3D LINEAR FEATURE EXTRACTION AND MATCHING

Gianfranco Forlani
Associate Professor, Department of Civil Engineering
University of Parma
Viale delle scienze, 43100 Parma
E-mail: forlani@parma1.eng.unipr.it
ITALY

Carla Nardinocchi
PhD, Department ITS
University of Rome
E-mail: carla@anvax1.cineca.it

Raffaella Rizzi
Civil Engineer, PhD student, Politecnico of Milan

KEY WORDS: 3D matching, Feature extraction, Object reconstruction

ABSTRACT

The aim of this work is to extract and match line segments from terns of images and to compute the spatial coordinates of the endpoints of their intersection in object space, to improve the automatic reconstruction of objects, complementing area-based methods. To this aim, three images are used to increase reliability. The feature extraction strategy is based on the (Burns et al., 1986) approach. The line segments so found are first grouped in terns made of likely homologous, based on the a priori infomation on exterior orientation and on object shape (approximate DEM of the object). The set of candidates is then checked to highlight inconsistencies between the hypothesized terns, by intersecting the image rays of representative points selected on the line segments. The method, applied to simulated examples, works correctly; its application to a set of images in an architectural environment is satisfactory, though errors of omission and commission are still present.

1. INTRODUCTION

Algorithms and techniques for 3D object reconstruction, namely the determination of the object shape by mono, stereo and multiple images, are grouped in two broad areas: signal based and feature based. Experience showed performance, merits and limits of both methods in different areas of application. It is well established that correlation methods are very accurate in industrial photogrammetry, where targets are routinely used to squeeze the highest accuracy, and very productive in aerial photogrammetry, where DTM can be produced from measurements of tens of thousands of points per image. The performance and the fidelity of the object representation, though, is less satisfactory the more we move towards large scale images, specially in urban areas, where the implicit assumption of smoothness of the object is violated. This has a twofold impact over the results obtained by correlation methods: the hypothesis of locally plane object surface of the l.s. matching doesn't hold, therefore the localization of the homologous point is less reliable; besides, the number and the distribution of the points measured on the surface may not be enough to define the object breaklines (edges, corners, etc...). In this respect, the choice of an appropriate interpolation method is also very important, since the degree of smoothness of the method will generally affect the result. No matter the method, though, it must be noted that the object shape will be correctly reproduced only if the sampling frequency is adjusted to object roughness: in other words, there is no interpolation approach capable of make up for lacking of data.

In man made objects and particularly in buildings most edges are in fact straight lines, therefore very simple to delineate (just the endpoints are necessary) by a human operator. In contrast to that, if the same edge is to be delineated by area based methods with the same accuracy, hundreds of points may be required over the two sides of the edge. Matching algorithms based on feature extraction (mainly of linear features), are therefore potentially more economic and accurate in object delineation (Haala,1995; Perdersini et al., 1996; Medioni, Nevatia, 1984). The common scheme of all these strategies begins with some feature extraction technique, where the edges in the image are detected and classified, leading to a vector image description. Then, based on the available a-priori information (e.g. imaging geometry, object shape) line segments are coupled or grouped in potentially matching sets and, defining an appropriate cost function to weigh each set, a minimum cost configuration is achieved, which should correspond to the correct line association. Here one more such algorithm is presented, where, to increase reliability, line segments are extracted and matched from three images.

2. FEATURE EXTRACTION

2.1 Image segmentation

The purpose of image segmentation is to group the image pixels in regions satisfying a given criterium, based on texture or edge properties. Here the latter approach is used, taking into account either gradient magnitude and gradient direction and is performed in three steps.

1. Noise reduction by some operators. This is necessary because the small mask used is very sensitive to noise. Among the many operators available, capable of dealing with image noise and at the same time to preserve edge sharpness (Lemmens, Han, 1988; Canny, 1986), we found the Conditional Averaging Filter (CAF) satisfactory. To run CAF over the image, a threshold must be fixed: this is done based on the visual inspection of the results.
2. Image convolution with the gradient operator. Edge pixels will be therefore identified by large gradient magnitudes and a threshold must be introduced to separate them from the background
3. Grouping pixels belonging to the same edge. To this aim it can be noted that, as long as the gradient orientation is constant between contiguous edge pixels, the corresponding edge portion belongs to a line segment.

No matter the mask used to approximate the g.v. gradient vector, different threshold have to be fixed to select edge pixels. Thoug the performance of edge operators is sometimes very good, the segmentation output still will be affected by image characteristics and by the choice of these threshold values, either making life easy for grouping algorithms or preventing them from getting any acceptable outcome. Operators with large masks tend to increase smoothing, loosing details; small masks instead are very sensitive to noise. We used the edge operator suggested by (Burns et al., 1986) with a small 2x2 mask which allows also to recover gradient orientation.

2.2 Linear feature extraction

We look for a description of the image contents based on lines. This may be achieved in many ways, e.g. by line following, Hough transform etc. (Ballard and Brown, 1982); we opted for an alternative suggested by (Burns et al., 1986), with minor changes. The concept is the following: we get a line segment from each region where the gradient magnitude is above a specified threshold and its orientation is within a certain range. The straight line to which the segment belongs will be defined by the gravity centre of the area and by the direction perpendicular to gradient orientation. The end points of the segment will be determined by projecting the points of the area over the straight line (Forlani et al., 1996).

Since the most critical parameter in this stage is the threshold for the gradient magnitude, we tried to get an empirical rule capable of tying it to the actual dataset: on the images we processed, the mean value of the gradient magnitude proved to be satisfactory.

After the image gradient has been computed, the orientation of the gradient in each pixel where the magnitude exceed the threshold is computed and assigned to a partition of the interval [0-2π]. The choice of the number of partitions is often critical: too many partitions lead to a very fragemented image; too few partitions yield a rough approximation of the edge. Either 12 or 24 partitions were used.

The set of all contiguous pixels belongin to the same partition make the so-called support region of the edge. The segment orientation is computed by the L-1 norm (the median value of all orientation) while the gravity centre is computed as a weighted mean, by using the gradient magnitude to weight each contribution.

We have now a vector representation of the edges, where each line segment is defined by its orientation, its gravity centre, its end points; additional information on the goodness of fit of the straight line to the support region is provided by the dispersion of the orientation values and the dispersion of the pixels about the straight line.

3. 3D FEATURE MATCHING

Matching the features extracted in each image in the previous processing stage is accomplished in two steps: first, a preliminary list of terns of objects is built; then the terns are collected in disconnected groups and their mutual consistency is verified. To this aim, as already stated, auxiliary information on object are necessary: we assume the knowledge of the exterior orientation of the images and an approximate object shape, given by a DEM of the object itself and its surroundings.

3.1 The preliminary list of correspondencies

In order to select the line segments likely to be corresponding, we begin comparing the images pairwise. In turn, each image of the tern becomes the reference and its line segments are projected over the two other images, to define a local search window where the homologous segment should lie. Let be a the reference image, b and c the second and third images. The limits of the search window relative to a specific line i are defined as follows (see Fig. 1):

1) the endpoints of the line are projected on the DEM and their approximate object coordinates are computed. To this aim, based on the pixel-to-image coordinate

transformation, and correcting for lens distortion and atmospheric refraction if required, the pixel coordinates are converted to image coordinates and the intersection of the projecting ray with the DEM closest to the projection centre is computed on the way down. Likewise, when projecting over *a* and *b*, occlusions are accounted for.

2) Since the DEM is only approximate, based on an estimate DZ of its uncertainty, the points over the image rays from the endpoints of *i* are defined. All four points are then projected over the image *b*, defining a quadrilateral including the endpoints of the segment *i* in image *b*; the same procedure is applied to image *c*. Provided that the uncertainty of the DEM has been properly set and that occluding object parts are represented in DEM, both quadrilaterals will certainly include the true endpoints of *i*. Indeed the two opposite sides of the quadrilateral lie along the epipolar lines from the endpoints of *i* and account for the uncertainty of the DEM (that of the exterior orientation is not explicitly taken into account but it may be assumed negligible compare to the uncertainty of the DEM). Besides, if the line *i* exist in image *a* and *b*, his slope in the image system must be in the range defined by the angle formed by joining the opposite corners of the quadrilateral.

3) All lines *j* and *k* in image *a* and *b* respectively, which cross or originate from the quadrilateral are checked against two conditions, in order to be considered as candidates likely to match with *i*:

- their slope must be in the above defined range;
- the gradient orientation is in the same direction of the projected vector.

4) Once the two lists are completed, they are searched for terns of candidates: line *i* is coupled with all *j*'s and each pair *i,j* with all *k*'s in *c*: this builds up the preliminary list of terns with respect to *a*.

5) After the procedure runs over all three images, we end with 3 ordered list of terns. The lists are then searched for to find all terns which are common to all three lists. This should at least partially get rid of "false" proximities between lines due to the effect of the perspective: if the images are taken with different angles, at least in one image the true spatial separation should become apparent and, if larger than the window search, exclude the wrong coupling.

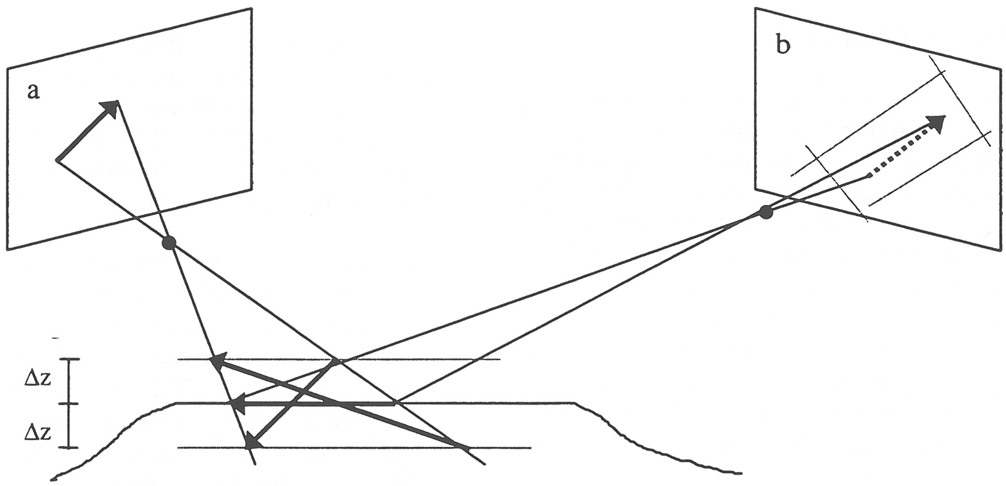


Figure 1 – Definition of the search area on image *b*

3.2 Grouping terns in the preliminary list.

The next stage of the procedure establishes groups of terns, by extracting from the above three lists sets of segments which are not connected with each other: a group will therefore include all terns of the three images which share one or more line with other terms and therefore are likely to match with more than one segment.

The number of groups depends on the object characteristics: if the edges in the object are not evenly distributed (e.g. there are clusters of edges far apart from each other, or at least well separated), then each group will refer to a sub-structure of the object and will be made of a relatively small number of terns. On the contrary case, when edges can be found everywhere on the whole object, it may happen that a single group will include all terns (representing the union of the search windows of each tern element).

By construction, each group contains ambiguities to be solved, but the search for matches, if any, can be restricted to group elements.

As a first step towards this goal, we try to bridge over small gaps between strictly collinear segments, under the assumption that they were possibly caused by noise or other disturbances which prevented the feature extraction stage to recover them as a single entity (one and the same edge). The obvious danger in this

operation is that arbitrary and wrong connections may arise, especially working on a single image. We therefore perform it only at this stage (after false connections due to perspective have been possibly eliminated), checking that the prolongation of one edge in an image gets support from the existence of a long edge in the other. This should robustify the operation, allowing at the same time to fix slightly large thresholds for the offset and the orientation difference between two segments to be joined.
We are now in a better position to discriminate false or impossible matching within the group: for each tern, a line *i* is selected. The overlap, computed by intersecting this line with the epipolar from the endpoints of *j* and *k* (see Fig. 2) , must be larger than a specified thereshold, on the assumption that true matchings must show some significant overlap between the corresponding lines, otherwise the tern is rejected.

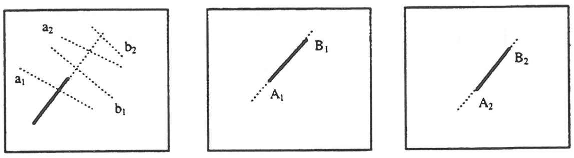
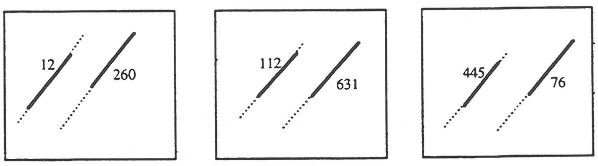


Figure 2 – Overlap between lines in a tern

3.3 Final selection of the terns

At this stage of the work, groups are divided into sub-groups. To understand how this is done and why, think of the case shown in Fig. 3: assume (a), (b) and (c) represent a group and that 12-112-445 and 260-631-76 represent two correct matches. If we would accept only one tern within a group, we will wrongly discard that with the smaller score, based on the final consistency criterium. Therefore we create sub-groups. Note that we must avoid that splitting the group may allow the possibility that two terns may be accepted, sharing one line: this would happen e.g. if in (b) only line 631 is contained. In such cases, only one tern must be accepted, if any.



(a) (b) (c)

Figure 3 - Splitting a group in subgroups

For each sub-group we find whether there is a tern of lines consistent with each other. To this aim, a reference image is selected and the intersections of *i* with the epipolars from the endpoints of *j* and *k* are computed. The midpoints of the two closest intersections is taken as check-point. Its nominal homologous in the other images are computed by inersecting *j* and *k* with the epipolar line from the check-point. For each tern of the subgroup we have therefore three pairs of image coordinates and we can check their consistency by the l.s. adjustment of the collinearity equations, getting also the 3D coordinates of the point. The consistency is measured by the estimates for σ_0 of the adjustment: based on image resolution and the error location of the edge, we can establish a threshold and reject all terns exceeding that value. Among the terns which are acceptable at the given the accuracy level, we pick the best as the most likely correct. Therefore each subgroup may end either with only one tern or without any tern.

4. EXTRACTION OF 3D SEGMENTS

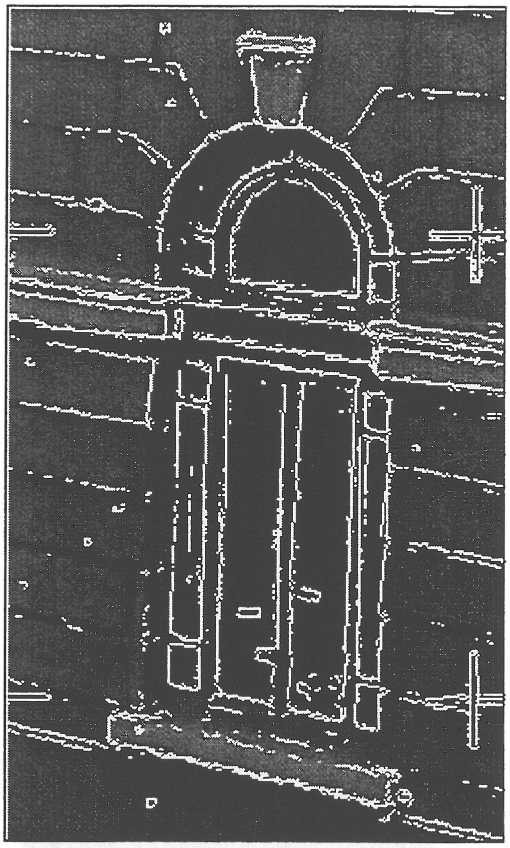
Once the triplets of segments have been recognized, they should determine a line segment in 3D. To this aim, since in general the endpoints of the segments will not be homologous, the intersection of the overlaps of *j* and *k* on *i* are computed and back projected on *a* and *b*. The object coordinates of the endpoints can now be computed, again by space intersection.

5. APPLICATIONS

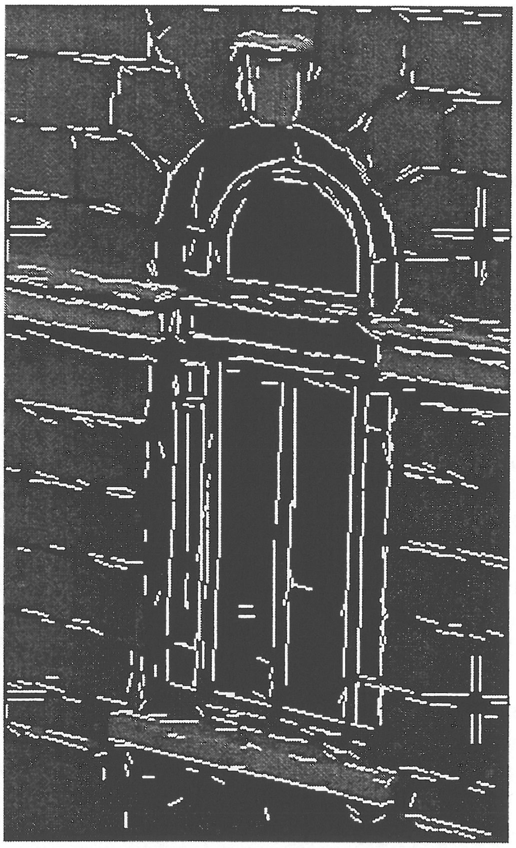
The mtohd has been first applied to simulated test cases, skipping the feature extraction stage and providing directly different sets of line segments in the images. No specific testing was performed on the influence of the approximation of the DEM; object shapes were relatively simple, without occlusions in the images.
Later the algorithm has been tested using the images of the facade of a building in the Politecnico of Milan. The images were acquired with a Rollei 6006 with a 40 mm lens. The image scale of the three images used varies from about 1:200 to 1:400; digitization was performed on a DTP scanner at 25 μm resolution.
Figure 4 shows the results of the feature extraction with different number of partitions, gradient operators and with or without displaying the segments less than 5 pixel long: it can be seen that there is much fragmentation and that some line do not appear to approximate the actual edge.
The DEM was simply taken by computing a plane through the control points on the facade used to orient the images. The uncertainty was assumed to be 25 cm.
Fig. 5 depict the result of the projection of the lines from the reference image to one of the slaves: the approximation is good for those on the facade, while those inside the window are obviously misplaced.
Finally, Fig. 6 shows the 3D lines succesfully matched.

6. CONCLUSIONS AND PERSPECTIVES

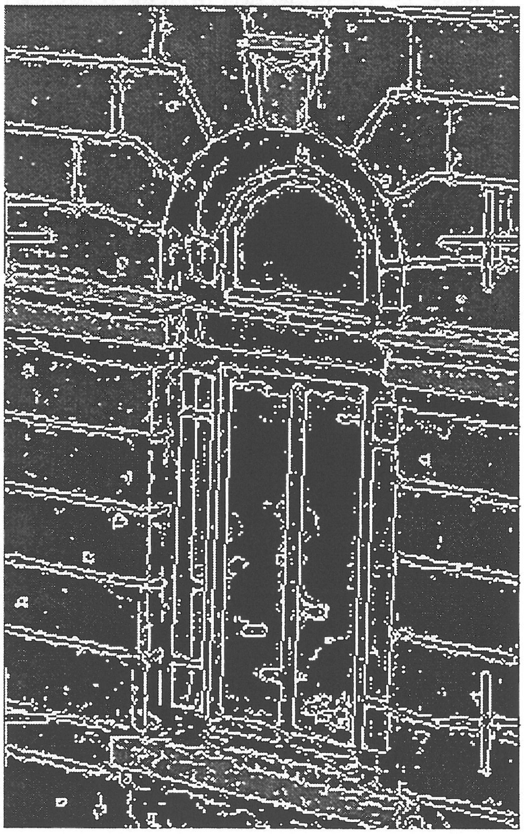
The results shown above are clearly not very satisfactory (but they are the first obtained on a real



a) Roberts operator, 12 partitions



b) Sobel operator, 24 partitions



c) Sobel operator, 12 partitions, with noise

d) Sobel operator, 12 partitions

Figure 4 - Feature extraction with different gradient operators and number of partitions

case) demonstrating that extensive testing and perhaps changes in the approach are necessary. Some error in the code may not be ruled out yet, since the position of some edges is clearly wrong. As it is often the case, much blame at the first sight may be attributed to the feature extraction stage. Two points in particular need more investigations: the straight line interpolation of the support region and the choice of the partitions that may lead to excessive fragmentation of the edges. Still to be verified is the influence of the DEM approximations (but note that small occlusions were successfully handled).

References from Journals

Burns J.B., Hanson A.R., Riseman E.M., 1986. Extracting straight lines. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, n. 4, pp. 425-455.

Medioni G., Nevatia R., 1984. Matching images using linear features. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, n. 6, pp. 675-685.

Canny J., 1986. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, N.6, pp.679-698.

References from Books

Ballard Brown D. H., 1982 .Computer Vision. Prentice Hall Englewood Cliffs New Jersey,1982.

References from Other Literature

Forlani G., Malinverni E.S., Nardinocchi C., 1996. Using perceptual grouping for road recognition. In: Int. Archives of Photogrammetry and Remote Sensing, Vol. 31, Part B3, Vienna, pp. 202-206.

Haala, N., 1995. 3D building reconstruction using linear edge segments. In: D. Fritsch, D: Hobbie, eds, "Photogrammetric Week '95", Herbert Wichmann Verlag, Heidelberg, pp.19-28.

Lemmens M.J.P.M., Han C. S. L. A., 1988. On the geometric accuracy of some differential-type of edge detectors and non-linear smoothing filters. Int. Archives of Photogrammetry and Remote Sensing, Vol. 27 Part B11, pp. 456-467.

Pedersini F., Sarti A., Tubaro S., 1996. A multi-view trinocular system for automatic 3-D object modeling and rendering. International Archives of Photogrammetry and Remote Sensing (IAPRS), Vol. 31, Part B5, pp. 506-511.

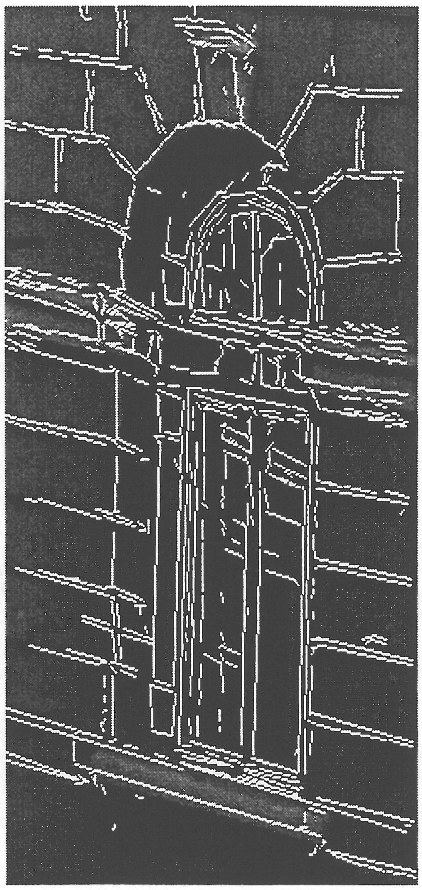


Figure 5 – The projection from the reference to the slave image

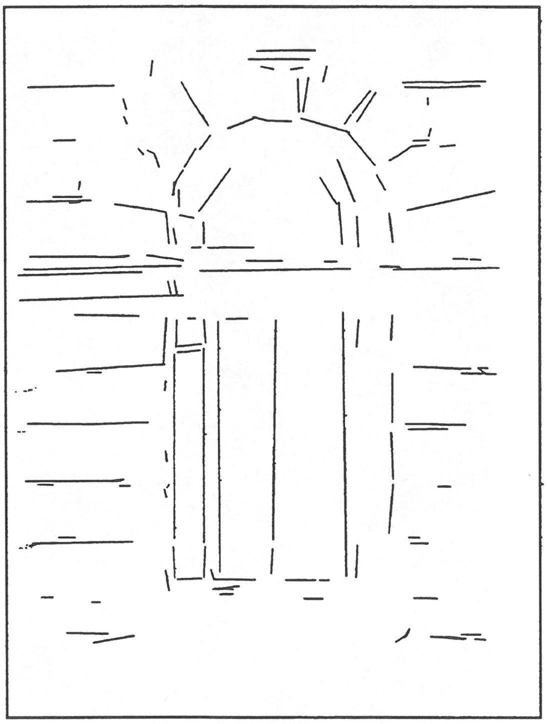


Figure 6 – The 3D lines matched