FORTH-ICS / TR-208 August 1997

Establishing Straight Line Correspondence

Manolis I. A. Lourakis

ABSTRACT: Straight line segments are common in images of man-made environments. Due to their mathematical simplicity and the ease with which they can be extracted from images, they have been widely used in vision applications. This paper presents a method for extracting and corresponding line segments between two views. Line segments are extracted by detecting edges and then fitting straight lines to the resulting edge points. For each of the line segments extracted from one image, a set of potential matches in the other image is formed. The strength of each potential match is measured by a matching function based on a few line attributes. A relaxation labeling technique that imposes global constraints on line matches is then used to disambiguate line matches. Results from the application of the method on real images are also presented.

Establishing Straight Line Correspondence

Manolis I. A. Lourakis

Computer Vision and Robotics Laboratory
Institute of Computer Science (ICS)
Foundation for Research and Technology --- Hellas (FORTH)
Science and Technology Park, Heraklio, Crete
POBox 1385, GR-711-10 Greece
http://www.ics.forth.gr/proj/cvrl
E-mail: lourakis@ics.forth.gr, Tel: +30 (81) 391704, Fax: +30 (81) 391601

Technical Report FORTH-ICS / TR-208 --- August 1997

ABSTRACT: Straight line segments are common in images of man-made environments. Due to their mathematical simplicity and the ease with which they can be extracted from images, they have been widely used in vision applications. This paper presents a method for extracting and corresponding line segments between two views. Line segments are extracted by detecting edges and then fitting straight lines to the resulting edge points. For each of the line segments extracted from one image, a set of potential matches in the other image is formed. The strength of each potential match is measured by a matching function based on a few line attributes. A relaxation labeling technique that imposes global constraints on line matches is then used to disambiguate line matches. Results from the application of the method on real images are also presented.

1 Introduction

Edges are among the most important features of objects. They are defined by large local intensity changes, which can be due to the the geometric structure of objects, the difference of surface reflectance properties or nonuniform illumination. Edges in the form of straight line segments are very common in man-made environments. Since straight lines have simple geometric properties, linear segments have been widely used in motion analysis, stereo and object recognition. Recently, the application of projective geometry to vision has led to new developments such as the notion of geometric projective invariants [12, 3, 16] and the geometry of multiple views [14, 15], that have reemphasized the significance of line segments as image features. Moreover, important practical reasons are in favor of the use of line segments to solve vision problems: First, line segments are easier to extract from digital images compared to nonlinear features such as points and curves. Second, they capture more of the information contained in images than that captured by point features. Finally, line segments can be localized in images with subpixel accuracy.

This paper proposes a method for solving the correspondence problem between two sets of line segments extracted from a pair of images. The method is intended to be used in analyzing image motion, although it can be useful for other applications such as feature-based binocular stereo and object recognition. Previous approaches for matching line segments can be classified into two broad categories. The approaches in the first category use line segments to form weighted graphs, which are then input to graph operations to determine segment matches. Such operations include shortest path estimation, the computation of graph isomorphism or the determination of maximal cliques. The major drawback of this type of techniques is that they are computationaly expensive. The second class of techniques use some similarity measure to rank local matches and then use a global iterative operation such as relaxation labeling to determine a global match. Such methods can be computationaly efficient, but lack guaranteed convergence. Thus, they can be trapped in undesirable local minima. A few representative approaches to line matching are briefly reviewed in the following.

In [5], Horaud and Skordas present a graph-based technique for line matching. A relational graph is built for each image, and then, for each segment in one image, a set of potential assignments in the other image is determined. Such assignments form nodes in a *correspondence graph*, whose arcs represent compatible assignments established on the basis of segment relationships. Stereo matching becomes equivalent to searching for sets of mutually compatible nodes in this graph. Such sets are found by identifying maximal cliques. Lee and Leou [6] propose a divide-and-conquer approach to line matching. Based on orientation, segments from each image are clustered into similar groups which are then matched using a dynamic programming search on a relational graph. Medioni and Nevatia [11] match straight line segments using a criterion they call ''minimal differential disparity'', which depends upon similarity of segment contrast and orientation together with a property of segment intersection under translation within a predefined disparity range. Selection of unambiguous matches is achieved through a form of relaxation that takes account of neighboring matches. McIntosh and Mutch [10] use attributes to describe the region from which lines are extracted. A match function determines the strength of the similarity of a line in one image with every line in the othen image. Matching pairs are then formed by lines that are mutually the best match in both directions. Liu and Huang [8] use a match function based on geometric as well as photometric properties of lines to characterize the similarity of a candidate pair. To reduce the cost of computation, a two-step matching process is applied. At the first step, correspondence is established between two small sets of long lines. These *kernel matches* are taken as references in the second step for matching the remaining lines. Li [7] uses the Hough transform to show that the line matching problem in image space can be converted into a point matching problem in the Hough space. A dynamic programming technique that combines multiple constraints is used to search for the optimal matching. Lloyd et al [9] match intensity edges extracted from a binocular pair having horizontal epipolar lines. They employ dynamic programming to produce candidate matches for each row which are then passed to a relaxation labeling process that selects matches based on an edge continuity constraint.

The present work employs a matching function for measuring the similarity of local matches, combined with a relaxation labeling technique that disambiguates local matches by imposing global constraints. In the remainder of this paper, Section 2 discusses line extraction, Section 3 presents local line matching and Section 4 details the relaxation technique used to resolve local matches. Experimental results from the application of the method on real images are presented in Section 5. The paper is concluded in Section 6.

2 Line Extraction

The proposed method starts by extracting a set of straight line segments from each of the input images. First, edge pixels are identified in each image using the Canny edge detector [2]. Edge pixels are then grouped into chains through an edge following procedure based on neighborhood relations. Following this, edge chains are segmented at high curvature locations. Finally, line segments are identified by fitting straight lines to the segmented edge chains as follows. Recall that points $\mathbf{p} = (x, y, 1)^t$ [1] lying on a line satisfy the equation

$$\mathbf{p}^t \cdot \mathbf{l} = 0 \quad , \tag{1}$$

where $\cdot$ denotes the dot product and $\mathbf{l}$ is the 3×1 vector of the line parameters. Considering the N points belonging to an edge segment, the N constraints on the line parameters that arise from Eq. (1) can be written more compactly as the following linear homogeneous equation

$$\mathbf{M}\,\mathbf{l} \;=\; \mathbf{0} \;,$$

where M is a $N \times 3$ matrix. The line parameters $\mathbf{l}$ are then estimated from the solution of the following minimization problem

$$min_{\mathbf{l}} ||\mathbf{M}\,\mathbf{l}||^2 \quad subject\ to \quad ||\mathbf{l}|| = 1 \quad , \tag{2}$$

where $||\ ||$ denotes the vector 2-norm. The solution to this constrained minimization problem is known to be the eigenvector of the matrix $\mathbf{M}^t\mathbf{M}$ that corresponds to the smallest eigenvalue [4].

An undesirable effect that might be observed after line extraction is that straight edges can be broken in several segments. This obviously complicates matching, requiring a linking operation if line breaking is too often. Line segment linking can be based on a criterion combining similarity of orientation and proximity of segment endpoints. However, in the present work, no attempt to link segments was made.

3 Line Matching

After extracting straight lines, the proposed method proceeds to the matching stage. A set of attributes is computed for each line segment and the matching score for a pair of lines is determined based on a combination of their attributes. The attributes considered for each line segment are the following:

- The orientation of the line segment
- The length of the line segment
- The average intensity along the line segment

[1] In the following, a^t will denote the transpose of the matrix or vector a.

The orientation of a line segment is defined to be the average gradient direction along the line segment, measured in the range $(-\pi, \pi]$. This definition of orientation is sensitive to whether the line segment is located on the boundary between a dark and a light image region or vice versa. The length of a line segment is defined to be the euclidean distance between its endpoints. The average intensity of a line segment is defined to be the average image intensity within a narrow stripe centered along the line segment. The matching score between a pair of lines is a number in the range $[0, 1]$, with large values implying closer matches. The matching score is computed by combining corresponding line segment attributes as follows:

- If the absolute difference in segment orientations is larger than 20^o, the matching score is zero.
- If the absolute difference in the lengths of the two lines is more than 50% of the length of the shorter line, the matching score is again considered to be zero.
- If none of the above two conditions is satisfied, corresponding line attributes are first normalized by dividing their absolute difference by the largest attribute. Thus, the orientation similarity O_{ij} for two segments l_i, l_j is defined to be equal to

$$O_{ij} \quad = \quad \frac{|orientation_i - orientation_j|}{|\max(orientation_i, orientation_j)|},$$

  where $orientation_i, orientation_j$ are the orientations of line segments i, j respectively. Similarly, the length similarity L_{ij} and intensity similarity I_{ij} are defined to be

$$L_{ij} = \frac{|length_i - length_j|}{\max(length_i, length_j)} \qquad I_{ij} = \frac{|intensity_i - intensity_j|}{\max(intensity_i, intensity_j)}$$

  O_{ij}, L_{ij} and I_{ij} take values from $[0, 1]$, with small values implying closer matches. The normalized attributes are then combined to yield the matching score using

$$score(l_i, l_j) = 1.0 - \max(O_{ij}, L_{ij}, I_{ij}).$$

  The maximum operator in the preceding expression selects the most dissimilar of the three attributes and defines the matching score based on it. This means that the resulting operator is conjuctive (i.e. corresponding to a logical AND operation), since all three attributes must be similar for a pair score to be high [1]. Matches with a score less than a predefined threshold (0.6 in our implementation), are eliminated from further consideration. On the other hand, matches having scores larger or equal to that threshold form *candidate matches.*

Using the above definition of a matching score, a line in one image can have several candidate matching lines in the second image. To work out a globally consistent match, pair matches are disambiguated using a relaxation labeling technique that employs neighborhood relationships to impose the uniqueness and continuity constraints. This technique is based on the scheme used in [17] to disambiguate corner matches obtained by correlation.

4 Relaxation Labeling

Let (l_i, l_j) be a candidate match, where l_i is a line in the first image and l_j a line in the second. The set of neighbors of l_i, $N(l_i)$, is defined as the set of line segments in the first image whose midpoints are at most $\mathcal{R}$ pixels away from the midpoint of l_i. $N(l_j)$, the set of neighbors of l_j in the second image is defined similarly. If (l_i, l_j) is a good match, then many matches (n_k, n_m), where $n_k \in N(l_i), n_m \in N(l_j)$, are expected to be present. Moreover, the position of n_k relative to l_i should be similar to the position of n_m relative to l_j. This observation is captured by the notion of the *strength of a match (SM)* which is defined as

$$SM(l_i, l_j) = score(l_i, l_j) \sum_{n_k \in N(l_i)} \max_{n_m \in N(l_j)} \frac{score(n_k, n_m)\delta(l_i, l_j, n_k, n_m)}{D(l_i, l_j, n_k, n_m) + 1}, \tag{3}$$

where $D(l_i, l_j, n_k, n_m) = (d(l_i, n_k) + d(l_j, n_m))/2$ is the average euclidean distance between the midpoints of neighboring lines and $\delta(l_i, l_j, n_k, n_m)$ is

$$\delta(l_i, l_j, n_k, n_m) = \begin{cases} e^{-\frac{r}{\epsilon}} & \text{if } r < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where r is the relative distance difference defined by

$$r = \frac{|d(l_i, n_k) - d(l_j, n_m)|}{D(l_i, l_j, n_k, n_m) + 1} \tag{5}$$

and ϵ is a threshold on r. This definition is similar to that used in the PMF stereo algorithm [13].

Intuitively, the above definition of a strength of match uses neighboring lines to gather evidence regarding the similarity of the line pair under consideration. The contribution of each line pair formed by neighboring lines depends on the relative distance difference, in a way that close pairs contribute more than pairs being further away. To account for the fact that a line in one image can have more than one candidate matches in the other, the max operator within the sum in Eq.(3) ensures that only the line resulting in the largest contribution is considered.

The definition in Eq.(3) has the undesirable drawback of not being symmetric. This means that if the roles of a pair of images are reversed, $SM(l_i, l_j) \neq SM(l_j, l_i)$. This is due to the fact that more than one lines $n_k \in N(l_i)$ may score their maximum value in Eq.(3) with the same line $n_m \in N(l_j)$ To remedy this, only the line that gives the largest maximum value is taken into account. This results in the strength of match being symmetric for the two images. In our implementation $\mathcal{R}$ is 0.16 of the image width and $\epsilon = 0.4$.

Summing the SMs defined by Eq.(3) for every line pair, an energy function can be defined as

$$E = \sum_{l_i, l_j} SM(l_i, l_j)$$

Then, the problem of disambiguating matches is equivalent to minimizing the above energy function. Relaxation labeling is one possible approach for minimizing the total energy. It is an iterative procedure that reorganizes matches until the total energy converges to a minimum. Various schemes for updating total energy have been proposed. One is the *winners-take-all* which works as follows: At each iteration, the matches that have the highest matching strengths for both lines that form them are considered to be correct. Then, as dictated by the uniqueness constraint, any other pairings that are associated with with the selected matches are eliminated from further consideration. This permits other line pairs to be considered as correct in the next iteration, provided that the lines forming them are mutually best matches. The winners-take-all scheme is fast, but can be caught in a local minimum. An alternative approach for minimizing the total energy is the so-called *looser-takes-nothing*. According to it, at each iteration the match that has received the smallest matching strength is eliminated. Progressively, this scheme ends up with each line having only one matching line, which is assumed to be the best. However, the looser-takes-nothing scheme is too slow, since it requires one iteration for each candidate match that is eliminated. In [17], Zhang et al propose a scheme called *some-winners-take-all*. It's main difference compared to the winners-take-all scheme is that in addition to the matching strengths, it also takes into consideration the ambiguity of each candidate match. More specifically, the line pairs that have the largest strength for both lines that form them are selected at each iteration. Such line pairs are called *potential* matches. Then, potential matches are sorted in decreasing order according to their strengths of match, forming a list L_{SM}. If (i, j) and (r, s) are two potential matches such that (r, s) follows (i, j) in decreasing order of the sorted strengths of match, the ratio

$$U = \frac{SM_{rs} - SM_{ij}}{SM_{rs}}$$

is a measure of the unambiguity of the match (i, j). Potential matches are also sorted in decreasing order according to their unambiguity, forming a list L_U. Then, the some-winners-take-all scheme proceeds by selecting as correct matches all potential matches that are among the first p percent of both L_{SM} and L_U. Thus, ambiguous matches are not selected even if they have high strengths of match, and matches having low strengths of match are not selected even if they are unambiguous. In the case that $p = 100$, the some-winners-take-all scheme reduces to the winners-take-all. To avoid premature termination, the percent p used in some-winners-take-all must be greater than 50, so that in case that there exist more than one potential matches, at least one match is selected at each iteration. In our implementation, $p = 60$. Some-winners-take-all is guaranteed to converge, since at least one potential match is selected at each iteration, thus reducing the total energy by its corresponding strength of match. The fact that some-winners-take-all uses bidirectional matching, i.e. only considers pairings that are mutually best matches, avoids many false matches that would result if the strength of match was computed in one direction only. The cost paid for having less false matches is that there is a possibility of missing true matches that do not have enough support in both directions.

5 Experimental Results

A set of experiments has been conducted in order to test the performance of the described method. Representative results from these experiments are given in this section.

Figure 1 shows a pair of images of an office that were used in one of the experiments [2]. The images have a resolution of 512 $\times$ 512 pixels. During this experiment, only lines longer than 30 pixels were considered for matching. After line extraction, line matching took less than 15 seconds on a Linux Pentium/133. Figure 2 shows the lines matched between the two images in Fig. 1. Matching lines are indicated with the same number in the two images of Fig. 2.

Another experiment regards the ''kiel'' motion sequence. Frames 0 and 10 of this sequence are shown in Fig. 3. Each frame of the ''kiel'' sequence is 720 $\times$ 576 pixels. The lines matched between these two frames are shown in Fig. 4. Only lines longer than 40 pixels were considered for matching. Since the ''kiel'' images contain many line

[2]Images courtesy of the INRIA RobotVis group

Figure 1: Two images of an office scene used to test the method

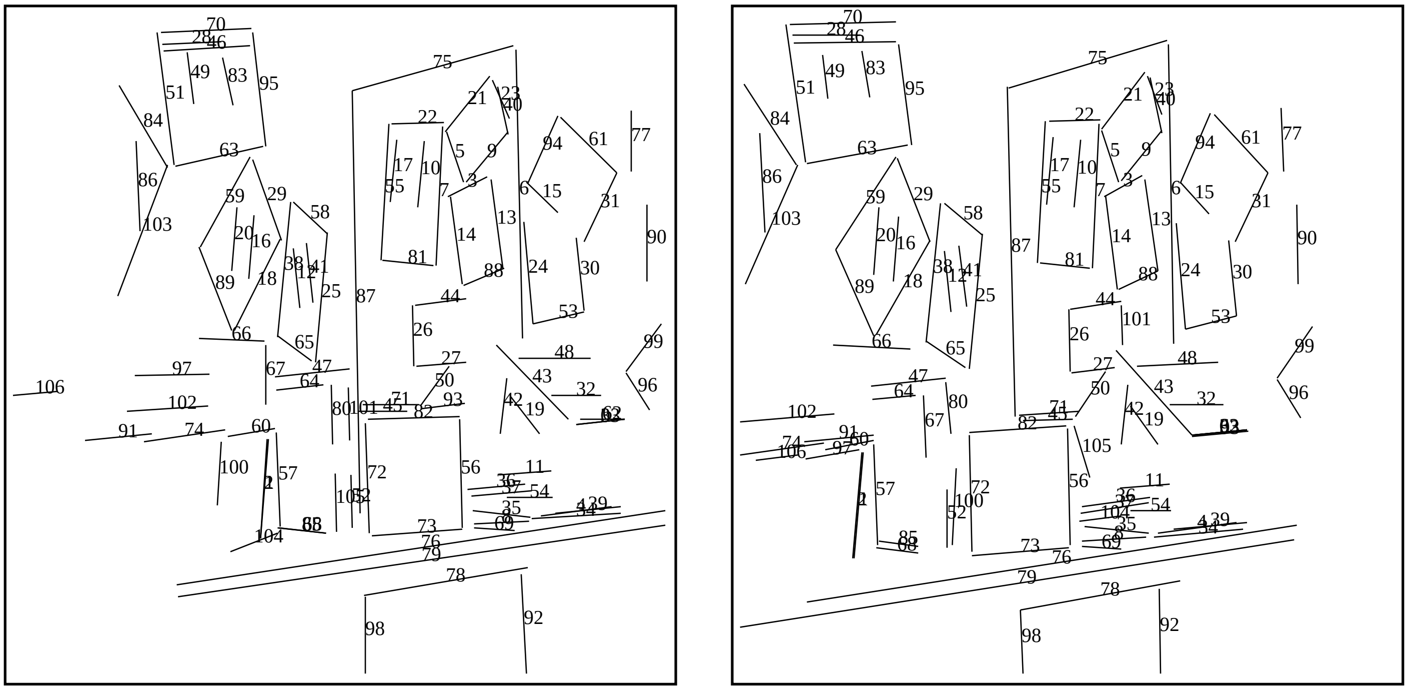

Figure 2: Matched line pairs for the office scene

Figure 3: Frames 0 and 10 of the ''kiel'' sequence

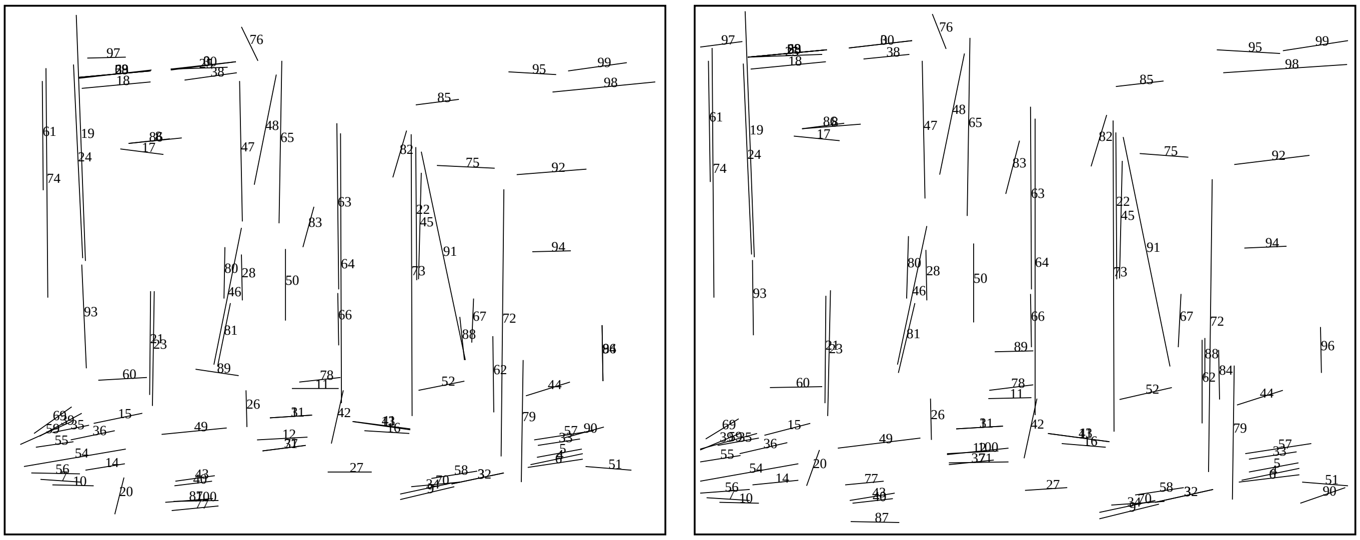

Figure 4: Matched line pairs for the ''kiel'' images

segments, matching lines between them is computationally expensive. For the two frames shown in Fig. 3, line matching needed approximately 5 minutes on a Linux Pentium/133.

6 Conclusions

This paper presented a method for corresponding line segments extracted from a pair of images. The set of corresponding segments produced by the method can be used by various vision algorithms dealing with problems such as motion analysis, stereo and object recognition. As it can be seen from the experimental results presented in the preceding section, the proposed method is quite successful in pairing line segments. However, as experiments with many different images indicated, it is very likely that the method will produce a few mismatches. This observation implies that any algorithm that makes use of the matched lines must expect and be robust to the existence of a few false matches.

References

[1] I. Bloch. Information Combination Operators for Data Fusion: A Comparative Review with Classification. *IEEE Transactions on SMC*, SMC-26(1):52--67, January 1996.

[2] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on PAMI*, PAMI-8:679--697, 1986.

[3] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant Descriptors for 3-D Object Recognition and Pose. *IEEE Transactions on PAMI*, PAMI-13:971--991, 1991.

[4] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 3rd edition, 1996.

[5] R. Horaud and T. Skordas. Stereo Correspondence Through Feature Grouping and Maximal Cliques. *IEEE Transactions on PAMI*, PAMI-11(11):1168--1180, 1989.

[6] S.H. Lee and J.J. Leou. A Dynamic Programming Approach to Line Segment Matching in Stereo Vision. *Pattern Recognition*, 27(8):961--986, 1994.

[7] Z.N. Li. Stereo Correspondence Based on Line Matching in Hough Space Using Dynamic Programming. *IEEE Transactions on SMC*, SMC-24(1):144--152, 1994.

[8] Y. Liu and T.S. Huang. Determining Straight Line Correspondences from Intensity Images. *Pattern Recognition*, 24(6):489--504, 1991.

[9] S.A. Lloyd, E.R. Haddow, and J.F. Boyce. A Parallel Binocular Stereo Algorithm Utilizing Dynamic Programming and Relaxation Labeling. *Computer Vision, Graphics and Image Processing*, 39:202--225, 1987.

[10] J.H. McIntosh and K.M. Mutch. Matching straight lines. *Computer Vision, Graphics, and Image Processing*, 43(3):386--408, September 1988.

[11] C. Medioni and R. Nevatia. Segment-Based Stereo Matching. *Computer Vision, Graphics and Image Processing*, 31:2--18, 1985.

[12] J.L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, MA, 1992.

[13] S.H. Pollard, J.E.W. Mayhew, and J.P. Frisby. PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit. *Perception*, 14:449--470, 1985.

[14] A. Shashua. Algebraic Functions for Recognition. *IEEE Transactions on PAMI*, PAMI-17(8):779--789, 1995.

[15] P.H.S. Torr and A. Zisserman. Robust Parameterization and Computation of the Trifocal Tensor. In R. Fisher and E. Trucco, editors, *Proc. British Machine Vision Conference*, pages 655--664. BMVA, Sept 1996.

[16] I. Weiss. Geometric Invariants and Object Recognition. *International Journal of Computer Vision*, 10:207--231, 1993.

[17] Z. Zhang. A new and efficient iterative approach to image matching. In *Proceedings of ICPR'94*, pages 563--565, Jerusalem, Israel, 1994.