The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

On-line estimation of the reference visual features Application to a vision based long range navigation task

Adrien Durand Petiteville, Michel Courdesses, Viviane Cadenat and Philippe Baillion

Abstract— **In this paper, we consider the problem of realizing a vision-based navigation task. To this aim, we present an algorithm allowing to automatically compute the necessary reference visual features. This algorithm relies on a predictor/estimator pair able to determine the visual features depth sufficiently rapidly with respect to the control law sampling period. The proposed method can then be used on-line. An example of such an application is presented through the realization of a vision-based long range navigation task. Both simulation and experimentation results validate the whole approach.**

I. INTRODUCTION

In the past decades, many works have addressed the problem of using information provided by a vision system to control a robot. Such techniques are commonly known as Visual Servoing [Corke, 1996], [Chaumette and Hutchinson, 2006] and [Hutchinson et al., 1996]. Visual servoing is roughly classified into two main categories: Image based visual servoing (IBVS) and Position based visual servoing (PBVS) [Chaumette and Hutchinson, 2006]. In the first approach, the goal to be reached is expressed only in the image space, whereas in the second one, it is given in terms of a desired camera pose [Chaumette and Hutchinson, 2006]. Some contributions combine these two basic approaches to improve the task execution [Fang et al., 2005] [Malis et al., 1999]. A complete survey can be found in [Chaumette and Hutchinson, 2006].

We focus in the sequel on the first kind of control. In this case, the controller relies on the current visual features and on their reference values. If the former can be directly obtained from the image, the literature provides two main approaches to determine the latter. Either the robot is positioned at the goal and then an image is shot; or the reference configuration to be reached and the target model are known and the perspective projection relations can be applied [Chaumette, 2002]. The first method requires to manually move the robot to the desired location, which may appear as a restriction of its autonomy as the task objective is to reach this configuration automatically. The second one allows to overcome this drawback, but requires a particular knowledge about the environment (target dimensions and location), which is a strong assumption. If the landmark model is not available, an estimation of the depth is necessary to reconstruct it in the world frame. Different approaches can be used. See for instance the works [Matthies et al., 1989] by Matthies who derived and compared several algorithms based on a Kalman filter. A correct depth value can be obtained only if the camera motion respects some very particular constraints. It would also be possible to use vision methods like the epipolar geometry [Ma et al., 2003] [Basri et al., 1999], stereovision [Cervera et al., 2002], or even structure from motion techniques [Jerian and Jain, 1991]. In our particular case, the estimation process is intended to be used in the control law computation. Therefore the depth calculation time must be the smallest possible. Although the previous methods are used in real time in the vision community, a worthwhile solution consists in using a technique which is consistent with the control law sampling time. De luca et al propose such a method which consists in using nonlinear observer to estimate the depth [De Luca et al., 2008]. However the obtained convergence time seems to be a bit large with respect to our purpose. As a conclusion, the above mentioned approaches cannot be used in a control context. This is the reason why we have developed a predictor/estimator pair based on a specific algorithm presented in [Folio and Cadenat, 2008] and described in the sequel.

The paper is organized as follows. Section 2 is dedicated to system modelling. Section 3 details the automatic computation of the reference visual features using a predictor/corrector pair. Experimental results validating the proposed algorithm are presented in the next section. Finally, the method is applied in the specific context of a long range vision-based navigation task. Some simulation results show the interest of the technique in this particular case.

II. MODELLING

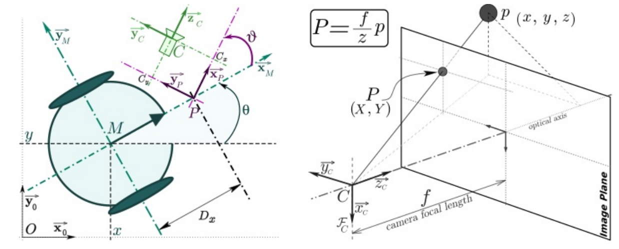
A. System Modelling

We consider the system presented in figure 1(a), which consists of a robot equipped with a camera mounted on a pan-platform. We describe the successive frames : $F_O(O,\vec{x}_O,\vec{y}_O,\vec{z}_O)$ attached to the world, $F_M(M,\vec{x}_M,\vec{y}_M,\vec{z}_M)$ linked to the robot, $F_P(P,\vec{x}_P,\vec{y}_P,\vec{z}_P)$ attached to the platform, and $F_C(C,\vec{x}_C,\vec{y}_C,\vec{z}_C)$ linked to the camera. Let θ be the direction of the robot wrt. $\vec{x}_O$, ϑ the direction of the pan-platform wrt. $\vec{x}_M$, P the pan-platform centre of rotation and D_x the distance between the robot reference point M and P.

Defining vector $q=(l,\theta,\vartheta)^T$ where l is the robot curvilinear abscissa, the control input is given by $\dot{q}=(\upsilon,\omega,\varpi)^T$, where υ and ω are the cart linear and angular velocities, and ϖ is the pan-platform angular velocity wrt. F_M. For

Durand Petiteville, Courdesses and Cadenat are with CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France; Email [adurandp, cadenat, courdess]@laas.fr

Baillion is with AIP-PRIMECA Université de Toulouse UPS 118 route de Narbonnes, F-31062 Toulouse, France; Email Philippe.Baillion@cict.fr



(a) The robotic system (b) The camera pinhole model

Fig. 1. System modeling

such a robot, the kinematic model is classically given by the following relations (1) where $\dot{M}_x(t)$ is the speed of M wrt. $\vec{x}_O$ and $\dot{M}_y(t)$ wrt. $\vec{y}_O$.

$$\begin{cases} \dot{M}_x(t) = \upsilon(t)\cos(\theta(t)) \\ \dot{M}_y(t) = \upsilon(t)\sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \\ \dot{\vartheta}(t) = \varpi(t) \end{cases} \tag{1}$$

The camera motion can be described by the kinematic screw:

$$T_{C/F_O} = \left[(V_{C/F_O})^T(\Omega_{F_C/F_O})^T)\right]^T \tag{2}$$

where V_{C/F_O} and Ω_{F_C/F_O} are the camera translation and rotation speeds wrt. the frame F_O. For this specific mechanical system, T_{C/F_O} is related to the control input by the robot jacobian J : $T_{C/F_O} = J\dot{q}$. As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw $T_r = (V_{\vec{y}_C}, V_{\vec{z}_C}, \Omega_{\vec{x}_C})^T$, and a reduced jacobian matrix J_r as follows: $T_r = J_r\dot{q}$ where J_r is given by:

$$J_r = \begin{pmatrix} -\sin(\vartheta(t)) & D_x\cos(\vartheta(t)) + C_x & C_x \\ \cos(\vartheta(t)) & D_x\sin(\vartheta(t)) - C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \tag{3}$$

B. Visual Data

The vision-based navigation task consists in positioning the camera with respect to a given static landmark using visual data. To this aim the landmark is characterized by n interest points which can be extracted by our image processing. The visual data are represented by a $2n$-dimensional vector s made of the coordinates (X_i, Y_i) of each point P_i, in the image plane as shown on figure 1(b). The value of s depends on the relative camera position with respect to the landmark. So the goal of image based visual servoing is to make converge the current visual signals s to their reference values s^*. s^* then corresponds to the value of s obtained at the desired camera pose with respect to the target.

For a fixed landmark, the variation of the visual signal $\dot{s}$ is related to the reduced camera kinematic screw T_r thanks to the interaction matrix $L_{(s,z)}$ as shown below [Espiau et al., 1992]:

$$\dot{s} = L_{(s,z)}T_r = L_{(s,z)}J_r\dot{q} \tag{4}$$

In the case of n points, $L_{(s,z)} = [L^T_{(P_1)}, \ldots, L^T_{(P_n)}]^T$ where $L_{(P_i)}$ is classically given by [Espiau et al., 1992]:

$$L_{(P_i)} = \begin{pmatrix} L_x(s_i, z_i) \\ L_y(s_i, z_i) \end{pmatrix} = \begin{pmatrix} 0 & \frac{X_i}{z_i} & \frac{X_iY_i}{f} \\ -\frac{f}{z_i} & \frac{Y_i}{z_i} & f + \frac{Y_i^2}{f} \end{pmatrix} \tag{5}$$

z_i represents the depth of the projected point p_i, and f is the camera focal (see figure 1(b)).

III. AUTOMATIC DETERMINATION OF VISUAL FEATURES S^*

As seen in the introduction, in the case where the landmark dimensions are unknown, the automatic determination of the visual features s^* requires an accurate value of the depth. For this reason, we first present a predictor/estimator pair to estimate z_i, before describing an optimal method allowing to determine s^*.

A. Reconstruction Of The Depth

Recently, to handle occlusion problems, [Folio, 2007] has proposed to solve the dynamic system (4) to obtain the expression of the visual data. However, the latter depends not only on s but also on depth z which must then be determined. As our robot is not equipped with any sensor able to measure this data, we have to reconstruct it. After some computations (see [Folio, 2007] for a detailed proof), it can be shown that, for any $t \in [t_{k-1}, t_k]$, X_i, Y_i and z_i express as:

$$\begin{cases} X_i(t) = \frac{z_i(k-1)X_i(k-1)}{z_i(t)} \\ Y_i(t) = \frac{f}{z_i(t)}\left\{D_x\sin(\vartheta(t)) + \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(t)) - C_y\right. \\ \qquad +c_1\cos(A(\dot{q}(k-1))(t - t_{k-1})) \\ \qquad \left.-c_2\sin(A(\dot{q}(k-1))(t - t_{k-1})\right\} \\ z_i(t) = -D_x\cos(\vartheta(t)) + \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(t)) - C_x \\ +c_1\sin(A(\dot{q}(k-1))(t - t_{k-1})) + c_2\cos(A(\dot{q}(k-1))(t - t_{k-1})) \end{cases} \tag{6}$$

where:

$$\begin{cases} A(\dot{q}(k-1)) = \omega(k-1) + \varpi(k-1) \\ c_1 = \frac{Y_i(k-1)z_i(k-1)}{f} - D_x\sin(\vartheta(k-1)) - \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1)) + C_y \\ c_2 = z_i(k-1) + D_x\cos(\vartheta(k-1)) - \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1)) + C_x \end{cases}$$

Thanks to (6), Folio has developed a recursive algorithm able to estimate X_i, Y_i and z_i provided that $\vartheta(t)$ has been previously determined[1]. However, it should be noted that initial conditions, namely $X_i(k-1)$, $Y_i(k-1)$ and $z_i(k-1)$, are required to determine $X_i(k)$, $Y_i(k)$ and $z_i(k)$. So, this algorithm cannot be used to properly estimate $z_i(k)$ without a precise initial value of $z_i(k-1)$. This is the reason why we have modified D. Folio's algorithm to estimate $z_i(k)$ using only the visual data.

In this paper, we propose to estimate the depth by building a predictor/estimator pair using data from m images, with $m \in \mathbb{N}^*$ to repair a too small signal/noise ratio [Durand Petiteville et al., 2009]. Our first objective is to express a predictor $\hat{X}_i(k|k-j), \hat{Y}_i(k|k-j)$ of one point P_i at instant k using the image at $k-j$, with $j \in [1, \ldots, m]$. To

[1] As one can see, solution (6) requires the determination of $\vartheta(t)$. This angle can be simply computed by integrating $\dot{\vartheta} = \varpi$ between t_{k-1} and t. A straightforward calculus leads to $\vartheta(t) = \varpi(k-1)(t - t_{k-1}) + \vartheta(k-1)$, where $\vartheta(k-1)$ is the pan-platform angular value at t_{k-1}, which is usually provided by the embedded encoder.

this aim, we rewrite equation (6) to relate $z_i(k-1|k-1)$ and $z_i(k|k-1)$. We obtain:

$$\hat{z}_i(k-1|k-1) = \frac{\hat{z}_i(k|k-1) - \beta}{\alpha_i} \tag{7}$$

where α_i is given in the appendix. Denoting by $\tilde{X}_i(k-1)$ and $\tilde{Y}_i(k-1)$ the visual data measured at instant $k-1$, we use (7) in (6) to obtain the predictor for the visual features:

$$\begin{cases} \hat{X}_i(k|k-1) = \frac{\hat{z}_i(k|k-1)\tilde{X}_i(k-1) - \beta\tilde{X}_i(k-1)}{\hat{z}_i(k|k-1)\alpha_i} \\ \hat{Y}_i(k|k-1) = \frac{f}{\hat{z}_i(k|k-1)}\left(\frac{\tilde{Y}_i(k-1)\hat{z}_i(k|k-1)}{f\alpha_i}\cos(A(\dot{q}(k-1))T)\right. \\ \qquad \left. - \frac{\hat{z}_i(k|k-1)}{\alpha_i}\sin(A(\dot{q}(k-1))T) + \kappa_i\right) \end{cases} \tag{8}$$

where β and κ_i are parameters whose expressions are given in the appendix for the sake of clarity and $T = t_k - t_{k-1}$. As shown by (8), the obtained predictor depends only on the last image. To use more than one image and improve the accuracy, a first natural solution would be to recursively use equations (7) and (8). However, this would lead to highly complex relations. This is the reason why we propose to find how image $k-j$ can be transformed into image k. Defining $\mathscr{X}(k) = [M_x(k), M_y(k), \theta(k), \vartheta(k)]^T$ as the system state at k, we propose to compute the smallest sequence of control inputs allowing to reach state at k starting from state at $k-j$. To this aim, we need to verify the controllability of the corresponding nonlinear discrete system $\mathscr{X}(k+1) = g(\mathscr{X}(k), \dot{q}(k))$ where $g(\mathscr{X}(k), \dot{q}(k))$ is obtained by analytically solving (1). Its expression is given by:

$$g: \left(\begin{array}{l} M_x(k) = M_x(k-1) \\ \quad + \frac{\upsilon(k-1)}{\omega(k-1)}(\sin(\theta(k-1) + \omega(k-1)*T) - \sin(\theta(k-1))) \\ M_y(k) = M_y(k-1) \\ \quad - \frac{\upsilon(k-1)}{\omega(k-1)}(\cos(\theta(k-1) + \omega(k-1)*T) - \cos(\theta(k-1))) \\ \theta(k) = \theta(k-1) + \omega(k-1)*T \\ \vartheta(k) = \vartheta(k-1) + \varpi(k-1)*T \end{array} \right) \tag{9}$$

when $\omega \neq 0$. Note that the problem is straightforward if $\omega = 0$. Such a system is controllable in p steps if the following matrix P is full rank [Djeridane, 2004].

$$P = \left[\begin{array}{c} \frac{\partial g(\mathscr{X}(p-1), \dot{q}(p-1))}{\partial \dot{q}(p-1)} \\ \frac{\partial g(\mathscr{X}(p-1), \dot{q}(p-1))}{\partial \mathscr{X}(p-1)} \frac{\partial g(\mathscr{X}(p-2), \dot{q}(p-2))}{\partial \dot{q}(p-2)} \\ \dots \\ \frac{\partial g(\mathscr{X}(p-1), \dot{q}(p-1))}{\partial \mathscr{X}(p-1)} \\ \dots \\ \frac{\partial g(\mathscr{X}(1), \dot{q}(1))}{\partial \mathscr{X}(1)} \frac{\partial g(\mathscr{X}(0), \dot{q}(0))}{\partial \dot{q}(0)} \end{array} \right]^T \tag{10}$$

It can be shown that P is not full rank for $p = 1$. For $p = 2$, this property is fulfilled if $\omega \neq 2\eta\pi$ for $\eta \in \mathbb{N}$. Now, thanks to (9), we can compute the two control inputs allowing to reach the system state at k from the one at $k-j$. The first one aims at positionning the robot at $[M_x(k), M_y(k)]$. The second one orientates the robot and the platform towards $[\theta(k), \vartheta(k)]$. We obtain the following equations:

$$\begin{cases} \upsilon_{e1} = \frac{\omega_{e1}}{2\sin(\frac{\omega_{e1}*T}{2})} * R \\ \omega_{e1} = \frac{-2\theta(k-j)}{T} + 2\arctan\left(\frac{(M_y(k) - M_y(k-j))}{(M_x(k) - M_x(k-j))}\right) \\ \varpi_{e1} = 0 \end{cases}$$
$$\begin{cases} \upsilon_{e2} = 0 \\ \omega_{e2} = \frac{\theta(k) - (\theta(k-j) + \omega_{e1}T)}{T} \\ \varpi_{e2} = \frac{\vartheta(k) - \vartheta(k-j)}{T} \end{cases} \tag{11}$$

where: $R = \sqrt{(M_x(k) - M_x(k-j))^2 + (M_y(k) - M_y(k-j))^2}$.

Now, thanks to the two control inputs $\dot{q}_{e1} = (\upsilon_{e1}, \omega_{e1}, \varpi_{e1})^T$ and $\dot{q}_{e2} = (\upsilon_{e2}, \omega_{e2}, \varpi_{e2})^T$ given by (11), we are able to reach the image at instant k from any image at $k-j$. It should be noted that the robot trajectory in the world frame computed with $[\dot{q}_{e1}, \dot{q}_{e2}]$ is not the same as the one calculated with the sequence $[\dot{q}(k-j), ..., \dot{q}(k-1)]$. Therefore we have to introduce an intermediate state $\mathscr{X}(j')$ to compute our estimator. It corresponds to the system state which has been reached at the virtual instant j' by applying $\dot{q}_{e1}$. Using (7) recursively, we obtain the following results:

$$\begin{array}{l} \hat{z}_i(k-j'|k-j) = \hat{z}_i(k-j|k-j)\phi_i + \varphi_i \\ \hat{z}_i(k|k-j') = \hat{z}_i(k-j'|k-j)\phi'_i + \varphi'_i \\ \hat{z}_i(k|k-j) = \hat{z}_i(k-j|k-j)\mu_i + \nu_i \end{array} \tag{12}$$

Now, using (12) and (8), we express a predictor using image at $k-j$ as follows:

$$\begin{cases} \hat{X}_i(k|k-j) = \frac{\tilde{X}_i(k-j)\hat{z}_i(k|k-j) - \nu_i}{\hat{z}_i(k|k-j)\mu_i} \\ \hat{Y}_i(k|k-j) = f\left\{\frac{\tilde{Y}_i(k-j)\cos(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{f\mu_i}\right. \\ \qquad - \frac{\sin(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ \qquad \left. - \frac{\sin(A(\dot{q}_{e2})T)}{\phi'_i} + \frac{\gamma_i}{\hat{z}_i(k|k-j)})\right\} \end{cases} \tag{13}$$

The different parameters $\phi_i, \varphi_i, \phi'_i, \varphi'_i, \mu_i, \nu_i$ and κ_i are given in the appendix. Once the predictors have been obtained, we address the estimators determination problem. To this aim, we propose to minimize the following criterion which represents the error (for one point P_i) between the above predictors and the corresponding measures at t_k. We get:

$$C^* = \sum_{j=1}^{m} (\hat{X}_i(k|k-j) - \tilde{X}_i(k))^2 + (\hat{Y}_i(k|k-j) - \tilde{Y}_i(k))^2 \tag{14}$$

Derivating this cost function with respect to the depth leads to :

$$\begin{array}{l} \frac{\partial C^*}{\partial \hat{z}_i(k|k-j)} = \sum_{j=1}^{m} \left\{2(\hat{X}_i(k|k-j) - \tilde{X}_i(k))\frac{\partial \hat{X}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)}\right. \\ \qquad \left. + 2(\hat{Y}_i(k|k-j) - \tilde{Y}_i(k))\frac{\partial \hat{Y}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)}\right\} \end{array} \tag{15}$$

where: $\frac{\partial \hat{X}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} = \frac{\tilde{X}_i(k-j)\nu i}{\hat{z}_i^2(k|k-j)\mu_i}$ and $\frac{\partial \hat{Y}_i(k+2|k)}{\partial \hat{z}_i(k|k-j)} = \frac{-f\gamma_i}{\hat{z}_i^2(k|k-j)}$. Our estimator $\hat{z}_i(k|k)$ is then given by:

$$\hat{z}_i(k|k) = \frac{\sum_{j=1}^{m} Num_i^j}{\sum_{j=1}^{m} Den_i^j} \tag{16}$$

with

$$Num_i^j = \frac{\nu_i^2 \tilde{X}_i^2(k-j)}{\mu_i^2} + f^2\gamma^2 \tag{17}$$

and

$$
\begin{aligned}
Den_i^j = & (\frac{\tilde{X}_i(k-j)}{\mu_i} - \tilde{X}_i(k))\frac{\tilde{X}_i(k-j)\nu_i}{\mu_i} \\
& - \Big\{ \frac{\tilde{Y}_i(k-j)\cos(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{\mu_i} \\
& - \frac{f\sin(A(\dot{q}_{e1})T))\cos(A(\dot{q}_{e2})T)}{\mu_i} \\
& - \frac{f\sin(A(\dot{q}_{e2})T)}{\phi_i'} - \tilde{Y}_i(k) \Big\} f\gamma_i(k)
\end{aligned}
\qquad (18)
$$

We have then computed an estimator for the depth using data provided by m previous images. This depth will be used to determine the reference visual feature s^*.

B. Estimation Of The Reference Visual Data s^*

In this part, we present a method allowing to automatically build s^*. First, we reconstruct the target coordinates in the frame F_O using z. We define $H_{O/M}$, $H_{M/P}$, $H_{P/C}$ and $H_{I/C}$ wich are respectively the homogeneous transformation matrices between F_M and F_O, F_P and F_M, F_C and F_P, F_C and the image plane (Fig 1(b)). Noted that $H_{I/C}$ is function of z. Now we relate the coordinates of one point of the target $p_i = [\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i]^T$ in F_O, to its projection $P_i = [X_i, Y_i, z_i]^T$ in the image plane as follows:

$$P_i = H_{I/C} * H_{P/C}^{-1} * H_{M/P}^{-1} * H_{O/M}^{-1} * p_i \qquad (19)$$

To improve the accuracy of our estimation, it is important to take into account an important number of images. To this aim, the camera moves to n different poses taking an image for each of them. In this way, n points P_i correspond to the original 3D point p_i, increasing the number of measures. Relation (19) becomes:

$$Y = \Theta p_i + R \qquad (20)$$

with

$$Y = \begin{bmatrix} P_i(1) \\ \vdots \\ P_i(n) \end{bmatrix} \quad \Theta = \begin{bmatrix} \Theta_i(1) \\ \vdots \\ \Theta_i^{(}n) \end{bmatrix} \quad R = \begin{bmatrix} R_i(1) \\ \vdots \\ R_i(n) \end{bmatrix}$$

The expressions of $\Theta_i(j)$ and $R_i(j)$, for $j \in [1, ...n]$, are given in the appendix. Solving equation (20) in the least squares sense leads to:

$$p_i = (\Theta^T \Theta)^{-1} \Theta^T (Y - R) \qquad (21)$$

Now, once p_i is known, it is possible to build the reference visual features s^* using relation (19). In this way, we avoid the drawback of manually moving the robot to take an image at the goal. Our approach is more accurate and more general than the classical one mentioned in the introduction because a knowledge of the target model is no longer required. Finally, it is sufficiently rapid and efficient to be used on-line to realize a long range navigation task (see section V).

IV. EXPERIMENTATION

We have implemented the proposed approach on our experimental testbed. We first present the latter before describing the obtained results.

Experiments were conducted in the AIP-PRIMECA laboratory. The considered testbed is Pekee II developed by Wany^{TM}Robotics (see figure 2). Its standard equipment has been extended with an Axis^{TM}214 PTZ network camera which has been coarsely calibrated, without considering the lens distortion. As tilt and zoom are not considered in our set of experiments, Pekee II can be represented by the model given in section II-A. Sensors acquisitions and actuators management are performed using a specific $C++/C\#$ software platform developed in AIP-PRIMECA.

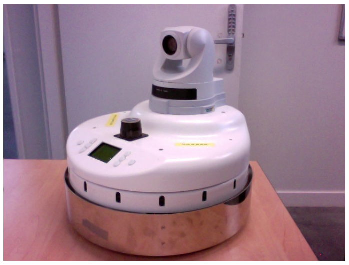


Fig. 2. Pekee II

We consider a vision-based navigation task consisting in positioning the camera with respect to a given landmark located at the world frame origin. This landmark is a 16cm by 16cm target made of four circles. The visual features are then defined by the four circle centers which are extracted using our image processing (see figure 4(b)). The goal is to automatically compute the reference visual features before the vision-based navigation starts. To this aim, the robot is initially positioned at point $(2.2, 2.2)$ with respect to the world frame. Its camera is orientated so that the target lies in its field of view. Then it turns on itself, making a 180 degrees scan and taking an image every 10 degrees. This motion is used in our reconstruction algorithm to estimate both the target depth and the value of s^* as mentioned above. Any robot motion could have been considered but in this case we have chosen a sole rotation. The obtained results are represented on figures 3 and 4(b).

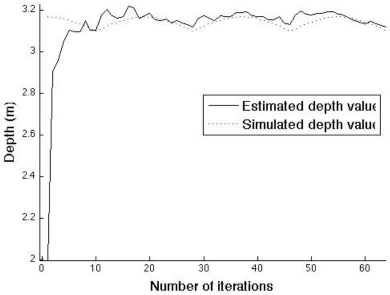
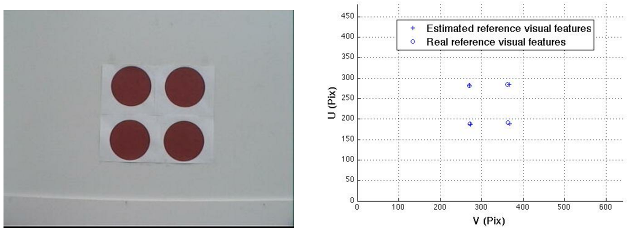


Fig. 3. Estimated and simulated depth of one point

Figure 3 shows the estimated depth for one target point for the sake of clarity, as the results are similar for the whole landmark. As one can see, the estimator, using at most $m = 20$ images, reaches the correct value[2] after about 10 images.

[2] This value is obtained by simulating the robot position with respect to the target.



(a) Landmark (b) Reference visual features

Fig. 4. Landmark and visual features

Thus, our algorithm seems to converge more rapidly than the one proposed in [De Luca et al., 2008], although the number of necessary images is not explicitly mentioned in the latter. This estimated depth is then used to compute the reference visual features s^*. As one can see on figure 4(b), the obtained accuracy is quite satisfying, as the estimated visual features are very close to their real values obtained by taking an image at the desired position(1.21, 0) in the world frame.

Thus, we have validated our reconstruction algorithm. Up to now, we have used the depth value to compute the reference visual features during an initialization phase. We can also benefit from this estimation to calculate the interaction matrix [Chaumette and Hutchinson, 2006] and to treat the target occlusion [Durand Petiteville et al., 2010]. In the sequel, we show that this method can be useful to perform a vision-based long range navigation task.

V. AN APPLICATION EXAMPLE

The visual servoing technique, described in [Espiau et al., 1992] and applied to mobile robots, as in [Pissard-Gibollet and Rives, 1995] allows to execute only short range navigation missions. Recently, Cherubini and Chaumette have elaborated a new method to perform a long range outdoor navigation [Cherubini and Chaumette, 2009]. This technique is based on time-dependent visual features and requires a learning step. In this section, our goal is to show that our reference visual features estimation tool can be also used on-line to execute an indoor long range navigation task. Thus, this part of our work is intended to exhibit an application example of our technique. This is the reason why the mission is defined by an ad hoc set of known targets to be successively reached. We associate to each of them visual features and a task function . Thus the global navigation mission is expressed as a pre-defined sequence of subtasks. Associating a task function to each of them, we compute a controller able to make it vanish. To ensure the control smoothness when switching from one subtask to the other, we apply the dynamical sequencing formalism [Souères and Cadenat, 2003].

To complete the navigation mission, the robot has to compute the reference visual features corresponding to the first target, then to realize the vision based subtask. When the robot is in the first landmark neighbourhood, the camera looks for the next target to calculate the associated reference visual features. At the same time, the mobile base keeps on performing the current task thanks to the visual features estimation method proposed in [Folio and Cadenat, 2008]. When the next reference visual features are computed, the robot switches to the next vision based navigation task. This algorithm is repeated until the last target is reached.

We have simulated this reasoning using Matlab$^{\mathrm{TM}}$software. The task consists in positioning the camera with respect to the final landmark $\mathscr{L}_3$ which is not visible from the robot initial configuration (see figure 5). To this aim, the global mission has been divided into three subtasks. Therefore, three landmarks made of four interest points must be successively reached. Their positions have been defined ad hoc so that no problem of collision may occur. The obtained results are presented on figures 5 and 6.

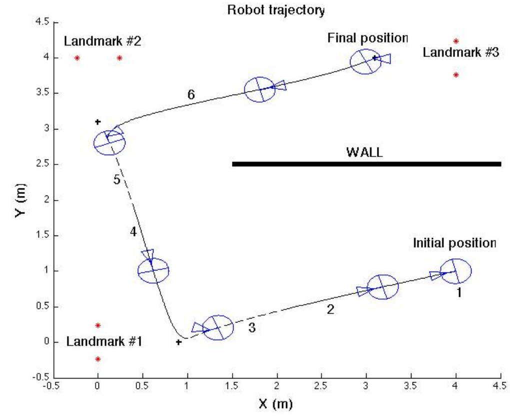


Fig. 5. Robot trajectory

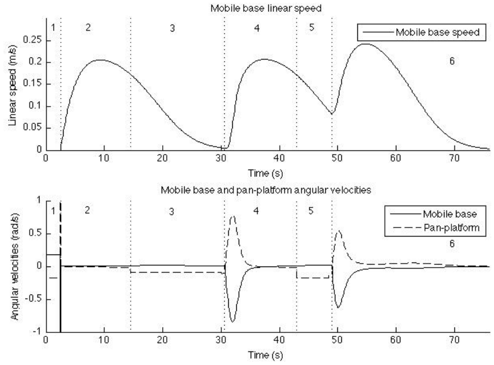


Fig. 6. Evolution of robot velocities

In figure 5 the trajectories done using the real and estimated visual features are respectively represented by the continuous and dotted lines. The crosses correspond to the desired camera position with respect to the different landmarks. Figure 5 shows that the task is successfully performed, as the camera is correctly positioned with respect to the third landmark. Furthermore, figure 6 demonstrates that the control law remains smooth, despite the switches between the different controllers. Phase 1 corresponds to the initialization phase described above (see IV). Steps 2, 4 and 6 are related to the visual servoing with respect to landmarks $\mathscr{L}_1, \mathscr{L}_2$ and $\mathscr{L}_3$. Finally, during phases 3 and 5, the pan-platform is controlled to detect $\mathscr{L}_2$ and $\mathscr{L}_3$, while the mobile base is driven using the virtual visual features provided by D. Folio's algorithm. Thus, these results show the relevancy of both the estimators of z and s^* in the particular context of a long range vision-based navigation task.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a method allowing to automatically compute the reference visual features. This method relies on the estimation of the depth value thanks to a predictor/corrector pair. We have validated the proposed approach in simulation and in an experimental context. The obtained results have shown its efficiency and its relevancy at the beginning of a vision-based mission. As s^* can be computed sufficiently rapidly to be used on-line, this algorithm can be coupled to a visual servoing control. An example of such a successful association has been shown through the realization of a long range navigation task.

For future works, we first aim at theoretically proving the estimator convergence and its sensitivity to the incertitudes (image noise and odometry errors). In a second step, we will more deeply address the long range navigation problem to define a dedicated formalism.

REFERENCES

[Basri et al., 1999] Basri, R., Rivlin, E., and Shimshoni, I. (1999). Visual homing: Surfing on the epipoles. *Int. J. Comput. Vision*, 33(2):117–137.

[Cervera et al., 2002] Cervera, E., Martinet, P., and Berry, F. (2002). Robotic manipulation with stereo visual servoing. *Robotics and Machine Perception, SPIE International Group Newsletter*, 1(1):3.

[Chaumette, 2002] Chaumette (2002). *La commande des robots manipulateurs*, chapter Asservissement visuel. Hermés.

[Chaumette and Hutchinson, 2006] Chaumette and Hutchinson (2006). Visual servo control, part 1 : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4).

[Cherubini and Chaumette, 2009] Cherubini, A. and Chaumette, F. (2009). Visual navigation with a time-independent varying reference. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'09*, pages 5968–5973, St Louis, USA.

[Corke, 1996] Corke, P. (1996). *Visual control of robots : High performance visual servoing*. Research Studies Press LTD.

[De Luca et al., 2008] De Luca, Oriolo, and Giordano (2008). Features depth observation for image based visual servoing: theory and experiments. *Int. Journal of Robotics Research*, 27(10).

[Djeridane, 2004] Djeridane (2004). *Sur la commandabilité des systèmes non linéaires à temps discret*. PhD thesis, Université Paul Sabatier - Toulouse III.

[Durand Petiteville et al., 2009] Durand Petiteville, A., Courdesses, M., and Cadenat, V. (2009). Reconstruction of the features depth to improve the execution of a vision-based task. In *9th International workshop on Electronics, Control, Modelling, Measurement and Signals 2009*, Mondragon, Spain.

[Durand Petiteville et al., 2010] Durand Petiteville, A., Courdesses, M., and Cadenat, V. (2010). A new predictor/corrector pair to estimate the visual features depth during a vision-based navigation task in an unknown environment. In *7th International Conference on Informatics in Control, Automation and Robotics*, Funchal, Portugal.

[Espiau et al., 1992] Espiau, Chaumette, and Rives (1992). A new approach to visual servoing in robotics. *IEEE Trans. Robot. Automat.*, 8:313–326.

[Fang et al., 2005] Fang, Y., Dixon, W. E., Dawson, M., and Shawda, P. (2005). Homography based visual servo regulation of mobile robots. *IEEE Trans. on System, Man and Cybernetics*, 35(5):1041–1050.

[Folio, 2007] Folio (2007). *Stratégies de commandes référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile*. PhD thesis, Université Paul Sabatier - Toulouse III.

[Folio and Cadenat, 2008] Folio and Cadenat (2008). *Computer Vision*, chapter 4. Xiong Zhihui; IN-TECH.

[Hutchinson et al., 1996] Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Trans. on Rob. and Automation*, 12(5):651–670.

[Jerian and Jain, 1991] Jerian, C. and Jain, R. (1991). Structure from motion: a critical analysis of methods. *IEEE Transactions on systems, Man, and Cybernetics*, 21(3):572–588.

[Ma et al., 2003] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. (2003). *An invitation to 3-D vision: from images to geometric models*. New York: Springer-Verlag.

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

[Matthies et al., 1989] Matthies, Kanade, and Szeliski (1989). Kalman filter-based algorithms for estimating depth in image sequences. *Int, Journal of Computer Vision*, 3(3):209–238.

[Pissard-Gibollet and Rives, 1995] Pissard-Gibollet and Rives (1995). Applying visual servoing techniques to control a mobile handeye system. In *IEEE Int., Conf. on Robotics and Automation*, Nagoya, Japan.

[Souères and Cadenat, 2003] Souères, P. and Cadenat, V. (2003). Dynamical sequence of multi-sensor based tasks for mobile robots navigation. In *SYROCO*, Wroclaw, Poland.

APPENDIX

Parameters for equations (7) and (8):

$$\left\{\begin{array}{l}
\alpha_i = \frac{\tilde{Y}_i(k-1)}{f}\sin(A(\dot{q}(k-1))T) + \cos(A(\dot{q}(k-1))T) \\
\beta = \left\{-D_x\sin(\vartheta(k-1)) - \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1)) + C_y\right\}\sin(A(\dot{q}(k-1))T) \\
\quad + \left\{D_x\cos(\vartheta(k-1)) - \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1)) + C_x\right\}\cos(A(\dot{q}(k-1))T) \\
\quad -D_x\cos(\vartheta(k)) + \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k)) - C_x \\
\kappa_i = \left\{\frac{-\tilde{Y}_i(k-1)\beta}{f\alpha_i} - D_x\sin(\vartheta(k-1)) - \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1)) + C_y\right\}\cos(A(\dot{q}(k-1))T) \\
\quad - \left\{\frac{-\beta}{\alpha_i} + D_x\cos(\vartheta(k-1)) - \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1)) + C_x\right\}\sin(A(\dot{q}(k-1))T) \\
\quad +D_x\sin(\vartheta(k)) + \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k)) - C_y
\end{array}\right.$$

Parameters for equations (12) and (13):

$$\left\{\begin{array}{l}
\phi_i = \frac{\tilde{Y}_i(k-j)}{f}\sin(A(\dot{q}_{e1})T) + \cos(A(\dot{q}_{e1})T) \\
\varphi_i = \left\{-D_x\sin(\vartheta(k)) - \frac{\upsilon_{e1}}{\varpi_{e1}}\cos(\vartheta(k-j)) + C_y\right\}\sin(A(\dot{q}_{e1})T) \\
\quad + \left\{D_x\cos(\vartheta(k-j)) - \frac{\upsilon_{e1}}{\varpi_{e1}}\sin(\vartheta(k-j)) + C_x\right\}\cos(A(\dot{q}_{e1})T) \\
\quad -D_x\cos(\vartheta(j')) + \frac{\upsilon_{e1}}{\varpi_{e1}}\sin(\vartheta(j')) - C_x
\end{array}\right.$$

$$\left\{\begin{array}{l}
\phi_i' = \frac{\tilde{Y}_i(k-j)}{f\phi_i}\cos(A(\dot{q}_{e1})T)\sin(A(\dot{q}_{e2})T) - \frac{\sin(A(\dot{q}_{e1})T)\sin(A(\dot{q}_{e2})T)}{\phi_i} + \cos(A(\dot{q}_{e2})T) \\
\varphi_i' = \Big[\Big\{\frac{\tilde{Y}_i(k-j)\varphi_i}{f\phi_i} - D_x\sin(\vartheta(k-j)) - \frac{\upsilon_{e1}}{\omega_{e1}}\cos(\vartheta(k-j)) + C_y)\Big\}\cos(A(\dot{q}_{e1})T) \\
\quad - \left\{\frac{-\varphi_i}{\phi_i} + D_x\cos(\vartheta(k-j)) - \frac{\upsilon_{e1}}{\omega_{e1}}\sin(\vartheta(k-j)) + C_x\right\}\sin(A(\dot{q}_{e1})T) \\
\quad +(\frac{\upsilon_{e1}}{\omega_{e1}} - \frac{\upsilon_{e2}}{\omega_{e2}})\cos(\vartheta(j'))\Big]\sin(A(\dot{q}_{e2})T) \\
\quad + \left\{D_x\cos(\vartheta(j')) - \frac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(j')) + C_x\right\}\cos(A(\dot{q}_{e2})T) \\
\quad -D_x\cos(\vartheta(k)) - \frac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(k)) + C_x
\end{array}\right.$$

$$\left\{\begin{array}{l}
\mu_i = \phi_i\phi_i' \\
\nu_i = \varphi_i\phi_i' + \varphi_i'
\end{array}\right.$$

$$\left\{\begin{array}{l}
\gamma_i = \Big[\Big\{\frac{-\tilde{Y}_i(k-j)\nu_i}{f\mu_i} - D_x\sin(\vartheta(k-j)) - \frac{\upsilon_{e1}}{\omega_{e1}}\cos(\vartheta(k-j)) + C_y\Big\}\cos(A(\dot{q}_{e1})T) \\
\quad - \left\{\frac{-\nu_i}{\mu_i} + D_x\cos(\vartheta(k-j)) - \frac{\upsilon_{e1}}{\omega_{e1}}\sin(\vartheta(k-j)) + C_x\right\}\sin(A(\dot{q}_{e1})T) \\
\quad +(\frac{\upsilon_{e1}}{\omega_{e1}} - \frac{\upsilon_{e2}}{\omega_{e2}})\cos(\vartheta(j'))\Big]\cos(A(\dot{q}_{e2})T) \\
\quad - \left\{\frac{-\varphi_i'}{\phi_i'} + D_x\cos(\vartheta(j')) - \frac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(j')) + C_x\right\}\sin(A(\dot{q}_{e2})T) \\
\quad +D_x\sin(\vartheta(k)) + \frac{\upsilon_{e2}}{\omega_{e2}}\cos(\vartheta(k)) - C_y
\end{array}\right.$$

$$\left\{\begin{array}{l}
\vartheta(j') = \vartheta(k-j) + \varpi_{e1}T \\
\vartheta(k) = \vartheta(k-j) + (\varpi_{e1} + \varpi_{e2})T
\end{array}\right.$$

Parameters for equation (19):

$$\Theta_i(j) = \begin{bmatrix} 0 & 0 & \frac{-f}{z_i} \\ [-\cos(\theta)\sin(\vartheta) - \sin(\theta)\cos(\vartheta)]\frac{f}{z_i} & [-\sin(\theta)\sin(\vartheta) + \cos(\theta)\cos(\vartheta)]\frac{f}{z_i} & 0 \\ \cos(\theta)\cos(\vartheta) - \sin(\theta)\sin(\vartheta) & \sin(\theta)\cos(\vartheta) + \cos(\theta)\sin(\vartheta) & 0 \end{bmatrix}$$

$$R_i(j) = \begin{bmatrix} \frac{hf}{z_i} \\ \{-[-M_x\cos(\theta) - M_y\sin(\theta)]\sin(\vartheta) + [M_x\sin(\theta) - M_y\cos(\theta)]\cos(\vartheta) + D_x\sin(\vartheta) - C_y\}\frac{f}{z_i} \\ [-M_x\cos(\theta) - M_y\sin(\theta)]\cos(\vartheta) + [M_x\sin(\theta) - M_y\cos(\theta)]\sin(\vartheta) - D_x\cos(\vartheta) - C_x \end{bmatrix}$$