2 1/2 D visual servoing with respect to planar contours having complex and unknown shapes

Ezio Malis, G. Chesi, R. Cipolla

2 1/2 D visual servoing with respect to planar contours having complex and unknown shapes

E. Malis*, G. Chesi† and R. Cipolla‡

Abstract

In this paper we present a complete system for segmenting, matching, tracking, and visual servoing with respect to unknown planar contours. 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 reference position relative to the planar contour. The system has been successfully tested on several contours with very complex shapes such as leaves, keys and coastal outlines of islands. Experiments using a ship mockup without any artificial marker proves that the system can be applied to ship building industry.

*E. Malis is with INRIA, ICARE Project, Sophia Antipolis, France.
†G. Chesi is with Department of Information Engineering, University of Siena, Italy.
‡R. Cipolla is with Department of Engineering, University of Cambridge, UK.

1 Introduction

Standard visual servoing systems (Hashimoto, 1993) (Hutchinson et al., 1996) are generally contrived to make the vision problem easy by using simple geometric visual features, or by using some "a priori" knowledge on the 3D geometry of the objects. From both theoretical and practical point of views, it would be preferable to use visual features extracted from images of complex unknown objects. In this case, the choice of the vision-based controller is extremely important since it strongly depends on the amount of knowledge about the considered scene. For example, if the 3D structure of the environment is unknown then position-based visual servoing (Wilson et al., 1996) can not be used. Image-based visual servoing (Espiau et al., 1992) can deal with complex objects and it has been recently applied to contours with unknown shape (Collewet and Chaumette, 2000). However, image-based visual servoing is not completely model-free since the depths of the observed objects are needed to compute the control law. The depths are generally unknown and only approximations can be used (Espiau et al., 1992) (Collewet and Chaumette, 2000). In order to deal with completely unknown environments, model-free approaches have been proposed recently (Basri et al., 1998) (Malis et al., 1999). These methods, based on projective reconstruction of the scene from only features matching, are robust to cali-

bration errors (Malis and Chaumette, 2002) and need only a rough estimate of the distance of the camera to the object. This single depth parameter can be chosen very approximately. Indeed, it has been proven in (Malis et al., 1999) that the distance has little influence on the stability of the system. The 2 1/2 D visual servoing system proposed in (Malis et al., 1999) was specially designed to work without any knowledge about the 3D structure of the target and to exploit the information provided by a projective reconstruction between two views of any rigid object. Despite its generality, the approach has been tested on simple visual features. In that case, the objects were specially marked with white points on black backgrounds and the matching problem was assumed to be solved. The primary objective of this paper is to propose a 2 1/2 D visual servoing which allows to use unknown complex planar contours. When considering a planar contour, the projective reconstruction consists in recovering the homography matrix between two images of the contour. Thus, in addition to the robustness of the approach to calibration errors (Malis et al., 1999), we can benefit also from recent planning techniques (Mezouar and Chaumette, 2000) which can be used to keep the object in the field of view of the camera. Another objective of this paper is to integrate all the steps of a visual servoing system: image segmentation, matching, tracking and servoing. In particular, we focus on the matching of two images of a contour with an unknown complex shape

since we need to find the correspondences between two contours in order to compute the homography. Despite matching is a problem which is common to all visual servoing systems it is often supposed to be solved. On the other hand, matching two views of an object under full perspective projection and for large camera displacements is a difficult problem. Even using "a priori" knowledge of the 3D model of the objects, finding the correspondences between the projection of the model and a real image remains a difficult problem. 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 (Taubin and Cooper, 1992) (Reiss, 1993) (Rothwell, 1995). However, existing invariants suffer from occlusion and sensitivity to image noise. To cope with these problems, semi-local integral invariants were proposed in (Sato and Cipolla, 1996). 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. 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. Thus, we propose a matching method which allows us to deal with both previous cases. The contour matching is completely automatic and if different contours are present in the scene then the good one is automatically selected.

2 Matching and homography estimation

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 (Canny, 1986). The second part takes as input all the edges found in the first part and links them in order to form closed curves. It must be underlined that we suppose that occlusions can only occur during tracking (see Section 3). During the initial matching the full contour must be visible in the current and reference image.

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 1):

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

where $\mathbf{G}$ is the collineation matrix of dimension 3×3, $\mathbf{p} = (u, v, 1)$ and $\mathbf{p}^* = (u^*, v^*, 1)$ are vectors containing the projective coordinates (in pixels) of the points $\mathcal{P}$ and $\mathcal{P}^*$ respectively. The problem is thus to find jointly the

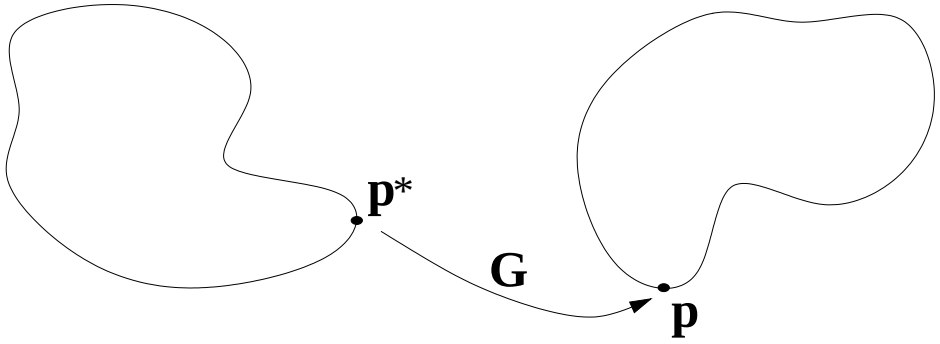


Figure 1: point to point correspondence

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 a triangular matrix containing the camera internal parameters of dimension 3×3:

$$\mathbf{A} = \begin{bmatrix} f & s & u_0 \\ 0 & fr & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where f is the focal length in pixels, s is the skew, r is the aspect ratio, u_0 and v_0 are the pixel coordinates of the principal point. In general, the camera internal parameters are only roughly known. Thus, we use an estimation $\widehat{\mathbf{A}}$ instead of the true matrix.

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 recognize 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.

2.1 The matching algorithm

The algorithm proposed in (Chesi et al., 1999) 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:

$$\begin{aligned} \widehat{u}' &= \frac{\widehat{g}_{11}u^* + \widehat{g}_{12}v^* + \widehat{g}_{13}}{\widehat{g}_{31}u^* + \widehat{g}_{32}v^* + \widehat{g}_{33}} \\ \widehat{v}' &= \frac{\widehat{g}_{21}u^* + \widehat{g}_{22}v^* + \widehat{g}_{23}}{\widehat{g}_{31}u^* + \widehat{g}_{32}v^* + \widehat{g}_{33}} \end{aligned} \tag{4}$$

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-parameterized 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.

$$\begin{array}{rclrcl} \widehat{\mathcal{U}} & = & \mathcal{F}(\widehat{\mathbf{u}}) & \widehat{\mathcal{V}} & = & \mathcal{F}(\widehat{\mathbf{v}}) \\ \mathcal{U} & = & \mathcal{F}(\mathbf{u}) & \mathcal{V} & = & \mathcal{F}(\mathbf{v}) \end{array} \tag{5}$$

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

$$\min_{\mathbf{G}} \sum_{i=1}^{n} \Big(|\widehat{\mathcal{U}}(i)| - |\mathcal{U}(i)|\Big)^2 + \Big(|\widehat{\mathcal{V}}(i)| - |\mathcal{V}(i)|\Big)^2 \tag{6}$$

This cost function does not depend on the choice of the starting point (provided that the parameterization is the same for the two curves). Indeed, the vectors $\widehat{\mathbf{u}}, \widehat{\mathbf{v}}$ computed with $\widehat{\mathbf{G}}$ (and after the re-parameterization) differ from $\mathbf{u}, \mathbf{v}$ by only a shift and, hence, the magnitude of their DFT is the same. The optimization problem defined in (6) takes a longer time than the algorithm proposed in (Chesi et al., 1999) 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.

2.2 Object identification

Unlike the method proposed in (Chesi et al., 1999) where the reference object was supposed to have already been identified, here we consider the problem of recognizing 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 contour 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 $\widehat{C}_i^*$ minimizing the distance $d(C^*, \widehat{C}_i^*)$ from C^*, where:

$$d(C^*, \widehat{C}_i^*) = \max\{\delta(C^*, \widehat{C}_i^*), \delta(\widehat{C}_i^*, C^*)\} \tag{7}$$

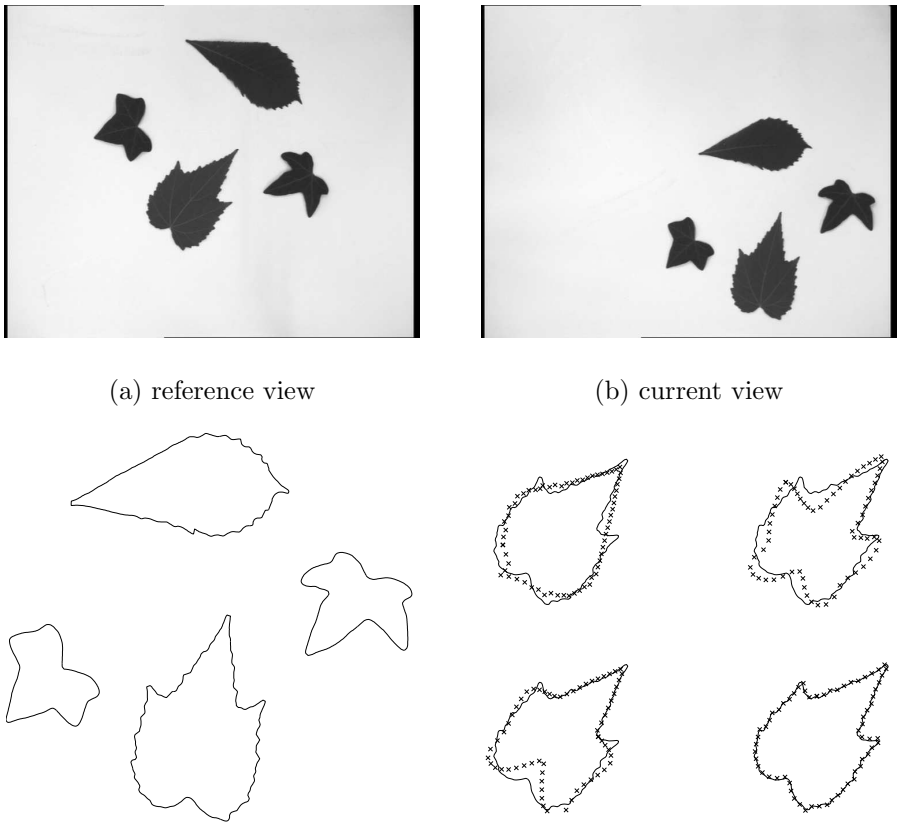
and

$$\delta(C^*, \widehat{C}^*) = \frac{1}{n} \sum_{j=1}^{n} \min_k \left\| \mathbf{p}_j^* - \widehat{\mathbf{p}}_k^* \right\|. \tag{8}$$

2.3 Examples

The segmentation and matching algorithms were tested on several planar objects with polygonal, smooth or fractal-like shapes. Figure 2 shows an example using four leaves with similar shapes. The selected leaf in the reference view is the one on the bottom. Figure 2(c) shows the closed contours

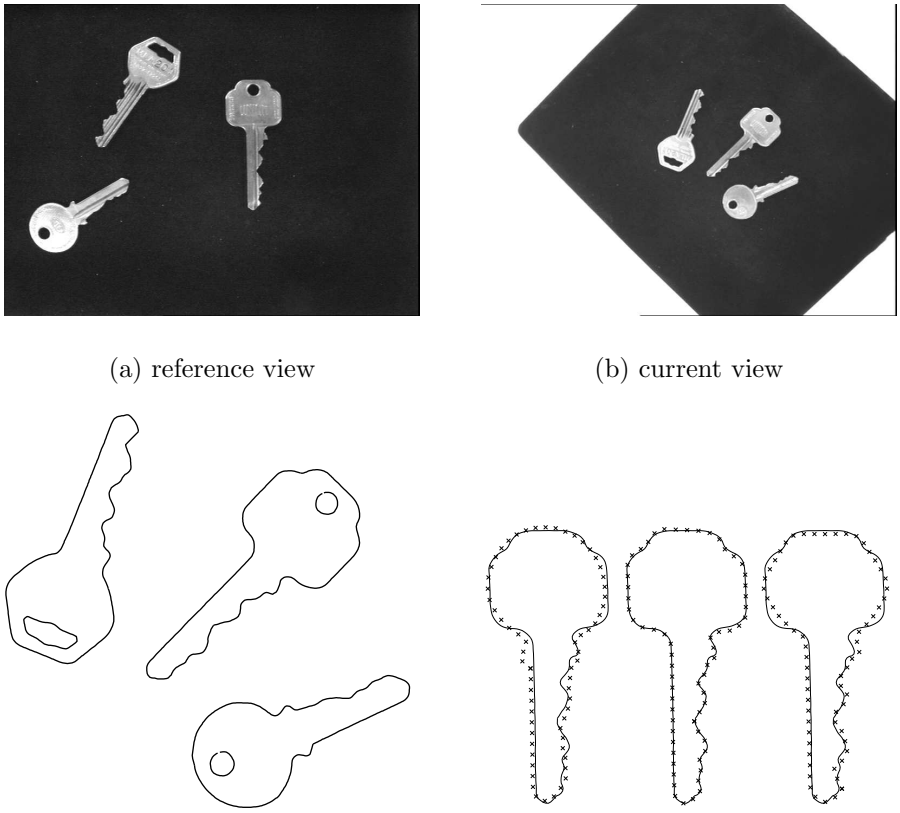
found in the current view. For all them the collineation matrix is estimated and the results are shown in Figure 2(d). The calculated distances (from the left-bottom leaf in the anti-clockwise sense) are 5.92, 1.07 (the selected leaf), 58 and 51.2 pixels which means that the reference leaf was identified with a great margin. If the contours are very similar, as in Figure 3 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 recognized 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.



(a) reference view (b) current view

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

Figure 2: Matching of a leaf. Despite leaves have very similar shapes, the matching is sufficiently accurate to identify the good leaf.



(a) reference view (b) current view

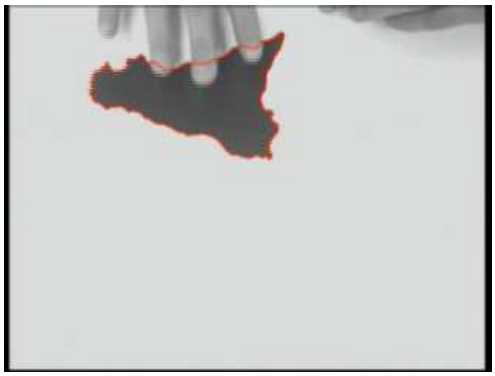
(c) contours C_i (d) errors $\widehat{C}_i^* - C^*$

Figure 3: Matching of a key. Despite keys have very similar shapes, the matching is sufficiently accurate to identify the good key.

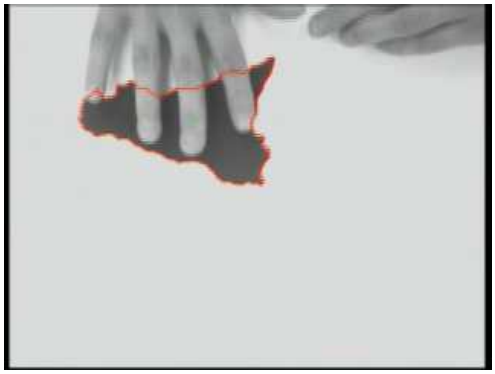
3 Contour tracking robust to occlusions

During the visual servoing the contour is tracked with the real time active contours tracking system proposed in (Drummond and Cipolla, 2000). The contour is represented by 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 measurements are projected down onto the space defined by the projective transformations group using the Lie Algebra and, successively, the active contour updated. This method is robust to occlusion even if only local information is used. Note that the tracking is not constrained only to affine transformations. Indeed, a projective compensation extends the range deformation to the full projective group in such a way that the affine component of the observed deformation is essentially intact. Figure 4 shows six images of a sequence where the contour of Sicily is tracked and Extension 1 shows the complete sequence. The robustness to occlusion is very important since a part of the contour can eventually get out of the image during servoing. Active contours are used to extract contours even in the presence of occlusions since the contour matching do not work if the contour is occluded. Since the active contour is constrained to follow a projective deformation, the occluded parts can be reconstructed. Once the full contour is available, the

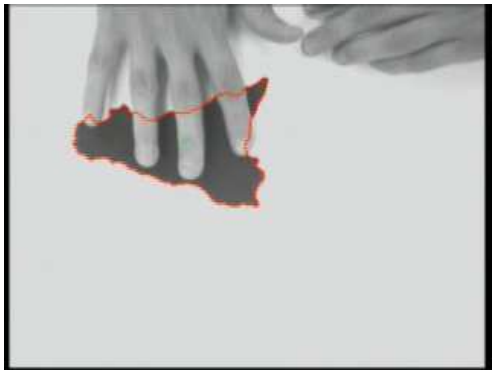
(a) Image 1



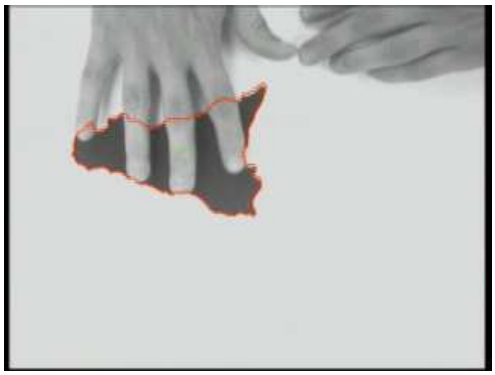
(b) Image 2



(c) Image 3



(d) Image 4



(e) Image 5

(f) Image 6

Figure 4: Tracking robust to occlusion (see also Extension 1).

matching between two consecutive views of the contour can be successfully achieved in real time.

The the local tracking could be improved by taking into account the multiple-view constraint existing between collineations. For example, if a reference image of the contour is available, as for standard teaching-by-showing techniques, then it is possible to re-project the occluded parts from the reference image to the current image using the collineation. Obviously, if there are not enough points of the contour in the image to compute the homography then the reprojection cannot be done.

4 Visual Servoing

The matching algorithm alone does not work in the presence of occlusions. Thus, the (full projective) contour tracking in (Drummond and Cipolla, 2000) is very useful since it allows to reconstruct the occluded parts (once the full contour is available the matching algorithm can be used). On the other hand, the vision-based control proposed in (Drummond and Cipolla, 2000) is designed for affine transformation only and cannot be used in the presence of full projective deformations. Thus, the visual servoing approach proposed in the paper is close to the 2 1/2 D visual servoing (Malis et al., 1999). This approach exploits the information extracted from an homography matrix. The

homography matrix computed during the matching step can be decomposed as follows (Faugeras and Lustman, 1988):

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*T} \tag{9}$$

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. 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 converges 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.

The positioning task controlling the 6 camera d.o.f. is described as the regulation to zero of the following task function (Malis et al., 1999):

$$\mathbf{e}^T = \begin{bmatrix} \mathbf{e}_\nu^T & \mathbf{e}_\omega^T \end{bmatrix} = \begin{bmatrix} \mathbf{m}_e^T - \mathbf{m}_e^{T*} & \mathbf{u}^T\theta \end{bmatrix} \tag{10}$$

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:

$$\mathbf{m}_e^T = \begin{bmatrix} x & y & z \end{bmatrix} = \begin{bmatrix} \frac{X}{Z} & \frac{Y}{Z} & \log(Z) \end{bmatrix} \tag{11}$$

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 $\log(Z) - \log(Z^*)$ (Malis et al., 1999)). Any point on the contour can be used as a reference point. However, it is not always possible to find a particular point on the contour (for example consider the case of smooth contours without any high curvature given in Extension 2).

In order to make the control law more general and to avoid the choice of a point we proposed in (Chesi et al., 2000) to use the centroid of the contour as a reference point. However, the center of the contour produces a supplementary error in the estimation of the coordinates of the reference point (it is well known that the projection of the centroid of a planar contour is not the centroid of the projection of the contour). It is therefore necessary to analyze the stability of the new control law. The extended images coordinates of the centroid of the contour are defined as follows:

$$\mathbf{m}_e^\top = \begin{bmatrix} \overline{x} & \overline{y} & \overline{z} \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n} x_i & \frac{1}{n}\sum_{i=1}^{n} y_i & \frac{1}{n}\sum_{i=1}^{n} z_i \end{bmatrix}$$

where $(\overline{x}, \overline{y})$ are the coordinates of the centroid of the contour and $\overline{z}$ 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 again possible to compute $\overline{z} - \overline{z}^*$). The derivative of the task function is related to the velocity $\mathbf{v}$ of the camera by:

$$\dot{\mathbf{e}} = \mathbf{L}\,\mathbf{v} \tag{12}$$

where the interaction matrix $\mathbf{L}$ is upper block triangular:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_\nu & \mathbf{L}_{\nu\omega} \\ 0 & \mathbf{L}_\omega \end{bmatrix} \tag{13}$$

The block $\mathbf{L}_\omega$ contains the same matrix given in (Malis et al., 1999), while the two other blocs have different form:

$$\mathbf{L}_\nu = \begin{bmatrix} -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{Z_i} & 0 & \frac{1}{n}\sum_{i=1}^{n}\frac{x_i}{Z_i} \\ 0 & -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{Z_i} & \frac{1}{n}\sum_{i=1}^{n}\frac{y_i}{Z_i} \\ 0 & 0 & -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{Z_i} \end{bmatrix}$$

$$\mathbf{L}_{\nu\omega} = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n} x_i y_i & -\frac{1}{n}\sum_{i=1}^{n} 1 + x_i^2 & \frac{1}{n}\sum_{i=1}^{n} y_i \\ \frac{1}{n}\sum_{i=1}^{n} 1 + y_i^2 & -\frac{1}{n}\sum_{i=1}^{n} x_i y_i & -\frac{1}{n}\sum_{i=1}^{n} x_i \\ -\frac{1}{n}\sum_{i=1}^{n} y_i & \frac{1}{n}\sum_{i=1}^{n} x_i & 0 \end{bmatrix}$$

In order to control the camera, we use the following simple proportional control law:

$$\mathbf{v} = -\widehat{\mathbf{L}}^{-1} \begin{bmatrix} \lambda_\nu \mathbf{I} & 0 \\ 0 & \lambda_\omega \mathbf{I} \end{bmatrix} \widehat{\mathbf{e}} = - \begin{bmatrix} \lambda_\nu \widehat{\mathbf{L}}_\nu^{-1} & -\lambda_\omega \widehat{\mathbf{L}}_\nu^{-1} \widehat{\mathbf{L}}_{\nu\omega} \widehat{\mathbf{L}}_\omega^{-1} \\ 0 & \lambda_\omega \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix} \widehat{\mathbf{e}} \qquad (14)$$

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, λ_ν and λ_ω two positive gains witch tunes the velocity of convergence. Without loss of generality, we can set $\widehat{\mathbf{L}}_\omega^{-1} = \mathbf{I}$ in the control law, since $\widehat{\mathbf{L}}_\omega^{-1}\widehat{\mathbf{e}}_\omega = \widehat{\mathbf{e}}_\omega$ (Malis et al., 1999). The matrix $\widehat{\mathbf{L}}_{\nu\omega}$ can be computed from image measurements and a rough approximation of camera intrinsics parameters $\widehat{\mathbf{m}}_i = \widehat{\mathbf{A}}^{-1}\mathbf{p}_i$ (where $\widehat{\mathbf{m}}_i = (\widehat{x}_i, \widehat{y}_i, 1)$). On the other hand, the matrix $\widehat{\mathbf{L}}_\nu$ cannot be easily computed since each depth Z_i is

unknown. For this reason, the matrix used in the control law is:

$$\widehat{\mathbf{L}}_{\nu}^{-1} = -\widehat{\overline{Z}} \begin{bmatrix} 1 & 0 & \widehat{\overline{x}} \\ 0 & 1 & \widehat{\overline{y}} \\ 0 & 0 & 1 \end{bmatrix}$$

where only an "average" depth parameter $\overline{Z}$ must be guessed. Similarly to standard 2 1/2 D visual servoing there is only one global parameter to be approximated and not all the depths Z_i relative to each point of the contour. Note that $\mathbf{L}\ \widehat{\mathbf{L}}^{-1} = \mathbf{I}$ if and only if $Z_i = Z\ \forall i$. Thus, even in the absence of calibration errors, the convergence of the task function to zero will not be exactly decoupled. However, the control law is locally asymptotically stable as it is proven in the next section.

5 Stability Analysis

In this section we prove the local stability of the control law. The local stability is extremely important since the trajectory of the contour can be planned using the method described in (Mezouar and Chaumette, 2000). Thus, only small displacements can be considered at each iteration of the control law.

Plugging the control law (14) into equation (12) we obtain the following

closed-loop differential equation:

$$\dot{\mathbf{e}} = -\mathbf{L}\,\widehat{\mathbf{L}}^{-1} \begin{bmatrix} \lambda_\nu \mathbf{I} & 0 \\ 0 & \lambda_\omega \mathbf{I} \end{bmatrix} \widehat{\mathbf{e}} \tag{15}$$

which can be separated into two different loops:

$$\begin{bmatrix} \dot{\mathbf{e}}_\nu \\ \dot{\mathbf{e}}_\omega \end{bmatrix} = \begin{bmatrix} -\lambda_\nu \mathbf{L}_\nu \widehat{\mathbf{L}}_\nu^{-1} \widehat{\mathbf{e}}_\nu + \lambda_\omega \left(\mathbf{L}_{\nu\omega} - \mathbf{L}_\nu \widehat{\mathbf{L}}_\nu^{-1} \widehat{\mathbf{L}}_{\nu\omega} \right) \widehat{\mathbf{e}}_\omega \\ -\lambda_\omega \mathbf{L}_\omega \widehat{\mathbf{e}}_\omega \end{bmatrix}$$

The control loop of the rotation is decoupled from the control loop of the translation. Thus, the stability can be analyzed in two steps.

5.1 Stability and robustness of the rotation control

The proof of the stability and robustness of the rotation control is exactly the same given in (Malis and Chaumette, 2002). The estimated task function $\widehat{\mathbf{e}}_\omega$ can be written as a function of the true task function $\mathbf{e}_\omega$:

$$\widehat{\mathbf{e}}_\omega = \mathbf{E}_\omega \mathbf{e}_\omega = \mu \tilde{\mathbf{A}} \mathbf{e}_\omega \tag{16}$$

where $\mu = 1/\|\tilde{\mathbf{A}}\mathbf{u}\|$ and $\tilde{\mathbf{A}} = \widehat{\mathbf{A}}^{-1}\mathbf{A}$ is the following upper triangular matrix:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \frac{f}{\widehat{f}} & \frac{1}{\widehat{f}}\left(s - \widehat{s}\frac{fr}{\widehat{f}\widehat{r}}\right) & \frac{u-\widehat{u}_0}{\widehat{f}} - \frac{\widehat{s}\,(v-\widehat{v}_0)}{\widehat{f}^2\widehat{r}} \\ 0 & \frac{fr}{\widehat{f}\widehat{r}} & \frac{v-\widehat{v}_0}{\widehat{f}} \\ 0 & 0 & 1 \end{bmatrix}$$

The closed loop equation for the rotational subsystem is thus:

$$\dot{\mathbf{e}}_\omega = -\lambda_\omega \mu \mathbf{L}_\omega \tilde{\mathbf{A}} \mathbf{e}_\omega \tag{17}$$

The stability of this system can be analyzed independently on $\mathbf{e}_\nu$ and its robustness domain have been already analyzed in (Malis et al., 1999). The equilibrium point $\mathbf{e}_\omega = 0$ of the differential system (17) is locally asymptotically stable *if and only if* $\tilde{\mathbf{A}}$ has eigenvalues with positive real part. The equilibrium point is globally asymptotically stable if $\tilde{\mathbf{A}} > 0$. In that case, $\|\mathbf{e}_\omega\|$ decreases at each iteration of the control law. Since $\mu \geq 1/\|\mathbf{A}\|$, the solution of the differential equation (17) can be bounded as follow:

$$\|\mathbf{e}_\omega(t)\| \leq \|\mathbf{e}_\omega(0)\| e^{-\lambda'_\omega t} \tag{18}$$

where $\lambda'_\omega = \lambda_\omega \sigma / \|\tilde{\mathbf{A}}\|$, σ being the unknown minimum singular value of $\frac{1}{2}(\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T)$. Thus, $\|\mathbf{e}_\omega(t)\|$ will converge exponentially to zero. Finally, since $\mathbf{e}_\omega(t)$ is completely defined by the differential system (17) and by its initial condition $\mathbf{e}_\omega(0)$, we can compute $\widehat{\mathbf{e}}_\omega(t)$ from equation (16) and plug it in equation (15) obtaining a new closed-loop system for $\mathbf{e}_\nu$.

5.2 Stability and robustness of the translation control

The estimated task function $\widehat{\mathbf{e}}_\nu$ can be written as a function of the true task function $\mathbf{e}_\nu$:

$$\widehat{\mathbf{e}}_\nu = \mathbf{E}_\nu \mathbf{e}_\nu \tag{19}$$

where:

$$\mathbf{E}_\nu = \begin{bmatrix} \frac{f}{\widehat{f}} & \frac{1}{\widehat{f}}\left(s - \widehat{s}\frac{fr}{\widehat{f}\widehat{r}}\right) & 0 \\ 0 & \frac{fr}{\widehat{f}\widehat{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The new closed-loop equation for $\mathbf{e}_\nu$ is the sum of two nonlinear terms:

$$\dot{\mathbf{e}}_\nu = -\lambda_\nu \mathbf{Q}(\mathbf{e}_\nu)\mathbf{e}_\nu + \lambda_\omega \mathbf{f}(\mathbf{e}_\nu, t) \tag{20}$$

where:

$$\mathbf{Q} = \mathbf{L}_\nu \widehat{\mathbf{L}}_\nu^{-1} \mathbf{E}_\nu \tag{21}$$

$$\mathbf{f} = \left(\mathbf{L}_{\nu\omega} - \mathbf{L}_{\nu}\widehat{\mathbf{L}}_{\nu}^{-1}\widehat{\mathbf{L}}_{\nu\omega}\right)\widehat{\mathbf{e}}_{\omega}(t) \tag{22}$$

Equation (20) is a differential equation where $\mathbf{e}_{\nu}$ has to be determined and $\widehat{\mathbf{e}}_{\omega}(t)$ is a function of time. Consider the system linearized around $\mathbf{e}_{\nu} = 0$, $\mathbf{e}_{\omega} = 0$ (i.e. $\mathbf{m}_e = \mathbf{m}_e^*$):

$$\dot{\mathbf{e}}_{\nu} = -\lambda_{\nu}\mathbf{Q}(0)\mathbf{e}_{\nu} + \lambda_{\omega}\mathbf{f}(0,t) \tag{23}$$

where:

$$\mathbf{Q}(0) = \frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix} \frac{f}{\widehat{f}}\frac{\widehat{\overline{Z}^*}}{Z_i^*} & \frac{1}{\widehat{f}}\left(s - \widehat{s}\frac{f\,r}{\widehat{f}\widehat{r}}\right)\frac{\widehat{\overline{Z}^*}}{Z_i^*} & \frac{\widehat{\overline{Z}^*}}{Z_i^*}(\widehat{\overline{x}^*} - x_i^*) \\ 0 & \frac{f\,r}{\widehat{f}\widehat{r}}\frac{\widehat{\overline{Z}^*}}{Z_i^*} & \frac{\widehat{\overline{Z}^*}}{Z_i^*}(\widehat{\overline{y}^*} - y_i^*) \\ 0 & 0 & \frac{\widehat{\overline{Z}^*}}{Z_i^*} \end{bmatrix}$$

is a non-singular triangular matrix and the function $\mathbf{f}(0,t)$ is a decreasing function:

$$\lim_{t\to\infty}\mathbf{f}(0,t) = 0$$

The equilibrium point $\mathbf{e}_{\nu} = 0$ of the differential system (23) is locally asymptotically stable *if and only if* $\mathbf{Q}(0)$ has eigenvalues with positive real part.

The eigenvalues of $\mathbf{Q}(0)$ are:

$$\lambda_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{\overline{Z}}{Z_i} \tag{24}$$

$$\lambda_2 = \frac{f}{\widehat{f}}\frac{1}{n}\sum_{i=1}^{n}\frac{\overline{Z}}{Z_i} \tag{25}$$

$$\lambda_3 = \frac{fr}{\widehat{f}\widehat{r}}\frac{1}{n}\sum_{i=1}^{n}\frac{\overline{Z}}{Z_i} \tag{26}$$

Since $Z_i > 0\ \forall i$, $\lambda_1 > 0$ if and only if $\overline{Z} > 0$. Thus, we can theoretically use any positive scalar $\overline{Z}$ in the control law. Since $f > 0$, $\lambda_2 > 0$ if and only if $\widehat{f} > 0$ and $\lambda_3 > 0$ if and only if $\widehat{r} > 0$. Again, we can use any positive approximation of the camera intrinsic parameters in the control law.

A sufficient condition for the asymptotic stability of the system is $\mathbf{Q}(\mathbf{e}_\nu) > 0\ \forall \mathbf{e}_\nu \neq 0$. This condition is equivalent to $\tilde{\mathbf{A}} > 0$ and $\det(\mathbf{Q} + \mathbf{Q}^\top) > 0$. The sufficient condition $\tilde{\mathbf{A}} > 0$ is the same for the rotational control and it depends only on camera intrinsic parameters. On the other hand, the sufficient condition $\det(\mathbf{Q}+\mathbf{Q}^\top) > 0$ depends also on $\mathbf{e}_\nu$. Thus, the condition $\det(\mathbf{Q} + \mathbf{Q}^\top) > 0$ define an open set $\mathcal{S}$ containing $\mathbf{e}_\nu = 0$ such that for any starting point $\mathbf{e}_\nu(0) \in \mathcal{S}$ the system is asymptotically stable.

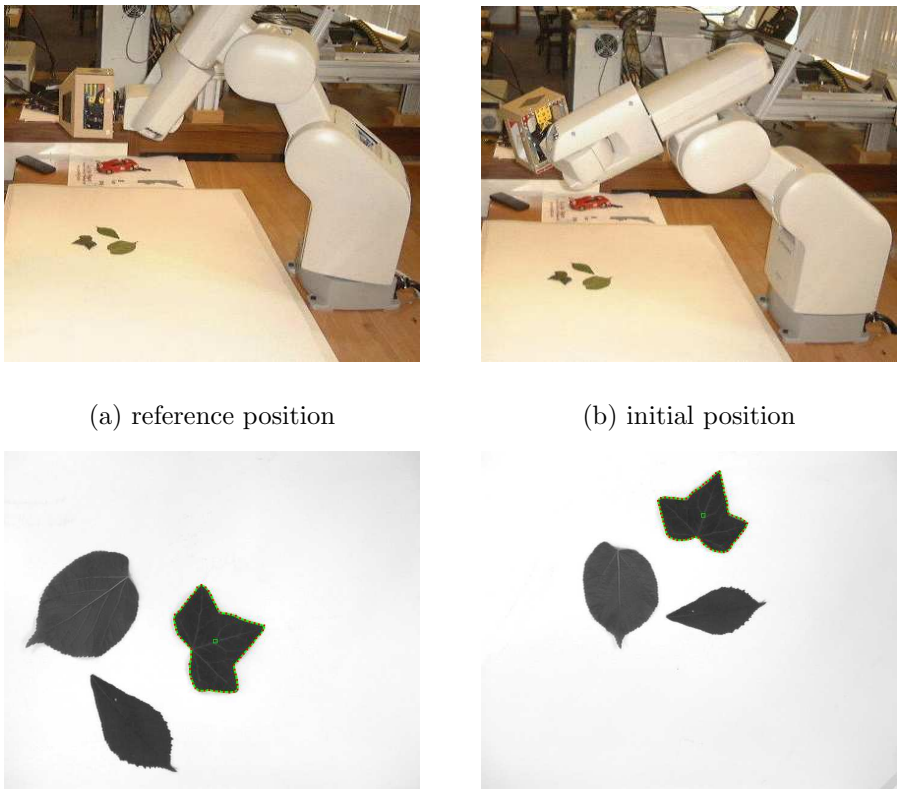
6 Experimental results

In our experiments we have used a Mitsubishi robot RV-E2 Movemaster with 6 degrees of freedom (see Figure 5). 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. The aim of the first experiment is to test the system when the initial displacement of the camera is large. The second experiment tests the system in a situation very close to a real industrial application.

6.1 Positioning with respect to complex unknown contours

Visual servoing experiments have been realized using the complex non smooth contours presented in the paper. The system has also been successfully tested on different smooth contours with generic shapes (see Extension 2). Figure 5 shows an experiment using leaves. Several experiments have been done, using different leaves as reference target, obtaining similar results. In this experiment, the selected leaf in the reference view (Figure 5(c)) is ivy. After the robot and/or the leaves have been moved away from the reference position, the system is able to select the right leaf during the matching step.

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.



(a) reference position (b) initial position

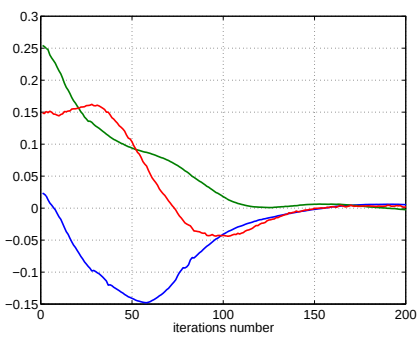
(c) reference view (d) initial view

Figure 5: Experimental results with a leaf. The contour is not smooth and the system is uncalibrated.

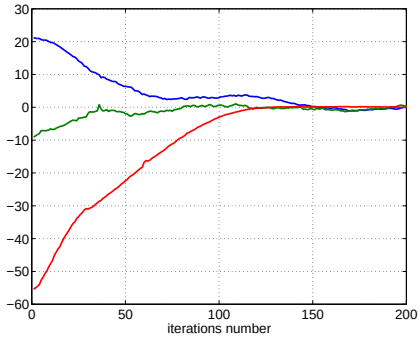
Figure 6(a) and 6(b) show the behavior of the extended image coordinates and orientation of the camera, while Figure 6(c) and 6(d) show the translational and rotational velocity (i.e. the control law). Note that to compute

the control law we do not use any knowledge on the depths of the points belonging to the contour. The control law is stable and the error converges to zero but not exponentially (as it should in ideal conditions) since the system is uncalibrated. Furthermore, the center 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 is stopped when the error between all corresponding points on the contour is less than 0.5 pixel.

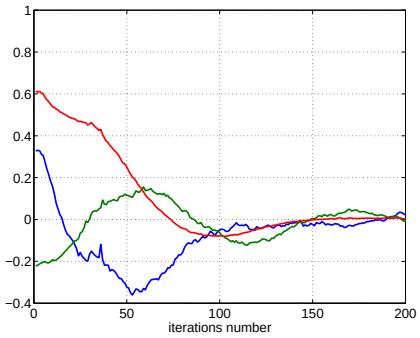
The experimental result is obtained with a rotation $r_x = 20°$ and $r_x = 10°$ since the robot workspace is limited (see Figures 5 (a) and (b)). However, the induced projective distortion is large enough to prove that the visual tracking proposed in (Drummond and Cipolla, 2000) is working correctly. Indeed, an affine tracking without any projective compensation would fail in this case. Note also that in the reference position the image plane is nearly parallel to the contour plane. This corresponding to a singular position for any affine controller. Thus, this experiment is a meaningful test case to prove both the validity of the full projective tracking in (Drummond and Cipolla, 2000) and the improvements provided by the 2 1/2 D visual servoing method with respect to the affine controller proposed in (Drummond and Cipolla, 2000).



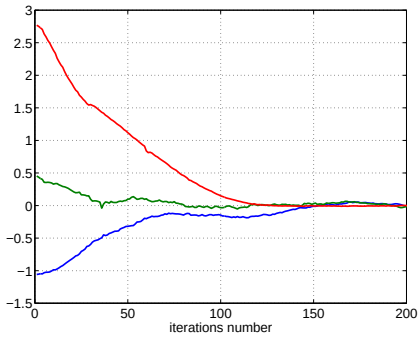
(a) extended image coordinates



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



(c) translation velocity (cm/s)

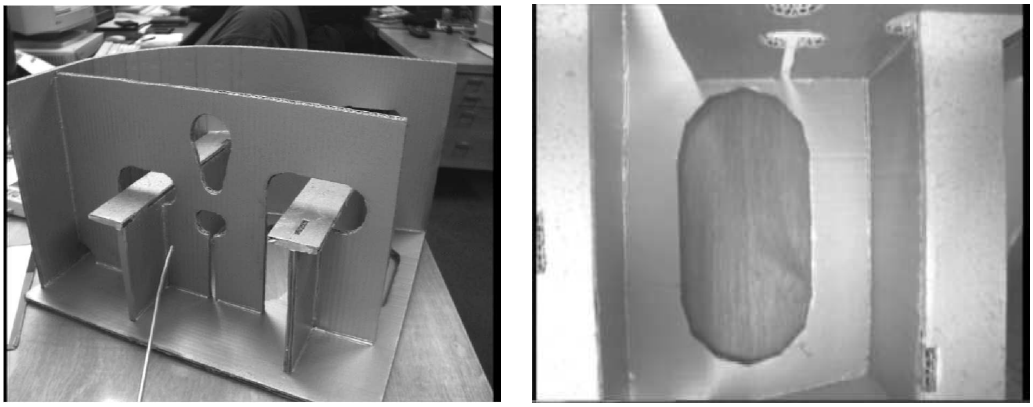


(d) rotation velocity (deg/s)

Figure 6: Experimental results with a leaf.

6.2 Application to ship building

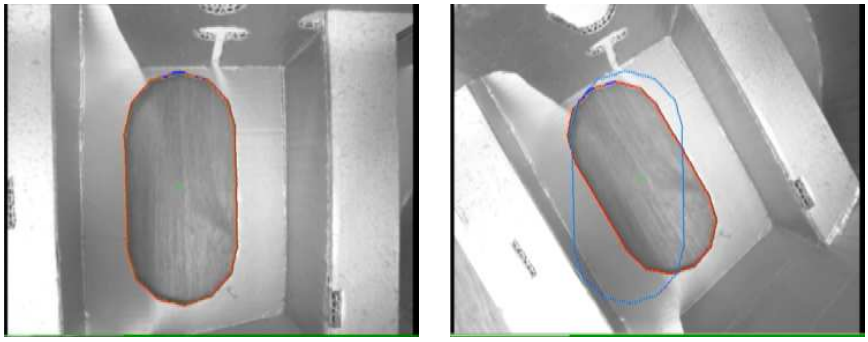
The work presented in the paper has been funded by the Long Term Research European Project VIGOR (Visually Guided Robots using Uncalibrated Cameras). The goal of the project was to remove the calibration bottleneck in visually-guided robots, in order to perform industrial tasks with minimal modeling. One of the area concerned by the project is the shipbuilding industry. Today, the robot-based shipbuilding technology is limited in that robots are exclusively programmed off-line. When in the actual environment, the CAD-generated trajectories have to be computed depending on the exact locations and orientations of all the relevant parts in the workspace, for example, to perform a welding task on the hull. With the proposed approach the relevant trajectories of the robot are automatically transformed on line using the visual input. The experiment illustrated here demonstrates the feasibility of vision-based control with neither prior camera calibration nor hand-eye calibration and without prior knowledge of the precise location of the robot with respect to an unknown and complex contour on the workpiece. Figure 7 shows the front view (a) and a bird's eye view (b) of a mockup of the ship. Several closed contours are visible and the approach presented in the paper can be used to position a welding torch with respect to a ship part. Consider for example the image in Figure 7(b) as the reference image. The



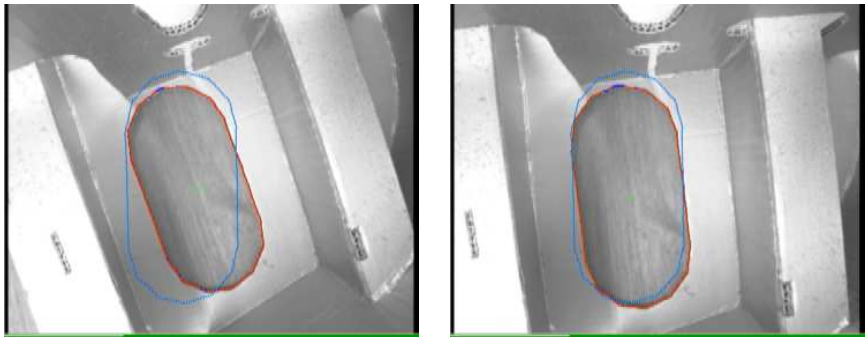
(a) Front view (b) Bird's eye view

Figure 7: Mock-up of the ship used in the experiment.

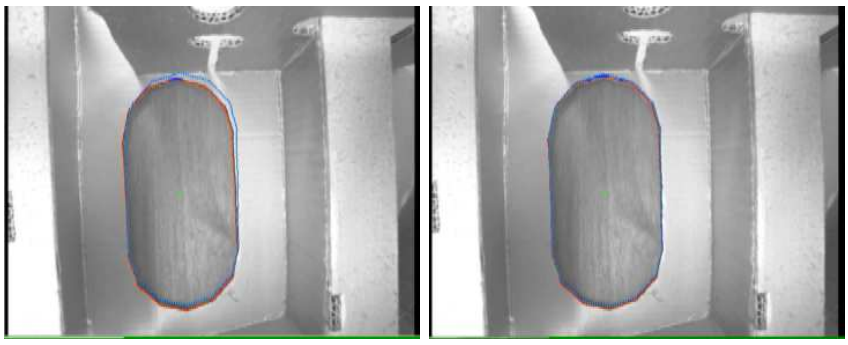
reference contour is extracted from the reference image (see Figure 8(a)). The robot is displaced to its initial position. As it was shown in the previous experiment, the proposed visual servoing approach is able to converge from any initial position. However, we present here the results obtained for a small displacement. Indeed, in order to meet real-time requirements (sample rate is $50ms$), the sequence acquired by the camera is stored on disk only at the end of the servoing. Thus, the total amount of available memory is limited. From the initial image, the system extracts the initial contour (the red contour in Figure 8(b)) which is automatically matched with the reference contour (the blue contour in Figure 8(b)). Figures 8(b)-(f) show the behavior of the current contour during the servoing. At the convergence, Figure 8(f), the current and reference contours are superimposed. The complete sequence of the experiment is available in Extension 3.



(a) Reference image (b) Initial image



(c) Image I (d) Image II



(e) Image III (f) Final image

Figure 8: Experimental results with the mock-up of the ship (see also Extension 3)

7 Conclusion

In this paper, we have presented a complete visual servoing system for positioning a camera with respect to a planar contour. All the steps necessary to visual servoing have been addressed: segmentation, matching, tracking and servoing. The vision controller is based on a 2 1/2 D visual servoing technique. It exploits the homography between the initial and the reference view of the contour, which is found by the matching algorithm. The results show that the system is able to match contours with complex shapes for which invariants cannot be used. Moreover, the selected object is recognized 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 good precision even for unmarked complex planar objects.

8 Acknowledgments

The work was supported by an EC (ESPRIT) grant no. LTR26247 (VIGOR). We would thank Tom Drummond for providing the tracking software.

Appendix: Index to multi-media Extensions

The multi-media extensions to this article can be found online by following the hyperlinks from www.ijrr.org.

Extension	Media type	Description
1	video	Tracking of a contour robust to occlusion
2	video	Visual servoing using an unknown smooth contour
3	video	Visual servoing using a contour on the ship mockup

References

Basri, R., Rivlin, E., and Shimshoni, I. (1998). Visual homing: Surfing on the epipoles. In *IEEE Int. Conf. on Computer Vision*, pages 863–869, Bombay, India.

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

Chesi, G., Malis, E., and Cipolla, R. (1999). Collineation estimation from two unmatched views of an unknown planar contour for visual servoing. In *British Machine Vision Conference*, volume 1, pages 224-233, Nottingham, United Kingdom, September.

Chesi, G., Malis, E., and Cipolla, R. (2000). Automatic segmentation and matching of planar contours for visual servoing. In *International Conference on Robotics and Automation*, volume 3, pages 2753-2758, San Francisco, USA, April.

Collewet, C. and Chaumette, F. (2000). A contour approach for image-based control on objects with complex shape. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 751-756, Takamatsu, Japan.

Drummond, T. and Cipolla, R. (2000). Application of Lie Algebras to Visual Servoing. In *Int. Journal of Computer Vision*, 37(1):21–41, 2000.

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326.

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

Hashimoto, K. (1993). *Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback*, vol. 7 of *World Scientific Series in Robotics and Automated Systems*. World Scientific Press, Singapore.

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

Malis, E. and Chaumette, F. (To appear, 2002). Theoretical improvements in the

stability analysis of a new class of model-free visual servoing methods. *IEEE Transaction on Robotics and Automation.*

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

Mezouar, Y. and Chaumette, F. (2000). Path planning in image space for robust visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2759–2764, San Francisco, CA.

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

Rothwell, C. (1995). *Object recognition through invariant indexing.* Oxford Science Publications.

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

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

Wilson, W. J., Hulls, C. C. W., and Bell, G. S. (1996). Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696.