A NEW PREDICTOR/CORRECTOR PAIR TO ESTIMATE THE VISUAL FEATURES DEPTH DURING A VISION-BASED NAVIGATION TASK IN AN UNKNOWN ENVIRONMENT
A Solution for Improving the Visual Features Reconstruction During an Occlusion

A. Durand Petiteville, M. Courdesses and V. Cadenat
CNRS, LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France

Keywords: Visual servoing, Visual data estimation, Predictor/estimator pair.

Abstract: This papers deals with the problem of estimating the visual features during a vision-based navigation task when a temporary total occlusion occurs. The proposed approach relies on an existent specific algorithm. However, to be efficient, this algorithm requires highly precise initial values for both the image features and their depth. Thus, our objective is to design a predictor/estimator pair able to provide an accurate estimation of the depth value, even when the visual data are noisy. The obtained results show the efficiency and the interest of our technique.

1 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). 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 (Corke, 1996). 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 control law depends only on the visual features. Therefore, if they are lost because of an occlusion or any other unexpected event, the desired task cannot be realized anymore. Here, we are working in a mobile robotics context. In this context, the realization of a vision-based navigation task in a given environment requires to preserve not only the image data visibility, but also the robot safety. In that case, techniques allowing to avoid simultaneously collisions and visual data losses appear to be limited, because they are restricted to missions where an avoidance motion exists without leading to local minima (Folio and Cadenat, 2008). As many robotic tasks cannot be performed if the visual data loss is not tolerated, it is necessary to provide methods which accept that occlusions may effectively occur without leading to a task failure.

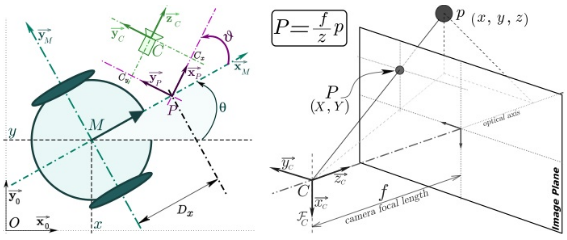
Folio has proposed such an approach (Folio and Cadenat, 2008). The idea is to reconstruct the visual features whenever necessary. The developed algorithm is based upon the vision/motion link which relates the variation of the visual features in the image to the camera motion. However, this method needs accurate initial values for the visual features and the depth. If the first ones can be obtained from the last image available before the occlusion occurs, determining a precise initial value for the depth sufficiently rapidly to correctly handle the occlusion remains a challenging problem. Different approaches are proposed in the literature. See for instance the works by Matthies who derived and compared several algorithms based on a Kalman filter (Matthies et al., 1989). However, a correct depth value can be obtained only if the camera motion respects some very particular constraints. These solutions are not suitable for our particular case. It would be also possible to use the epipolar geometry (Ma et al., 2003), stereovision (Cervera et al., 2002), or even structure from motion techniques (Jerian and Jain, 1991). However, this kind of approaches are time-consuming and can-

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 - A Solution for Improving the Visual Features Reconstruction During an Occlusion.
In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 268-274
DOI: 10.5220/0002948902680274

not be used for our purpose. Finally, (De Luca et al., 2008) have proposed to estimate the depth using a non linear observer. But, unfortunately, the convergence time seems to be too large to be useful in our specific context.

In this paper, we have developed a non recursive algorithm based on a predictor/corrector pair allowing to provide an accurate value of the depth using visual data. This value will then be used to feed D. Folio's algorithm in order to improve its efficiency.

The paper is organized as follows. Section II is dedicated to some preliminaries regarding the system modelling, the visual servoing control law design and the description of D. Folio's algorithm. Section III details and analyzes the predictor/estimator pair which has been developed. Finally, simulation results validating our approach are shown in section IV.



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

Figure 1: System modeling.

2 PRELIMINARIES

2.1 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 such a robot, the kinematic model is classically given by the following relations:

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

where $\dot{M}_x(t)$ is the speed of M wrt. $\vec{x}_O$ and $\dot{M}_y(t)$ wrt. $\vec{y}_O$.

The camera motion can be described by the kinematic screw T_{C/F_O}:

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

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:

$$T_r = J_r\dot{q} \tag{3}$$

Defining C_x and C_y as the coordinates of C along axes $\vec{x}_P$ and $\vec{y}_P$ (see figure 1(a)), J_r is given by:

$$J_r = \begin{pmatrix} -\sin(\vartheta(t)) & D_x\cos(\vartheta(t)) + C_x & C_x \\ \cos(\vartheta(t)) & D_x\sin(\vartheta(t)) - C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \tag{4}$$

It should be noted that J_r can be inverted ($det(J_r) = D_x$).

2.2 The Vision-based Navigation Task

The vision-based navigation task consists in positioning the camera with respect to a given static landmark. We assume that this landmark can be characterized by n interest points which are extracted by our image processing. Therefore, 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). 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):

$$\dot{s} = L_{(s,z)}T_r = L_{(s,z)}J_r\dot{q} \tag{5}$$

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):

$$L_{(P_i)} = \begin{pmatrix} L_x(s_i,z_i) \\ L_y(s_i,z_i) \end{pmatrix} = \begin{pmatrix} 0 & \frac{X_i}{z_i} & \frac{X_iY_i}{f} \\ -\frac{f}{z_i} & \frac{Y_i}{z_i} & f + \frac{Y_i^2}{f} \end{pmatrix} \tag{6}$$

where z_i represents the depth of the projected point p_i, and f is the camera focal (see figure 1(b)). At the beginning of the visual servoing, z_i is not available yet. Thus, only an approximation of the interaction matrix noted $\hat{L}_{(s^*,z^*)}$ computed at the desired position will be used to build the control law as in (Chaumette and Hutchinson, 2006).

To perform the desired vision-based task, we apply the visual servoing technique given in (Espiau et al., 1992) to mobile robots as in (Pissard-Gibollet and Rives, 1995). The proposed approach relies on the task function formalism (Samson et al., 1991) and consists in expressing the visual servoing task by the following task function to be regulated to zero:

$$e = C(s - s^*) \tag{7}$$

where s^* represents the desired value of the image data. Matrix C, called combination matrix, allows to take into account more visual features than available degrees of freedom. Several choices are possible: see for example (Comport et al., 2004). A classical idea consists in defining $C = \hat{L}^+_{(s^*,z^*)} = (\hat{L}^T_{(s^*,z^*)}\hat{L}_{(s^*,z^*)})^{-1}\hat{L}^T_{(s^*,z^*)}$ because the expression of the control law is simplified (see (8) hereafter). Now, it remains to determine a controller allowing to make e vanish. Such a controller is classically designed by imposing an exponential decrease, that is $\dot{e} = -\lambda e$, where λ is a positive scalar or a positive definite matrix. Then, the visual servoing controller can be written as follows:

$$\dot{q}_{(s)} = -(C\hat{L}_{(s^*,z^*)}J_r)^{-1}\lambda C(s - s^*) = J_r^{-1}\lambda C(s - s^*) \tag{8}$$

2.3 Visual Data Estimation

However, the above controller can only be used if the visual data are available. If they are not because of a landmark occlusion or a camera failure for example, the task cannot be realized anymore. To remedy this critical situation, (Folio and Cadenat, 2008) has recently proposed to solve the dynamic system (5) 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 and Cadenat, 2008) 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:

$$\left\{\begin{array}{l} X_i(t) = \frac{z_i(k-1)X_i(k-1)}{z_i(t)} \\ Y_i(t) = \frac{f}{z_i(t)}\Big\{D_x \sin(\vartheta(t)) + \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(t)) - C_y \\ \qquad +c_1 \cos(A(\dot{q}(k-1))(t - t_{k-1})) \\ \qquad -c_2 \sin(A(\dot{q}(k-1))(t - t_{k-1})\} \\ z_i(t) = -D_x \cos(\vartheta(t)) + \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(t)) - C_x \\ \qquad +c_1 \sin(A(\dot{q}(k-1))(t - t_{k-1})) \\ \qquad +c_2 \cos(A(\dot{q}(k-1))(t - t_{k-1})) \end{array}\right. \tag{9}$$

where:

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

Thanks to (9), Folio has developped 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 $z_i(k-1)$.

3 METHODOLOGY

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 $j \in [1, ..., m]$. To this aim, we rewrite equation (9) to relate $z_i(k-1|k-1)$ and $z_i(k|k-1)$. We obtain:

$$\hat{z}_i(k-1|k-1) = \frac{\hat{z}_i(k|k-1) - \beta}{\alpha_i} \tag{10}$$

where α_i and β are 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 (10) in (9) to obtain the following predictor for the visual features:

[1] As one can see, solution (9) 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.

$$\begin{cases} \hat{X}_i(k|k-1) = \frac{\hat{z}_i(k|k-1)\tilde{X}_i(k-1) - \beta\tilde{X}_i(k-1)}{\hat{z}_i(k|k-1)\alpha_i} \\ \hat{Y}_i(k|k-1) = \frac{f}{\hat{z}_i(k|k-1)} \\ \qquad (\frac{\tilde{Y}_i(k-1)\hat{z}_i(k|k-1)}{f\alpha_i}\cos(A(\dot{q}(k-1))T) \\ \qquad -\frac{\hat{z}_i(k|k-1)}{\alpha_i}\sin(A(\dot{q}(k-1))T) + \kappa_i) \end{cases} \tag{11}$$

where κ_i is given in the appendix $T = t_k - t_{k-1}$. As shown by (11), the obtained predictor depends only on the last image. To use more than one image and improve the accuracy, a first natural solution is to recursively use equations (10) and (11). 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 $\mathcal{X}(k) = [M_x(k), M_y(k), \theta(k), \vartheta(k)]^T$ 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 first need to verify the controllability of the corresponding nonlinear discrete system $\mathcal{X}(k+1) = g(\mathcal{X}(k), \dot{q}(k))$ where $g(\mathcal{X}(k), \dot{q}(k))$ is obtained by analytically solving equation (1). Its expression is given by:

$$g : \begin{pmatrix} M_x(k) = M_x(k-1) \\ \qquad + \frac{\upsilon(k-1)}{\omega(k-1)}(\sin(\theta(k-1) + \omega(k-1)*T) \\ \qquad - \sin(\theta(k-1))) \\ M_y(k) = M_y(k-1) \\ \qquad - \frac{\upsilon(k-1)}{\omega(k-1)}(\cos(\theta(k-1) + \omega(k-1)*T) \\ \qquad - \cos(\theta(k-1))) \\ \theta(k) = \theta(k-1) + \omega(k-1)*T \\ \vartheta(k) = \vartheta(k-1) + \varpi(k-1)*T \end{pmatrix} \tag{12}$$

when $\omega \neq 0$ (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).

$$P = \begin{bmatrix} \frac{\partial g(\mathcal{X}(p-1), \dot{q}(p-1))}{\partial \dot{q}(p-1)} \\ \frac{\partial g(\mathcal{X}(p-1), \dot{q}(p-1))}{\partial \mathcal{X}(p-1)} \frac{\partial g(\mathcal{X}(p-2), \dot{q}(p-2))}{\partial \dot{q}(p-2)} \\ \dots \\ \frac{\partial g(\mathcal{X}(p-1), \dot{q}(p-1))}{\partial \mathcal{X}(p-1)} \\ \dots \\ \frac{\partial g(\mathcal{X}(1), \dot{q}(1))}{\partial \mathcal{X}(1)} \frac{\partial g(\mathcal{X}(0), \dot{q}(0))}{\partial \dot{q}(0)} \end{bmatrix}^T \tag{13}$$

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 (12), 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 positioning 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:

$$\begin{cases} \upsilon_{e1} = \frac{\omega_{e1}}{2\sin(\frac{\omega_{e1}*T}{2})} * R \\ \omega_{e1} = \frac{-2\theta(k-j)}{T} + 2\arctan\left(\frac{(M_y(k) - M_y(k-j))}{(M_x(k) - M_x(k-j))}\right) \\ \varpi_{e1} = 0 \end{cases}$$
$$\begin{cases} \upsilon_{e2} = 0 \\ \omega_{e2} = \frac{\theta(k) - (\theta(k-j) + \omega_{e1}T)}{T} \\ \varpi_{e2} = \frac{\vartheta(k) - \vartheta(k-j)}{T} \end{cases} \tag{14}$$

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 (14), 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 $\mathcal{X}(j')$ to compute our predictor. It corresponds to the system state which has been reached at the virtual instant j' by applying $\dot{q}_{e1}$. Using (10) recursively, we obtain the following results:

$$\begin{aligned} \hat{z}_i(k-j'|k-j) &= \hat{z}_i(k-j|k-j)\phi_i + \varphi_i \\ \hat{z}_i(k|k-j') &= \hat{z}_i(k-j'|k-j)\phi'_i + \varphi'_i \\ \hat{z}_i(k|k-j) &= \hat{z}_i(k-j|k-j)\mu_i + \nu_i \end{aligned} \tag{15}$$

The different parameters $\phi_i, \varphi_i, \phi'_i, \varphi'_i, \mu_i, \nu_i$ and κ_i involved in these equations are given in the appendix. Now, using (15) and (11), we express a predictor using image at $k-j$ as follows:

$$\begin{cases} \hat{X}_i(k|k-j) = \frac{\tilde{X}_i(k-j)\hat{z}_i(k|k-j) - \nu_i}{\hat{z}_i(k|k-j)\mu_i} \\ \hat{Y}_i(k|k-j) = f\left\{\frac{\tilde{Y}_i(k-j)\cos(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{f\mu_i}\right. \\ \qquad -\frac{\sin(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ \qquad \left. -\frac{\sin(A(\dot{q}_{e2})T)}{\phi'_i} + \frac{\gamma_i}{\hat{z}_i(k|k-j)})\right\} \end{cases} \tag{16}$$

Once the predictors have been obtained, in a second step, 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 instant t_k. We get:

$$C^* = \sum_{j=1}^{m} (\hat{X}_i(k|k-j) - \tilde{X}_i(k))^2 + (\hat{Y}_i(k|k-j) - \tilde{Y}_i(k))^2 \tag{17}$$

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

$$\frac{\partial C^*}{\partial \hat{z}_i(k|k-j)} = \sum_{j=1}^{m} \Big\{ 2(\hat{X}_i(k|k-j) - \tilde{X}_i(k)) \frac{\partial \hat{X}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} + 2(\hat{Y}_i(k|k-j) - \tilde{Y}_i(k)) \frac{\partial \hat{Y}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} \Big\} \tag{18}$$

where:

$$\frac{\partial \hat{X}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} = \frac{\tilde{X}_i(k-j)\nu i}{\hat{z}_i^2(k|k-j)\mu_i} \tag{19}$$

and

$$\frac{\partial \hat{Y}_i(k+2|k)}{\partial \hat{z}_i(k|k-j)} = \frac{-f\gamma_i}{\hat{z}_i^2(k|k-j)} \tag{20}$$

Our estimator $\hat{z}_i(k|k)$ is then given by:

$$\hat{z}_i(k|k) = \frac{\sum_{j=1}^{m} Num_i^j}{\sum_{j=1}^{m} Den_i^j} \tag{21}$$

with

$$Num_i^j = \frac{\nu_i^2 \tilde{X}_i^2(k-j)}{\mu_i^2} + f^2\gamma^2 \tag{22}$$

and

$$\begin{aligned} Den_i^j = {} & \big(\tfrac{\tilde{X}_i(k-j)}{\mu_i} - \tilde{X}_i(k)\big) \tfrac{\tilde{X}_i(k-j)\nu_i}{\mu_i} \\ & - \Big\{ \tfrac{\tilde{Y}_i(k-j)\cos(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ & - \tfrac{f\sin(A(\dot{q}_{e1})T))\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ & - \tfrac{f\sin(A(\dot{q}_{e2})T)}{\phi_i'} - \tilde{Y}_i(k) \Big\} f\gamma_i(k) \end{aligned} \tag{23}$$

We have then computed an estimator for the depth using data provided by m previous images. This depth will be used as an initial condition in Folio's algorithm to reconstruct the visual data.

4 SIMULATION

We have simulated the proposed estimation method during a visual servoing task using Matlab$^{\text{TM}}$software. We present hereafter the obtained results. The considered landmark has been characterized by four interest points. The initial robot configuration has been defined by $\mathcal{X} = (5, 5, -2.35, 0)^T$ and the reference visual features s^* have been determined for the configuration $\mathcal{X} = (1.2, 0, \pi, 0)^T$. For all the presented simulations, the control input $\dot{q}$ given by (8) has been computed using $\lambda = 0.7$ and $T = 0.1s$. $z_i(k)$ is reconstructed using equation (21). The corresponding results for one point are represented on figure 2.

In the first simulation, all the data are supposed to be perfectly known. As one can see, the obtained estimation of the depth of each point p_i is perfect, which validates estimator (21) in the ideal case. Notice that a good estimated value of depth is immediately obtained, which is not the case of the method proposed in (De Luca et al., 2008). Moreover, we can see the depth estimation using Folio's algorithm. The initial condition is not the real value. The error between them is preserved during the entire simulation.

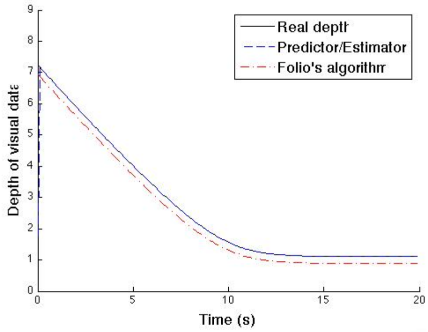


Figure 2: Evolution of both estimated and real depths of one point P_i.

Now, we aim at validating estimator (21) when the visual data are noisy. A one pixel noise has been added on $\tilde{X}_i$ and $\tilde{Y}_i$. In this case the estimator uses at most $m = 20$ images. The corresponding results for one point are represented on figure 3. In a noisy context, estimator (21) converges towards the real depth value within an acceptable time. Indeed the estimated depth value is correct after about 2 s. The number of images m used in the estimation process can be tuned to fit the performances of the considered testbed. Finally, to show that our algorithm provides an adequate value of z sufficiently rapidly, we have coupled it to D. Folio's estimation method. Thus, in the same conditions as previously, we have simulated a loss of visual data between the seventh and ninth seconds during a visual servoing task. As shown in figure 4, the estimated depth values allow to correctly reconstruct the visual features. The navigation task can then be correctly realized, although the controller has been computed with the estimated data instead of the real ones. This last result demonstrates the efficiency of the proposed approach in a noisy context. D. Folio's algo-

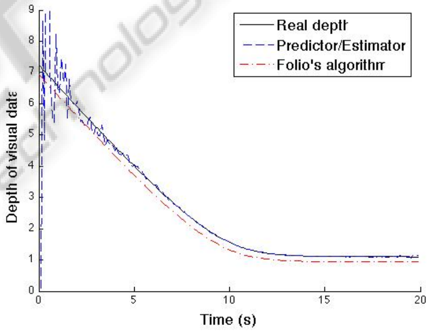


Figure 3: Evolution of both estimated and real depths.

rithm accuracy is then significantly improved thanks to our approach.

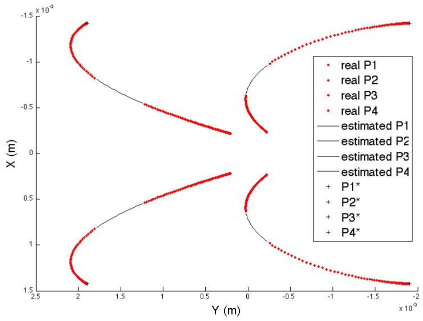


Figure 4: Evolution of estimated and real visual features.

5 CONCLUSIONS

In this paper, we have presented a method allowing to estimate the depth z_i during a vision-based navigation task. The proposed approach relies on a predictor/estimator pair able to provide an estimation of z_i, even when the visual data are noisy. The advantage of the proposed approach is that it relies on a parameterizable number of images, which can be adjusted depending on the computation abilities of the considered processor. The reconstructed depth value is then used to feed Folio's algorithm, increasing its accuracy. The obtained results have proven the efficiency of our technique in a noisy context. Up to now, we have only used the estimated value of z_i to improve Folio's work. In the future, we plan to benefit from this value at two different levels. The first one concerns the control law design with the computation of $L_{(s,z)}$. The approximations classically made in the visual servoing area could then be overcome. The second level is related to the determination of the reference visual signals s^*. This term is computed either experimentally by taking an image at the desired position or theoretically by means of models. These solutions significantly reduce autonomy. We believe that a precise estimation of the depth can be very helpful to automatically on-line compute the value of s^*, suppressing the above mentioned drawbacks. Finally, another challenging aspect of our future work will consist experimenting our approach on a real robot.

REFERENCES

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

Comport, Pressigout, Marchand, and Chaumette (2004). Une loi de commande robuste aux mesures aberrantes en asservissement visuel. In *Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France.

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

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

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.

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

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

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

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). Applying visual servoing techniques to control a mobile handeye system. In *IEEE Int., Conf. on Robotics and Automation*, Nagoya, Japan.

Samson, Borgne, and Espiau (1991). Robot control : The task function approach. *Oxford science publications*.

APPENDIX

Parameters for equation (11):

$$\left\{\begin{aligned}
&\alpha_i = \tfrac{\tilde{Y}_i(k-1)}{f}\sin(A(\dot{q}(k-1))T)+\cos(A(\dot{q}(k-1))T)\\
&\beta = \Big\{-D_x\sin(\vartheta(k-1))-\tfrac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1))+C_y\Big\}\\
&\qquad \sin(A(\dot{q}(k-1))T)+\\
&\qquad \Big\{D_x\cos(\vartheta(k-1))-\tfrac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1))+C_x\Big\}\\
&\qquad \cos(A(\dot{q}(k-1))T)\\
&\qquad -D_x\cos(\vartheta(k))+\tfrac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k))-C_x\\
&\kappa_i = \Big\{\tfrac{-\tilde{Y}_i(k-1)\beta}{f\alpha_i}-D_x\sin(\vartheta(k-1))-\tfrac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1))+C_y\Big\}\\
&\qquad \cos(A(\dot{q}(k-1))T)-\\
&\qquad \Big\{\tfrac{-\beta}{\alpha_i}+D_x\cos(\vartheta(k-1))-\tfrac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1))+C_x\Big\}\\
&\qquad \sin(A(\dot{q}(k-1))T)+D_x\sin(\vartheta(k))+\tfrac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k))-C_y
\end{aligned}\right.$$

Parameters for equations (20) and (21):

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

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

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

$$\left\{\begin{aligned}
&\mu_i = \phi_i\phi'_i\\
&\nu_i = \varphi_i\phi'_i+\varphi'_i
\end{aligned}\right.$$

$$\left\{\begin{aligned}
&\gamma_i = \Big[\Big\{\tfrac{-\tilde{Y}_i(k-j)\nu_i}{f\mu_i}-D_x\sin(\vartheta(k-j))-\tfrac{\upsilon_{e1}}{\omega_{e1}}\cos(\vartheta(k-j))+C_y\Big\}\\
&\qquad \cos(A(\dot{q}_{e1})T)\\
&\qquad -\Big\{\tfrac{-\nu_i}{\mu_i}+D_x\cos(\vartheta(k-j))-\tfrac{\upsilon_{e1}}{\omega_{e1}}\sin(\vartheta(k-j))+C_x\Big\}\\
&\qquad \sin(A(\dot{q}_{e1})T)\\
&\qquad +(\tfrac{\upsilon_{e1}}{\omega_{e1}}-\tfrac{\upsilon_{e2}}{\omega_{e2}})\cos(\vartheta(j'))\Big]\cos(A(\dot{q}_{e2})T)\\
&\qquad -\left\{\tfrac{-\varphi'_i}{\phi'_i}+D_x\cos(\vartheta(j'))-\tfrac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(j'))+C_x\right\}\\
&\qquad \sin(A(\dot{q}_{e2})T)\\
&\qquad +D_x\sin(\vartheta(k))+\tfrac{\upsilon_{e2}}{\omega_{e2}}\cos(\vartheta(k))-C_y
\end{aligned}\right.$$