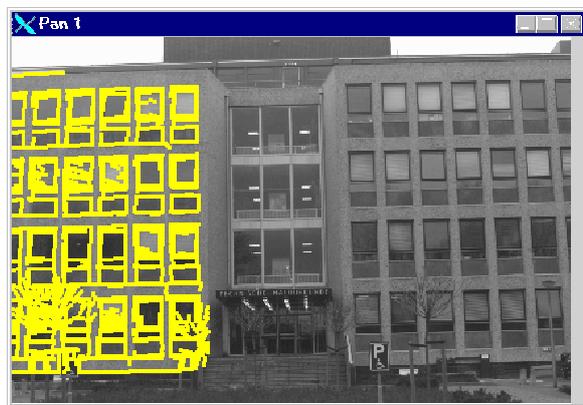


Figure 7: Detected edges on image 2 within the facade of interest

The projection of the detected edges from image 1 onto image 2 by intermediate projection onto the 3D façade propagates all the edges to the second image (Figure 8).

Figure 9 shows the difference between all the detected edges on image 2 and those that are matched with projected edges from image 1. It can be clearly seen that many fake edges from shadows, reflections or temporal conditions (e.g. open windows, the third window from right to left detected on image 1) are eliminated from the set. However, the correspondence between matched edges (Figure 9b) is not unique, i.e. each projected edge from image 1 is matched with more than one edge of image 2.

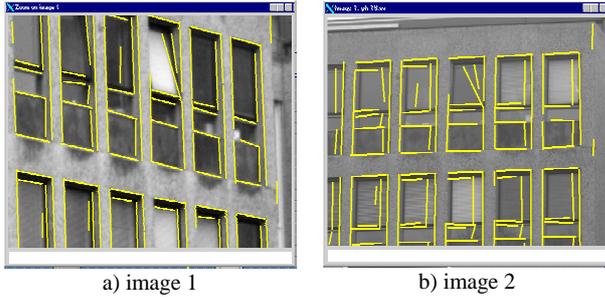


Figure 8: Detected edges (image 1) projected on image 2 (right)

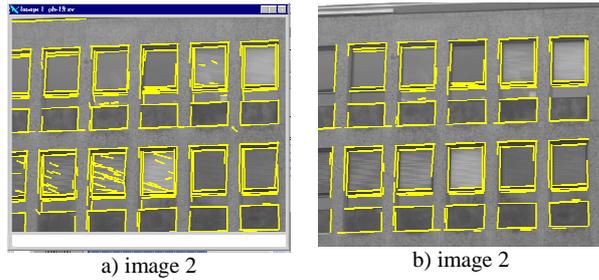


Figure 9: Detected edges (left) and matched edges (right)

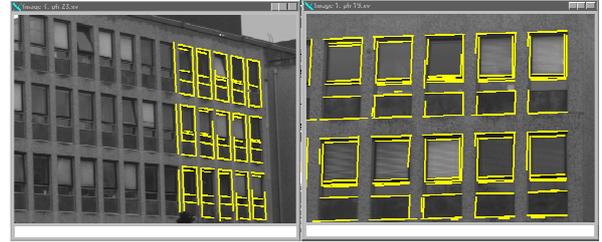
To investigate the quality of the match, we have applied several different thresholds. Table 1 shows the number of matched edges between the two images with respect to the different thresholds used.

Table 1: Thresholds and number of matched edges

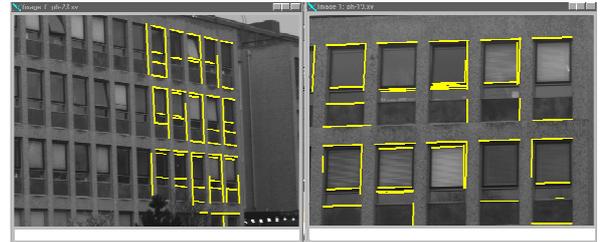
Case	1	2	3	4	5	6	7
Buffer (pix)	10	10	10	5	5	3	3
End-points	1	1	1	1	2	2	2
Length (mm)	-	0.8	0.8	0.8	0.8	0.8	0.5
Angle (degree)	3	3	1	1	1	1	1
Match. image 1	137	132	91	74	38	22	14
Match. image 2	315	257	157	96	45	26	16

Figure 10 portrays some of the matching results for both images according to the seven example cases presented in Table 1. As it can be seen from the table, the uniqueness between the matched edges varies significantly. In principle, the more restrictive constraints are applied, the more unique is the matching. In the last two cases, the correspondence is approximately one-to-one. The obtained results depend not only on the values of the thresholds but also on the order they are applied and the sequence of images that is used. For example, the threshold for the allowed difference in the length filters significant number of the matched edges (compare case 1 & case 2, and case 6 & case 7), but only if the detected edges of the first image are longer than the ones detected on the second image (as in our particular case). If the two images were

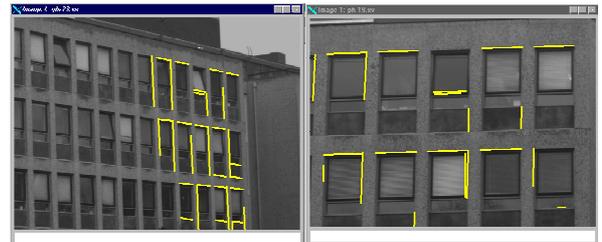
exchanged, the buffer threshold and the endpoints considered (i.e. 2) would have the same effect.



a) case 2: buff=10, end-points=1, length=0.8, angle=3



b) case 4: buff=5, end-points=1, length=0.8, angle=1



c) case 5: buff=5, end-points=2, length=0.8, angle=1



d) case 6: buff=3, end-points=2, length=0.8, angle=1



e) case 7: buff=3, end-points=2, length=0.5, angle=1

Figure 10: Matched edges on image 1 and image 2

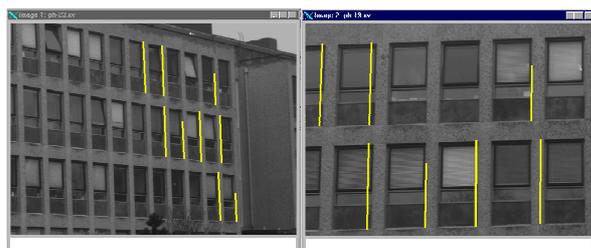
Finally, to compute the 3D line feature out of the matched edges, we have used all the candidates. As mentioned above, the two end points of the 3D line are the two most distant points on the 3D line of intersection (see Figure 4). Therefore all the 3D lines are longer than the edges detected on the images (see Figure 11). We applied only one constraint at this step, i.e.

maximum allowed 3D line feature length is restricted to 3 m. The constraint is selected with respect to the particular size of the line features that can be expected on the given façade.

Figure 12 shows some of the lines longer than 3 m. Predominantly those are a result of horizontal edges, which interpretation planes intersect in a very small angle and thus often far away from the façade they belong to. Since these lines are eliminated from the final set, vertical lines are the only ones visible on Figure 11. For example, in case 7 (despite the good match) only one line is left after the intersection (i.e. the vertical line in Figure 10e).



a) case 4: buff=5, end-points=1, length=0.8, angle=1



b) case 6: buff=3, end-points=2, length=0.8, angle=1

Figure 11: Vertical 3D lines back projected on the images

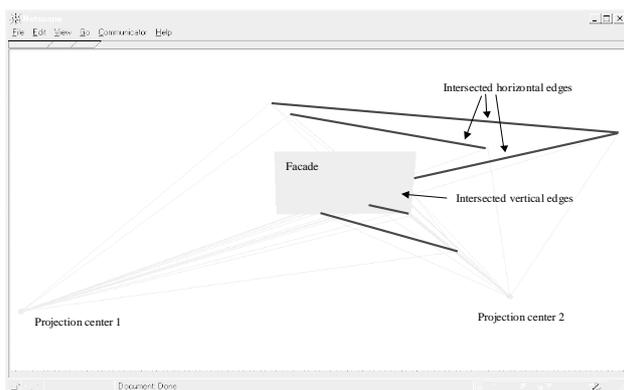


Figure 12: Intersection of interpretation planes.

#### 4. DISCUSSION

Currently, the experiments are concentrating on tuning thresholds to reduce the number of possible candidates for matching and improving the intersection of the matched candidates.

#### 4.1 Edge matching

The experiments clearly showed that the utilisation of a rough 3D model (e.g. façades of interest) significantly improves the quality and quantity of candidates for matching. The benefits of this approach can be summarised as follows:

- The number of edges to be processed is limited to those that do belong to the façade of interest.
- All the edges from the first image can be transformed to approximately the corresponding position on the second image
- The search of candidates for matching can be conducted in a very limited area of interest around detected edges. Compare to the epipolar line match, which fails to match edges of which the endpoints are in the same epipolar plane, the algorithm successfully finds matches regardless the direction of the edge.
- The angles of interest allows to eliminate fake edges detected in one of the images like shadows, reflections, branches of trees (a very common case for images taken from street level), etc.

However a large number of edges detected on one image (that may be considered as real 3D features) still cannot be matched due to a number of reasons:

- Lack of visibility (e.g. the façade is only partly visible on the second image) or occlusion, possibly by other objects (e.g. trees).
- The edges are not detected on the second image due to lower contrast.
- The position of the feature changed while the images were taken (e.g. a window or door is opened or closed).
- The accuracy of the rough model used for the depth assessment. Features that are in front of or behind the used plane of the façade are systematically shifted to the right or left on the second image. This shift may appear larger than the interest area used for finding candidates.
- The area of interest (buffer) depends very much on the size of the features that can be expected and has to be tuned very carefully by many experiments with different images.
- Since the visibility of edges is not equal on the different images, the same edge (even well visible) may appear with different length (covering even two features). For example, the edge on the upper-right window (Figure 8a) is wrongly detected as a very long edge and it will be matched with two edges (Figure 9c). The two constraints, i.e. end points and difference in the length of the candidates, will most commonly eliminate such edges (compare with Figure 10), although real 3D line features have to be encountered there. An eventual solution could be found by tuning the parameters that are used for the edge detection.

#### 4.2 Interpretation plane intersection

After matching the edges of the two images, the interpretation planes of the two corresponding edges are intersected to obtain the parameters of the related line in space. However, the quality of the parameters of this 3D line depends on several factors:

- The quality of the match. Are the matched edges indeed projections of the same object edge?

- The precision of the two interpretation planes that depends on:
  - The precision of the parameters of the edges in the images (again dependent on image quality, method of extraction, and line length).
  - The quality of interior and exterior orientation.
- The imaging geometry. Mainly the angle between the planes is of importance for interpretation plane intersection.

In the example of section 3 the angle between the interpretation planes of two corresponding horizontal lines is very small due to the fact that the line between the two projection centres is also horizontal and parallel to the building. Indeed the horizontal 3D lines show large errors (Figure 12). In conclusion, before intersection the angle between interpretation planes has to be checked for an acceptable minimum value.

In order to avoid losing many lines as in the example, rules of thumb have to be formulated for the image acquisition. For example, images taken at different height levels would improve significantly the intersections in section 3. Two other solutions can be chosen as well. Firstly, the 3D lines that result from bad intersections can be positioned in the a priori object plane when the precision of the plane position is better than the (depth) precision of the intersection. Secondly, the projections of horizontal and vertical edges of the building could be linked when their endpoints are only a few pixels apart, assuming that their intersection corresponds to a point on the building. Then the four related interpretation planes (two for the horizontal and two for the vertical edge) could be intersected using least-squares adjustment (van den Heuvel, 1999). More general, this adjustment involves a number of interpretation planes that equals the number of images (possibly more than two, see next section) times the number of linked edges.

#### 4.3 Multiple image matching

In the current approach, the intersection of the two interpretation planes is not redundant. Reliability can be obtained by taking into consideration one or more additional interpretation planes utilising more images. In general, the quality of the intersected 3D line in terms of precision and reliability improves with each additional image in which the same object edge is extracted. Furthermore, the quality of the matching improves, i.e. the number of erroneous matches is expected to drop. Whilst in close-range photogrammetry considerable research efforts have been directed to multiple image point matching (Maas, 1992), multiple image line matching is still insufficiently explored. In this respect we consider the image line matching a promising topic for future research.

### 5. CONCLUSIONS

We have presented our preliminary results on 3D line extraction to provide line features needed for the accurate positioning for an augmented reality application. Although the approach was inspired by the 3D model reconstruction procedures applied within the UbiCom project, it contributes to the more general research on 3D line extraction. The obtained results exhibit a number of positive findings as the edge matching is concerned. Further research has to be carried out for improving the interpretation plane intersections to obtain the 3D line feature

parameters. Utilisation of multiple images should be one of the first steps towards improvement.

### REFERENCES

- Foerstner, W., 1994, A framework for low level feature extraction, in: *Computer Vision - ECCV94*, Springer Verlag, Berlin, vol.2, pp.283-394
- van den Heuvel, F.A., 1998, Vanishing point detection for architectural photogrammetry, *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXII part 5, Hakodate, Japan, pp.652-659
- van den Heuvel, F.A., 1999, A Line-photogrammetric mathematical model for the reconstruction of polyhedral objects. in *Videometrics VI, Sabry F. El-Hakim (ed.), Proceedings of SPIE Vol.3641*, ISBN 0-8194-3112-5, pp. 60-71
- Maas, H.-G., 1992, Robust Automated Surface Reconstruction with Structured Light. in: *International Archives of Photogrammetry and Remote Sensing*, Vol.29, Part B5, pp.709-713.
- Pasman, W., and F. W. Jansen, 2001, Distributed Low-latency Rendering for Mobile AR, in: *Proceedings of IEEE & ACM International Symposium on Augmented Reality*, 29-31 October, Columbia University, NY, pp. 107-113.
- Pasman, W., S. Zlatanova, S. Persa, J. Caarls, 2001, Alternatives for optical tracking, *UbiCom internal report 30/5/1*, <http://www.cg.its.tudelft.nl/~wouter/publications/>, (accessed June, 15<sup>th</sup> 2002), 12p.
- Persa, S. and P.P. Jonker, 2000, Hybrid Tracking System for Outdoor Augmented Reality, in: *Proceedings of 2nd Int. Symposium on Mobile Multimedia Systems and Applications MMSA2000*, 9-10 November, Delft, The Netherlands, pp. 41-47.
- Suveg, I. and G. Vosselman, 2002, Automatic 3D Reconstruction of Buildings from Aerial Images, in: *Proceedings of SPIE Photonics West, Electronic Imaging*, 19-25 January, San Jose, USA, CDROM
- UbiCom project, 2002, <http://bscw.ubicom.tudelft.nl/> (accessed June, 15<sup>th</sup> 2002)
- Vermeij, M. and S. Zlatanova, 2001, Semi-automatic 3D building reconstruction using *Softplotter*, in: *Proceedings of International Symposium on "Geodetic, Photogrammetric and Satellite technologies: development and integrated applications"*, 8-9 November, Sofia, Bulgaria, pp. 305-314
- Zlatanova, S. 2001, 3D modelling for augmented reality, in: *Proceedings of the 3<sup>rd</sup> ISPRS Workshop on Dynamic and Multi-Dimensional GIS*, 23-25 May, Bangkok, Thailand, pp. 415-420
- Zlatanova S. and F. van den Heuvel, 2001, 3D city modelling for mobile augmented reality, in: *Proceedings of International Symposium on Architectural Photogrammetry*, 8-21 September, Potsdam, Germany, 8p. (to be published).