3D MOTION RECOVERY WHILE ZOOMING USING ACTIVE CONTOURS

Elisa Martínez Marroquín
Enginyeria La Salle
Universitat Ramon Llull
Pge. Bonanova, 8
08022 Barcelona. Spain
elisa@salleurl.edu

Carme Torras Genís
Institut de Robòtica i Informàtica Industrial
CSIC-UPC
Llorens i Artigas, 4-6
08028 Barcelona. Spain
ctorras@iri.upc.es

Abstract. This paper considers the problem of 3D motion recovery from a sequence of monocular images while zooming. Unlike the common trend based on point matches, the proposed method relies on the deformation of an active contour fitted to a reference object. We derive the relation between the contour deformation and the 3D motion components, assuming time-varying focal length and principal point. This relation allows us to present a method to extract the rotation matrix and the scaled translation along the optical axis.

1 Introduction

The ability to zoom provides an image definition that eases a range of visual tasks common in robot vision, such as structure recovery or recognition. However, camera zooming invalidates most of the current solutions to computer vision problems (e.g., tracking or calibration), which assume constant intrinsic camera parameters, and therefore demands new approaches [6, 5]. Zooming does not only change the focal length but also the principal point, due to optical and mechanical misalignments in the lens system of the camera [4, 17]. The rest of intrinsic camera parameters (e.g., pixel size and aspect ratio) remain constant for long periods of time [16] and may be assumed known.

The process of calibration with the aid of a calibration pattern [18, 19] is inapplicable in real time or in cases where the camera optical parameters undergo frequent changes. Different approaches have recently emerged for autocalibrating the camera assuming time-varying internal parameters [17, 14, 1]. They are based only on point matches. The present work is based on an active contour and aims to recover the 3D motion parameters.

It is known that the 3D structure and motion can be recovered from a sequence of images [7, 13]. This requires a measure of the visual motion on the image plane and a model that relates this motion to the real 3D motion. The bottleneck when trying to bring this into practice is the computation of visual motion, which requires at least a set of feature matches between frames. Moreover, common methods for feature matching perform particularly poorly when zooming. Noting that the cumulative research on active contours [3, 8, 2] provides an efficient tracking of objects, this work has been motivated by the idea of building an algorithm for 3D motion recovery upon an active contour tracker.

Previous works by the authors highlight the feasibility of recovering 3D structure and motion from the analysis of an active contour fitted to a reference object. This is shown for different degrees of camera calibration [11, 10] and for uncalibrated cameras with constant intrinsic parameters [12, 9]. Here we extend the analysis to the case of time-varying internal calibration parameters due to zooming.

The paper is organized as follows. Section 2 relates the deformation of a contour to the 3D motion components and the internal calibration parameters. Then, Section 3 describes the process followed to recover the 3D motion components. Section 4 shows two examples of the experiments conducted to test the method. Finally, we draw some conclusions in Section 5.

2 Projection of 3D contour on the image plane

An active contour is fitted to the contour $\mathbf{D}(s)$ of a reference object (i.e. the target), which is marked on-line by the operator and may have any shape. It is automatically tracked along the sequence, and its corresponding shape vector is updated at each frame [12]. The shape vector provides a direct measure of image deformation that, as we will show, permits deriving the 3D relative motion between the camera and the target.

Keeping the attention on the small region used as reference, allows one to assume a simplified camera model for the tracked region, no matter if this model does not fit the rest of the image. Using a weak-perspective camera model, the projection $\mathbf{d_0}(s)$ (hereafter, the template) of the 3D contour $\mathbf{D}(s)$ in the initial frame is

$$\mathbf{d_0}(s) = \frac{f^{(0)}}{Z_0} \begin{bmatrix} K_u & 0 \\ 0 & K_v \end{bmatrix} \begin{bmatrix} X_0(s) \\ Y_0(s) \end{bmatrix} + \begin{bmatrix} u_0^{(0)} \\ v_0^{(0)} \end{bmatrix}, \tag{1}$$

where $f^{(i)}, u_0^{(i)}, v_0^{(i)}$ are the focal length and principal point for frame i, K_u, K_v denote the pixel size and Z_0 is the distance from the camera to the target at the reference frame.

Assuming rigid motion between frames, the projection of the 3D curve in frame i is

$$\begin{aligned} \mathbf{d}(s) = & \frac{f^{(i)}}{Z_0 + T_z} \begin{bmatrix} K_u & 0 \\ 0 & K_v \end{bmatrix} \left(\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} X_0(s) \\ Y_0(s) \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \right) + \\ & + \begin{bmatrix} u_0^{(i)} \\ v_0^{(i)} \end{bmatrix}, \end{aligned} \tag{2}$$

where R_{ij} are the elements of the rotation matrix and T_i are the elements of the translation vector. Combining equations (1) and (2),

$$\begin{aligned} \mathbf{d}(s) - & \begin{bmatrix} u_0^{(i)} \\ v_0^{(i)} \end{bmatrix} = \\ & = \frac{f^{(i)}}{f^{(0)}} \frac{Z_0}{Z_0 + T_z} \begin{bmatrix} K_u & 0 \\ 0 & K_v \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} \frac{1}{K_u} & 0 \\ 0 & \frac{1}{K_v} \end{bmatrix} \left(\mathbf{d_0}(s) - \begin{bmatrix} u_0^{(0)} \\ v_0^{(0)} \end{bmatrix} \right) + \\ & + \frac{f^{(i)}}{Z_0 + T_z} \begin{bmatrix} K_u & 0 \\ 0 & K_v \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix}. \end{aligned}$$

The above equation can be rewritten as

$$\mathbf{d}(s) - \begin{bmatrix} u_0^{(i)} \\ v_0^{(i)} \end{bmatrix} = \mathbf{L} \left(\mathbf{d_0}(s) - \begin{bmatrix} u_0^{(0)} \\ v_0^{(0)} \end{bmatrix} \right) + \mathbf{p},$$

where

$$\mathbf{L} = \frac{f^{(i)}}{f^{(0)}} \frac{Z_0}{Z_0 + T_z} \begin{bmatrix} K_u & 0 \\ 0 & K_v \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} \frac{1}{K_u} & 0 \\ 0 & \frac{1}{K_v} \end{bmatrix} =$$
$$= \frac{f^{(i)}}{f^{(0)}} \frac{Z_0}{Z_0 + T_z} \begin{bmatrix} R_{11} & R_{12}\frac{K_u}{K_v} \\ R_{21}\frac{K_v}{K_u} & R_{22} \end{bmatrix}$$

and

$$\mathbf{p} = \frac{f^{(i)}}{f^{(0)}} \frac{1}{Z_0 + T_z} \begin{bmatrix} K_u & 0 \\ 0 & K_v \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix}. \tag{3}$$

The difference between the curve at a particular instant and the template is

$$\mathbf{d}(s) - \begin{bmatrix} u_0^{(i)} \\ v_0^{(i)} \end{bmatrix} - \mathbf{d_0}(s) + \begin{bmatrix} u_0^{(0)} \\ v_0^{(0)} \end{bmatrix} = (\mathbf{L} - \mathbf{I}) \left(\mathbf{d_0}(s) - \begin{bmatrix} u_0^{(0)} \\ v_0^{(0)} \end{bmatrix} \right) + \mathbf{p}, \tag{4}$$

where $\mathbf{I}$ is the 2×2 identity matrix.

Without loss of generality, the center of the template is assumed to be equal to the principal point, then equation (4) can be rewritten in terms of $\mathbf{d'_0}(s)$ and $\mathbf{d'}(s)$, that is, the projected contours referred to the template's centroid, as

$$\mathbf{d'}(s) - \mathbf{d'_0}(s) = (\mathbf{L} - \mathbf{I})\mathbf{d'_0}(s) + \mathbf{p} - \Delta\mathbf{u}, \tag{5}$$

where $\Delta\mathbf{u} \triangleq \begin{bmatrix} u_0^{(i)} - u_0^{(0)} \\ v_0^{(i)} - v_0^{(0)} \end{bmatrix}$. Equation (5) shows that the changes in the contour at each frame correspond to affine deformations of the template.

The affine parameters are $\mathbf{L}$ and $\mathbf{r} \triangleq \mathbf{p} - \Delta\mathbf{u}$, and are recovered from the shape of the contour, as follows. An active contour tracker is used to estimate the contour shape at each frame. This tracker is based on a Kalman filter and provides estimates of the contour's shape vector, which contains the affine parameters that relate the current shape of the contour with the template's shape (see [12] for details).

Assuming a constant aspect ratio $\mathcal{A} = \frac{K_u}{K_v}$, the $\mathbf{L}$ matrix can be rewritten as

$$\mathbf{L} = \frac{f^{(i)}}{f^{(0)}} \frac{Z_0}{Z_0 + T_z} \begin{bmatrix} R_{11} & R_{12}\mathcal{A} \\ R_{21}\frac{1}{\mathcal{A}} & R_{22} \end{bmatrix},$$

Then, taking $\mathcal{A} = 1$, a simplified matrix $\mathbf{L_s}$ can be computed

$$\mathbf{L_s} = \frac{f^{(i)}}{f^{(0)}} \frac{Z_0}{Z_0 + T_z} \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}. \tag{6}$$

3 Extraction of 3D motion parameters

In this section we derive the relation between the affine parameters described above and the 3D motion components: 3D rotation $\mathbf{R}$ and 3D translation $\mathbf{T}$. The rotation matrix can be written in terms of the Euler angles,

$$\mathbf{R} = \mathbf{R_z}(\phi)\mathbf{R_x}(\theta)\mathbf{R_z}(\psi) \tag{7}$$

where $\mathbf{R_z}(\psi)$ and $\mathbf{R_z}(\phi)$ are rotation matrices about the Z axis and $\mathbf{R_x}(\theta)$ is a rotation matrix about the X axis.

Using the Euler notation to represent the rotation matrix, equation (6) can be rewritten as

$$\begin{aligned}\mathbf{L_s} = & \frac{f^{(i)}}{f^{(0)}}\frac{Z_0}{Z_0+T_z}\mathbf{R_z}|_{\mathbf{2}}(\phi)\mathbf{R_x}|_{\mathbf{2}}(\theta)\mathbf{R_z}|_{\mathbf{2}}(\psi) = \\ & = \frac{f^{(i)}}{f^{(0)}}\frac{Z_0}{Z_0+T_z}\mathbf{R_z}|_{\mathbf{2}}(\phi)\begin{bmatrix}1 & 0\\ 0 & cos\theta\end{bmatrix}\mathbf{R_z}|_{\mathbf{2}}(\psi), \qquad (8)\end{aligned}$$

where $\mathbf{R}|_{\mathbf{2}}$ denotes the 2×2 submatrix of $\mathbf{R}$. Then,

$$\mathbf{L_s}\mathbf{L_s}^{\mathbf{T}} = \mathbf{R_z}|_{\mathbf{2}}(\phi)\begin{bmatrix}\left(\frac{f^{(i)}}{f^{(0)}}\frac{Z_0}{T_z+Z_0}\right)^2 & 0\\ 0 & \left(\frac{f^{(i)}}{f^{(0)}}\frac{Z_0}{T_z+Z_0}\right)^2 cos^2\theta\end{bmatrix}\mathbf{R_z}|_{\mathbf{2}}{}^{-\mathbf{1}}(\phi).$$

This last equation shows that θ can be computed from the ratio of eigenvalues of $\mathbf{L_s}\mathbf{L_s}^{\mathbf{T}}$, namely (λ_1, λ_2),

$$cos\theta = \sqrt{\frac{\lambda_2}{\lambda_1}},$$

where $\lambda_1 = \left(\frac{f^{(i)}}{f^{(0)}}\frac{Z_0}{Z_0+T_z}\right)^2$ is the largest eigenvalue. The angle ϕ can be extracted from the eigenvectors of $\mathbf{L_s}\mathbf{L_s}^{\mathbf{T}}$. The eigenvector $\mathbf{v_1}$ with largest eigenvalue equals the first column of $\mathbf{R_z}|_{\mathbf{2}}(\phi)$,

$$\mathbf{v_1} = \begin{bmatrix}cos\phi\\ sin\phi\end{bmatrix}.$$

At this stage, isolating $\mathbf{R_z}|_{\mathbf{2}}(\psi)$ in equation (8),

$$\mathbf{R_z}|_{\mathbf{2}}(\psi) = \frac{f^{(0)}}{f^{(i)}}(1+\frac{T_z}{Z_0})\begin{bmatrix}1 & 0\\ 0 & \frac{1}{cos\theta}\end{bmatrix}\mathbf{R_z}|_{\mathbf{2}}(-\phi)\mathbf{L_s},$$

and observing that

$$\frac{f^{(0)}}{f^{(i)}}\left(1+\frac{T_z}{Z_0}\right) = \frac{1}{\sqrt{\lambda_1}}, \qquad (9)$$

we can find $sin\psi$ and then ψ. Once the angles ψ, θ, ϕ are known, the rotation matrix $\mathbf{R}$ can be computed as in equation (7).

¿From equation (9) the scaled depth is recovered as

$$1+\frac{T_z}{Z_0} = \frac{1}{\sqrt{\lambda_1}}\frac{f^{(i)}}{f^{(0)}}.$$

This recovered depth depends on the relation between the focal lengths in consecutive frames. In robot vision applications one may assume that the robot controls the zooming factor. Hence, the relation between focal lengths at different time instants may be assumed known even when the exact focal length is unknown.

From equations (3) and (9),

$$\frac{f^{(0)}}{Z_0}\begin{bmatrix}T_x\\ T_y\end{bmatrix} = \frac{\mathbf{r}+\Delta\mathbf{u}}{\sqrt{\lambda_1}}\begin{bmatrix}\frac{1}{K_u} & 0\\ 0 & \frac{1}{K_v}\end{bmatrix}.$$

Thus, we observe that the recovered scaled translation depends on the difference between the principal points in consecutive frames.

4 Experimental results

Previosly to incorporating the technique to the visual system of the robot ARGOS [15], for which it has been developed, we have performed some experiments in a more controlled setting. Two examples of the experiments conducted to test the method in the laboratory are presented. Figure 1 shows an example of the results of the 3D motion estimation while zooming. An active contour has been fitted to the target (i.e. the square). A virtual object is drawn in the middle of the image with the estimated 3D rotation and translation along Z between the camera and the target. As expected, the estimated 3D rotation is invariant to zooming, while the estimated translation along Z changes proportionally to the zoom factor.

Figure 2 shows the results obtained for different 3D motions. Again a virtual object is drawn in the middle of the screen following the motions of the target. We verify that the proposed method provides qualitatively correct results.

5 Concluding remarks

We have analysed how the deformation of an active contour can be used to extract the 3D motion components while zooming. The basis of the method draws on ideas from previously published papers by the authors [10, 9], and fills the remaining hole in the analysis of the deformation of an active contour for different assumptions about the intrinsic camera parameters.

The theoretical deduction along with the experimental results show that the 3D rotation matrix can be reliably recovered while zooming. However, as one could expect, the scaled depth is distorted by the camera zoom. On the other hand, the change in the principal point due to the zoom affects the recovery of the other two components of the 3D translation vector. The experiments have been conducted with a monocular camera, hence the results keep the ambiguities common in this case. However, the deduction may be easily extended to a stereo rig.

Acknowledgments

This research has been partially supported by the research grant "Navegación autónoma de robots guiados por objetivos visuales" CICYT DPI2000-1352-C02-01 of the Spanish Science and Technology Council.

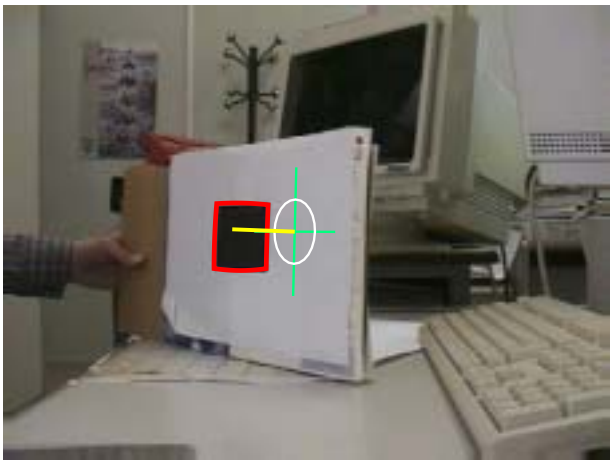
References

[1] L. Agapito, R. Hartley, and E. Hayman. Linear calibration of a rotating and zooming camera. In *Proc. of the Computer Vision and Pattern Recognition Conference*, 1999.

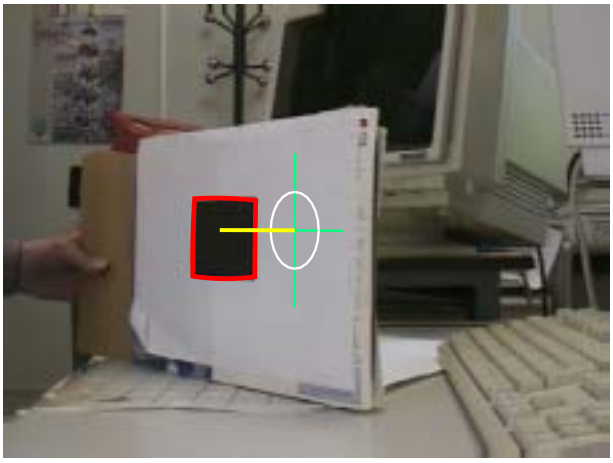
[2] A. Blake and M. Isard. *Active contours*. Springer, 1998.

[3] A. Blake, M.A. Isard, and D. Reynard. Learning to track the visual motion of contours. *J. Artificial Intelligence*, 78:101–134, 1995.

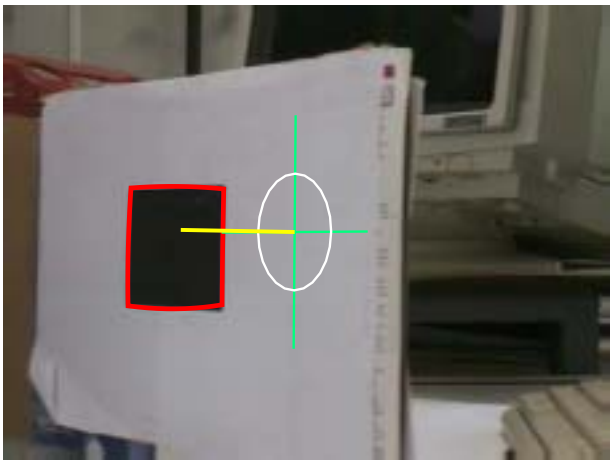
[4] J.L. Crowley, P. Bobet, and C. Schmidt. Maintaining stereo calibration by tracking image points. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 483–488, 1993.



A.



B.



C.

Figure 1: *3D motion recovery while zooming. The estimation of the rotation is invariant to camera zooming.*

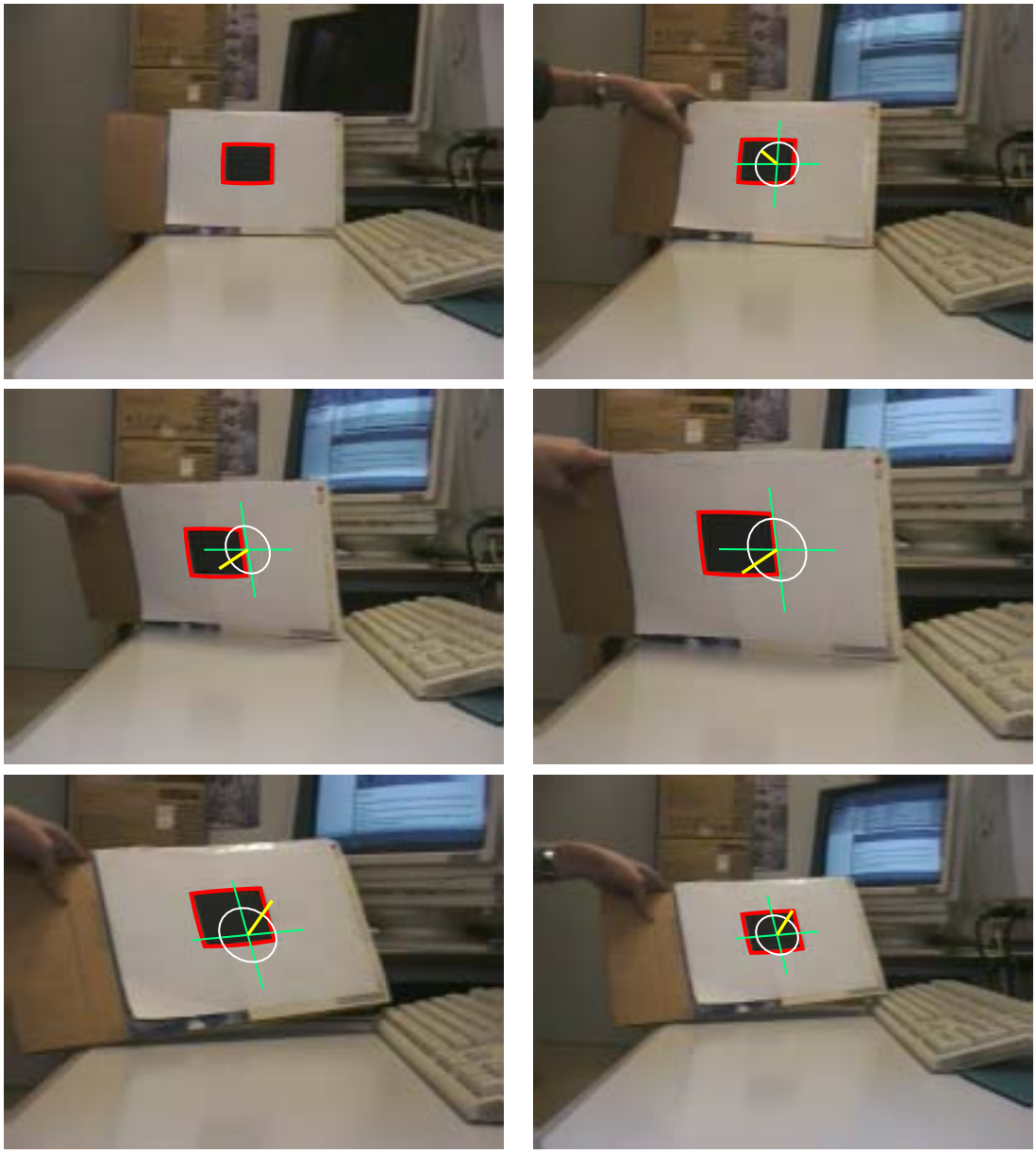


Figure 2: *3D motion recovery with an uncalibrated camera while zooming. Six samples of a video sequence taken by a static camera observing a moving target. The first image is the initial view, which is taken as the template. The subsequent images show the recovered motion after different target motions.*

[5] E. Hayman, T. Thrhallsson, and D. Murray. Zoom-invariant tracking using points and lines in affine views an application of the affine multifocal tensors. In *Proc. 7th Int. Conf. on Computer Vision*, September 1999.

[6] A. Heyden and K. Astrom. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1997.

[7] B.K.P. Horn. *Robot vision.* MIT Press, 1986.

[8] M.A. Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. Computer Vision*, pages 343–356, Cambridge, England, Apr 1996.

[9] E. Martínez and C. Torras. Depth map from the combination of matched points with active contours. In *Proc. of the IEEE International Conference on Intelligent Vehicles.*, pages 332–338, Dearborn, Michigan, USA, October, 2000.

[10] E. Martínez and C. Torras. Epipolar geometry from the deformation of an active contour. In *Proc. International Conference on Pattern Recognition.*, pages 534–537, Barcelona, Spain. September, 2000.

[11] E. Martínez and C. Torras. Integration of appearance and geometric methods for the analysis of monocular sequences. In *Proc. IST/SPIE 12th Annual Symp. on Electronic Imaging*, pages 62–70, San Jose. California. USA. January, 2000.

[12] E. Martínez and C. Torras. Qualitative vision for the guidance of legged robots in unstructured environments. *Pattern Recognition.*, 34(8):1585–1600, August 2001.

[13] D.W. Murray and B.F. Buxton. *Experiments in the machine interpretation of visual motion.* MIT Press, 1990.

[14] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. 6th Int. Conf. on Computer Vision, Bombay, India*, pages 90–95, January 1998.

[15] Argos Robot. *www-iri.upc.es/people/porta/robots/argos/index.html.*

[16] S. Soatto and P. Perona. Recursive estimation of camera motion from uncalibrated image sequences. In *Proc. 1st IEEE International Conference on Image Processing (ICIP)*, pages II–58–62, 1994.

[17] P. Sturm. Self calibration of a moving camera by pre–calibration. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1996.

[18] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[19] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.