Automatic segmentation and matching of planar contours for visual servoing

G. Chesi*, E. Malis and R. Cipolla

*Dipartimento di Ingegneria dell'Informazione - Università di Siena
Via Roma, 56 - 53100 Siena, ITALY

Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK

Abstract

In this paper we present a complete system for segmenting, matching and tracking planar contours for use in visual servoing. Our system can be used with arbitrary contours of any shape and without any prior knowledge of their models. The system is first shown the target view. A selected contour is automatically extracted and its image shape is stored. The robot and object are then moved and the system automatically identifies the target. The matching step is done together with the estimation of the homography matrix between the two views of the contour. Then, a 2 1/2 D visual servoing technique is used to reposition the end-effector of a robot at the target position relative to the planar contour. The system has been successfully tested on several contours with very complex shapes such as leaves, keys and the coastal outlines of islands.

1 Introduction

Matching two views of an object under full perspective projection and for large camera displacements is a difficult problem. Many visual servoing systems are contrived to make the vision problem simple (usually using points and lines as visual features) [5][9]. Even using "a priori" knowledge of the 3D model of the objects as in [6], finding the correspondences between the projection of the model and a real image remains a difficult problem. Recently, image-based visual servoing with respect to more complex objects has been studied assuming that the matching has been done [4]. In this paper, a visual servoing system using complex features such as contours is proposed dealing with the problem of finding correspondences. In order to simplify the matching problem, the only hypotheses made here are that the contour is planar and that occlusions can occur only during the tracking stage.

The main approaches for curve matching are based on finding invariants to the transformation linking two images of the curve. Geometric invariants have been studied extensively [15] [11] [12]. However, existing invariants suffer from occlusion and sensitivity to image noise. To cope with these problems, semi-local integral invariants were proposed in [13]. They showed that it is possible to define invariants semi-locally which require a lower order of derivatives and hence are less sensitive to noise. Although semi-local integral invariants reduce the order of derivatives required, it is still high in the general affine case. A quasi-invariant parametrisation was introduced in [14] which makes it possible to use second order derivatives instead of fourth and fifth but is only approximately invariant.

Not only are derivatives of high order difficult to calculate since sensitive to noise, but for some curves the derivatives do not give any information (e.g. polygonal curves) or contain many discontinuities. Hence, for these contours, it is not possible to use differential or semi local-integral invariants, while it would be possible to use invariants based on features like corners that, however, are not present in smooth curves. In this paper, the model of the objects is supposed to be unknown and the contours are neither smooth nor polygonal. Our method allows us to deal with both previous cases and the contour matching is completely automatic. If different objects are present in the scene, the right one is automatically selected.

Together with computing the correspondences between the points of the two contours, the homography matrix between the two views is estimated in order to use the visual servoing method proposed in [10]. This approach is called 2 1/2 D visual servoing since the input is expressed in part in the 3D Cartesian space and in part in the 2D image space. More precisely, it is based on the camera displacement estimation from the homography matrix (the rotation and the scaled translation of the camera) between the current and reference views of an object. The visual servoing system proposed in [10] was tested on simple images. In that case the objects were specially marked with white points on black backgrounds and the matching problem was assumed solved. In this paper, the system is considerably improved by using complex shapes with the 2 1/2 D visual servoing. This method was specially designed to work without any knowledge about the 3D structure of the target and to exploit the information provided by the homography matrix alone. However, it must be noticed that after the matching step the 2D visual servoing [2] [4] can also be applied without using the estimated homography.

The paper is organised as follows. In the first section, the contours segmentation is briefly described. In the second section, the algorithm for matching two views of a contour and, at the same time, for finding the collineation between them is presented. In the third section, the collineation is used to servo the robot with the 2 1/2 D visual servoing technique. In the fourth section, the experimental results demonstrate the validity of our method.

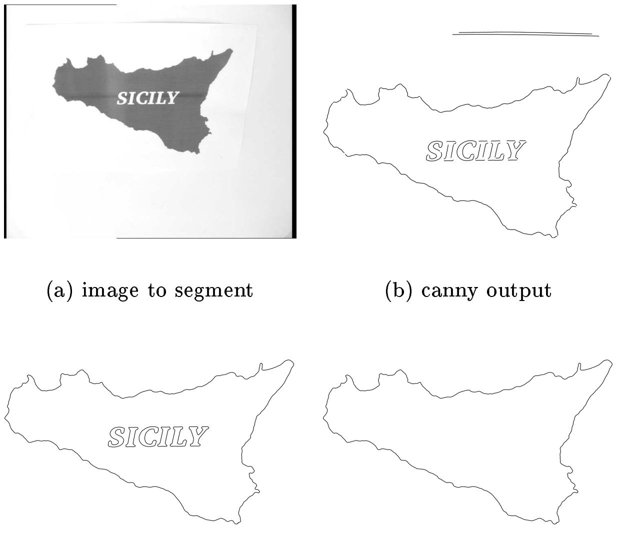
2 Contour segmentation

The segmentation of closed contours is made in two steps: edge detection and edge linking. The first step is done by using the Canny edge detection algorithm [1]:

1. Image smoothing: a Gaussian filter is convolved with the image to reduce the noise.
2. Discontinuities detection: the image gradient is computed to find edges.
3. Non-maxima suppression: gradient magnitude ridges are thinned to obtain edges.
4. Thresholding: an hysteresis method is used to identify and connect edges.

The second part takes as input all the edges found in the first part (the hysteresis algorithm provides a list of edges that, in general, represent discontinuous contours) and links them in order to form closed curves. The linking is done by connecting the closest edges, only if the distance is below a predefined threshold.

Figure 1 shows an example for the detection of the closed contour of the island of Sicily. In the image shown in Figure 1(a) there are seven closed contours given by the island and the letters of "SICILY", as shown in Figure 1(c). Figure 1(b) shows the output of the Canny edge detector for our image. Once all the contours in the image have been found, one of them can be selected as a reference contour for the visual servoing (for example in Figure 1(d) only Sicily's contour was selected). This selection is here done by hand but it could be done automatically.



(a) image to segment (b) canny output

(c) linker output (d) selected contour

Figure 1: Closed contours detection

3 Matching and homography estimation

After the segmentation of a reference image and the selection of the reference contour C^*, the camera is displaced to another position and the segmentation is repeated for the current image. A point $\mathcal{P}$ of the contour C corresponding to the point of C^* in the reference image can be obtained from $\mathcal{P}^*$ since (see figure 2):

$$\mathbf{p} \propto \mathbf{G}\mathbf{p}^* \tag{1}$$

where $\mathbf{G}$ is the collineation matrix of dimension 3×3, $\mathbf{p} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ and $\mathbf{p}^* = \begin{bmatrix} u^* & v^* & 1 \end{bmatrix}^T$ are vectors containing the projective coordinates (in pixels) of the points $\mathcal{P}$ and $\mathcal{P}^*$ respectively.

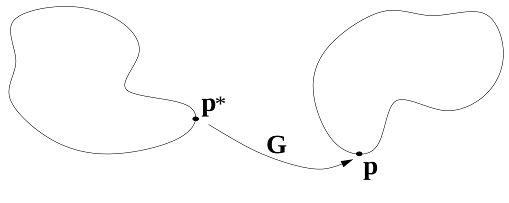


Figure 2: point to point correspondence

The problem is thus to find jointly the collineation $\mathbf{G}$ and which point of C^* is transformed in which point of C. Once the collineation matrix has been found, the homography matrix $\mathbf{H}$ can be easily computed:

$$\mathbf{H} = \mathbf{A}\mathbf{G}\mathbf{A}^{-1} \tag{2}$$

where $\mathbf{A}$ is the triangular matrix of the camera internal parameters of dimension 3×3. Let us notice that, in the vision community, the words "collineation" and "homography" are both used to indicate a projective transformation between two hyper-planes (in our case two dimensional). However, it is important to distinguish between the uncalibrated transformation $\mathbf{G}$ and the calibrated one $\mathbf{H}$ from which it is possible to extract the Euclidean information.

In the next subsection, an algorithm for the estimation of the collineation matrix is described. In the second subsection, this algorithm is used to recognise the selected object when there are more than one closed contours in the current image. Finally, some results obtained with our method are presented in the third subsection.

3.1 Collineation estimation algorithm

The algorithm proposed in [3] is initially used to match the contours and find the collineation matrix. It consists of two main parts: the correspondence finder and the collineation finder. The first part determines the best point on C corresponding to the starting point on the contour C^*. In this part of the algorithm the last row of the matrix $\mathbf{G}$ is supposed to be constant. This means that we are considering a weak perspective transformation between the two contours that allow us to use a very fast least-square technique to compute the solution. The second part of the algorithm determines an estimation of $\mathbf{G}$ based on the correspondence found in the first part. Now the transformation is supposed to be full perspective. The two steps are repeated until the matching error does not decrease any more.

This algorithm works well even for large perspective transformations. However, if perspective is strong, the starting point found by the algorithm is not precise enough to obtain good correspondences. The matching precision can be improved by using the properties of the Discrete Fourier Transform (DFT). Let (u^*, v^*) be the image coordinates of a point belonging to the contour C^*. If we have a guess $\widehat{\mathbf{G}}$ of the collineation matrix, a new contour $C^{'}$ can be obtained from C^* since:

$$\begin{aligned} \widehat{u}^{'} &= \frac{\widehat{g}_{11}u^* + \widehat{g}_{12}v^* + \widehat{g}_{13}}{\widehat{g}_{31}u^* + \widehat{g}_{32}v^* + \widehat{g}_{33}} \\ \widehat{v}^{'} &= \frac{\widehat{g}_{21}u^* + \widehat{g}_{22}v^* + \widehat{g}_{23}}{\widehat{g}_{31}u^* + \widehat{g}_{32}v^* + \widehat{g}_{33}} \end{aligned} \tag{3}$$

where $\widehat{g}_{ij}$ is an element of the guessed collineation matrix $\widehat{\mathbf{G}}$. The curve obtained applying the guessed collineation matrix is re-parametrised in such a way that the points on the respective contours be uniformly spaced (i.e. $(\widehat{u}^{'}, \widehat{v}^{'}) \rightarrow (\widehat{u}, \widehat{v})$). Now, let's consider the DFT of the vectors $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}})$ and $(\mathbf{u}, \mathbf{v})$ containing respectively the coordinates of the estimated contour $\hat{C}$ and of the contour C.

$$\begin{array}{lllllll} \widehat{\mathcal{U}} &=& \mathcal{F}(\widehat{\mathbf{u}}) & \quad & \widehat{\mathcal{V}} &=& \mathcal{F}(\widehat{\mathbf{v}}) \\ \mathcal{U} &=& \mathcal{F}(\mathbf{u}) & \quad & \mathcal{V} &=& \mathcal{F}(\mathbf{v}) \end{array} \tag{4}$$

Then a more accurate estimation of $\mathbf{G}$ can be found as the one minimising the difference between the magnitude of $\mathcal{U}, \mathcal{V}$ and the magnitude of $\widehat{\mathcal{U}}, \widehat{\mathcal{V}}$:

$$\min_{\mathbf{G}} \sum_{i=1}^{n} \left(|\widehat{\mathcal{U}}(i)| - |\mathcal{U}(i)|\right)^2 + \left(|\widehat{\mathcal{V}}(i)| - |\mathcal{V}(i)|\right)^2 \tag{5}$$

This cost function does not depend on the choice of the starting point (provided that the parametrisation is the same for the two curves). Indeed, the vectors $\widehat{\mathbf{u}}, \widehat{\mathbf{v}}$ computed with $\widehat{\mathbf{G}}$ (and after the reparametrisation) differ from $\mathbf{u}, \mathbf{v}$ by only a shift and, hence, the magnitude of their DFT is the same. The optimisation problem defined in (5) takes a longer time than the algorithm proposed in [3] and is used only if the initial matching error is too high.

Once the best collineation matrix is found, the reference contour C^* is reprojected in the current image and the nearest points to the points of the current contour are matched.

3.2 Object identification

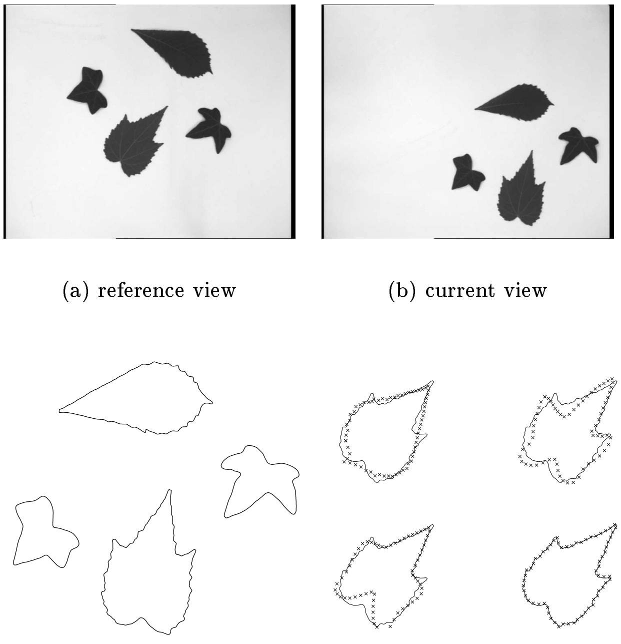
Unlike the method proposed in [3] where the reference object was supposed to have already been identified, here we consider the problem of recognising the contour in the current view in the presence of other objects and finding the collineation matrix between the reference view. Let C_i be a contours in the current view. For all i an estimation $\widehat{\mathbf{G}}_i$ of $\mathbf{G}$ is calculated with the collineation estimation algorithm (supposing C_i the corresponding contour of C^*) and C_i is projected on the contour $\widehat{C}_i^*$ in the reference view by the inverse collineation $\mathbf{G}_i^{-1}$. Then the selected object is represented by the contour $\widehat{C}_i^*$ minimising the distance $d(C^*, \widehat{C}_i^*)$ from C^*, where:

$$d(C^*, \widehat{C}_i^*) = \max\{\delta(C^*, \widehat{C}_i^*), \delta(\widehat{C}_i^*, C^*)\} \tag{6}$$

and

$$\delta(C^*, \widehat{C}^*) = \frac{1}{n} \sum_{j=1}^{n} \min_k \left\| \mathbf{p}_j^* - \widehat{\mathbf{p}}_k^* \right\| \tag{7}$$

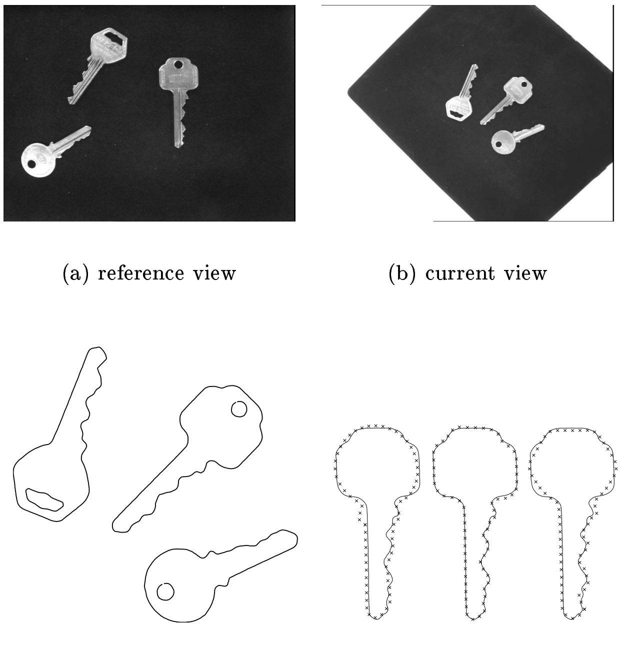
3.3 Examples



(a) reference view (b) current view

(c) contours C_i (d) errors $\widehat{C}_i^* - C^*$

Figure 3: Matching of a leaf.



(a) reference view (b) current view

(c) contours C_i (d) errors $\widehat{C}_i^* - C^*$

Figure 4: Matching of a key.

The segmentation and matching algorithms were tested on several planar objects with polygonal, smooth or fractal-like shapes. Figure 3 shows an example using four leaves with similar shapes. The selected leaf in the reference view is the one on the bottom. Figure 3(c) shows the closed contours found in the current view. For all them the collineation matrix is estimated and the results are shown in Figure 3(d). The calculated distances (from the left-bottom leaf in the anti-clockwise sense) are 5.92, 1.07 (the selected leaf), 58 and 51.2 pixels which means that the reference leaf was identified with a great margin. If the contours are very similar, as in Figure 4 where three keys are used, the errors could be very close. In this example, the relative position of the objects in the initial view and in the reference view is not the same. The selected key is the one on the right in the reference view. The distances obtained with the estimated collineation matrices (from the bottom key in the anti-clockwise sense) are 2.95, 0.99 (the selected key) and 2.68 pixels. Even if the distances are closer than in the previous case the reference key is recognised with a good margin.

It must be noticed that the matching precision depends also on the number of points used to describe the contours. In our experiments, 256 points are used to describe very complex shapes. The distance between two points is, in our examples, greater than two pixels. A good result is thus to obtain a matching error smaller than the sampling step. It could be possible to increase the precision using more points but only during the matching step. Indeed, the system does not allow us to update the homography matrix at video-rate with more than 256 points during the servoing stage. Moreover, the precision depends also on the planarity of the objects. In the presented examples, the leaves are not exactly planar (the shadow can also modify the contours) which explain why the matching error is smaller in the case of the keys.

4 Visual Servoing

The homography matrix computed during the matching step can be decomposed as follows [8]:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*T} \tag{8}$$

where $\mathbf{R}$ and $\mathbf{t}$ are respectively the rotation and the translation of the frame F attached to the contour C with respect to the frame F^* attached to the contour C^*, $\mathbf{n}^*$ is the unit vector normal to the plane of the contour π expressed in F^* and d^* is the distance between the origin of F^* and π. Then, the positioning task controlling the 6 camera d.o.f. is described as the regulation to zero of the following task function [10]:

$$\mathbf{e}^T = \left[\begin{array}{cc} \mathbf{m}_e^T - \mathbf{m}_e^{T*} & \mathbf{u}^T\theta \end{array} \right] \quad (9)$$

where $\mathbf{u}$ is the rotation axis and θ the rotation angle computed from $\mathbf{R}$ and $\mathbf{m}_e$ are the extended image coordinates of a point on the plane:

$$\mathbf{m}_e^T = \left[\begin{array}{ccc} x & y & z \end{array} \right] = \left[\begin{array}{ccc} \frac{X}{Z} & \frac{Y}{Z} & \log(Z) \end{array} \right] \quad (10)$$

where $z = \log(Z)$ is a supplementary coordinate controlling the relative depth of the camera from the plane (even if Z is unknown, from the homography matrix it is possible to compute $\log(Z) - \log(Z^*)$ [10]). Any point on the contour can be used as a reference point. However, the centre of the contour is used in the experiments even if it produces a supplementary error in the estimation of the coordinates of the point (it is well known that the projection of the centre of gravity of a planar contour is not the centre of gravity of the projection of the contour). Indeed, the performances of the system are not really affected since the 2 1/2 D visual servoing has proved to be robust to big calibration errors [10] using the following simple proportional control law:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^{-1} \widehat{\mathbf{e}} \quad (11)$$

where $\mathbf{v}$ is the camera velocity sent to the robot controller, $\widehat{\mathbf{L}}$ is an approximation of the interaction matrix related to the time variation of the task function $\mathbf{e}$, $\widehat{\mathbf{e}}$ is the estimated task function and λ a positive gain witch tunes the velocity of convergence.

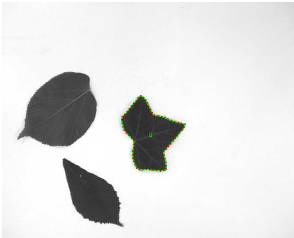
During the visual servoing the homography matrix is updated using the points of the contour tracked with the real time active contours tracking system proposed in [7]. Indeed, the reference object is tracked with an active contour constrained to undergo only deformations due to 2D projective transformations. This is achieved by searching for the contour in the image along the normal to the contour tangent at each point. The measurement so found is then projected down onto the space defined by the projective transformations group using the Lie Algebra and, successively, the active contour updated. At each iteration, only the homography matrix is updated supposing the initial match to be good. If there is a big matching error, the system still converge near the final position but it will oscillate around it. The precision of the system can be improved by repeating the matching procedure during the servoing especially near the convergence when the two contours are very close.

5 Experimental results

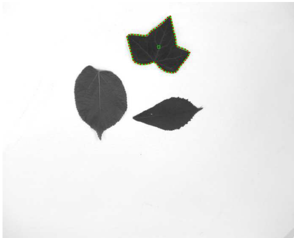
In our experiments we have used a Mitsubishi robot RV-E2 Movemaster with 6 degrees of freedom. A coarsely calibrated camera (the parameters given by the manufacturer are used) is mounted on the end-effector. The transformation between the end-effector frame and the camera frame is only roughly known.

(a) reference position

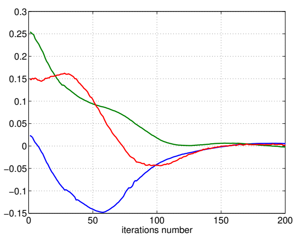
(b) initial position



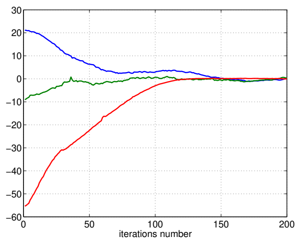
(c) reference view



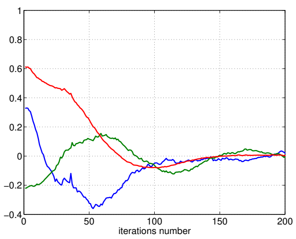
(d) initial view



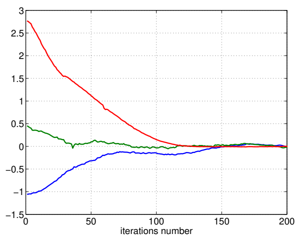
(e) ext. image coord.



(f) rotation $\mathbf{u}\theta$ (deg)



(g) transl. vel. (cm/s)



(h) rot. vel. (deg/s)

Figure 5: experimental results with a leaf

Visual servoing experiments have been realised using the contours presented in the paper. Figure 5 shows an experiment using leaves. The selected leaf in the reference view (Figure 5(c)) is ivy (several experiments have been done, using different leaves as reference target, obtaining similar results). Figure 5(d) shows the initial view of the leaves and the ivy matched with our algorithm. In both views the ivy is represented by an active contour that is used during the servoing to track the contour. In Figure 5(a) and 5(b) are shown respectively the reference and initial robot position. Figure 5(e) and 5(f) show the behaviour of the extended image coordinates and orientation of the camera, while Figure 5(g) and 5(h) show the translational and rotational velocity (i.e. the control law). The control law is stable and the error converge to zero but not exponentially (as it should in ideal conditions) since the system is coarsely calibrated. Furthermore, the centre of gravity of the contour was used instead of a point of the contour. The convergence of the visual servoing demonstrates that the initial matching was good in spite of the noise and the camera displacement. The visual servoing stopped when the error between corresponding points is less than 0.5 pixel. It is surprising that such a precision can be reached if the initial matching error was bigger. However, the matching error depends on the camera displacement between the two views. As mentioned previously, it would be then a great improvement of the system precision to repeat the matching step online near the convergence.

6 Conclusion

In this paper a system for segmenting and matching a planar contour was presented. The matching algorithm find the homography between the initial and the target view, which is used to position a robot with respect to the contour with a 2 1/2 D visual servoing technique. The experimental results show that the system is able to match objects with complex shapes for which invariants cannot be used. Moreover, the selected object is recognised even in presence of other similar ones. The accuracy of the estimated homography is tested servoing a camera mounted on the end-effector of a 6 d.o.f. robot. The experiments show that the system can position the robot end-effector with a great precision even for complex planar objects and large camera displacements.

Acknowledgements

This work was supported by an EC (ESPRIT) grant no. LTR26247 (VIGOR) and an EPSRC grant no. GR/K84202. We would thank Tom Drummond for providing the tracking software.

References

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

[2] F. Chaumette. *La relation vision-commande: théorie et application à des tâches robotiques.* PhD thesis, Universit de Rennes I, IRISA, July 1990.

[3] G. Chesi, E. Malis, and R. Cipolla. Collineation estimation from two unmatched views of an unknown planar contour for visual servoing. In *British Machine Vision Conference*, Nottingham, UK, Sep. 1999.

[4] C. Collewet. *Contributions á l'élargissement du champ applicatif des asservissements visuel 2D.* PhD thesis, Universit de Rennes I, IRISA, February 1999.

[5] P. I. Corke. Visual control of robot manipulators - A review. In K. Hashimoto, editor, *Visual servoing*, vol. 7 of *World Scientific Series in Robotics and Automated Systems*, pp. 1–31. World Scientific Press, 1993.

[6] T. Drummond and R. Cipolla. Real-time tracking of complex structures with on-line camera calibration. In *British Machine Vision Conference*, Nottingham, United Kingdom, September 1999.

[7] T. Drummond and R. Cipolla. Visual tracking and control using Lie algebras. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 652–657, Fort Collins, Colorado, June 1999.

[8] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[9] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[10] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):234–246, April 1999.

[11] T. Reiss. *Recognizing planar object using invariant image features.* (LNCS 676), Springer Verlag, 1993.

[12] C. Rothwell. *Object recognition through invariant indexing.* Oxford Science Publications, 1995.

[13] J. Sato and R. Cipolla. Affine integral invariants and matching of curves. In *Int. Conf. on Pattern Recognition*, volume 1, pages 915–919, Vienna, Austria, 1996.

[14] J. Sato and R. Cipolla. Quasi-invariant parametrisations and matching of curves in images. *Int. Journal of Computer Vision*, 28(2):117–136, June/July 1998.

[15] G. Taubin and D. Cooper. *Object recognition based on moment (or algebraic) invariants.* In Geometric Invariance in Computer vision. MIT Press, 1992.