Visual servoing based on structure from controlled motion or on robust statistics

Christophe Collewet, François Chaumette

Visual Servoing Based on Structure From Controlled Motion or on Robust Statistics

Christophe Collewet, *Member, IEEE*, and François Chaumette, *Member, IEEE*

***Abstract*—This paper focuses on the way to achieve accurate visual servoing tasks when the shape of the object being observed as well as the desired image are unknown. More precisely, we want to control the camera orientation with respect to the tangent plane at a certain object point corresponding to the center of a region of interest. We also want to observe this point at the principal point to fulfil a fixation task. A 3-D reconstruction phase must, therefore, be performed during the camera motion. Our approach is then close to the structure-from-motion problem. The reconstruction phase is based on the measurement of the 2-D motion in a region of interest and on the measurement of the camera velocity. Since the 2-D motion depends on the shape of the objects being observed, we introduce a unified motion model to cope both with planar and nonplanar objects. However, since this model is only an approximation, we propose two approaches to enlarge its domain of validity. The first is based on active vision, coupled with a 3-D reconstruction based on a continuous approach, and the second is based on statistical techniques of robust estimation, coupled with a 3-D reconstruction based on a discrete approach. Theoretical and experimental results compare both approaches.**

***Index Terms*—Active vision, 3-D reconstruction, dynamic vision, M-estimators, robust estimation, structure-from-motion, visual servoing.**

I. Introduction

VISUAL servoing is now a widely used technique in robot control [1]. It allows robots to perform tasks from visual features acquired by a camera. However, synthesizing the control law usually requires a model of the scene observed by the camera and also the knowledge of the desired features. In some cases, however, such knowledge is not available. Let us cite, for example, applications in the surgical domain, agriculture, agrifood industry, or in unknown environments like underwater or space (see, for example, [2]). This problem can also appear when considering specific tasks, like perception tasks, where the camera has to move w.r.t. an object of interest to perform automatically an optical character recognition task. In that case, the desired image is also unknown since the goal of the task is precisely to move the camera to see clearly the characters to decode [3]. Classically, visual servoing approaches cannot cope with such applications. Concerning the image-based approach, the main problem is that the desired features are needed. Indeed, in our case where a model is not available, the camera should be moved to the desired position to learn it. But this position is not known in our case. Contrary to the image-based approach, the position-based approach does not require the knowledge of the 2-D desired features since features expressed in the 3-D Cartesian space are required to perform the task. To do that, the pose between the camera and the object is needed [4], which cannot be obtained without a precise model of the object. Recent techniques called model-free visual servoing (or hybrid visual servoing) [5] as well as extended 2-D visual servoing [6] avoid the knowledge of such a model. Unfortunately, they cannot also be used since they are based on the matching of 2-D features between the current and the desired image, which is unknown. On the contrary, visual servoing based on dynamic visual features does not need the knowledge of the desired image. Indeed, the control law is based on parameters that characterize the 2-D motion [7], [8]. This allows the achievement of a positioning task consisting in moving the camera to a position parallel to an object of unknown shape. However, this approach is currently restricted to planar objects.

The main contribution of this paper is that the 3-D parameters obtained by structure-from-motion (SfM) are explicitly used in the control scheme. This makes it possible to easily synthesize the control law, in particular, to take into account any desired orientation of the camera w.r.t. the object and to ensure that it remains in the camera field of view. More precisely, the camera orientation is controlled w.r.t. the tangent plane at a certain point on an unknown object corresponding to the center of a region of interest (ROI). The camera position is controlled to observe this point at the principal point to fulfil a fixation task. The reconstruction phase is based on the measurement of the 2-D motion in a region of interest and on the measurement of the camera velocity. Our emphasis here is on a unified motion model to cope as well with planar as with nonplanar objects, contrary to what has been proposed in our previous paper [9], [10], where only planar objects could be considered.

The paper is organized as follows: first, we present in Section II a review of previous work relevant to SfM. We show how to recover the structure of the object in Section III, either by a continuous or a discrete approach. These approaches are compared in Section IV. We describe how to obtain the 2-D motion in Section V, while Section VI details the approaches that we propose to enlarge the validity domain of our unified motion model. The control law is presented in Section VII. Experimental results on a 6-DOF eye-in-hand robot are given in Section VIII.

II. Previous Work on Structure-From-Motion

Numerous papers have addressed the problem of structure-from-motion, that is recovering both the 3-D structure of an object and the 3-D motion of a camera observing the scene of

Manuscript received February 1, 2007; revised September 25, 2007. This paper was recommended for publication by Associate Editor B. Nelson and Editor L. Parker upon evaluation of the reviewers' comments.

The authors are with IRISA/INRIA Rennes Bretagne Atlantique, 35042 Rennes, France (e-mail: christophe.collewet@irisa.fr; francois.chaumette@irisa.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2007.913996

interest. Nevertheless, two basic approaches emerge, *continuous approaches* and *discrete approaches*. The former are based on the optical flow, while the latter are based on the matching of visual features between two (or more) views.

A. Continuous Approaches

Let us consider a point M of the object described by $\mathbf{X} = (X, Y, Z)$ in the camera frame, with the Z-axis the camera optical axis. Assuming, without loss of generality, a unit focal length, this point projects to the point m, described by $\mathbf{x} = (x, y, 1)$, according to $\mathbf{x} = \mathbf{X}/Z$.

Moreover, let us assume that the camera is subjected to the velocity $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v} = (v_x, v_y, v_z)$ and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ are its translational and rotational components respectively. Therefore, $\dot{\boldsymbol{X}}$ can be expressed as

$$\dot{\mathbf{X}} = -\boldsymbol{v} - \boldsymbol{\omega} \times \mathbf{X} \tag{1}$$

which yields to the well-known relation [12]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -1-x^2 & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \mathbf{v}. \tag{2}$$

In (2), only the depth Z is unknown if $\mathbf{x}$, $\dot{\mathbf{x}}$, and $\mathbf{v}$ can be measured.

Various approaches to estimate Z exist. They are based on how $\dot{\mathbf{x}}$ is used in the estimation. Much work is based on the assumption that the brightness of m remains constant during the motion, yielding the well-known additional constraint [12]

$$\dot{x} I_x + \dot{y} I_y + I_t = 0 \tag{3}$$

where I_x, I_y, and I_t represent the spatiotemporal derivatives of the intensity at m. By substituting $\dot{\mathbf{x}}$ given by (2) in (3), an analytic solution can be obtained [13] when a fixation point is used. Such approaches, known as *direct approaches*, have the main advantage of avoiding the computation of the optical flow but require, in return, accurate estimations of I_x, I_y, and I_t, and therefore, are not very accurate in practice (see, however, [14], where a trilinear brightness constraint instead of (3) is used, leading to more robust results).

Another approach is to compute the optical flow and, if necessary, its spatial derivatives. Such approaches are known as *indirect approaches*. These approaches can be classified into two main categories, *dense* or *sparse*. In the former case, the depth of all the points is estimated [15]. In the latter, only points in the neighborhood of a set of points [16] are reconstructed, leading to an estimation of the orientation of the tangent plane. Most often, hypotheses about the scene are formulated, in particular, whether or not planar surfaces exist [15], [16].

B. Discrete Approaches

Whereas the previous approaches were based on differential aspects, the discrete approaches use a discrete way to describe the displacement of a point M between two frames by integrating (1) during Δt

$$\mathbf{X}_{k+1} = \mathbf{R}_k \mathbf{X}_k + \mathbf{t}_k \tag{4}$$

where $\mathbf{R}_k$ is a rotation matrix depending on $\boldsymbol{\omega}$, and $\mathbf{t}_k$ is a translation vector depending on $\boldsymbol{v}$ and $\boldsymbol{\omega}$ at time k.

First, since such an approach is based on feature matching, it will necessarily lead to a sparse reconstruction (see nevertheless [17], where quasi-dense reconstruction is performed). Because the scenes considered in this paper contain very little structure, we focus only on point features. However, we invite the interested reader to [18] where other features are used.

The first work in this domain led to nonlinear relations by exploiting the temporal constraint of the projection of M into two frames [19]. A linear formulation can also be obtained by using a decomposition of the essential [20] or the fundamental [21] matrices if at least eight matched points are provided from two views. However, some degenerate cases may occur if either the camera motion is a pure rotation, or the scene is planar [22]. Indeed, the relation between the two frames expresses in that case as a homography. Therefore, specific algorithms have been proposed to handle such degenerate situations by switching from epipolar feature matching to a homography approach [23]. Note also that the SfM problem can be solved from a homography matrix computed from a virtual plane attached to the object [5].

Nevertheless, such approaches become very unstable when the camera undergoes a small motion between k and $k+1$. Such cases can arise in robot control, where the acquisition rate has to be as high as possible. A solution to this problem is either to consider that the two views involved in the reconstruction process are the current and the desired views, as proposed in [5], or to use all the past frames up to time k. Most often, such approaches are based on the reconstruction of a state vector by using an extended Kalman filter [24]. In any case, the main disadvantage of discrete techniques remains that they need to maintain feature correspondences over multiple frames.

III. A Unified Approach

We have seen that the continuous approaches using a local formulation are more appropriate than the discrete approaches to recover the normal at a point because of their local nature. We have also seen that discrete approaches are not suited to perform locally dense reconstruction since they are based on matching sparse features. In the next sections, we propose a solution for the continuous and the discrete approaches to reconstruct the normal at the center of a ROI whatever be the shape of the object. In addition, contrary to what is often proposed, we do not have to assume explicit constraints about the scene [15], [16] or to use a selection mechanism of models [25], [26].

A. Reconstruction of the Normal

Let us consider a point P described by $\mathbf{X_P} = (X_P, Y_P, Z_P)$ in the camera frame. This point is chosen so that its projection p described by $\mathbf{x_P} = (x_p, y_p, 1)$ lies in the center of the ROI. Note that we will not assume, as is usually done, that it lies near the principal point (see [8] and [16], for example).

The tangent plane at P can be expressed as follows

$$Z = Z_P + A_{10}(X - X_P) + A_{01}(Y - Y_P) \tag{5}$$

where $A_{10} = \frac{\partial Z}{\partial X}|_P$ and $A_{01} = \frac{\partial Z}{\partial Y}|_P$. This leads to the normal $\mathbf{n} = (A_{10}, A_{01}, -1)$ at P, which is required to compute the control law (see Section VII). We can rewrite (5) in a more

compact form

$$Z = A_{00} + A_{10}\, X + A_{01}\, Y \tag{6}$$

where $A_{00} = -\mathbf{n}^\top \mathbf{X_P}$.

By perspective projection, it is also possible to rewrite (6) with respect to the normalized coordinates $\mathbf{x}$

$$\frac{1}{Z} = \boldsymbol{\alpha}^\top \mathbf{x} \tag{7}$$

with $\boldsymbol{\alpha} = (\alpha_{10}, \alpha_{01}, \alpha_{00})$, where $\alpha_{10} = -A_{10}/A_{00}$, $\alpha_{01} = -A_{01}/A_{00}$, and $\alpha_{00} = 1/A_{00}$. We can also rewrite (6) by introducing $u = x - x_p$ and $v = y - y_p$

$$\frac{1}{Z} = \boldsymbol{\beta}^\top \mathbf{u} \tag{8}$$

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 $\beta_{00} = \boldsymbol{\alpha}^\top \mathbf{x_P}$. Therefore, if we can estimate $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$, the unit normal $\tilde{\mathbf{n}}$ at P can be deduced as $\tilde{\mathbf{n}} = -\boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|$.

In the following sections, we will see how to explicitly obtain $\boldsymbol{\beta}$ either by a continuous or by a discrete approach.

B. Continuous Approach

Let us note $\varphi(\mathbf{u})$ the function (with $\varphi(\mathbf{0}) = 0$) such that the true depth can be expressed by

$$\frac{1}{Z} = \boldsymbol{\beta}^\top \mathbf{u} + \varphi(\mathbf{u}) \tag{9}$$

for any point belonging to the object. We have $\varphi(\mathbf{u}) = 0\ \forall \mathbf{u}$ for a planar object, but $\varphi(\mathbf{u}) \neq 0$ in the other cases. Note that this function only describes the terms higher than order 1 in $\mathbf{u}$.

Thereafter, a general 2-D motion model can be obtained by substituting (9) in (2)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a'(\boldsymbol{\beta}^\top \mathbf{u} + \varphi(\mathbf{u})) + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ b'(\boldsymbol{\beta}^\top \mathbf{u} + \varphi(\mathbf{u})) + (1+y^2)\omega_x - xy\omega_y - x\omega_z \end{bmatrix} \tag{10}$$

where $a' = xv_z - v_x$ and $b' = yv_z - v_y$.

However, this model depends on the surface being observed through the function $\varphi(\mathbf{u})$. To not depend on the object shape, a way to proceed is to consider only a small neighborhood of p so that the depths given by (9) coincide with the ones given by the tangent plane (8). However, since (8) is only true locally, we perform a first-order Taylor series expansion of (10) around p, leading to the following affine 2-D displacement model

$$\dot{\mathbf{x}} = \mathbf{M}\, \mathbf{u} \tag{11}$$

where

$$\begin{cases} M_{11} = y_p\omega_x - 2x_p\omega_y + \beta_{00}v_z + \beta_{10}a \\ M_{12} = x_p\omega_x + \omega_z + \beta_{01}a \\ M_{13} = x_p y_p \omega_x - \left(1 + x_p^2\right)\omega_y + y_p\omega_z + \beta_{00}a \\ M_{21} = \beta_{10}b - y_p\omega_y - \omega_z \\ M_{22} = 2y_p\omega_x - x_p\omega_y + \beta_{00}v_z + \beta_{01}b \\ M_{23} = \left(1 + y_p^2\right)\omega_x - x_p y_p \omega_y - x_p\omega_z + \beta_{00}b \end{cases} \tag{12}$$

with $a = x_p v_z - v_x$ and $b = y_p v_z - v_y$. Throughout this paper, we refer to this motion model (11) as *the unified motion model* since it can cope with planar or nonplanar objects.

Consequently, if we estimate the parameters of this motion model (see Section V) and if the 3-D velocity is assumed to be known, an estimation $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ can be obtained by solving a linear system. Indeed, (12) can be rewritten as follows

$$\mathbf{C}\boldsymbol{\beta} = \boldsymbol{\Gamma} \tag{13}$$

with

$$\mathbf{C}^\top = \begin{bmatrix} a & 0 & 0 & b & 0 & 0 \\ 0 & a & 0 & 0 & b & 0 \\ v_z & 0 & a & 0 & v_z & b \end{bmatrix} \tag{14}$$

and

$$\boldsymbol{\Gamma} = \begin{bmatrix} M_{11} - y_p\omega_x + 2x_p\omega_y \\ M_{12} - x_p\omega_x - \omega_z \\ M_{13} - x_p y_p \omega_x + \left(1 + x_p^2\right)\omega_y - y_p\omega_z \\ M_{21} + y_p\omega_y + \omega_z \\ M_{22} - 2y_p\omega_x + x_p\omega_y \\ M_{23} - \left(1 + y_p^2\right)\omega_x + x_p y_p \omega_y + x_p\omega_z \end{bmatrix} \tag{15}$$

leading to the following least-squares solution

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{C}^\top \mathbf{C}\right)^{-1} \mathbf{C}^\top \boldsymbol{\Gamma}. \tag{16}$$

However, this solution is only correct if the matrix $\mathbf{C}^\top \mathbf{C}$ is well conditioned, that is if the condition number ν of $\mathbf{C}^\top \mathbf{C}$ is low enough. Since $\mathbf{C}^\top \mathbf{C}$ is very simple, the analytical form of ν can be determined

$$\nu = \frac{a^2 + b^2 + v_z^2 + |v_z|\sqrt{a^2 + b^2 + v_z^2}}{a^2 + b^2 + v_z^2 - |v_z|\sqrt{a^2 + b^2 + v_z^2}}. \tag{17}$$

We will use this relation in Section VI-A to ensure that ν does not become too high during the camera motion.

Remarks:

1) Let us recall the well-known result that a 3-D reconstruction is not possible if no translation occurs or if the camera moves in the direction of the point to reconstruct $(a = b = 0)$. Indeed, in these cases, $\mathbf{C}^\top \mathbf{C}$ becomes singular. We will return to this problem in Section VI-A.
2) Instead of performing a first-order Taylor series expansion of (10) at p, one may perform a second-order Taylor series expansion leading to a quadratic model. In that case, if the object is planar, this model will coincide with the true 2-D motion. However, previous work on motion estimation using parametric models has shown that, in practice, one cannot expect to obtain reliable second-order terms [16], [27]. This paper will also show that using the unified affine motion model leads to very good results (see Section VIII).
3) If we set p as the principal point in (12), we recover the classical affine motion model of a planar object when the terms of order 2 in x and y are neglected. That is what is usually used (see, for example, [16] and [27]). However, our model is more general and fits better the true motion. To illustrate this result, a simulation has been carried out on a planar object (see Fig. 1). The initial orientation between the camera and the plane was $\boldsymbol{\Phi} \approx (-32.0^\circ, 20.4^\circ)$

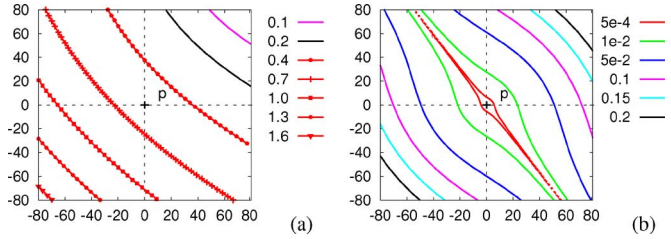

Fig. 1. Norm of the modeling error around p between the true motion and: (a) the classical affine motion model, (b) the unified affine motion model (in pixels).

(pitch and yaw); the initial depth in P was $Z_P = 61.2$ cm; P projected in the CCD plane at (131, 98); the velocities were $\boldsymbol{v} = (-0.05911, -0.07908, -0.02692)$ (m/s) and $\boldsymbol{\omega} = (-0.11747, 0.08056, -0.00059)$ (rad/s); the acquisition rate was 200 ms. Fig. 1(a) represents the norm of the modeling error around p between the true motion and the classical affine motion model, while Fig. 1(b) represents the norm of the modeling error between the true motion and the unified affine motion model (the theoretical parameters have been used for all the models). As can be seen, our motion model is much more accurate than the classical one.

4) Since $\widehat{\boldsymbol{\beta}}$ depends on measurements of $\mathbf{v}$ and the matrix $\mathbf{M}$, it is important to evaluate how sensitive is $\widehat{\boldsymbol{\beta}}$ w.r.t. uncertainties related to these measurements. More precisely, we can compute $\partial\beta_i/\partial v_m$ and $\partial\beta_i/\partial M_{mn}$ in order to evaluate

$$\sigma^2_{\beta_i} = \sum_{m=1}^{6}\left(\frac{\partial\beta_i}{\partial v_m}\right)^2\sigma^2_{v_m} \text{ and } \sigma^2_{\beta_i} = \sum_{m,n}\left(\frac{\partial\beta_i}{\partial M_{mn}}\right)^2\sigma^2_{M_{mn}} \tag{18}$$

where $\sigma^2_{v_m}$ describes the uncertainty in one of the components v_m of $\mathbf{v}$, and $\sigma^2_{M_{mn}}$ the uncertainty in the (m,n) component of the matrix $\mathbf{M}$. These terms have been computed in [28] for planar objects. They show that, as expected, the angular velocity has no influence on $\partial\beta_i/\partial M_{mn}$ or $\partial\beta_i/\partial v_m$; only the translational velocity $\boldsymbol{v}$ is involved in these computations. More precisely, it is shown that $\partial\beta_i/\partial M_{mn}$ or $\partial\beta_i/\partial v_m$ are low when $\boldsymbol{v}$ is high. They also show that $v_z = 0$ makes these parameters lower.

C. Discrete Approach

First, we integrate (1) between $k\Delta t$ and $(k+1)\Delta t$ to obtain (4). 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 have an expression for the 2-D displacement between k and $k+1$

$$\begin{cases} x_{k+1} = \dfrac{R_{11}\,x_k + R_{12}\,y_k + R_{13} + t_x/Z_k}{R_{31}\,x_k + R_{32}\,y_k + R_{33} + t_z/Z_k} \\ y_{k+1} = \dfrac{R_{21}\,x_k + R_{22}\,y_k + R_{23} + t_y/Z_k}{R_{31}\,x_k + R_{32}\,y_k + R_{33} + t_z/Z_k} \end{cases}. \tag{19}$$

From (19), we recover again the well-known result that without any translation $\mathbf{t}_k$, the depth cannot be recovered.

If we substitute the depth (9) in (19), we obtain a general model of the 2-D displacement

$$\begin{cases} x_{k+1} = \dfrac{R'_{11}\,u_k + R'_{12}\,v_k + R'_{13} + \left(\boldsymbol{\beta}^\top\mathbf{u}_k + \varphi(\mathbf{u}_k)\right)t'_x}{R'_{31}\,u_k + R'_{32}\,v_k + 1 + \left(\boldsymbol{\beta}^\top\mathbf{u}_k + \varphi(\mathbf{u}_k)\right)t'_z} \\ y_{k+1} = \dfrac{R'_{21}\,u_k + R'_{22}\,v_k + R'_{23} + \left(\boldsymbol{\beta}^\top\mathbf{u}_k + \varphi(\mathbf{u}_k)\right)t'_y}{R'_{31}\,u_k + R'_{32}\,v_k + 1 + \left(\boldsymbol{\beta}^\top\mathbf{u}_k + \varphi(\mathbf{u}_k)\right)t'_z} \end{cases} \tag{20}$$

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

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

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

We also introduce the following matrix

$$\mathbf{R}'' = \begin{bmatrix} R'_{11} + \beta_{10}t'_x & R'_{12} + \beta_{01}t'_x & R'_{13} + \beta_{00}t'_x \\ R'_{21} + \beta_{10}t'_y & R'_{22} + \beta_{01}t'_y & R'_{23} + \beta_{00}t'_y \\ R'_{31} + \beta_{10}t'_z & R'_{32} + \beta_{01}t'_z & 1 + \beta_{00}t'_z \end{bmatrix} \tag{22}$$

that can be rewritten when $t'_z \neq 0$ as follows by denoting $\boldsymbol{\gamma} = t'_z\boldsymbol{\beta}$ and $\boldsymbol{k} = \mathbf{t}'/t'_z$

$$\mathbf{R}'' = \begin{bmatrix} R'_{11} + \gamma_x k_x & R'_{12} + \gamma_y k_x & R'_{13} + \gamma_z k_x \\ R'_{21} + \gamma_x k_y & R'_{22} + \gamma_y k_y & R'_{23} + \gamma_z k_y \\ R'_{31} + \gamma_x & R'_{32} + \gamma_y & 1 + \gamma_z \end{bmatrix}. \tag{23}$$

Thereafter, either by using (22) or (23), the model given in (20) becomes

$$\begin{cases} x_{k+1} = \dfrac{R''_{11}\,u_k + R''_{12}\,v_k + R''_{13} + \varphi(\mathbf{u}_k)t'_x}{R''_{31}\,u_k + R''_{32}\,v_k + R''_{33} + \varphi(\mathbf{u}_k)t'_z} \\ y_{k+1} = \dfrac{R''_{21}\,u_k + R''_{22}\,v_k + R''_{23} + \varphi(\mathbf{u}_k)t'_y}{R''_{31}\,u_k + R''_{32}\,v_k + R''_{33} + \varphi(\mathbf{u}_k)t'_z} \end{cases}. \tag{24}$$

As in the continuous case, this model depends on the surface being observed. To not depend on the object shape, we also consider that the depths given by (9) coincide with the ones given by the tangent plane (8). Thus, here again, we have to perform a Taylor series expansion of (24) at p leading to the following affine 2-D displacement model

$$\mathbf{x}_{k+1} = \mathbf{M}\,\mathbf{u}_k \tag{25}$$

with

$$\mathbf{M} = \frac{1}{R''_{33}}\begin{bmatrix} \dfrac{\mathrm{Min}_{22}(\mathbf{R}'')}{R''_{33}} & \dfrac{\mathrm{Min}_{21}(\mathbf{R}'')}{R''_{33}} & R''_{13} \\ \dfrac{\mathrm{Min}_{12}(\mathbf{R}'')}{R''_{33}} & \dfrac{\mathrm{Min}_{11}(\mathbf{R}'')}{R''_{33}} & R''_{23} \end{bmatrix} \tag{26}$$

where the notation $\mathrm{Min}_{ij}(\mathbf{A})$ denotes the (i,j)th minor of the matrix $\mathbf{A}$. We refer to this displacement model (25) as *the unified displacement model*.

Thereafter, as in the continuous case, if we estimate the parameters of this displacement model (see Section V) and if the 3-D velocity is supposed to be known, an estimation of $\boldsymbol{\beta}$ can

be obtained. To do that, we first suppose that $t'_z \neq 0$ (the case where $t'_z = 0$ will be discussed afterward) and explicitly use the definition of $\mathbf{R}''$ given by (23) in (26) leading after simple manipulations to the following relations:

$$\begin{cases} M_{11} = \dfrac{\mathrm{Min}_{22}(\mathbf{R}') + (R'_{11} - k_x R'_{31})\gamma_z - K_x\gamma_x}{(1+\gamma_z)^2} \\ M_{12} = \dfrac{\mathrm{Min}_{21}(\mathbf{R}') + (R'_{12} - k_x R'_{32})\gamma_z - K_x\gamma_y}{(1+\gamma_z)^2} \\ M_{13} = \dfrac{R'_{13} + k_x\gamma_z}{1+\gamma_z} \\ M_{21} = \dfrac{\mathrm{Min}_{12}(\mathbf{R}') + (R'_{21} - k_y R'_{31})\gamma_z - K_y\gamma_x}{(1+\gamma_z)^2} \\ M_{22} = \dfrac{\mathrm{Min}_{11}(\mathbf{R}') + (R'_{22} - k_y R'_{32})\gamma_z - K_y\gamma_y}{(1+\gamma_z)^2} \\ M_{23} = \dfrac{R'_{23} + k_y\gamma_z}{1+\gamma_z} \end{cases} \tag{27}$$

where $K_x = R'_{13} - k_x$ and $K_y = R'_{23} - k_y$. From these relations, an estimation of $\boldsymbol{\gamma}$ can be obtained by solving the nonlinear system

$$\begin{cases} M_{11}\gamma_z^2 + (2M_{11} + k_x R'_{31} - R'_{11})\gamma_z + K_x\gamma_x + \Gamma_1 = 0 \\ M_{12}\gamma_z^2 + (2M_{12} + k_x R'_{32} - R'_{12})\gamma_z + K_x\gamma_y + \Gamma_2 = 0 \\ (M_{13} - k_x)\gamma_z + \Gamma_3 = 0 \\ M_{21}\gamma_z^2 + (2M_{21} + k_y R'_{31} - R'_{21})\gamma_z + K_y\gamma_x + \Gamma_4 = 0 \\ M_{22}\gamma_z^2 + (2M_{22} + k_y R'_{32} - R'_{22})\gamma_z + K_y\gamma_y + \Gamma_5 = 0 \\ (M_{23} - k_y)\gamma_z + \Gamma_6 = 0 \end{cases} \tag{28}$$

where Γ_i are the components of $\boldsymbol{\Gamma}$ defined as follows:

$$\boldsymbol{\Gamma} = \begin{bmatrix} M_{11} & - & \mathrm{Min}_{22}(\mathbf{R}') \\ M_{12} & - & \mathrm{Min}_{21}(\mathbf{R}') \\ M_{13} & - & R'_{13} \\ M_{21} & - & \mathrm{Min}_{12}(\mathbf{R}') \\ M_{22} & - & \mathrm{Min}_{11}(\mathbf{R}') \\ M_{23} & - & R'_{23} \end{bmatrix}. \tag{29}$$

More precisely, we rewrite (28) in a least-squares sense (since we have six equations and three unknowns) leading finally to a third-order polynomial in γ_z, which can be easily solved.

When $t'_z = 0$, instead of using (23) in (26), we use (22) leading to

$$\begin{cases} M_{11} = \mathrm{Min}_{22}(\mathbf{R}') + (\beta_{10} - \beta_{00}R'_{31})\,t'_x \\ M_{12} = \mathrm{Min}_{21}(\mathbf{R}') + (\beta_{01} - \beta_{00}R'_{32})\,t'_x \\ M_{13} = R'_{13} + \beta_{00}t'_x \\ M_{21} = \mathrm{Min}_{12}(\mathbf{R}') + (\beta_{10} - \beta_{00}R'_{31})\,t'_y \\ M_{22} = \mathrm{Min}_{11}(\mathbf{R}') + (\beta_{01} - \beta_{00}R'_{32})\,t'_y \\ M_{23} = R'_{23} + \beta_{00}t'_y \end{cases} \tag{30}$$

that can be expressed under a linear form with respect to $\boldsymbol{\beta}$

$$\mathbf{C}\boldsymbol{\beta} = \boldsymbol{\Gamma} \tag{31}$$

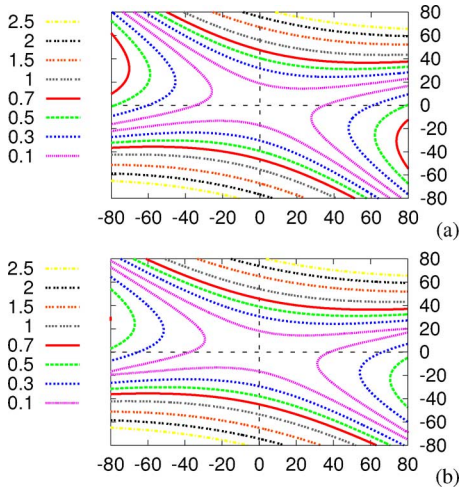

Fig. 2. Norm of the modeling error around p between the true displacement and: (a) the homographic displacement model, (b) the unified affine displacement model (in pixels).

with

$$\mathbf{C}^\top = \begin{bmatrix} t'_x & 0 & 0 & t'_y & 0 & 0 \\ 0 & t'_x & 0 & 0 & t'_y & 0 \\ -t'_x R'_{31} & -t'_x R'_{32} & t'_x & -t'_y R'_{31} & -t'_y R'_{32} & t_y \end{bmatrix} \tag{32}$$

leading to the following least-squares solution $\widehat{\boldsymbol{\beta}} = \mathbf{C}^+\,\boldsymbol{\Gamma}$.

Similar to the continuous case, one may ask why not use a homographic model [i.e., (20)] when $\varphi(\mathbf{u}_k) = 0$) instead of using the unified displacement model, since for a planar object, the homographic model fits the true motion. The main reason is that, here again, it is difficult to compute the parameters of this model accurately when the acquisition rate is high (as required for control issues) without matching points between two consecutive frames. In [10], this matrix has been reliably computed only for planar objects and when the ROI was very large, penalizing, therefore, the dynamic behavior of the robot.

Besides, in case of nonplanar objects, nothing says that a homographic model yields accurate results. To show the efficiency of our model, a simulation has been carried out on a nonplanar object (see Fig. 2). 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 $\boldsymbol{\Phi} = (-22^\circ, 25^\circ)$, the depth in P was $Z_P = 61$ cm. Fig. 2(a) represents the norm of the modeling error between the true displacement and the homographic displacement model, while Fig. 2(b) the one between the true displacement and the unified displacement model. Note that the theoretical parameters have been used for all the models. We clearly see that the unified model provides similar results than the homographic model, and even slightly better. It is a second reason to prefer our displacement model to the homographic one.

Two approaches have been proposed in this section to compute the structure, a continuous and a discrete one. We now compare them.

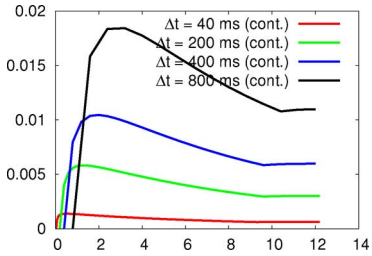

Fig. 3. Relative error between the true value of the depth in P versus time (in seconds) for various choices of Δt.

IV. Comparison Between the Continuous and the Discrete Approaches

As has been shown in Section III, the continuous approach is based on an instantaneous relation between the 3-D velocity, the structure of the object, and the 2-D motion through the matrix $\mathbf{M}$. The problem with such an approach is how to obtain precisely this matrix since it is valid only at a given time. Indeed, it depends on $\boldsymbol{\beta}$, which is not constant in the camera frame. Therefore, regardless of the way we obtain $\mathbf{M}$, a high acquisition rate or a low 3-D velocity is required. In practice, we will see in Section V that $\mathbf{M}$ is not directly measured, but rather the relation obtained by integrating (11) during Δt. Then, $\mathbf{M}$ is obtained by considering $\dot{\mathbf{x}} = (\mathbf{x}_{k+1} - \mathbf{x}_k)/\Delta t$, which is only valid if Δt is small. Moreover, the relation really observed takes into account that $\boldsymbol{\beta}$ is not constant, which is not modeled by the matrix $\mathbf{M}$. Contrary, to the continuous approach, the discrete one does not depend at all on Δt since it takes into account the variation of the structure expressed in the camera frame by integrating the 3-D velocities between two consecutive instants [see (19)].

To illustrate the role played by Δt using a continuous approach, simulations of positioning tasks with respect to a planar object have been carried out (see Fig. 3). The initial pose was ${}^{i}\boldsymbol{\Phi} = (-15^\circ, -15^\circ, 0)$ with ${}^{i}Z_P = 90$ cm, the desired pose was ${}^{d}\boldsymbol{\Phi} = (20^\circ, 20^\circ, 0)$, and the desired depth in P was $Z^* = 65$ cm. The control law is described in Section VII. Using simulations allows us to compare the exact value of $\boldsymbol{\beta}$ with the one obtained through (16). However, to provide concise results, we rather compare the true depth Z in P with the estimated one $\hat{Z}$ given by (8) when $\hat{\boldsymbol{\beta}}$ is computed from a continuous approach for various choices of Δt. More precisely, Fig. 3 represents the relative error $(Z - \hat{Z})/Z$ versus Δt. This figure clearly shows that, in practice, using a continuous approach provides accurate estimations when the acquisition rate is high (with $\Delta t \approx$ 200 ms, the relative error is less than 0.5 %).

Consequently, when Δt is low, we will use the continuous approach since, in that case, the structure is obtained from a linear system; otherwise, the discrete approach will be used.

V. Estimation of the Frame-to-Frame Displacement

As seen in Section III, recovering the structure needs to compute the 2-D motion between two consecutive frames. This 2-D motion has been modeled as an affine motion. Using $\mathbf{u} = \mathbf{x} - \mathbf{x_p}$ in (25), we have

$$\mathbf{x_{k+1}} = \mathbf{M}\,(\mathbf{x_k} - \mathbf{x_{pk}}) = \boldsymbol{\delta}\,(\mathbf{x_k}, \boldsymbol{\mu}) \tag{33}$$

with $\boldsymbol{\mu} = (M_{11}, M_{12}, M_{13}, M_{21}, M_{22}, M_{23})$.

Therefore, if we denote f and g two consecutive frames and assume that the brightness of m remains unchanged during the motion, we can write

$$f(m) = g(\boldsymbol{\delta}(m, \boldsymbol{\mu})). \tag{34}$$

Because of the noise, (34) is generally not satisfied. Therefore, the solution is to move the problem to an optimization one to find the parameters that minimize the following criterion

$$J(\boldsymbol{\mu}) = \sum_{m \in W} \left(f(m) - \boldsymbol{g}(\boldsymbol{\delta}(m, \boldsymbol{\mu}))\right)^2 \tag{35}$$

where W denotes the windows of interest centered in p.

Note that in practice, we perform a photometric normalization, as described in [29], for example. Indeed, since the camera is moving w.r.t. the object and because it is not planar, lighting changes occur. In that case, (34) is no longer perfectly valid.

To carry out the optimization, the classical approach [30] assumes that the acquisition rate and the displacements are sufficiently small. If so, a Taylor series expansion of g can be performed. However, if we want to access large displacements between two frames, this approach cannot be used. To cope with this problem, multi-scale [2] or multi-resolution approaches [31] can be used. Nevertheless, these solutions remain time consuming.

Since $\mathbf{v}$ can be approximately known, it can be used to provide an estimation $\widehat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ according to (12) or (26) (note that once $\widehat{\boldsymbol{\beta}}$ is known, it can also be introduced in (12) or (26) to improve $\widehat{\boldsymbol{\mu}}$, otherwise a coarse approximation of $\boldsymbol{\beta}$ is used). Therefore, we do not have to assume that the acquisition rate is high or the displacements are sufficiently small to ensure that the variation of $\boldsymbol{\mu}$ between f and g is small. Thereafter, it is possible to perform a first-order Taylor series expansion of $g(\boldsymbol{\delta}(m, \boldsymbol{\mu}))$ in a neighborhood of $\widehat{\boldsymbol{\mu}}$ such that $\boldsymbol{\mu} = \widehat{\boldsymbol{\mu}} + \boldsymbol{\varsigma}$

$$g(\boldsymbol{\delta}(m, \boldsymbol{\mu})) = g(\boldsymbol{\delta}(m, \widehat{\boldsymbol{\mu}})) + \nabla g^{\top}(\boldsymbol{\delta}(m, \widehat{\boldsymbol{\mu}})).J_{\boldsymbol{\delta}}{}^{\boldsymbol{\mu}}.\boldsymbol{\varsigma} \tag{36}$$

where $J_{\boldsymbol{\delta}}{}^{\boldsymbol{\mu}}$ represents the Jacobian of $\boldsymbol{\delta}$ with respect to $\boldsymbol{\mu}$. Note that we only have to assume here that $\boldsymbol{\varsigma}$ is low and not, as is usually the case, that $\boldsymbol{\mu}$ is low.

Therefore, using (36) in (35) and differentiating with respect to $\boldsymbol{\varsigma}$ leads to a linear system in $\boldsymbol{\varsigma}$. As usual, this system is inverted by using an iterative Newton–Raphson style algorithm to account for the error introduced by the Taylor series expansion (see [10] for more details).

VI. Enlarging the Validity Domain of the Unified Motion Model

Since $\widehat{\boldsymbol{\beta}}$ depends on the measurement of $\mathbf{M}$, the unified displacement model derived in Section III has to fit as best as possible the true displacement to provide an accurate value for $\widehat{\boldsymbol{\beta}}$. However, since this model has been obtained from a Taylor series expansion, we focus on the way to improve its domain of validity so that it can be valid even far from p. To cope with this problem, we study the modeling error $\mathbf{E}(\mathbf{u})$ between the true and the unified displacement model in order to minimize it.

First, let us consider the case of a planar object, i.e., $\varphi(\mathbf{u}) = 0$ in (9). In that case, $\mathbf{E}(\mathbf{u}) = \mathbf{E}_{\text{planar}}(\mathbf{u})$ can be deduced by

subtracting (25) from (24)

$$\mathbf{E}_{\text{planar}}(\mathbf{u}) = -\zeta(\mathbf{u}) \begin{bmatrix} \dfrac{\text{Min}_{22}(\mathbf{R}'')u + \text{Min}_{21}(\mathbf{R}'')v}{(\zeta(\mathbf{u}) + R''_{33})\, {R''_{33}}^2} \\ \dfrac{\text{Min}_{12}(\mathbf{R}'')u + \text{Min}_{11}(\mathbf{R}'')v}{(\zeta(\mathbf{u}) + R''_{33})\, {R''_{33}}^2} \end{bmatrix} \quad (37)$$

with $\zeta(\mathbf{u}) = R''_{31}u + R''_{32}v$.

In the general case of nonplanar objects, we introduce the following matrices

$$\mathbf{A} = \begin{bmatrix} R''_{11} & R''_{12} & t'_x \\ R''_{21} & R''_{22} & t'_y \\ R''_{31} & R''_{32} & t'_z \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} R''_{11} & t'_x & R''_{13} \\ R''_{21} & t'_y & R''_{23} \\ R''_{31} & t'_z & R''_{33} \end{bmatrix} \quad (38)$$

leading to a simple expression of the modeling error

$$\mathbf{E}(\mathbf{u}) = \mathbf{E}_{\text{planar}}(\mathbf{u}) + \delta\mathbf{E}(\mathbf{u}) \quad (39)$$

with

$$\delta\mathbf{E}(\mathbf{u}) = \varphi(\mathbf{u})\boldsymbol{\xi}(\mathbf{u}) \quad (40)$$

where

$$\boldsymbol{\xi}(\mathbf{u}) = \begin{bmatrix} \dfrac{\text{Min}_{21}(\mathbf{B}) - \text{Min}_{22}(\mathbf{A})u - \text{Min}_{21}(\mathbf{A})v}{D(\mathbf{u})} \\ \dfrac{\text{Min}_{11}(\mathbf{B}) - \text{Min}_{12}(\mathbf{A})u - \text{Min}_{11}(\mathbf{A})v}{D(\mathbf{u})} \end{bmatrix} \quad (41)$$

and $D(\mathbf{u}) = (R''_{31}u + R''_{32}v + R''_{33})(R''_{31}u + R''_{32}v + R''_{33} + t'_z\varphi(\mathbf{u}))$.

Two solutions are now proposed to minimize the modeling error $\mathbf{E}(\mathbf{u})$. The first is based on the choice of a specific camera motion, i.e., active vision [32]. In that case, our goal is to find a motion that minimizes $\mathbf{E}_{\text{planar}}(\mathbf{u})$ and $\boldsymbol{\xi}(\mathbf{u})$ involved in $\delta\mathbf{E}(\mathbf{u})$, leading to the minimization of $\mathbf{E}(\mathbf{u})$. Instead of minimizing $\boldsymbol{\xi}(\mathbf{u})$ in $\delta\mathbf{E}(\mathbf{u})$, the second solution is based on the selection of points $\mathbf{u}$ for which $\varphi(\mathbf{u})$ is low, leading also to the minimization of $\delta\mathbf{E}(\mathbf{u})$.

Finally, note that 1) since a 3-D motion is needed to compute $\widehat{\boldsymbol{\beta}}$ and 2) since the control law is based on the knowledge of $\widehat{\boldsymbol{\beta}}$ (see Section VII), a preliminary step is required before the servoing step, i.e., when $\widehat{\boldsymbol{\beta}}$ is not known at all. Both these steps are studied in the next sections.

A. Using Active Vision

1) First Step: The goal of this first step is to provide an initial value for $\widehat{\boldsymbol{\beta}}$ that will be used in the second step to perform the positioning task while improving this value. Since $\widehat{\boldsymbol{\beta}}$ is initially not known, only constant translations are performed (as seen in Section VII, $\widehat{\boldsymbol{\beta}}$ is required to achieve the task). These translations are chosen so that p will move toward the principal point at constant velocity.

1) Let us first consider the case where the object is planar. Since we have $\mathbf{R}_k = \mathbf{I}_3$, $\zeta(\mathbf{u})$ involved in (37) becomes $t'_z(\beta_{10}u + \beta_{01}v)$. Thus, if $t'_z = 0$, the modeling error vanishes.
2) Second, let us consider the case where the object is not planar. In that case, we have

$$\boldsymbol{\xi}(\mathbf{u}) = \frac{1}{D(\mathbf{u})} \begin{bmatrix} t'_x - (u + R'_{13})\, t'_z \\ t'_y - (v + R'_{23})\, t'_z \end{bmatrix}. \quad (42)$$

Since $t'_z = 0$, $D(\mathbf{u}) = 1$, and the modeling error is simply

$$\mathbf{E}(\mathbf{u}) = \varphi(\mathbf{u})\mathbf{t}'. \quad (43)$$

Consequently, to minimize this expression, we have to choose a low value for $\mathbf{t}'$. This can be easily done by minimizing $\boldsymbol{v}$. Nevertheless, a compromise must be made since we have seen in Section III-B that $\partial\beta_i/\partial M_{mn}$ and $\partial\beta_i/\partial v_m$ are high when $\boldsymbol{v}$ is low (see Section VIII).

2) Second Step: During this step, both the reconstruction and the servoing are performed in order to achieve the positioning task and to improve the estimate $\widehat{\boldsymbol{\beta}}$ provided by the first step of the algorithm. In that case, if we performed low 3-D rotations between two consecutive frames (even if the desired rotation is high), the modeling error expresses as during the first step. Thus, if we set $t'_z = 0$ and ensure that the other translations are low, we will also minimize both the planar and the nonplanar parts of $\mathbf{E}$.

Besides, if the computation of the 2-D motion is not time consuming, leading thus to a low value for Δt (we will see that it is true afterward), a continuous approach can be used to recover $\boldsymbol{\beta}$, as seen in Section IV. In that case, the constraint $t'_z = 0$ becomes simply $v_z = 0$. Note that this choice $v_z = 0$ not only minimizes the modeling error but also leads to an optimal value for the condition number $\nu = 1$ [see (17)]. In addition, since the other translations are not null, we are sure that the matrix $\mathbf{C}^\top\mathbf{C}$ will never be singular and we also avoid the degenerate motions for the SfM (see Section III-B).

To illustrate those theoretical issues, a simulation concerning a positioning task has been carried out on the hyperboloid described in Section III-C with $R = 7$ cm (see Fig. 4). The initial pose w.r.t. the tangent plane in P was ${}^i\boldsymbol{\Phi} = (-6^\circ, 25^\circ, 0^\circ)$ and the initial depth of P was ${}^iZ_P = 80$ cm. Fig. 4(a) and (b) depict, respectively, the modeling error $E_W^2 = \sum_W \mathbf{E}^2$ during the first and the second steps for various choices of v_z. As can be seen, E_W^2 is really minimum in both cases when $v_z = 0$. Fig. 4(c) and (d) show, respectively, the behavior of the condition number for the first and the second step. In both cases, one can show that it can be very bad when $v_z \neq 0$, especially during the second step, near the desired position. In fact, the system (13) to solve becomes ill-conditioned. Besides, Fig. 4(e) and (f) confirm that using low 3-D velocities is better than using high values (under the condition that $v_z = 0$). Fig. 4(e) depicts the behavior of E_W^2 for various choices of $\|\boldsymbol{v}\|$, while Fig. 4(f) shows the behavior of E_W^2 for various choices of λ (as we shall see in Section VII, the parameter λ tunes the velocities: the higher λ is, higher are the velocities).

B. Using a Selection of Points

Instead of using active vision, we propose in this section to find a locus of points $\mathcal{L}(\mathbf{u})$, even far from the principal point, for which the modeling error $\mathbf{E}(\mathbf{u})$ vanishes as well for planar

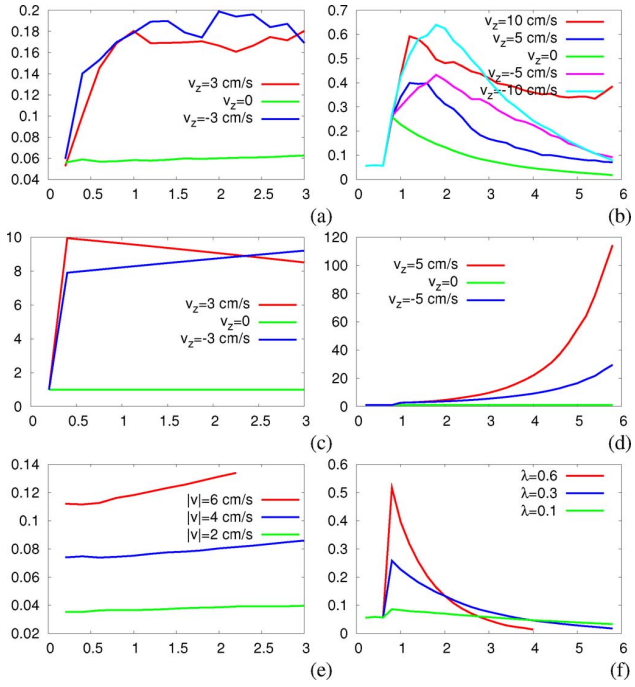

Fig. 4. Modeling error E_W^2 for various values of v_z versus time (x axes in seconds): (a) first step of the algorithm, (b) second step. Condition number ν for various values of v_z versus time: (c) first step, (d) second step. Modeling error E_W^2 for various values of $\|\boldsymbol{v}\|$ versus time (e). Modeling error E_W^2 for various values of $\boldsymbol{\lambda}$ versus time (f).

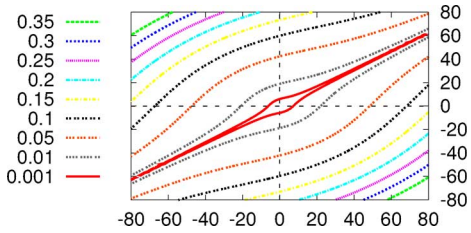

Fig. 5. $\|\mathbf{E}_{\text{planar}}(\mathbf{u})\|$ versus $\mathbf{u}$ (in pixels).

as for nonplanar objects. Once it will be found, we shall exploit it directly during the computation of the 2-D displacement as detailed next.

If we know that the camera is observing a planar object, then (37) shows that the locus where $\mathbf{E}(\mathbf{u}) = 0$ is simply the following straight line $\zeta(\mathbf{u}) = 0$ leading, by using the definition of $\mathbf{R}''$, to

$$\mathcal{L}(\mathbf{u}) : (R'_{31} + \beta_{10} t'_z)\, u + (R'_{32} + \beta_{01} t'_z)\, v = 0. \tag{44}$$

This straight line can be clearly seen in Fig. 1(b).

In the general case of an unknown object, vanishing $\mathbf{E}(\mathbf{u})$ is more difficult since $\zeta(\mathbf{u})$ is not necessarily factorized in $\delta\mathbf{E}(\mathbf{u})$. However, in practice, $\|\mathbf{E}_{\text{planar}}(\mathbf{u})\| \ll \|\delta\mathbf{E}(\mathbf{u})\|$. Fig. 5 depicts $\|\mathbf{E}_{\text{planar}}(\mathbf{u})\|$ in the same conditions as the simulation described in Section III-C. This figure has to be compared with Fig. 2(b) where $\|\mathbf{E}(\mathbf{u})\|$ is described. Thus, $\mathbf{E}_{\text{planar}}(\mathbf{u})$ can be neglected w.r.t. $\delta\mathbf{E}(\mathbf{u})$ in (39) and we have $\mathbf{E}(\mathbf{u}) \approx \delta\mathbf{E}(\mathbf{u})$. In addition, $\mathbf{E}(\mathbf{u})$ becomes proportional to $\varphi(\mathbf{u})$ [see (40)]. Now, we show that there is, most of the time, a locus $\mathcal{L}(\mathbf{u})$, where $\varphi(\mathbf{u})$, and thus $\mathbf{E}(\mathbf{u})$ is low, even far from $\mathbf{u} = \mathbf{0}$, and despite the fact that the unified model has been obtained from a Taylor series expansion around $\mathbf{u} = \mathbf{0}$.

For simplicity, we consider the class of objects where $\varphi(\mathbf{u}) = e_\beta(\mathbf{u})$ with

$$\varphi(\mathbf{u}) = e_\beta(\mathbf{u}) = \beta_{20} u^2 + \beta_{11} uv + \beta_{02} v^2 \tag{45}$$

where the parameters β_{ij} depend on the pose of the camera w.r.t. the object and on the object curvature. Note that $e_\beta(\mathbf{u})$ can also be seen as the second-order terms of the true depth (9). Consequently, (45) is valid for any object as soon its shape around P is $\mathcal{C}^2$.

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 (45) by expressing it in the frame of the eigenvectors of the matrix associated with it. Let s_1 and s_2 be the eigenvalues of this matrix. It is possible to rewrite (45) in the new frame as follows:

$$s_1 u'^2 + s_2 v'^2 - c = 0. \tag{46}$$

Three cases occur. The worst case appears when (46) describes an ellipse, i.e., when $s_1 s_2 > 0$. If $\text{sign}(s_1 s_2) \neq \text{sign}(c)$, $\mathcal{L}(\mathbf{u})$ is empty since (46) describes an imaginary ellipse. Otherwise, (45) 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

$$\mathcal{L}(\mathbf{u}) : \{v' + \sqrt{-s_1/s_2}\, u'\} \cup \{v' - \sqrt{-s_1/s_2}\, u'\}. \tag{47}$$

This case is very interesting since there is an infinite number of couples (u, v) for which (45) 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 (45) vanishes even if m is far from p. This locus is either the u' or the v' axis.

To illustrate this nice result, we return to the simulation described in III-C. We clearly see on Fig. 2(b) that (45) describes here the hyberbolas. We also guess the two straight lines $\mathcal{L}(\mathbf{u})$ given by (47) in Fig. 2(b).

Therefore, when we want to select points instead of using active vision, we propose to modify the way to compute the matrix $\mathbf{M}$ in Section V. Indeed, as can be seen in Fig. 2(b), selecting valid points m is very important, since even far from the center of the ROI, there is a locus where $\mathbf{E}(\mathbf{u})$ is null or very low (except in the case where (45) describes an ellipse). Therefore, our idea is to choose a value as high as possible for W according to the computation time (in order to not penalize the dynamics of the robot and the stability of the control scheme) and to select the points $\mathbf{u}$ that fit the true 2-D displacement, while the others are seen as outliers.

This can be done by using statistical techniques of robust estimation like the M-estimators [33]. Such approaches have been shown to be effective in various contexts (see, for example, [34] and [35]).

Indeed, it is commonly known that the ordinary least-squares estimator (35) produces a maximum likelihood estimation of

the parameters if the residuals are independent and normally distributed with constant standard variation. In that case, since the probability of occurrence of corrupted data is so small, the estimation of the parameters will be strongly affected by outliers to fit the whole set of data [36]. To cope with this problem, instead of minimizing (35), we prefer to compute $\hat{\boldsymbol{\mu}}$ as follows:

$$\widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \sum_{m \in W} \rho\left(e_m\right) \tag{48}$$

where the function ρ is interpreted as the negative logarithm of the probability density of residuals $e_m = r_m/\sigma$, where σ denotes the variance of residuals $r_m = f(m) - g(\boldsymbol{\delta}(m, \boldsymbol{\mu}))$ that fits with the model. Solving (48) is possible using the so-called *iteratively reweighted least-squares* algorithm (IRLS) [33]. To do that, an *influence function* [36] has to be chosen; we used the Tuckey's biweight function as proposed in [33]. Moreover, since the data contains outliers, computing σ is not a simple task. To do it robustly, it is usually done by computing the *median absolute deviation* (MAD) [37]. In fact, computing the MAD is very time consuming. Therefore, according to Section IV, a discrete 3-D reconstruction approach is required when selecting points are used. Indeed, the duration of the whole algorithm (computations of $\mathbf{M}$, $\hat{\boldsymbol{\beta}}$, and $\mathbf{v}$) is $\Delta t \approx 400$ ms (otherwise $\Delta t \approx 280$ ms).

C. Synthesis and Discussion

Two approaches have been described to enlarge the validity domain of the unified motion model. We have also seen that, in practice, $\mathbf{E}_{\text{planar}}(\mathbf{u}) \ll \delta\mathbf{E}(\mathbf{u})$ leading to $\mathbf{E}(\mathbf{u}) \approx \varphi(\mathbf{u})\boldsymbol{\xi}(\mathbf{u})$. The active vision approach minimizes $\boldsymbol{\xi}(\mathbf{u})$, while the selection of points chooses $\mathbf{u}$ so that $\varphi(\mathbf{u})$ is low. However, if the β_{ij}'s involved in $\varphi(\mathbf{u})$ are high (depending on the pose and on the object curvature), active vision requires slow 3-D velocities to minimize $\boldsymbol{\xi}(\mathbf{u})$ even if they cannot be too slow since a compromise has to be done (see Section VI-A1). Conversely, selecting points less depends on the value of the β_{ij}'s (if they are high or not) since a locus $\mathcal{L}(\mathbf{u})$ exists for which $\mathbf{E}(\mathbf{u})$ is null or very low (if $\varphi(\mathbf{u})$ is not an ellipse). Thus, this approach less depends on the pose and on the object curvature and also does not depend on the camera motion. Moreover, it also means that selecting points and using active vision simultaneously is not useful since selecting points is based on weaker assumptions.

On the other hand, note that during the realization of the task, it is not possible to evaluate the positioning error. Indeed, in the case of active vision, this error is proportional to $\varphi(\mathbf{u})$ (which is unknown), while in the case of selecting points, it depends on whether or not $\varphi(\mathbf{u})$ describes an ellipse.

Since we know how to enlarge the validity domain of the unified displacement model and thus to obtain the structure of the object reliably, it only remains to synthesize the control law. This is the subject of the next section.

VII. Control Law

First, let us remember the task to achieve. The goal is to ensure a given final orientation of the camera with respect to the tangent plane (π) described by (5) and, also to ensure that P will be observed at the principal point. Once $\boldsymbol{\beta}$ is estimated,

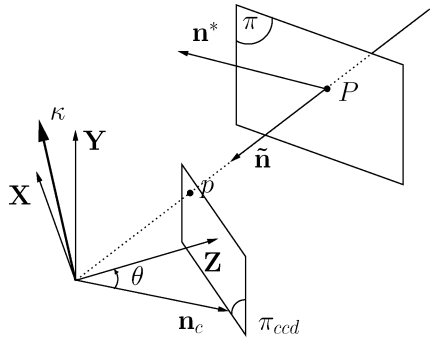

Fig. 6. Rotation to perform by the camera.

and therefore, $\boldsymbol{\alpha}$, the unit normal $\tilde{\mathbf{n}}$ of plane (π) at P in the camera frame can be derived (see Section III-A). To cope with a desired orientation of the camera w.r.t. the plane (π) different than parallel, we introduce $\mathbf{n}^*$ such that $\mathbf{n}^* = \mathbf{R}^*\tilde{\mathbf{n}}$, where the matrix $\mathbf{R}^*$ described the rotation between $\tilde{\mathbf{n}}$ and $\mathbf{n}^*$ (see Fig. 6). Therefore, we have to move the camera so that $\mathbf{Z} = \mathbf{n}_c$ with $\mathbf{n}_c = -\mathbf{n}^*$ and $\mathbf{Z}$ the unit vector carried by the optical axis. The rotation between $\mathbf{Z}$ and $\mathbf{n}_c$ can be expressed under the form $\boldsymbol{\kappa}\theta$ where $\boldsymbol{\kappa}$ represents the unit rotation axis vector and θ the rotation angle around this axis

$$\boldsymbol{\kappa} = \frac{\mathbf{n}_c \wedge \mathbf{Z}}{\|\mathbf{n}_c \wedge \mathbf{Z}\|} \tag{49}$$

and $\theta = \arccos(\mathbf{n}_c^\top \mathbf{Z})$.

The camera orientation being known, it is possible to compute the control law. We use the one described in [38]. Indeed, it ensures that P remains in the camera field of view since the trajectory of p is a straight line between the current position p and the desired position p^* (which has been chosen as the principal point of the image). We describe here briefly this approach known as *hybrid visual servoing*.

First, $\mathbf{x}_r$ is defined as follows

$$\mathbf{x}_r = \frac{1}{Z^*}\mathbf{X}_\mathbf{P} = \frac{Z_P}{Z^*}\mathbf{X}_\mathbf{P} \tag{50}$$

with Z^* the desired depth for P in final position.

In few words, this approach is based on the regulation to zero of the following task function $\mathbf{e} = (\mathbf{x}_r - \mathbf{x}_r^*, \boldsymbol{\kappa}\theta)$ yielding the camera velocity $\mathbf{v} = -\lambda\widehat{\mathbf{L}}^{-1}\mathbf{e}$, λ being a positive gain and $\widehat{\mathbf{L}}^{-1}$ the inverse of an approximation of the interaction matrix given by [38]

$$\widehat{\mathbf{L}}^{-1} = \begin{bmatrix} -Z^*\mathbf{I}_3 & Z^*[\mathbf{x}_r]_\times \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix} \tag{51}$$

with $[\mathbf{x}_r]_\times$ the antisymmetric matrix associated to $\mathbf{x}_r$.

Note that if active vision is used, we set $v_z = 0$, and thus, $Z^* = Z_P$. However, if a motion along the z-axis is required, we propose first to realize the positioning task regardless of the desired depth Z^*, and once it has been realized, to realize a second task to reach Z^* and to ensure that p still coincides with the principal point.

Let us note that the value of Z_P required for the computation of $\mathbf{x}_r$ is obtained by using (7), while $\mathbf{x}_\mathbf{p}$ is obtained by the tracking algorithm described in [39] since it can cope with difficult lighting conditions.

TABLE I
SYNTHESIS OF THE EXPERIMENTS

Goal: comparison between	Object	Figure nb.	Initial orientation (in °)	Initial depth (in cm)	Desired orientation (in °)	Final orientation (in °)
Using or not using active vision						
◇ Experiment 1	Planar	7 8	-1.1, -15.1, 4.7	93.3	0, 0	using : 0.3, -0.9 not using : -1.4, -0.9
◇ Experiment 2	Cylinder	- -	13.4, 8.7, 4	82.7	0, 0	using : 0.8, 1.1 not using : 1.6, 1.5
◇ Experiment 3	Cylinder	9 -	-1.5, -4.8, 4.7	86.7	15, 0	using : 14.2, 1.9 not using : 13.0, 3.0
Selecting points or not						
◇ Experiment 1	Planar	- 8	-1.1, -15.1, 4.7	93.3	0, 0	selecting : 0.2, -0.7 not selecting : 1.4, -0.9
◇ Experiment 4	Sphere	10 -	-17.0, -17.2, 1.9	71.5	0, 0	selecting : 0.4, 0.4 not selecting : -4.8, 2.8
Active vision or selecting points						
◇ Experiment 4	Sphere	10 -	-17.0, -17.2, 1.9	71.5	0, 0	selecting points : 0.4, 0.4 active vision :-0.1, 0.5
◇ Experiment 5	Sphere	- -	4.2, -5.2, 3.2	70.9	20,20	selecting : 19.5, 20.3 active vision : 16, 16

VIII. EXPERIMENTAL RESULTS

In order to validate the proposed algorithm, we present here some experimental results. First, we detail the way we measure the orientation error with respect to the tangent plane in P to validate our results. We assume that a model of the object being observed by the camera exists in a certain frame $\mathcal{F}_f$. Knowing this model and the pose between $\mathcal{F}_f$ and the camera frame $\mathcal{F}_c$ (using four dots, as can be seen in the next figures), it is possible to obtain from p the coordinates of P by intersecting the line of view with the object. Consequently, it becomes easy to compute the orientation of the tangent plane at P with respect to $\mathcal{F}_c$. This orientation is described, as in simulation, by the Cardan's angles $\boldsymbol{\Phi}$ (respectively, pitch, yaw, and roll). However, we are not interested in the roll angle. We introduce the following notations concerning the superscript of $\boldsymbol{\Phi}$: i for the initial rotation, d for the desired one, and m for the measured one.

The experimental system consists of a 6-DOF robot with an eye-in-hand charge-coupled device CCD camera. 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 is chosen from the initial image by the well-known Harris detector.

Note that the accuracy of $\boldsymbol{\alpha}$, required for the control law, can be improved in practice. Indeed, since the object is motionless, one can express a value $\boldsymbol{\alpha}^f$ in a fixed frame that can be filtered, since a fixed value has to be obtained. Thereafter, this filtered value is then expressed in the camera frame to be used in the control law. Once $\boldsymbol{\alpha}^f$ is sufficiently stable (it is the goal of the first step described in Section VI-A1; its duration is typically 7 iterations), the servoing step can begin. This step ends when the mean of the 2-D motion is lower than 1/4 pixel (this criterion has been used to easily compare the different approaches and to clearly show that $\sigma^2_{\beta_i}$ [see (18)]) becomes high when the velocities are too low) and a last step begins. It consists in a servoing step without any new 3-D reconstruction. This step uses $\boldsymbol{\alpha}^f$, considered as constant, which is expressed in the camera frame.

The following constants have been used during all the experiments: $W = 111 \times 111$ (that means 55 pixels each side of p), $\|\boldsymbol{v}\| = 3$ cm/s during the first step, $\lambda = 0.3$ during the second step (this value ensures that the 3-D velocities are high enough to minimize $\sigma^2_{\beta_i}$ [see (18)]) at the begining of the motion and low enough to minimize the modeling error $\mathbf{E}(\mathbf{u})$ while $\lambda = 1$ during the last one.

Three objects have been used for the experiments, a planar object, a cylinder, and a sphere of radius 7 cm. Of course, our algorithm does not know which object is being observed by the camera. Note that the curvature of the nonplanar objects is high. Consequently, since $\varphi(\mathbf{u})$ increases when the curvature increases, the experiments describe complex cases according to (40). Table I details all these experiments.

A. Comparison Between Using or Not Using Active Vision

The goal of this section is to show the efficiency of active vision by comparing it to when it is not used. Thus, selecting points is not used at all (the 2-D motion is computed as described in Section V). In addition, the 3-D reconstruction is performed by using a continuous approach since $\Delta t \approx 280$ ms.

The first experiment concerns the planar object and consists in positioning the camera parallel to the object when using active vision. Fig. 7(a) depicts the components of the camera velocity; Fig. 7(b) the norm of the task function $\mathbf{e}$; Fig. 7(c) the magnitude of the rotation θ to reach the desired orientation; and Fig. 7(d) the behavior of $\boldsymbol{\alpha}$ (filtered and non-filtered) expressed in a fixed frame. Finally, the initial and final images are reported, respectively, in Fig. 7(e) and (f), where the trajectory of p is also depicted (the square shows the window where the 2-D motion is computed). First, Fig. 7(b) confirms that the control law converges since $\|\mathbf{e}\|$ tends toward zero. One can also clearly note in Fig. 7(a) the three steps of the algorithm (the last step begins near 10 s). The orientation after servoing was ${}^m\boldsymbol{\Phi} = (0.3°, -0.9°)$ (recall that we are not interested in the last component of $\boldsymbol{\Phi}$). Consequently, we obtained an accurate positioning. In addition, we performed the same task without using active vision in the second step [Fig. (8)]. Here again, the control law converged [Fig. 8(b)]. Nevertheless, we obtained a higher positioning error since we had ${}^m\boldsymbol{\Phi} = (-1.4°, -0.9°)$. In addition, Fig. 8(d) shows that $\widehat{\boldsymbol{\alpha}}$ is more noisy than when active vision is used [see Fig. 7(d)]. This result was expected since active vision ensures

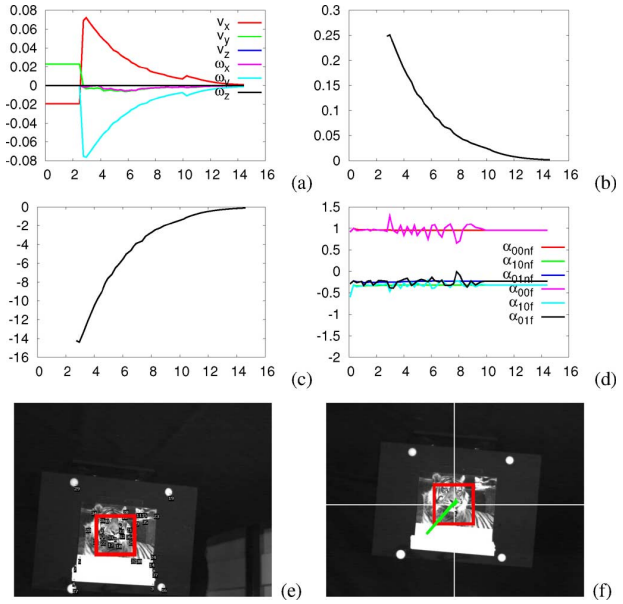

Fig. 7. 1st experiment (x-axis in seconds). Active vision. ${}^{d}\mathbf{\Phi} = \mathbf{0}$. (a) $\mathbf{v}$ (meters/second or radians/second). (b) Error defined as $\|\mathbf{e}\|$. (c) Magnitude θ of the rotation (deg.) (d) Vector $\boldsymbol{\alpha}$ in a fixed frame (filtered and non-filtered). (e) Initial image. (f) Final image.

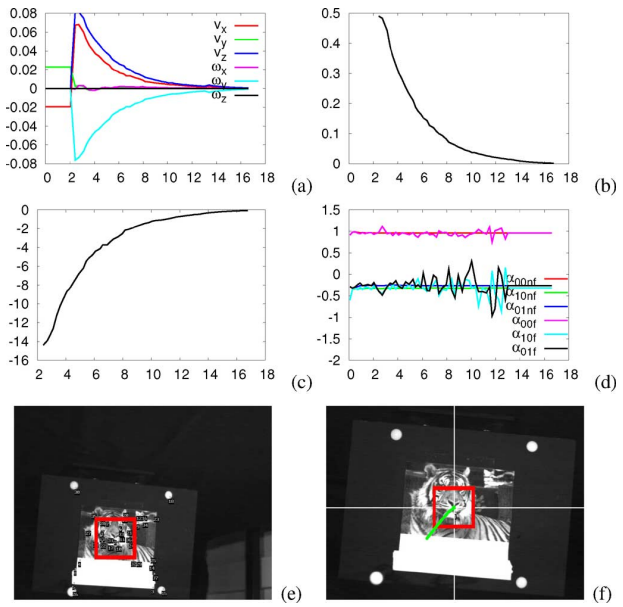

Fig. 8. First experiment. Without active vision. ${}^{d}\mathbf{\Phi} = \mathbf{0}$.

an optimal condition number of the system involved in the computation of $\hat{\boldsymbol{\beta}}$.

The second experiment concerns a positioning task with respect to the cylinder when ${}^{d}\mathbf{\Phi} = \mathbf{0}$ and when using active vision. After servoing, we obtained ${}^{m}\mathbf{\Phi} = (0.8^{\circ}, 1.1^{\circ})$. Here again, this result is better than whithout active vision since we obtained in that case ${}^{m}\mathbf{\Phi} = (1.6^{\circ}, 1.5^{\circ})$. We also performed a third experiment by setting ${}^{d}\mathbf{\Phi} = (15^{\circ}, 0^{\circ})$. Fig. 9 depicts the same parameters as in Fig. 7 and confirms that the control law converges without any problem. For this experiment, we obtained ${}^{m}\mathbf{\Phi} = (14.2^{\circ}, 1.9^{\circ})$. Without active vision we also obtained a bad result since we had ${}^{m}\mathbf{\Phi} = (13.0^{\circ}, 3.0^{\circ})$.

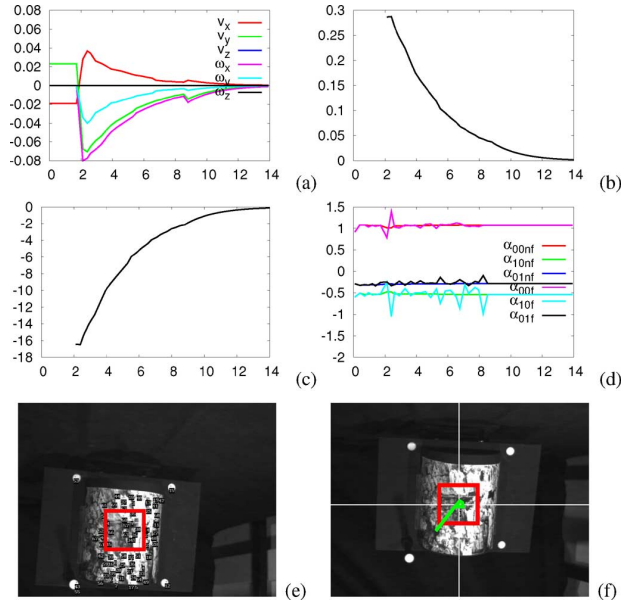

Fig. 9. Third experiment. Active vision. Cylinder. ${}^{d}\mathbf{\Phi} = (15^{\circ}, 0^{\circ})$.

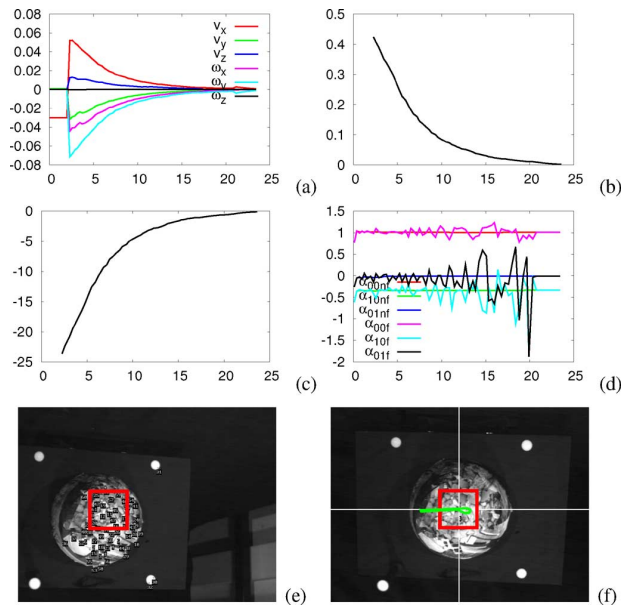

Fig. 10. Fourth experiment. Robust estimation. Sphere. ${}^{d}\mathbf{\Phi} = \mathbf{0}$.

B. Comparison Between Selecting Points or Not

Here we compare the benefit of selecting points or not. Thus, we compare the selection of points by robust estimation coupled with a discrete approach (since Δt is high) with the computation of the 2-D motion, as described in Section V, and coupled with a continuous approach (since Δt is low).

First, let us return to the first experiment. In that case, selecting points yields better results since we obtained ${}^{m}\mathbf{\Phi} = (0.2^{\circ}, -0.7^{\circ})$.

A fourth experiment has been carried out and concerns the sphere. It consists in positioning the camera parallel to the tangent plane at P so that $Z_P^* = 65$ cm. Fig. 10 describes the behavior of the algorithm when using the selection of points.

Here again, the control law converges without any problem [see Fig. 10(b)]. The orientation after servoing was ${}^m\mathbf{\Phi} = (0.4^\circ, 0.4^\circ)$ when using robust estimation. Without using robust estimation, a higher orientation error had been obtained since we had ${}^m\mathbf{\Phi} = (-4.8^\circ, 2.8^\circ)$. The benefit of using a robust estimation is clear. However, we can see in Fig. 10(d) that $\widehat{\boldsymbol{\alpha}}$ is more noisy than when active vision is used [see Fig. 7(d) or 9(d)] or not [Fig. 8(d)]. This can be easily explained. These approaches use a continuous approach, and thus, $\widehat{\boldsymbol{\alpha}}$ is obtained from a linear system, while the selection of points is based on a discrete approach, which requires the resolution of a nonlinear system. As seen in Fig. 10(d), this system becomes ill-conditioned when the 3D velocities become low. However, our filter (mentioned at the beginning of this section) is also robust and does not take into account outliers. That also shows that a valid criterion to stop the second step is required (see Section IX).

C. Comparison Between Using Active Vision or Selecting Points

The previous Section VIII-A and VIII-B have shown that both approaches enlarge the validity domain of the unified 2-D motion model. In this section, we compare them. Recall that active vision is used with a continuous approach, $Z^* = {}^iZ_P$ and with the 2-D motion computed, as described in Section V. The selection of points is coupled with a discrete approach and with $Z^* = 65$ cm.

Thus, we performed again the fourth experiment but using active vision. In that case, we obtained also a very low positioning error since we had ${}^m\mathbf{\Phi} = (-0.1^\circ, 0.5^\circ)$. Note that theoretically, selecting points does not require low 3-D velocities. However, in practice, the computation cost required for the selection is so heavy that Δt is high. Therefore, low 3-D velocities have to be used. In practice, they are even lower than those used with active vision [Fig. 10(a) has to be compared with Fig. 7(a) and Fig. 9(a)].

The fifth and last experiment has been carried out on the same sphere with ${}^d\mathbf{\Phi} = (20^\circ, 20^\circ)$. Unfortunately, in the case of active vision, we obtained bad results, the orientation error being around 4°. In contrast, we obtained ${}^m\mathbf{\Phi} = (19.5^\circ, 20.3^\circ)$ using the selection of points. Thus, this result is very good. Indeed, this result validates the theoretical issues of Section VI-C: when the β_{ij}'s are high, active vision cannot perform 3-D velocities low enough to minimize the modeling error. Conversely, selecting points does not care if their values are high or not.

IX. Discussion and Conclusion

This paper has presented a way to achieve accurate visual servoing tasks with respect to unknown objects, planar or not, and in the complex case when the desired visual features are also assumed to be unknown. To do that, we recover the parameters of the tangent plane at a certain point of the object and introduce them in a control law. Our approach is based on the computation of the 2-D displacement between two consecutive frames contrary to other approaches where, either the current and the desired frames are required or a feature matching between two consecutive frames is needed, as for discrete approaches. Consequently, our technique enlarges the application domain of visual servoing to complex scenes as, for example, natural scenes where matching is known to be a difficult task.

More precisely, our algorithm is based on a unified 2-D displacement model to cope as well with planar as with nonplanar objects. To our knowledge, previous work always required explicit constraints about the scene or used complex model selection mechanisms. Since our unified motion model is only an approximation of the true one, we have proposed two ways to enlarge its domain of validity.

The first one is based on active vision. Theoretical issues have shown that low 3-D displacements (especially $t'_z = 0$) between two consecutive frames minimizes the modeling error between the true and the approximated motion model. Experiments have validated that this approach leads to good results. Nevertheless, when the curvature of the object and the desired orientation are high, we have seen that a poor accuracy could be obtained. In addition, since the computation of the 2-D motion dedicated to this approach is not time consuming, active vision is used with a 3-D reconstruction based on a continuous approach. It allows to recover the structure from a linear system with an optimal condition number. Consequently, it provides less noisy results than the other approaches.

The other approach is based on the selection of points involved in the computation of the 2-D displacement. Indeed, we have proved that there is always a locus where the difference between the true and the approximated displacement models is very low or vanishes. To select the points, M-estimators have been introduced. However, this approach leads to a low acquisition rate (around 400 ms) and requires, therefore, a discrete approach to be really effective. Thus, the 3-D reconstruction is performed through the resolution of a nonlinear system and provides, consequently, more noisy results than a continuous approach like active vision. In addition, with a low acquisition rate, only slow camera motion can be considered. It can be seen as the main drawback of this approach. Conversely, the computation cost of using active vision is low (around 280 ms), and thus, higher 3-D velocities can be reached in practice. On the other hand, selecting points is a nice way to cope with the problem of the choice of the window size required to compute the 2-D displacement. We simply chose a large one to be more robust against noisy images and used the robust estimation process to select points that fit the true model. Besides, this approach has led to better results than active vision under complex conditions (high curvature and desired orientation).

Concerning future work, an important issue is to know when the 3-D reconstruction has to be stopped since we have seen that the parameters of the tangent plane become noisy otherwise. Therefore, a criterion concerning the 3-D motion has to be found. It could be based on a measurement of $\partial\beta_i/\partial M_{mn}$ or $\partial\beta_i/\partial v_m$ to ensure a low uncertainty on $\widehat{\boldsymbol{\beta}}$.

Acknowledgment

The authors would like to thank S. Hutchinson for his contribution to the improvement of the readability of this paper. The authors would also like to thank the anonymous reviewers for their corrections and their constructive criticism.

REFERENCES

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[2] F.-X. Espiau, E. Malis, and P. Rives, "Robust features tracking for robotic applications: Towards 2d1/2 visual servoing with natural images," presented at the IEEE Int. Conf. Robot. Autom. (ICRA 2002), Washington, DC, May 11–15.

[3] M. Gouiffès, C. Fernandez-Maloigne, A. Trémeau, and C. Collewet, "Color segmentation of inked characters: Application to meat traceability control," in *Proc. IEEE Int. Conf. Imag. Process. (ICIP 2004)*, Singapore, Oct. 24–27, pp. 195–198.

[4] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, Oct. 1996.

[5] E. Malis and F. Chaumette, "2 1/2d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 79–97, 2000.

[6] F. Schramm, G. Morel, A. Micaelli, and A. Lottin, "Extended-2d visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA 2004)*, New Orleans, LA, Apr. 26–May 1, pp. 267–273.

[7] J. Santos-Victor and G. Sandini, "Visual behaviors for docking," *Comput. Vis. Imag. Understanding*, vol. 67, no. 3, pp. 223–238, Sep. 1997.

[8] A. Crétual and F. Chaumette, "Visual servoing based on image motion," *Int. J. Robot. Res.*, vol. 20, no. 11, pp. 857–877, Nov. 2001.

[9] A. Alhaj, C. Collewet, and F. Chaumette, "Visual servoing based on dynamic vision," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA 2003)*, Taipei, Taiwan, Sep. 14–19, pp. 3055–3060.

[10] C. Collewet, A. Alhaj, and F. Chaumette, "Model-free visual servoing on complex images based on 3d reconstruction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA 2004)*, New Orleans, LA, Apr. 26–May 1, pp. 751–756.

[11] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," in *Proc. R. Soc. Lond.*, vol. B208, pp. 385–397, 1980.

[12] B. K. P. Horn and B. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 16, no. 1–3, pp. 185–203, Aug. 1981.

[13] M. A. Taalebinezhaad, "Direct recovery of motion and shape in the general case by fixation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 847–853, Aug. 1992.

[14] G. P. Stein and A. Shashua, "Model-based brightness constraints: On direct estimation of structure and motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 992–1015, Sep. 2000.

[15] M. Irani, B. Rousso, and S. Peleg, "Recovery of ego-motion using region alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 268–272, Mar. 1997.

[16] A. Calway, "Recursive estimation of 3d motion and surface structure from local affine flow parameters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 562–574, Apr. 2005.

[17] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418–433, Mar. 2005.

[18] T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," *Proc. IEEE*, vol. 82, no. 2, pp. 252–268, Feb. 1994.

[19] J. Roach and J. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 6, pp. 554–562, Nov. 1980.

[20] R. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.

[21] Q. T. Luong and O. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *Int. J. Comput. Vis.*, vol. 17, no. 1, pp. 43–75, 1996.

[22] H. Longuet Higgins, "The visual ambiguity of a moving plane," in *Proc. R. Soc. Lond.*, vol. B223, pp. 165–175, 1984.

[23] P. Torr, A. W. Fitzgibbon, and A. Zisserman, "Maintaining multiple motion model hypotheses over many views to recover matching and structure," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV 98)*, pp. 485–491.

[24] S. Soatto and P. Perona, "Reducing 'structure from motion': A general framework for dynamic vision part 1: Modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 9, pp. 933–942, Sep. 1998.

[25] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 577–589, Jun. 1998.

[26] K. Schindler and D. Suter, "Two-view multibody structure-and-motion with outliers through model selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 983–995, Jun. 2006.

[27] M. J. Black and A. D. Jepson, "Estimating optical-flow in segmented images using variable-order parametric models with local deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 972–986, Oct. 1996.

[28] A. Alhaj, "Apports de la vision dynamique en asservissement visuel" Ph.D. dissertation, Univ. Rennes I, Rennes, France, Jun. 2004, (in French).

[29] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Improving feature tracking with robust statistics," *Pattern Anal. Appl.*, vol. 2, pp. 312–320, 1999.

[30] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR 94)*, Seattle, WA, Jun.1994, pp. 593–600.

[31] G. D. Hager and K. Toyama, "Incremental focus of attention for robust visual tracking," *Int. J. Comput. Vis.*, vol. 35, no. 1, pp. 45–63, Nov. 1999.

[32] Y. Aloimonos, "What I have learned," *CVGIP: Imag. Understanding*, vol. 60, no. 1, pp. 74–85, 1994.

[33] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[34] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. Vis. Commun. Imag. Represent.*, vol. 6, no. 4, pp. 348–365, Dec. 1995.

[35] J.-P. Tarel, S.-S. Ieng, and P. Charbonnier, "Using robust estimation algorithms for tracking explicit curves," in *Proc. Eur. Conf. Comput. Vis., Part I (ECCV 02)*, 2002, pp. 492–507.

[36] F. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley, 1986.

[37] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[38] E. Malis, F. Chaumette, and S. Boudet, "2 1/2d visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[39] M. Gouiffès, C. Collewet, C. Fernandez-Maloigne, and A. Trémeau, "Feature points tracking: Robustness to specular highlights and lighting changes," in *Proc. Eur. Conf. Comput. Vis. (ECCV 2006)*, Graz, Austria, May 7–13.

Christophe Collewet (M'07) received the Graduate degree in automatic control from École Nationale Supérieure d'Ingénieurs de Caen, Caen, France, in 1986, and the Ph.D. degree in signal processing from the University of Rennes, Rennes, France, in 1999.

From May 1988 to October 2005, he was with Cemagref, the French institute of agricultural and environmental engineering research as "Chargé de Recherche". He is currently on secondment at IRISA/INRIA, Rennes, France, in the Lagadic Group (http://www.irisa.fr/lagadic). His current research interests include robotics, image processing, especially visual features tracking and vision-based control.

François Chaumette (M'98) received the Graduate degree from École Nationale Supérieure de Mécanique, Nantes, France, in 1987, and the Ph.D. degree in computer science from the University of Rennes, Rennes, France, in 1990.

Since 1990, he has been with IRISA/INRIA Rennes Bretagne Atlantique, where he is "Directeur de recherche" and Head of the Lagadic Group. His research interests include robotics and computer vision, especially visual servoing and active perception. He is the coauthor of more than 150 papers published in international journals on the topics of robotics and computer vision.

Dr. Chaumette has served over the last 5 years on the program committees for the main conferences related to robotics and computer vision. He has been the Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS from 2001 to 2005. He was the recipient of several awards including the AFCET/CNRS Prize for the Best French Thesis on Automatic Control in 1991, King-Sun Fu Memorial Best IEEE TRANSACTIONS ON ROBOTICS and AUTOMATION Paper Award with Ezio Malis in 2002.