Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 1995-V06

Monocular Computation of Motion in a Semi-Structured Environment[1]

J.J. Guerrero, C. Sagüés

If you want to cite this report, please use the following reference instead:
Monocular Computation of Motion in a Semi-Structured Environment. J.J. Guerrero, C. Sagüés, *VI Spanish Symposium on Pattern Recognition and Image Analysis*, pages 590-597, April 1995.

[1] This work was partially supported by project TAP-94-0390 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).

Monocular Computation of Motion in a Semi-Structured Environment

J.J. Guerrero & C. Sagüés

Dpto. de Ingeniería Eléctrica e Informática

Centro Politécnico Superior, UNIVERSIDAD DE ZARAGOZA

María de Luna 3, E-50015 ZARAGOZA, SPAIN

Phone 34-76-517274, Fax 34-76-512932

email: jjguerrero@mcps.unizar.es

Abstract

An algorithm to determine camera motion when the structure of the environment is not known, is presented. It is based on straight segments that are plentiful features in man made environments. Contour lines are directly extracted from intensity images. The correspondence of segments are made by tracking them in the image, using a Kalman filter. To make more robust the motion computation, tips of the segments besides the infinite line are taken when they can be identified.

1 Introduction

A robot works with objects in a real, prone to uncertainty environment. To the end of reducing the engineering which supplies the objects in prefixed locations, the robot has to be provided with sensorial capabilities. Vision is the sensor which provides more information, but as in other sensors, its information is incomplete and noisy. When the camera moves, some advantages are introduced. So, motion and third dimension can be extracted; camera can be directed towards the best position of observation; observed features can be fixed in the image.

Methods for extracting shape and motion information from vision can be classified as optical flow-based, direct and correspondence-based. Related to the correspondence-based methods many approaches to recover structure and motion from points and/or lines have been widely studied and revised in the last years [Tsai 84], [Liu 88], [Spetsakis 92]. We have treated this problem by using lines with a tip [Guerrero 94]. Working with lines it is well known that three images, at least, are needed in order to determine both the camera motion and the 3D scene structure. In [Huang 94] a review of the motion and structure problem from correspondent features is presented. Normally, these methods allow larger relative motion from an image to the next than optical flow based or direct methods, but they add the correspondence determination problem.

Edges are fundamental object features. In computer vision image edges usually correspond to some properties of 3D objects such as boundaries. In object recognition and motion analysis, edges of intensity images may be used as key features. Straight edges have a simple mathematical representation and involve more information than other features like points. Besides that, lines are easier and more accurate to extract in a noisy image than points, and is easier to match lines than points. In semi-structured environments, straight edges usually appear and sometimes their tips are well identified.

Segment extraction is followed by the search of correspondences over a sequence of images to carry out more higher level processes (object recognition, motion analysis, structure perception). In many works it is assumed that correspondences between features extracted from one image and those extracted from the next image, are available. However establishing and maintaining the correspondence is not an easy job [Aggarwal 88].

In this paper we propose to work with image features like edge segments with the image bright regions which supports them. The correspondence search is based in tracking in the image segments with some bright parameters. With correspondent lines in the images, a method to obtain camera motion is presented. Besides that, points over them (their tips) are considered in order to obtain a more robust determination of motion.

The organization of the paper is as follows: The first section presents the extractor of lines in the image. Section §3 gives a general overview of the tracking solution adopted to solve the correspondence problem. In §4 the method to determine the camera motion from lines using also their tips is explained. In §5 some experiments are shown. Finally §6 is devoted to expose some conclusions.

2 Extraction of segments

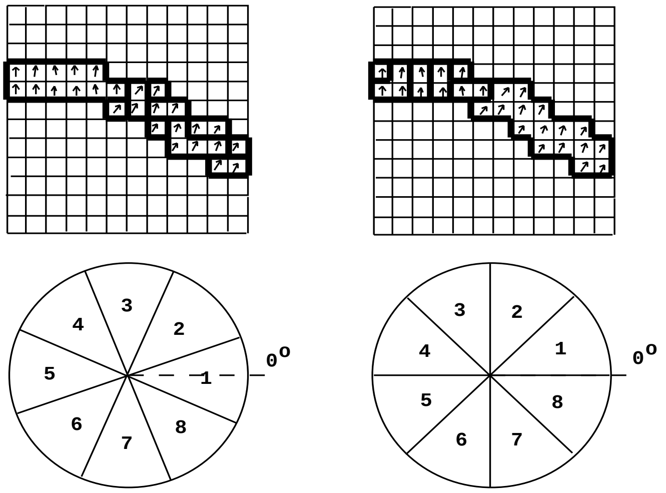

Figure 1: Segmentation in LSR

Extraction of edge lines from intensity images is a basic and powerful process to select the information of the image. Two main kinds of methods for straight segments detection can be used:

- The first one gives segments after an edge detector has been applied to the image. The most popular edge extractor is the proposed in [Canny 86]. The link of edge points into a line or a segment can be made with Hough transform or other linking methods [Giraudon 87].
- Lines can directly be extracted from intensity images. The work of [Burns 86] is perhaps the most important on extracting straight edges with this kind of methods.

We obtain straight segments as proposed by [Burns 86]. The first step in the procedure is the extraction of spatial gradients. Pixels are grouped into regions of similar direction of brightness gradient, with the gradient magnitude larger than a threshold. Two overlapping sets of partitions, with a post grouping process, are used in order to avoid the problems related with the arbitrary boundary of fixed partitions (see figure 1). Thus, we have line-support regions (LSR) containing all information available of the straight contours.

To obtain the line in the image, a planar brightness surface is fitted to the LSR by a least-squares approach, predicting the brightness (E) as a function of the image coordinates. In this fitting, a weighting norm $N_w(x,y)$ proportional to the gradient magnitude is considered. The measure minimized along the LSR in function of the parameters of the brightness surface (A,B,C) is expressed as:

$$\sum_{x}^{LSR} \sum_{y}^{LSR} [A\,x + B\,y + C - E]^2\; N_w(x,y)$$

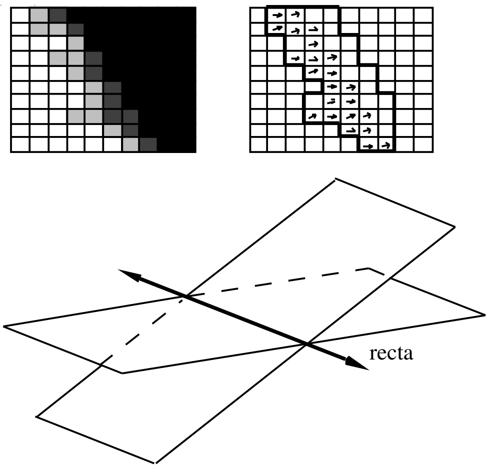

Figure 2: Line extraction in the image

The straight line is obtained, with subpixel accuracy, as the intersection of this brightness plane and the horizontal plane of mean brightness E_m in the LSR (weighted with the gradient magnitude)(see figure 2).

$$E_m \; = \frac{\sum_x^{LSR}\sum_y^{LSR} E \; N_w(x,y)}{\sum_x^{LSR}\sum_y^{LSR} \; N_w(x,y)}$$

The line-support region provides an excellent chance to extract not only geometrical information of the edge, but also some attributes related to the bright of the contour that will be used in the matching. This parameters are the average gray level, contrast, steepness and deviation from straightness that provide useful information to identify and classify the segments.

3 Matching of segments

The problem we want to address here is to determine correspondences between linear segments in a monocular sequence of images. Nothing about the camera motion is assumed except we have an homogeneous sequence of images. We have treated the correspondence problem by tracking segments in the image plane, based on a prediction-matching-update loop, using a Kalman filter. The prediction step provides reasonable estimates of the region and range of attributes where the match can be determined (see figure 3).

A nearest neighbor tracking approach similar to [Deriche 90] has been developed. However we use in the tracking and matching process, besides the classical location values, two image bright attributes of the segment [Guerrero 95]. These attributes are the average grey level (agl) and the mean contrast (c). They allow to match segments using not only the geometrical information, but also the intensity information that is, in many cases, more relevant and selective. The matching using these bright attributes can be made nearly in parallel to the geometrical matching with a little computational overhead. Besides that, the bright parameters are crucial to match segments when neither the structure nor the camera motion are known, because geometrical constraints are only valid locally and they can not be imposed in a non heuristic way. On the other hand the bright conservation considered in the tracker is only locally imposed because their variation and state noise are considered in the kinematical model, as we will see below. This makes the bright constraint more reliable.

Besides these bright attributes, the classical location image parameters of the segment are considered. We choose the midpoint representation which takes three location parameters and the length of the image segment. Thus the segment representation in the image is composed of 6 parameters: x_m (X coordinate of the midpoint), y_m (Y coordinate of the midpoint), θ (segment orientation), l (segment length), agl (average grey level in the LSR) and c (contrast of the contour).

All of them could be considered uncorrelated except the coordinates of the midpoint. Therefore, as it is proposed in [Deriche 90], a separate Kalman filter with a reduced dimension can be used if we want to improve the efficiency.

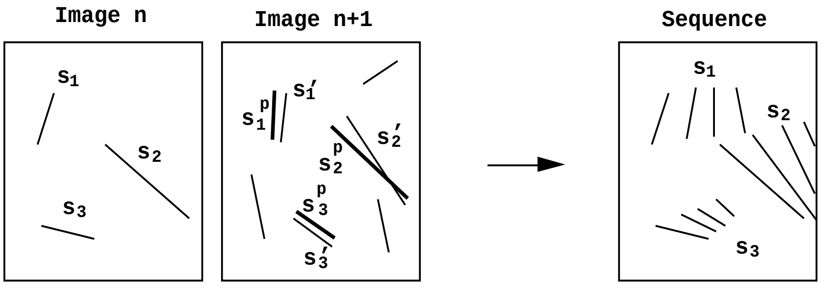

Figure 3: Matching in the sequence with segment tracking in the image (prediction S^p, observation S')

3.1 The System Model

It is well known that the Kalman Filter equations are computed directly from the linear space state model of a stochastic system, as:

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{w}(k) \qquad (1)\\
\mathbf{z}(k) &= \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \qquad (2)\\
\mathbf{E}\left[\mathbf{w}(k)\mathbf{w}(j)'\right] &= \mathbf{Q}(k)\delta_{kj} \qquad (3)\\
\mathbf{E}\left[\mathbf{v}(k)\mathbf{v}(j)'\right] &= \mathbf{R}(k)\delta_{kj} \qquad (4)
\end{aligned}$$

We have selected the constant velocity kinematic model proposed in [Bar-Shalom 88]. This model can be considered valid only locally. So, the state vector of the system is:

$$\mathbf{x} = \left(x_m, \dot{x}_m, y_m, \dot{y}_m, \theta, \dot{\theta}, l, \dot{l}, \mathrm{agl}, \dot{\mathrm{agl}}, c, \dot{c},\right)^T$$

Equations (1) and (3) represent the evolution of each segment. We assume that every segment parameter is decoupled from the others. For each pair parameter-derivative we define its dynamic equation by means of:

$$\begin{aligned}
\mathbf{F}_i &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}\\
\mathbf{Q}_i &= \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} q_i
\end{aligned}$$

Where Δt is taken to be the unit and i stands for each parameter. To sum up, the dynamic equations are completely defined except for one covariance value per each parameter: $q_{x_m}, q_{y_m}, q_\theta, q_l, q_{agl}, q_c$, that are derived from experimental results.

Equations (2) and (4) model how the camera detects the segment represented by the state. The matrix $\mathbf{H}$ selects the values corresponding to the segment parameter, the derivatives are not measured. It is a block diagonal matrix composed of six 1×2 blocks like:

$$\mathbf{H}_i = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The noise $\mathbf{v}(k)$ represents the measurement perturbation. In contrast to the state covariance, there is some correlation between the different segment parameters:

- The measurement noise along the segment direction is bigger than the noise in the orthogonal direction. The noises in these directions are represented by $\sigma_\parallel^2$ and $\sigma_\perp^2$ respectively. This model was proposed in [Deriche 90].

- The measurement noise corresponding to agl, c are independent.

Therefore the matrix $\mathbf{R}$ is defined as:

$$\begin{bmatrix} \sigma_{\|}^2\mathrm{C}^2 + \sigma_{\perp}^2\mathrm{S}^2 & \sigma_{\|}^2\mathrm{CS} - \sigma_{\perp}^2\mathrm{CS} & 0 & 0 & 0 & 0 \\ \sigma_{\|}^2\mathrm{CS} - \sigma_{\perp}^2\mathrm{CS} & \sigma_{\perp}^2\mathrm{C}^2 + \sigma_{\|}^2\mathrm{S}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\frac{\sigma_{\perp}^2}{l^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\sigma_{\|}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{agl}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_c^2 \end{bmatrix}$$

where $C = \cos\theta$ and $S = \sin\theta$.

So the measurement model is a function of the state parameter θ and is completely defined except for $\sigma_{\|}^2, \sigma_{\perp}^2, \sigma_{agl}^2, \sigma_c^2$ that are obtained experimentally.

3.2 Matching

The proposed matching technique can be seen as a Nearest Neighbor Standard Filter. Thus, the segment closest to the prediction using the Mahalanobis distance is considered a valid match. The Mahalanobis distance is computed as:

$$\mathbf{r}^T\mathbf{S}^{-1}\mathbf{r} \tag{5}$$

where $\mathbf{S}(k+1)$ is the covariance of the innovation $\mathbf{r}(k+1) = \mathbf{H}\hat{\mathbf{x}}(k+1|k) - \mathbf{z}(k+1)$

Other matching strategies that combine location parameters and bright parameters in a more efficient way are being tested in order to improve the match.

4 Motion from unknown lines

Methods to recover structure and motion from lines and/or points have been widely studied and revised in the last years [Huang 94]. We propose to solve the problem using lines and its tips when they can be identified. The sole use of lines involves three frames because as known, straight edges from two views do not provide motion information.

We adopt a pinhole camera model with a planar screen where the Z axis is aligned with the optical axis and the focal length is considered to be the unit. A point in the scene with $\mathbf{P} = (X, Y, Z)$ coordinates in the camera reference system is projected in the image with $\mathbf{p}' = (x, y, 1)$ coordinates, being:

$$x = \frac{X}{Z} \quad , \qquad y = \frac{Y}{Z} \tag{6}$$

Let us suppose the camera to move with respect to the scene and its motion to be composed by a translation ($\mathbf{t}$) and a rotation ($\mathbf{R}$). Attaching the main reference system to the first camera frame, we will use the following notation:

- $\mathbf{R}_{12}$, $\mathbf{R}_{13}$ and $\mathbf{t}_{12}$, $\mathbf{t}_{13}$ are the camera rotations and translations from the first to the second and from the first to the third camera locations, respectively.
- $\mathbf{n}_1^l$, $\mathbf{n}_2^l$ and $\mathbf{n}_3^l$ are the normal vectors of the projecting planes of the l-th line in the three camera references.
- $\mathbf{p}_1^l$, $\mathbf{p}_2^l$ and $\mathbf{p}_3^l$ are the normalized image vectors corresponding to a tip of the l-th line in the three camera references $\mathbf{p} = \frac{\mathbf{p}'}{||\mathbf{p}'||}$.

It is known that, the normals to the successive projecting planes of a straight edge when the camera moves are coplanar, because all of them are perpendicular to the edge. The successive normals can be expressed in other reference system by means of the rotation matrix which expresses the camera motion. If the camera rotates according to $\mathbf{R}_{12}$, the normal to the projecting plane in the second image ($\mathbf{n}_2$) can be expressed in the first reference system as $\mathbf{R}_{12}\ \mathbf{n}_2$. Similarly with the third image. Therefore, the normal vectors to the three projection planes corresponding to the l-th edge, in the first reference system are coplanar and therefore their vector triple product will be equal to zero [Liu 88].

$$\mathbf{n}_1^l \cdot (\mathbf{R}_{12}\ \mathbf{n}_2^l \times \mathbf{R}_{13}\ \mathbf{n}_3^l) = \ 0 \tag{7}$$

This equation is nonlinear with the unknowns $\mathbf{R}_{12}$ and $\mathbf{R}_{13}$ and can be solved by iterative methods when an initial guess is available. Since there are three unknowns in each rotation matrix, six or more line correspondences over three frames are needed to solve them. If we have identified some tip of the edges, we can construct some line by joining the tips of two lines.

Once the rotations are determined, the translations can be obtained by linear methods:

$$\mathbf{t}_{12} \cdot (\mathbf{R}_{12}\ \mathbf{n}_2^l) \ = \ \frac{||\mathbf{n}_1^l \times \mathbf{R}_{12}\ \mathbf{n}_2^l||}{||\mathbf{n}_1^l \times \mathbf{R}_{13}\ \mathbf{n}_3^l||}\ \mathbf{t}_{13} \cdot (\mathbf{R}_{13}\ \mathbf{n}_3^l) \tag{8}$$

We can also use the information of the tips of the edges when they have been identified. From each correspondent point in two frames we have one motion constraint. The image vectors of the tip of the edge meet in the space, and therefore these vectors and the translation vector, from one to other camera position, are coplanar [Faugeras 87]. This can be expressed for the first and second images and for the first and third images as :

$$\mathbf{t}_{12} \cdot (\mathbf{p}_1^l \times \mathbf{R}_{12}\mathbf{p}_2^l) = 0\ ; \quad \mathbf{t}_{13} \cdot (\mathbf{p}_1^l \times \mathbf{R}_{13}\mathbf{p}_3^l) = 0 \tag{9}$$

Using three frames, one third constraint can be taken. So, the three image vectors must meet in the space [Spetsakis 92]. This is expressed using the distance from the point to the origin of the first reference system. As a function of the motion from the first to the second frame, we can obtain this distance, and equating to the distance obtained from the first and third frame, we have:

$$d_1^l = \frac{(\mathbf{t}_{12} \times \mathbf{R}_{12}\mathbf{p}_2^l) \cdot (\mathbf{p}_1^l \times \mathbf{R}_{12}\mathbf{p}_2^l)}{||\mathbf{p}_1^l \times \mathbf{R}_{12}\mathbf{p}_2^l||^2} = \frac{(\mathbf{t}_{13} \times \mathbf{R}_{13}\mathbf{p}_3^l) \cdot (\mathbf{p}_1^l \times \mathbf{R}_{13}\mathbf{p}_3^l)}{||\mathbf{p}_1^l \times \mathbf{R}_{13}\mathbf{p}_3^l||^2} \tag{10}$$

The translations are solved with a scale factor. This scale factor is inherent to the use of one camera in motion. If there are more features than the minimum required a least-squares method allows to solve the problem more robustly.

5 Experiments

At the moment some experiments using two images have been made. In figure 4 the first image of the scene can be observed. The second image has been obtained making aproximately an horizontal translation of 20 cm. The straight segments extracted having a gradient less than 15 gray levels per pixel and a length less than 70 pixels are removed in order to have few but good segments (figure 5). From the 16 segments extracted in the first image, it turns out that only 11 have been matched in both images (some of those segments are thrown away because of its deviation of straightness is too large and some of them disappear in the second image).

The segments matched are used to compute the motion. When the structure is not known the translations and rotations are coupled in such a way that the solution is instable. In the proposed method a quite good guess of the motion is required to make the algorithm to converge. In the figure 6 a reconstruction with the computed motion of the scene is shown. Most of the lines are reasonably well reconstructed. The lines on the wall and the line on the laboratory table are nicely computed. The problems with the lines of the small table are due to its parallelism with the translation of the camera and also to that the tips are not well identified.

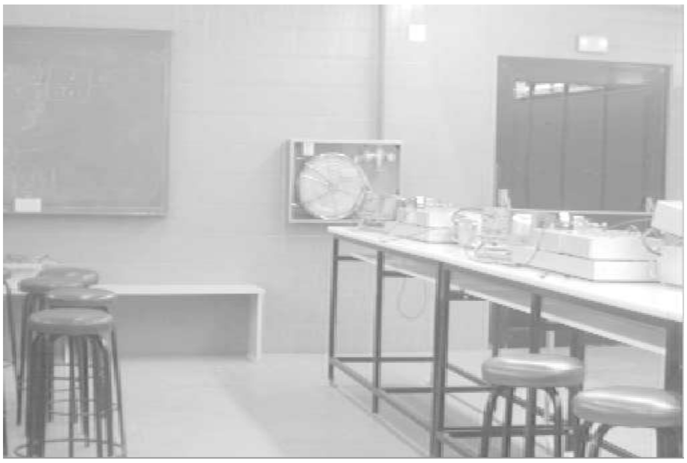

Figure 4: First image used in the experiment

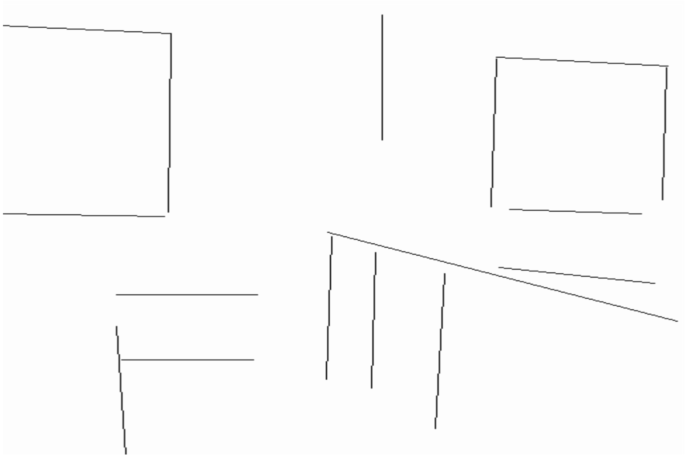

Figure 5: Segments extracted in the first image

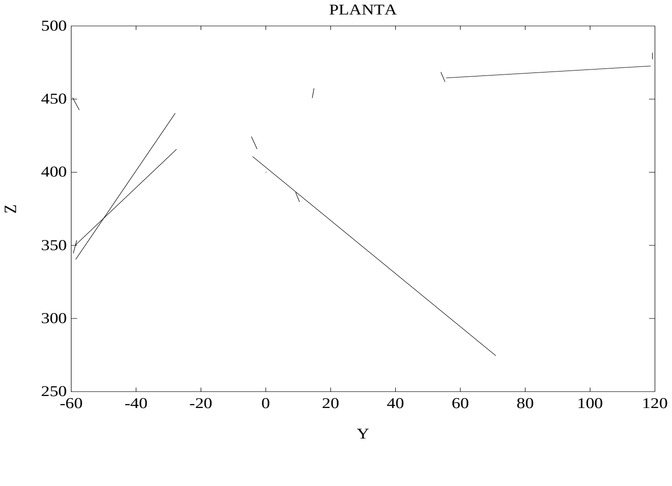

Figure 6: Reconstruction of lines in the reference system of the first camera. Top view

6 Conclusions

We have presented a method to obtain the camera motion in a semi-structured environment where straight contours can be obtained. The image lines are extracted using an approach that uses directly the image intensity. Segments are tracked in the image plane, with a Kalman filter to determine the correspondences. The motion is obtained from lines or its tips when they are identified.

A quite good guess of the motion is needed to obtain a coherent solution, because of the instability of the motion estimation when the structure is unknown.

Acknowledgments

This work was partially supported by project TAP-94-0390 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).

References

[Aggarwal 88] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. *Proceedings of the IEEE*, 76(8):917–935, 1988.

[Bar-Shalom 88] T. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press In., 1988.

[Burns 86] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.

[Canny 86] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[Deriche 90] R. Deriche and O. Faugeras. Tracking line segments. In *First European Conference on Computer Vision*, pages 259–268, Antibes, France, 1990.

[Faugeras 87] O. Faugeras, F. Lustman, and G. Toscani. Motion and strucure from motion from point and line matches. In *Proc. ICCV*, London, Jun. 1987.

[Giraudon 87] G. Giraudon. Chainage efficace de contour. Rapport de recherche RR-605, I.N.R.I.A., Sophia-Antipolis, France, 1987.

[Guerrero 94] J.J. Guerrero, C. Sagüés, and A. Lecha. Motion and structure from straight edges with tip. In *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, USA, Oct 1994.

[Guerrero 95] J.J. Guerrero and J.M. Martínez. Determination of corresponding segments by tracking both geometrical and brightness information. In *International Conference on Advanced Robotics*, Barcelona, September 1995. Submitted.

[Huang 94] T.S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.

[Liu 88] Y. Liu and T.S.Huang. Estimation of rigid body motion using straight line correspondences. *Computer Vision, Graphics And Image Processing*, (43):37–52, 1988.

[Spetsakis 92] Minas E. Spetsakis. A linear algorithm for point and line-based structure from motion. *CVGIP: Image Understanding*, 56(2):230–241, 1992.

[Tsai 84] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion paameters of rigid objects with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):13–27, 1984.