

# Using Robust Estimation for Visual Servoing Based on Dynamic Vision

Christophe Collewet and François Chaumette

**Abstract**—The aim of this article is to achieve accurate visual servoing tasks when the shape of the object being observed as well as the final image are unknown. More precisely, we want to control the orientation of the tangent plane at a certain point on the object corresponding to the center of a region of interest and to move this point to the principal point to fulfill a fixation task. To do that, we perform a 3D reconstruction phase during the servoing. It is based on the measurement of the 2D displacement in the region of interest and on the measurement of the camera velocity. Since the 2D displacement depends on the scene, we introduce an unified motion model to deal with planar as well with non-planar objects. Unfortunately, this model is only an approximation. In [1], we propose to use active vision to enlarge its domain of validity and a 3D reconstruction based on a continuous approach. In this paper, we propose to use robust estimation techniques and a 3D reconstruction based on discrete approach. Experimental results compare both approaches.

## I. INTRODUCTION

Visual servoing is now a widely used technique in robot control [2]. It allows to perform robotic tasks from visual features acquired by a camera. However, synthesizing the control law requires usually a model of the scene observed by the camera and also the knowledge of the desired features. In some cases it is not possible, let us cite applications for example in surgical domain [3] or when natural scenes are involved [4]. Classic image-based approach [5] cannot actually cope with such applications since the control law is based on the knowledge of desired visual features. Let us point out that the knowledge of the object is also needed to compute the interaction matrix and, therefore, the control law. It is also required for position-based approaches [6] to compute the relative pose of the camera w.r.t. the object. To avoid this a priori knowledge, some authors propose model-free approaches [7], [8] but note that the desired visual features are needed. Other works [9], [10], based on dynamic visual features, do not require the desired image but require a model of the object. Besides, note that, for now, these works are only dedicated to planar objects.

This work can be seen as an improvement of our previous work [11] where an approach restricted to planar objects had been proposed. Here we propose an approach to cope with planar or non-planar objects leading to an unified displacement model. However, since this model is only an approximation, we show how to compute it robustly. Since the shape of the object as well as the desired image are unknown, as in [11], a 3D reconstruction phase by dynamic vision is needed.

C. Collewet and F. Chaumette are with INRIA/IRISA, Campus de Beaulieu, 35042 Rennes cedex, France, {christophe.collewet, francois.chaumette}@irisa.fr.

The paper is organized as follows: first, we present in Section II a brief review on previous works relevant to 3D reconstruction by dynamic vision. The reconstruction of the parameters of the tangent plane is detailed in Section III while the way to compute robustly the unified 2D displacement model is presented in Section IV. Finally, experimental results and a comparison with the approach proposed in [1] are presented in Section V.

## II. PREVIOUS WORKS

Numerous papers have been addressing the problem of structure from motion, that is recovering both the 3D structure of an object and the 3D motion of a moving camera observing the considered object. Nevertheless, two basic approaches emerge, continuous approaches and discrete approaches. The former are based on the knowledge of the optical flow while the latter are based on the matching of visual features between two (or more) views. Since we are interested in natural scenes, we consider here only point features but we invite the interested reader to [12] where other features are considered. In addition, we focus on discrete approaches (the continuous approaches are investigated in [1]).

Usually 3D reconstruction approaches are based on the computation of the essential [13] or the fundamental [14] matrices from matched points between two views. However, some degenerate cases may occur, for example when the scene is planar [15]. In this case, specific algorithms have to be carried out. They are based on the computation of the homography matrix [16]. Note that in case of non-planar objects, an approach is proposed in [17] and [18] where an homography is computed from a virtual plane attached to the object. For visual servoing issues, let us point out that the two views that have to be considered are the current and the desired images. In this case, an important problem is to correctly match the points between those frames that can be very different. In addition, recall that in our case the desired frame is unknown. Therefore, we present here a solution based on two consecutive frames. Moreover, this approach also simplifies the matching problem. Note that a solution based on this way to proceed has been already proposed in [11] but it was restricted to planar objects. We will here extend this work.

## III. RECONSTRUCTION OF THE STRUCTURE

Let us consider a point  $P$  described by  $\mathbf{X}_P = (X_P, Y_P, Z_P)$  in the camera frame  $\mathcal{F}_c$ . This point is chosen so that its projection  $p$  described by  $\mathbf{x}_p = (x_p, y_p, 1)$  lies in the center of a region of interest (ROI). Let us consider also a point  $M$

of the object described by  $\mathbf{X} = (X, Y, Z)$  in  $\mathcal{F}_c$ . The tangent plane in  $P$  expresses as follows

$$Z = Z_P + A_{10}(X - X_P) + A_{01}(Y - Y_P) \quad (1)$$

where  $A_{10} = \left. \frac{\partial Z}{\partial X} \right|_P$  and  $A_{01} = \left. \frac{\partial Z}{\partial Y} \right|_P$  leading to the normal  $\mathbf{n}$  in  $P$

$$\mathbf{n} = (A_{10}, A_{01}, -1) \quad (2)$$

which is required to compute the control law (see [11]).

On the other hand, we can rewrite (1) in a more compact form

$$Z = A_{00} + A_{10}X + A_{01}Y \quad (3)$$

where  $A_{00} = -\mathbf{n}^\top \mathbf{X}_P$  such that (3) can be rewritten with respect to the normalized coordinates  $\mathbf{x}$  as follows

$$\frac{1}{Z} = \boldsymbol{\alpha}^\top \mathbf{x} \quad (4)$$

with  $\boldsymbol{\alpha} = (\alpha_{10}, \alpha_{01}, \alpha_{00})$  where  $\alpha_{10} = -A_{10}/A_{00}$ ,  $\alpha_{01} = -A_{01}/A_{00}$  and  $\alpha_{00} = 1/A_{00}$  or as follows by introducing  $u = x - x_p$  and  $v = y - y_p$

$$\frac{1}{Z} = \boldsymbol{\beta}^\top \mathbf{u} \quad (5)$$

with  $\mathbf{u} = (u, v, 1)$  and  $\boldsymbol{\beta} = (\beta_{10}, \beta_{01}, \beta_{00})$  where  $\beta_{10} = \alpha_{10}$ ,  $\beta_{01} = \alpha_{01}$  and  $\beta_{00} = \alpha_{00} + \boldsymbol{\alpha}^\top \mathbf{x}_P$ .

Therefore, if we can measure  $\boldsymbol{\alpha}$  or  $\boldsymbol{\beta}$ , the unit normal  $\tilde{\mathbf{n}}$  in  $P$  can be deduced

$$\tilde{\mathbf{n}} = -\frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|}. \quad (6)$$

On the other hand, we can compute the discrete displacement of a point  $M$  between two frames

$$\mathbf{X}_{k+1} = \mathbf{R}_k \mathbf{X}_k + \mathbf{t}_k \quad (7)$$

where  $\mathbf{R}_k$  is a rotation matrix depending on  $\boldsymbol{\omega}$  and  $\mathbf{t}_k$  a translation vector depending on  $\mathbf{v}$  and  $\boldsymbol{\omega}$  at time  $k$  with  $\mathbf{v} = (v_x, v_y, v_z)$  and  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$  being respectively the translational and rotational components of the camera velocity  $\mathbf{v}$ . Thus, if  $\mathbf{v}$  is measured we have an estimate for  $\mathbf{R}_k$  and  $\mathbf{t}_k$ . On the other hand, by perspective projection we access to an expression of the 2D displacement between  $k$  and  $k+1$

$$\begin{cases} x_{k+1} = \frac{R_{11}x_k + R_{12}y_k + R_{13} + t_x/Z_k}{R_{31}x_k + R_{32}y_k + R_{33} + t_z/Z_k} \\ y_{k+1} = \frac{R_{21}x_k + R_{22}y_k + R_{23} + t_y/Z_k}{R_{31}x_k + R_{32}y_k + R_{33} + t_z/Z_k} \end{cases} \quad (8)$$

To compute  $\boldsymbol{\alpha}$  we assume that the surface in the neighborhood of  $P$  is continuous so that we can write an expression of depths in function of image coordinates

$$\frac{1}{Z} = \sum_{p \geq 0, q \geq 0, p+q=n} \alpha_{pq} x^p y^q \quad (9)$$

where  $n$  is the degree of this polynomial.

Thereafter, if we substitute this expression in (8) we obtain a model of the 2D displacement. Unfortunately, the shape of this model depends on the surface being observed through

the value of  $n$ . To not depend on the object shape, a way to proceed is to consider only a small neighborhood in  $p$  so that the depths given by (9) coincide with the ones given by the tangent plane (5) leading to the following 2D displacement model

$$\begin{cases} x_{k+1} = \frac{R'_{11}u_k + R'_{12}v_k + R'_{13} + \boldsymbol{\beta}^\top \mathbf{u}_k t'_x}{R'_{31}u_k + R'_{32}v_k + 1 + \boldsymbol{\beta}^\top \mathbf{u}_k t'_z} \\ y_{k+1} = \frac{R'_{21}u_k + R'_{22}v_k + R'_{23} + \boldsymbol{\beta}^\top \mathbf{u}_k t'_y}{R'_{31}u_k + R'_{32}v_k + 1 + \boldsymbol{\beta}^\top \mathbf{u}_k t'_z} \end{cases} \quad (10)$$

where we have introduced the matrix  $\mathbf{R}'$  and the vector  $\mathbf{t}' = (t'_x, t'_y, t'_z)$  defined as follows

$$\begin{cases} \mathbf{t}' = \frac{1}{K} \mathbf{t} \\ \mathbf{R}' = \frac{1}{K} \begin{bmatrix} R_{11} & R_{12} & R_{11}x_p + R_{12}y_p + R_{13} \\ R_{21} & R_{22} & R_{21}x_p + R_{22}y_p + R_{23} \\ R_{31} & R_{32} & R_{31}x_p + R_{32}y_p + R_{33} \end{bmatrix} \end{cases} \quad (11)$$

where  $K = R_{31}x_p + R_{32}y_p + R_{33}$

First, let us suppose that  $t'_z \neq 0$ . In this case we can introduce  $\boldsymbol{\gamma} = t'_z \boldsymbol{\beta}$  and  $\mathbf{k} = \mathbf{t}'/t'_z$  which allow us to rewrite (10)

$$\begin{cases} x_{k+1} = \frac{R'_{11}u_k + R'_{12}v_k + R'_{13} + \boldsymbol{\gamma}^\top \mathbf{u}_k k_x}{R'_{31}u_k + R'_{32}v_k + 1 + \boldsymbol{\gamma}^\top \mathbf{u}_k} \\ y_{k+1} = \frac{R'_{21}u_k + R'_{22}v_k + R'_{23} + \boldsymbol{\gamma}^\top \mathbf{u}_k k_y}{R'_{31}u_k + R'_{32}v_k + 1 + \boldsymbol{\gamma}^\top \mathbf{u}_k} \end{cases} \quad (12)$$

where we denote  $\mathbf{k} = (k_x, k_y, 1)$  and  $\boldsymbol{\gamma} = (\gamma_x, \gamma_y, \gamma_z)$ .

On the other hand, since (5) is only true locally, we have to perform a Taylor series expansion of (12) in  $p$  leading to the following unified 2D displacement model

$$\mathbf{x}_{k+1} = \mathbf{M} \mathbf{u}_k \quad (13)$$

where

$$\begin{cases} M_{11} = \frac{\zeta_{11} + (R'_{11} - k_x R'_{31})\gamma_z + (k_x - R'_{13})\gamma_x}{\xi_z^2} \\ M_{12} = \frac{\zeta_{12} + (R'_{12} - k_x R'_{32})\gamma_z + (k_x - R'_{13})\gamma_y}{\xi_z^2} \\ M_{13} = \frac{R'_{13} + (k_x + R'_{13})\gamma_z + k_x \gamma_z^2}{\xi_z^2} \\ M_{21} = \frac{\zeta_{21} + (R'_{21} - k_y R'_{31})\gamma_z + (k_y - R'_{23})\gamma_x}{\xi_z^2} \\ M_{22} = \frac{\zeta_{22} + (R'_{22} - k_y R'_{32})\gamma_z + (k_y - R'_{23})\gamma_y}{\xi_z^2} \\ M_{23} = \frac{R'_{23} + (k_y + R'_{23})\gamma_z + k_y \gamma_z^2}{\xi_z^2} \end{cases} \quad (14)$$

with  $\zeta_{ij} = R'_{ij} - R'_{i3}R'_{3j}$  and  $\xi_z = 1 + \gamma_z$ .

Throughout this paper, we refer to this displacement model (13) as *the unified displacement model* since it can cope with planar or non-planar objects.

Consequently, if we estimate the parameters of this motion model (see Section IV) and if the 3D velocity is supposed to

be known, an estimation  $\hat{\gamma}$  of  $\gamma$  can be obtained by solving the following system

$$\begin{cases} M_{11}\hat{\gamma}_z^2 + (2M_{11} + k_x R'_{31} - R'_{11})\hat{\gamma}_z - K_x \hat{\gamma}_x + \Gamma_1 = 0 \\ M_{12}\hat{\gamma}_z^2 + (2M_{12} + k_x R'_{32} - R'_{12})\hat{\gamma}_z - K_x \hat{\gamma}_y + \Gamma_2 = 0 \\ (M_{13} - k_x)\hat{\gamma}_z^2 + (2M_{13} - k_x - R'_{13})\hat{\gamma}_z + \Gamma_3 = 0 \\ M_{21}\hat{\gamma}_z^2 + (2M_{21} + k_y R'_{31} - R'_{21})\hat{\gamma}_z - K_y \hat{\gamma}_x + \Gamma_4 = 0 \\ M_{22}\hat{\gamma}_z^2 + (2M_{22} + k_y R'_{32} - R'_{22})\hat{\gamma}_z - K_y \hat{\gamma}_y + \Gamma_5 = 0 \\ (M_{23} - k_y)\hat{\gamma}_z^2 + (2M_{23} - k_y - w_{23})\hat{\gamma}_z + \Gamma_6 = 0 \end{cases} \quad (15)$$

where  $K_x = k_x - R'_{13}$ ,  $K_y = k_y - R'_{23}$  and where  $\Gamma_i$  are the components of  $\Gamma$  defined as follows

$$\Gamma = \begin{bmatrix} M_{11} - \zeta_{11} \\ M_{12} - \zeta_{12} \\ M_{13} - R'_{13} \\ M_{21} - \zeta_{21} \\ M_{22} - \zeta_{22} \\ M_{23} - R'_{23} \end{bmatrix}. \quad (16)$$

More precisely, we rewrite (15) in a least-squares sense (since we have 6 equations and 3 unknowns) leading finally to a 3rd order polynomial in  $\hat{\gamma}_z$ .

In the case where  $t'_z = 0$  the unified displacement model becomes much simpler

$$\begin{cases} x_{k+1} = R'_{13} + t'_x \beta_{00} + (\zeta_{11} - t'_x R'_{31} \beta_{00} + t'_x \beta_{10}) u_k \\ \quad + (\zeta_{12} - t'_x R'_{32} \beta_{00} + t'_x \beta_{01}) v_k \\ y_{k+1} = R'_{23} + t'_y \beta_{00} + (\zeta_{21} - t'_y R'_{31} \beta_{00} + t'_y \beta_{10}) u_k \\ \quad + (\zeta_{22} - t'_y R'_{32} \beta_{00} + t'_y \beta_{01}) v_k \end{cases} \quad (17)$$

and leads to a linear form with respect to  $\beta$

$$\mathbf{C}\beta = \Gamma \quad (18)$$

with

$$\mathbf{C} = \begin{bmatrix} t_x & 0 & -t_x R'_{31} \\ 0 & t_x & -t_x R'_{32} \\ 0 & 0 & t_x \\ t_y & 0 & -t_y R'_{31} \\ 0 & t_y & -t_y R'_{32} \\ 0 & 0 & t_y \end{bmatrix} \quad (19)$$

leading to the following solution

$$\hat{\beta} = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \Gamma. \quad (20)$$

As recovering the structure requires the knowledge of the 2D displacement between two consecutive frames, we focus in the next section on the way to compute it.

#### IV. ESTIMATION OF THE FRAME-TO-FRAME DISPLACEMENT

The classic approach [19] is based on the minimization of the following criterion

$$J(\boldsymbol{\mu}) = \sum_{m \in W} (f(m) - g(\boldsymbol{\delta}(m, \boldsymbol{\mu})))^2 \quad (21)$$

where  $f$  and  $g$  denote the intensity in  $m$  of two consecutive frames,  $\boldsymbol{\delta}$  describes the unified displacement model parametrized by the vector  $\boldsymbol{\mu}$  (given by (14) or (17)) and  $W$  the windows of interest centered in  $p$ .

However, since (13) or (17) is only an approximation of the true displacement model, if we want to obtain an accurate value for  $\boldsymbol{\mu}$  we have to choose carefully the points  $m$  involved in (21). Indeed, we want to select those which fit the true displacement model.

In the general case of non-planar objects it is difficult to study the difference  $\mathbf{E}(\mathbf{u})$  between the approximated 2D displacement and the true one. Therefore, we assume that the true displacement model coincides with its second order Taylor series expansion and compute the difference  $\epsilon(\mathbf{u})$  between this model and the unified displacement model. In fact, we are interested to point out the locus  $\mathcal{L}(\mathbf{u})$  when  $\epsilon(\mathbf{u})$  is small, since in that case  $\mathbf{E}(\mathbf{u})$  will be also small. When the 3D rotations  $\mathbf{R}$  between two consecutive frames are small, one can show that  $\epsilon(\mathbf{u})$  can be expressed as follows

$$\epsilon(\mathbf{u}) = \epsilon_{planar}(\mathbf{u}) + \epsilon_{non-planar}(\mathbf{u}) \quad (22)$$

where

$$\epsilon_{planar}(\mathbf{u}) = \frac{t'_z (u\beta_{10} + v\beta_{01})}{\xi_z^3} \begin{bmatrix} u(a\beta_{10} - \xi_z) + v\beta_{01}a \\ b\beta_{10}u + v(\beta_{01}b - \xi_z) \end{bmatrix} \quad (23)$$

with  $a = t'_z R'_{13} - t'_x$ ,  $b = t'_z R'_{23} - t'_y$  and

$$\epsilon_{non-planar}(\mathbf{u}) = \frac{e_\beta(\mathbf{u})}{\xi_z^2} (t'_x, t'_y) \quad (24)$$

where  $e_\beta(\mathbf{u})$  describes the second order terms of  $1/Z$  in  $p$

$$e_\beta(\mathbf{u}) = \frac{1}{2}\beta_{20}u^2 + \beta_{11}uv + \frac{1}{2}\beta_{02}v^2 \quad (25)$$

such that

$$\frac{1}{Z} = \boldsymbol{\beta}^\top \mathbf{u} + e_\beta(\mathbf{u}). \quad (26)$$

If we know that the camera is observing a planar object (i.e.  $e_\beta(\mathbf{u}) = 0$ ), then (23) shows that the locus where  $\epsilon(\mathbf{u}) = 0$  is simply the following straight line

$$\mathcal{L}(\mathbf{u}) : u\beta_{10} + v\beta_{01} = 0. \quad (27)$$

In the general case of an unknown object, vanishing  $\epsilon(\mathbf{u})$  is more difficult since  $u\beta_{10} + v\beta_{01}$  is not necessary factorized in  $\epsilon_{non-planar}(\mathbf{u})$ . However, in practice,  $\|\epsilon_{planar}(\mathbf{u})\| \ll \|\epsilon_{non-planar}(\mathbf{u})\|$ . Thus, we have  $\epsilon(\mathbf{u}) \approx \epsilon_{non-planar}(\mathbf{u})$  and  $\epsilon(\mathbf{u})$  becomes proportional to  $e_\beta(\mathbf{u})$ . Therefore, we are interested to identify the locus  $\mathcal{L}(\mathbf{u})$  where  $e_\beta(\mathbf{u})$  is low.

To recover  $\mathcal{L}(\mathbf{u})$ , we are interested in the isocontours  $e_\beta(\mathbf{u}) = c$  where  $c$  is a low value. To study them, we reduce the quadratic form (25) by expressing it in the frame of the eigenvectors of the matrix associated with (25). Let  $s_1$  and  $s_2$  be the eigenvalues of this matrix, thus it is possible to rewrite (25) in the new frame as follows

$$s_1 u'^2 + s_2 v'^2 - c = 0 \quad (28)$$

Three cases occur, the worst case appears when (28) describes an ellipse, i.e. when  $s_1 s_2 > 0$ . If  $\text{sign}(s_1 s_2) \neq$

sign( $c$ ),  $\mathcal{L}(\mathbf{u})$  is empty since (28) describes an imaginary ellipse. Otherwise (25) will vanish only if  $\mathbf{u} = \mathbf{0}$ . However, note that the locus around the major axis will lead to lower errors than around the minor axis (since in that direction  $e_\beta(\mathbf{u})$  increases faster). When the conic is an hyperbola, i.e. when  $s_1 s_2 < 0$ ,  $\mathcal{L}(\mathbf{u})$  is obtained for  $c = 0$ . In that case, the hyperbolas degenerate into two straight lines

$$\mathcal{L}(\mathbf{u}) : \left\{ v' + \sqrt{-s_1/s_2} u' \right\} \cup \left\{ v' - \sqrt{-s_1/s_2} u' \right\}. \quad (29)$$

This case is very interesting since there is an infinite number of couples  $(u, v)$  for which (25) vanishes even far from  $p$ . The last case, also interesting, appears when the conic is a parabola, i.e. whenever  $s_1$  or  $s_2$  is null since there is also a locus where (25) vanishes even if  $m$  is far from  $p$ . This locus is either the  $u'$  or the  $v'$  axis.

To illustrate these theoretical issues, a simulation has been carried out (see Fig. 1). The object is an hyperboloid of one sheet described by  $(X/R)^2 - (Y/2R)^2 + (Z/2R)^2 = 1$  with  $R = 5$  cm. The orientation between the camera and the tangent plane was  $\Phi = (-22^\circ, 25^\circ)$  (respectively pitch, yaw)<sup>1</sup>, the depth in P was  $Z_p = 61$  cm. Fig. 1 represents  $\|\mathbf{E}(\mathbf{u})\|$  (considered equal to  $\|\epsilon(\mathbf{u})\|$ ). We clearly see on Fig. 1 that (25) describes here hyperbolas. We also guess the two straight lines  $\mathcal{L}(\mathbf{u})$  given by (29) in Fig. 1.

Thus, except the case where (25) describes an ellipse, we have seen that even far from the center of the ROI there is always a locus where the error between the true 2D displacement and the approximated one is very low. Therefore, our idea is to choose a value as higher as possible for  $W$  according to the computation time (in order to not penalizing the dynamics of the end-effector and the stability of the control scheme) and to select the points  $m$  which fits the true 2D displacement while the other are seen as outliers.

This can be done by using statistical techniques of robust estimation like the M-estimators [20]. Such approaches have been shown to be effective in various contexts, see for example [21]–[23]. Other approaches can also be used like the RANSAC algorithm or the Hough transform but since these methods are too much time consuming they have not been considered.

Indeed, it is commonly known that in the case of the ordinary least-squares estimator (as for now in (21)) the probability of occurrence of corrupted data is so small that the estimation of the parameters will be strongly affected when they occur [24]. To cope with this problem, instead of minimizing (21), we prefer to compute  $\hat{\boldsymbol{\mu}}$  as follows

$$\hat{\boldsymbol{\mu}} = \arg \min_{\boldsymbol{\mu}} \sum_{m \in W} \rho(e_m) \quad (30)$$

where the function  $\rho$  is interpreted as the negative logarithm of the probability density of residuals  $e_m = r_m/\sigma$  where  $\sigma$  denotes the variance of residuals  $r_m = f(m) - g(\delta(m, \boldsymbol{\mu}))$  that fits the model. Since the residuals contain outliers, computing  $\sigma$  is not a simple task. To do it robustly, it is

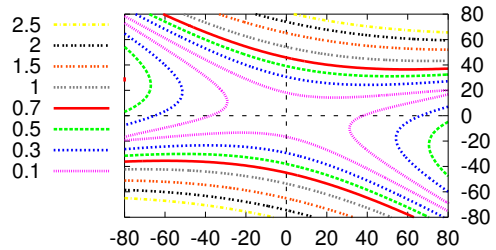


Fig. 1.  $\|\mathbf{E}(\mathbf{u})\|$  vs.  $\mathbf{u}$  (in pixels).

usually done by computing the Median Absolute Deviation (MAD) [25].

Solving (30) can be done by using the so-called Iteratively Reweighted Least-Squares algorithm (IRLS) [20] that moves the M-estimation problem to a weighted least-squares algorithm

$$\sum_m \rho(e_m) = \frac{1}{2} \sum_m w_m e_m^2 \quad (31)$$

with  $w_m = \psi(e_m)/e_m$  (see [20]) where  $\psi(x) = \partial\rho(x)/\partial x$  is called the influence function [24].

Various influence functions are reported in the literature. We used the Tuckey's biweight function as proposed in [20]. Indeed, since this function vanishes when gross errors appear, it ensures that (30) will be not affected by such errors.

## V. EXPERIMENTAL RESULTS

In order to validate our algorithm, we present here some experimental results. We first compare the benefit of using or not robust statistics, then, we compare the continuous approach (presented in [1]) with the discrete one and, finally, the use of active vision (also presented in [1]) with the use of robust statistics.

The experimental system consists of a 6 dof robot with an eye-in-hand CCD camera. Note that the transformation matrix between the end-effector and the camera has been calibrated. In contrast, the intrinsic parameters of the camera are roughly known. The point  $p$  has been chosen from the initial image by the Harris detector and has been tracked by using the algorithm described in [26]. The way to evaluate the orientation  $\Phi$  (respectively pitch, yaw and roll) between the camera frame and a frame attached to the tangent plane in  $P$  has been detailed in [1]. We introduce the following notations concerning the superscript of  $\Phi$ :  $i$  for the initial pose,  $d$  for the desired one and  $m$  for the measured one. The control law is synthesized from  $\alpha$  and is detailed in [11], its goal is to reach the desired orientation (expressed under the form  $\kappa\theta$ , where  $\kappa$  represents the unit rotation axis vector and  $\theta$  the rotation angle around this axis) at a given depth  $Z_p^*$  and to move the point  $p$  to the principal point. To do that, we have introduced a task function  $\mathbf{e}$  that we regulate to zero.

Since the object is motionless, one can improve the accuracy on  $\alpha$ . Indeed, in a fixed frame, one can express a value  $\alpha^f$  that can be filtered since a fixed value has to be obtained. Thereafter, this value is expressed in the camera frame to be used in the control law. Moreover, proceeding

<sup>1</sup>We are not interested in the roll angle.

this way allows to know when  $\alpha^f$  is stable enough to be used in the control law (typically 7 acquisitions are sufficient). Thus, a preliminary phase is required. Since  $\hat{\beta}$  is initially not known, only constant translations are considered (with  $v_z = 0$ , see [1]). They are chosen so that  $p$  will move towards the principal point. Finally, the camera motion consists of three phases, a first phase at constant velocity, a second phase when both reconstruction and servoing are performed, and a last phase where only the servoing operates. This last phase occurs when the mean of the 2D displacement in the second phase is lower than 1/4 pixel.

#### A. Comparison between using or not using robust estimation

The first experiment concerns a sphere of radius 7 cm and consists in positioning the camera parallel to the tangent plane in  $P$  so that  $Z_P^* = 65$  cm. Fig. 2 describes the behavior of the algorithm when using the discrete approach. Fig. 2a depicts the components of the camera velocity to reach the desired pose; Fig. 2b the norm of the task function  $e$ ; Fig. 2c the magnitude of the rotation  $\theta$  to reach the desired orientation; Fig. 2d the behavior of  $\alpha$  (filtered and non-filtered) expressed in a fixed frame. Finally, the initial and final images are reported respectively on Fig. 2e-f (the 4 white dots are used to compute  ${}^m\Phi$  off-line). First, Fig. 2b confirms that the control law converges since  $\|e\|$  tends towards zero. One can also remark on Fig. 2a the three phases of the algorithm, the last phase begins near 20 s when the 2D motion is lower than 1/4 pixel. For this experiment we had  ${}^i\Phi = (-17^\circ, -17.2^\circ, -1.9^\circ)$  and  ${}^iZ_P = 71.5$  cm. The orientation after servoing was  ${}^m\Phi = (0.4^\circ, 0.4^\circ)$  when using robust estimation (recall that we are not interested in the last component of  $\Phi$ ). Without using robust estimation, a higher orientation error has been obtained since we had  ${}^m\Phi = (-4.8^\circ, 2.8^\circ)$ . The benefit of using robust estimation is clear.

#### B. Comparison between the continuous and the discrete approach

Instead of using (13), it is possible to derive a continuous affine motion model which leads to a linear formulation for  $\hat{\beta}$  as detailed in [1]. Both approaches need the knowledge of the frame-to-frame displacement, i.e. the matrix  $M_c$ . We computed this matrix as described in Section IV.

In comparison, we performed the same task as described in Section V-A when using the continuous approach. We obtained worse results since we had  ${}^m\Phi = (-1.9^\circ, 1.3^\circ)$ . This can be easily explained by the acquisition rate which in the case of using robust estimation is low (due to the computation duration of the MAD). Therefore, the assumption of instantaneous displacement as required to derive a continuous approach is no more valid.

#### C. Comparison between the use of active vision and the use of robust estimation

Using a continuous or a discrete approach leads in both cases to an approximated 2D motion model. Whereas we have used here a robust estimation technique to enlarge

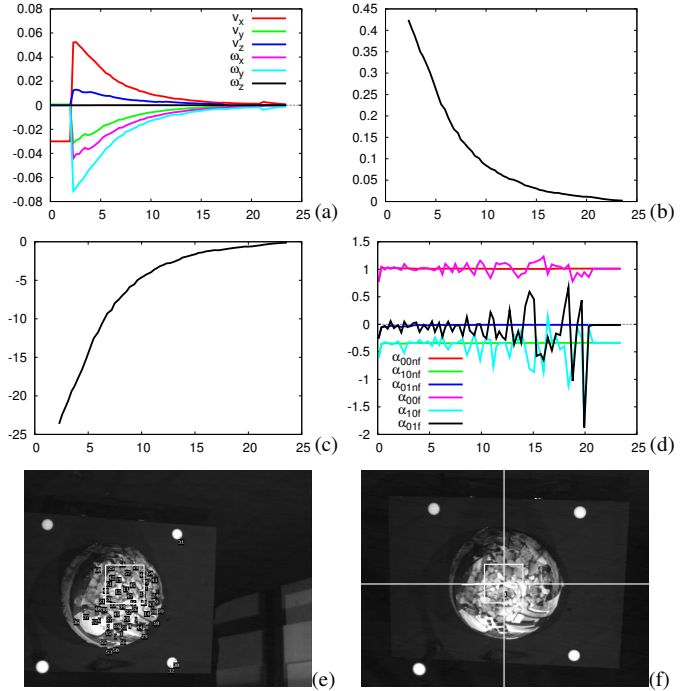


Fig. 2. 1<sup>st</sup> experiment ( $x$  axes in seconds).  ${}^d\Phi = \mathbf{0}$ . (a) Kinematic screw (m/s or rad./s). (b) Error defined as  $\|e\|$ . (c) Magnitude of the rotation  $\theta$  (deg.) (d) Vector  $\alpha$  in a fixed frame (filtered and non-filtered). (e) Initial image. (f) Final image.

the domain of validity of the 2D motion model, in [1] we have used active vision to do it. Remark that, since using active vision is not time consuming, we have used in [1] a continuous approach. Thereafter, we compare here the use of active vision coupled to a continuous approach with the use of robust estimation coupled to a discrete approach.

We performed again the same experiment as in Section V-A but when using active vision. In this case, we obtained also a very low positioning error since we had  ${}^m\Phi = (-0.1^\circ, 0.5^\circ)$ .

The second experiment has been carried out on the same sphere with  ${}^d\Phi = (20^\circ, 20^\circ)$  (see Fig. 3). Fig. 3 depicts the same parameters as in Fig. 2 and confirms that the control law converges without any problem. The initial orientation was  ${}^i\Phi = (4.2^\circ, -5.2^\circ, 3.2^\circ)$  and we had  ${}^iZ_P = 70.9$  cm. We obtained  ${}^m\Phi = (19.5^\circ, 20.3^\circ)$ . As can be seen, this result is very good. Unfortunately, in the case of active vision, we obtained bad results, the orientation error was around  $4^\circ$ .

When dealing with desired orientation  $\Phi \neq 0$  using robust estimation leads clearly to lower positioning errors than when using active vision. The main drawback of using robust estimation is that this technique is time consuming (the update rate is around 400 ms) and therefore penalizes the dynamic behavior of the robot. In contrast, a higher update rate is obtained (around 280 ms) when using active vision. Consequently, higher 3D velocities can be achieved. In addition, in the case of a less complex object, like a cylinder for example, good results can be obtained even when  $\Phi \neq 0$  (see [1]).



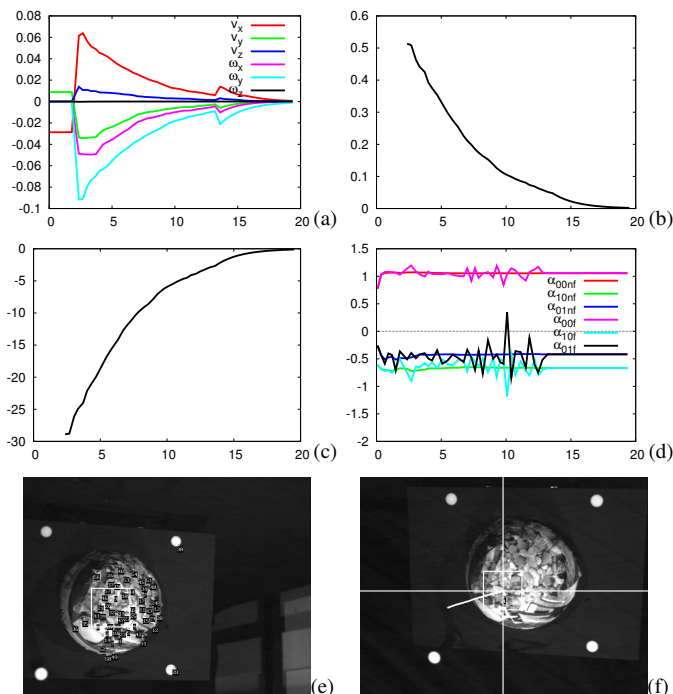


Fig. 3. 2<sup>nd</sup> experiment.  $d\Phi = (20^\circ, 20^\circ)$ .

## VI. CONCLUSION

We have presented in this paper a way to achieve visual servoing tasks when the desired visual features and the shape of the object being observed are unknown. To do that, we recover the parameters of the tangent plane at a certain point of an unknown object that are introduced in a control law to perform a positioning task. Our approach is based on the computation of the 2D displacement between two consecutive frames contrary to other approaches where the current and the desired frames are required. These approaches lead, therefore, to a more complex matching process without a priori knowledge of the scene. Our technique does not require the desired image and consequently enlarges the application domain of visual servoing to natural scenes. More precisely, our algorithm is based on an unified 2D displacement model to cope as well with planar as with non-planar objects. However, since this model is an approximation of the true 2D displacement we have introduced robust statistics to enlarge its validity domain. Experimental results have shown that accurate positioning can be achieved. However, because a low update rate is obtained (around 400 ms with a Pentium 4 at 2 Ghz) only slow camera motion can be considered, this is the main drawback of this approach.

## REFERENCES

- [1] C. Collewet and F. Chaumette, "Visual servoing on non-planar objects from active vision," in *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, Roma, Italy, April 10-14 2007.
- [2] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.
- [3] A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual

- servoing," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 5, pp. 842–853, October 2003.
- [4] F.-X. Espiau, E. Malis, and P. Rives, "Robust features tracking for robotic applications: towards 2d1/2 visual servoing with natural images," in *IEEE Int. Conf. on Robotics and Automation, ICRA'2002*, Washington, USA, May 11-15, 2002.
- [5] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.
- [6] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, October 1996.
- [7] F. Schramm, G. Morel, A. Micaelli, and A. Lottin, "Extended-2d visual servoing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'2004*, New Orleans, USA, April 26-May 1, 2004, pp. 267–273.
- [8] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, April 2002.
- [9] J. Santos-Victor and G. Sandini, "Visual behaviors for docking," *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 223–238, September 1997.
- [10] A. Crétual and F. Chaumette, "Visual servoing based on image motion," *Int. Journal of Robotics Research*, vol. 20, no. 11, pp. 857–877, November 2001.
- [11] C. Collewet, A. Alhaj, and F. Chaumette, "Model-free visual servoing on complex images based on 3d reconstruction," in *IEEE Int. Conf. on Robotics and Automation, ICRA'2004*, New Orleans, USA, April 26-May 1, 2004, pp. 751–756.
- [12] T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, February 1994.
- [13] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," in *Nature*, vol. 293, 1981, pp. 133–135.
- [14] Q. T. Luong and O. Faugeras, "The fundamental matrix: theory, algorithms, and stability analysis," *Int. Journal of Computer Vision*, vol. 17, no. 1, pp. 43–75, 1996.
- [15] H. Longuet-Higgins, "The visual ambiguity of a moving plane," in *IEEE Proc. of the Royal Society of London*, vol. B223, 1984, pp. 165–175.
- [16] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, pp. 485–508, 1988.
- [17] Z. Zhang and A. R. Hanson, "Scaled euclidean 3d reconstruction based on externally uncalibrated cameras," in *IEEE Symp. on Computer Vision*, Coral Gables, Florida, USA, 1995.
- [18] E. Malis and F. Chaumette, "2 1/2d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.
- [19] J. Shi and C. Tomasi, "Good features to track," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, Seattle, Washington, USA, June 1994, pp. 593–600.
- [20] P. J. Huber, *Robust statistics*. New York, NY, USA: Wiley, 1981.
- [21] J. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, December 1995.
- [22] J.-P. Tarel, S.-S. Ieng, and P. Charbonnier, "Using robust estimation algorithms for tracking explicit curves," in *European Conf. on Computer Vision-Part I, ECCV '02*, 2002, pp. 492–507.
- [23] A. Compot, E. Marchand, and F. Chaumette, "Statistically robust 2d visual servoing," *IEEE Trans. on Robotics*, vol. 22, no. 2, pp. 415–421, April 2006.
- [24] F. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. New York: John Wiley and Sons, 1986.
- [25] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. New York, NY, USA: John Wiley & Sons, Inc., 1987.
- [26] M. Gouffès, C. Collewet, C. Fernandez-Maloigne, and A. Trémeau, "Feature points tracking: robustness to specular highlights and lighting changes," in *European Conf. on Computer Vision, ECCV'2006*, Graz, Austria, May 7-13, 2006.