2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Management of visual signal loss during image based visual servoing

A. Durand Petiteville[1,2], S. Durola[1,2], V. Cadenat[1,2] and M. Courdesses[1,2]

Abstract— **In this paper, we address the problem of the total visual features loss during visual servoing. We present a new method allowing to reconstruct these features even if the image is completely unavailable. The proposed method has been developed for a 6 degree-of-freedom (DOF) calibrated camera and a static landmark of interest which can be characterized by point features. It relies on a predictor/corrector pair coupled with a depth estimation algorithm. Numerous simulation results are provided and show the relevance of the proposed approach.**

I. Introduction

In the past decades, many works have addressed the problem of using informations provided by a vision system to control a robot. Such techniques are commonly known as visual servoing [1]. Visual servoing is roughly classified into two main categories: Image Based Visual Servoing (IBVS) and Position Based Visual Servoing (PBVS). In the first approach, the visual features extracted from the image (*e.g.* points, lines and moments [2]) are directly used in the control loop, whereas in the second one they are used to compute the robot state. Therefore, for both approaches, the visual features are mandatory to calculate the control inputs. In this paper, we focus on IBVS, although the proposed methods can be applied to PBVS.

Several phenomenons can lead to the lack of visual features: camera temporary breakdown, image processing errors, or even occlusions of the landmarks by a piece of the robot or an environment obstacle. To guarantee the success of a visual servoing task, it is then necessary to develop a method managing the loss of the visual signal.

Since few years, the visual signal loss problem has been addressed in several works and various approaches related to different scientific domains (control theory, vision, planning) have been proposed. A brief overview is presented hereafter. In [3] and [4] authors propose some control techniques allowing to deal with the partial loss of the visual features. In [5], [6], [7] and [8] solutions based on tracking treat the problem of total occlusions of the landmark by using measures in the current image. Another approach based on planning techniques [9] and [10] proposes to compute a path free of occlusions by using a metric model of the environment. Finally, it is possible to compute the visual features knowing the landmark dimensions and the robot pose with respect to this latter. However, none of these methods allow to deal with all the phenomenons leading to the visual features loss (camera breakdown, occlusions, ...) and especially when the image is no longer available. To be able to deal with the most general cases, it is mandatory to develop a method which can be used without any current image. Moreover, to improve the robot autonomy, the method should not require any knowledge about the landmark dimensions and the robot pose in the scene or with respect to the landmark. Such a method has been presented in [11] for a static landmark, point visual features and a 3 DOF calibrated camera. It is based on a prediction of the next visual features knowing the previous ones, the associated depths and the camera motion. This method has to be completed with an estimator of the visual features depth.

The visual features depth problem has already been addressed in many works but, to the best of authors' knowledge, none of them is suitable to manage a visual signal loss during a visual servoing. Indeed, in this particular case a correct visual feature depth estimation has to be available as soon as possible to handle occlusions the best way. It will be also necessary to compute it sufficiently rapidly with respect to the control law sampling period, in order to insure a nice behavior of the feedback. A brief overview of some interesting works is presented herebelow. In [12], the authors derive and compare 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 as the epipolar geometry [13] [14], stereovision [15], or even structure from motion techniques [16]. 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 uses a nonlinear observer to estimate the depth [17]. A similar approach, relying on a minimal nonlinear observer, is proposed in [18]. In [19], a method based on the visual features prediction equations for a 3 DOF camera and a predictor/corrector pair has been developed. These techniques have been analyzed and compared in [20] and the results have shown that the two first methods are quite difficult to tune whereas the last one is suitable with the visual servoing context.

In this paper, we propose to extend the previously obtained results concerning the visual features prediction [11] and the associated depth [19] to a 6 DOF calibrated camera. The paper is organized as follows. In the sequel, we introduce the system model before presenting the method allowing to compute the current visual features using the previous ones and the associated depth. Next, we describe a predictor/corrector pair estimating the visual features depth. Finally, we present a set of simulations concerning the visual features depth reconstruction and the visual signal loss management.

[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2]Univ de Toulouse, UPS, LAAS, F-31400, Toulouse, France
[adurandp,sdurola,cadenat,courdess]@laas.fr

II. PRELIMINARIES

In this section, we first address modeling aspects and we present the calibrated pinhole camera, the camera state vector and the camera control inputs. We end this section recalling the basic principles of IBVS.

A. The camera pinhole model

The pinhole camera model consists of an image plane I attached to an orthonormal frame $F_C(C, \vec{x}_C, \vec{y}_C, \vec{z}_C)$. The image plane I is parallel to the $(\vec{x}_C, \vec{y}_C)$ plane and the distance between those two planes along $\vec{z}_C$ is known as the focal length f (see Fig. 1). Its value is provided by a preliminary calibration step.

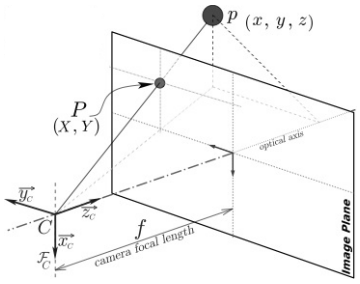


Fig. 1: The camera pinhole model

First of all, let us state the problem. To do so, we propose to define the camera state in the world frame $F_O(O, \vec{x}_O, \vec{y}_O, \vec{z}_O)$. The position of F_C will be represented by the cartesian coordinates x_C, y_C, z_C of point C in F_O. As for the frame orientation, different choices are possible: Euler's angles, quaternions, Bryant's angles, ... In this paper, we have chosen to express the orientation using the Euler's angles. However the work presented thereafter is independent of this choice. The same methodology can be applied using another type of orientation representation.

Using the Euler's angles the orientation of the frame F_C with respect to the frame F_O is defined by three angles: ψ around $\vec{z}_C$, θ around $\vec{x}_\psi$, and φ around $\vec{z}_\theta$ (see Fig. 2).

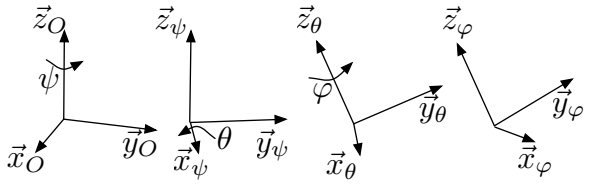


Fig. 2: Euler angles. $F_\psi(\vec{x}_\psi, \vec{y}_\psi, \vec{z}_\psi)$ is obtained after rotating F_O of ψ around $\vec{z}_O$. F_θ results of a rotation θ of F_ψ around $\vec{x}_\psi$. F_φ is obtained after rotating of φ around $\vec{z}_\theta$. Finally, $F_C = F_\varphi$.

Finally, we obtain the state vector $\chi = [x_C, y_C, z_C, \psi, \theta, \varphi]^T$, which allows to determine the camera pose in the world frame F_O.

To control the camera, we use an IBVS. This method allows to compute a kinematic screw $T^{F_C}_{C/F_O}$ calculated at point C with respect to F_O and expressed in F_C. This kinematic screw controls the six degrees of freedom of the camera to successfully realize the task. It can be decomposed as follows:

$$T^{F_C}_{C/F_O} = \begin{bmatrix} V^{F_C}_{\vec{x}_C} & V^{F_C}_{\vec{y}_C} & V^{F_C}_{\vec{z}_C} & \Omega^{F_C}_{\vec{x}_C} & \Omega^{F_C}_{\vec{y}_C} & \Omega^{F_C}_{\vec{z}_C} \end{bmatrix}^T \tag{1}$$

where $V^{F_C}_{\vec{x}_C}$, $V^{F_C}_{\vec{y}_C}$ and $V^{F_C}_{\vec{z}_C}$ are respectively the linear velocities with respect to $\vec{x}_C$, $\vec{y}_C$ and $\vec{z}_C$ expressed in F_C. Following the same idea, $\Omega^{F_C}_{\vec{x}_C}$, $\Omega^{F_C}_{\vec{y}_C}$ and $\Omega^{F_C}_{\vec{z}_C}$ are respectively the angular velocities with respect to $\vec{x}_C$, $\vec{y}_C$ and $\vec{z}_C$ expressed in F_C.

Remark: *Expressing the kinematic screw in F_O, it can be shown that the state variation $\dot{\chi}$ is given by:*

$$\dot{\chi} = \begin{bmatrix} V^{F_O}_{\vec{x}_C} \\ V^{F_O}_{\vec{y}_C} \\ V^{F_O}_{\vec{z}_C} \\ \Omega^{F_O}_{\vec{z}_C} - \dot{\varphi}\cos(\theta) \\ \Omega^{F_O}_{\vec{x}_C}\cos(\psi) + \Omega^{F_O}_{\vec{y}_C}\sin(\psi) \\ \left(\Omega^{F_O}_{\vec{x}_C}\sin(\psi) - \Omega^{F_O}_{\vec{y}_C}\cos(\psi)\right)/\sin(\theta) \end{bmatrix} \tag{2}$$

By integrating (2)*, it is possible to calculate the state evolution depending on the applied kinematic screw.*

B. Image based visual servoing

Image based visual servoing consists in controlling the camera using visual features as feedback data. The following error is generally defined [1] by $e_{vs} = s - s^*$ where s and s^* are vectors consisting respectively of the current and desired visual features. The choice of the quantity and the nature of visual features determines the type of task that is performed (see for example [2]). In this work, we consider a static landmark described by a set of points and a constant s^*. For this particular case, to make e_{vs} vanish the following classical controller is designed:

$$T^{F_C}_{C/F_O} = -L^+ \lambda_{vs}(s - s^*) \tag{3}$$

where L^+ is the pseudo-inverse of the interaction matrix L [1] related to s and λ_{vs} a positive scalar or a definite positive matrix which imposes the exponential decrease [1]. Equation (3) clearly shows that visual features s must be available at each instant to perform an image based visual servoing. If they become unavailable due to a landmark occlusion for example, the task fails. For this reason, it is mandatory to consider the visual signal loss issue.

III. VISUAL SIGNAL LOSS MANAGEMENT

Now we focus on our visual signal loss management method. In this paper, we extend the works presented in [11] and [19] which allow to deal with this problem for a 3 DOF calibrated camera. In these works, we have proposed the following algorithm which has two running modes, depending on the visual features availability [19]:

- If the image can be provided by the camera, we first extract the current visual features s_k. Then, once a sufficient

number of images n_{pc} is available, a depth estimation process is launched. It allows to compute the depth d_k associated to s_k from the visual features $s_{k-n_{pc}\to k}$ obtained with the last n_{pc} images. Thus, the data collected when the image is available are used to estimate the depth d_k.

- If the image is no more available, a visual features reconstruction process is launched. It allows to predict the current visual features s_k and their associated depth d_k using the previous data s_{k-1} and d_{k-1}. It is then possible to get a value for the visual features despite a total loss of the image.

A sketch of this algorithm is presented herebelow.

Algorithm 1 Visual Signal Loss Management

```
k = 0
n_pc > 0
while VisualServoing do
    if ImageAvailable then
        s_k = ImageProcessing(im_k)
        if k − n_pc ≥ 0 then
            d_k = DepthEstimation(s_{k−n_pc→k})
        else
            d_k = NaN
        end if
    else if k > 0 then
        s_k = FeaturesPrediction(s_{k−1}, d_{k−1})
        d_k = DepthPrediction(d_{k−1})
    end if
    k = k + 1
end while
```

In [11] and [19], the two above mentioned processes were restricted to the case of a 3 DOF camera. Here, we extend the methods used in the processes to the most general case of a 6 DOF camera. We present in the following subsections the two corresponding methods.

A. Visual features prediction

To build the predictor, we have first defined two homogeneous transformation matrices. The first one links the camera frame to the image plane, and the second one the two camera frames at two separate instants.

Concerning the first matrix, the coordinates (X, Y) of a point P on the image plane I, which correspond to the projection of a point p with coordinates (x, y, z) in the camera frame F_C, are obtained using the perspective projection equations for a pinhole camera (see Fig. 1). Defining the homogeneous coordinates $\underline{p} = (x, y, z, 1)^T$ and $\underline{P} = (X, Y, z, 1)^T$, it is possible to define the homogeneous transformation matrix $H_{I/C}$ between the frame F_C and the image plane I as follows:

$$\begin{bmatrix} X(t) \\ Y(t) \\ z(t) \\ 1 \end{bmatrix} = H_{I/C(t)} \times \underline{p} = \begin{bmatrix} \frac{f}{z(t)} & 0 & 0 & 0 \\ 0 & \frac{f}{z(t)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ 1 \end{bmatrix} \tag{4}$$

Now we focus on the homogeneous transformation matrix between two camera states at instants t_1 and t_2. To this aim, we first compute the homogeneous transformation matrix $H_{O/C(t)}$ between the camera frame F_C at instant t and the world frame F_O as follows[1]:

$$H_{O/C(t)} = \begin{bmatrix} H^{11}_{O/C}(t) & H^{12}_{O/C}(t) & H^{13}_{O/C}(t) & x_C(t) \\ H^{21}_{O/C}(t) & H^{22}_{O/C}(t) & H^{23}_{O/C}(t) & y_C(t) \\ H^{31}_{O/C}(t) & H^{32}_{O/C}(t) & H^{33}_{O/C}(t) & z_C(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

with

$$\begin{gathered} H^{11}_{O/C}(t) = \cos(\psi(t))\cos(\varphi(t)) - \sin(\psi(t))\cos(\theta(t))\sin(\varphi(t)) \\ H^{12}_{O/C}(t) = -\cos(\psi(t))\sin(\varphi(t)) - \sin(\psi(t))\cos(\theta(t))\cos(\varphi(t)) \\ H^{13}_{O/C}(t) = \sin(\psi(t))\sin(\theta(t)) \\ H^{21}_{O/C}(t) = \sin(\psi(t))\cos(\varphi(t)) + \cos(\psi(t))\cos(\theta(t))\sin(\varphi(t)) \\ H^{22}_{O/C}(t) = -\sin(\psi(t))\sin(\varphi(t)) + \cos(\psi(t))\cos(\theta(t))\cos(\varphi(t)) \\ H^{23}_{O/C}(t) = -\cos(\psi(t))\sin(\theta(t)) \\ H^{31}_{O/C}(t) = \sin(\theta(t))\sin(\varphi(t)) \\ H^{32}_{O/C}(t) = \sin(\theta(t))\cos(\varphi(t)) \\ H^{33}_{O/C}(t) = \cos(\theta(t)) \end{gathered}$$

Using (5), we can compute $H_{C(t_1)/C(t_2)}$:

$$H_{C(t_1)/C(t_2)} = H^{-1}_{O/C(t_1)} \times H_{O/C(t_2)} \tag{6}$$

The tools to build the image predictor are now available. Using (4) and (6) we obtain the following prediction of $P(t_2)$ knowing $P(t_1)$:

$$\underline{P}(t_2) = H_{I/C(t_2)} H^{-1}_{C(t_1)/C(t_2)} H^{-1}_{I/C(t_1)} \underline{P}(t_1) \tag{7}$$

Defining H^{ij} the element in the i^{th} row and j^{th} column of $H_{C(t_1)/C(t_2)}$ and recalling that $\underline{P}(t_1) = (X(t_1), Y(t_1), z(t_1), 1)^T$ and $\underline{P}(t_2) = (X(t_2), Y(t_2), z(t_2), 1)^T$, this equation can be rewritten as:

$$X(t_2) = \frac{z(t_1)}{z(t_2)} \left[H^{11} X(t_1) + H^{21} Y(t_1) + H^{31} f + \frac{f T_1}{z(t_1)} \right] \tag{8}$$

$$Y(t_2) = \frac{z(t_1)}{z(t_2)} \left[H^{12} X(t_1) + H^{22} Y(t_1) + H^{32} f + \frac{f T_2}{z(t_1)} \right] \tag{9}$$

$$z(t_2) = z(t_1) \left[H^{13} \frac{X(t_1)}{f} + H^{23} \frac{Y(t_1)}{f} + H^{33} \right] + T_3 \tag{10}$$

where

$$\begin{aligned} T_1 &= -H^{11}H^{14} - H^{21}H^{24} - H^{31}H^{34} \\ T_2 &= -H^{12}H^{14} - H^{22}H^{24} - H^{32}H^{34} \\ T_3 &= -H^{13}H^{14} - H^{23}H^{24} - H^{33}H^{34} \end{aligned}$$

In order to use (8) and (9) as image predictors, we need to provide $X(t_1)$, $Y(t_1)$ and $z(t_1)$. The two first ones can

[1]This matrix is valid only for a state vector using the Euler's angles.

be obtained from the last image before the occlusion. The depth[2] $z(t_1)$ has to be estimated using the available images set. To do so, we have developed a predictor/corrector pair which is described hereafter.

B. Visual features depth estimation

To estimate the visual features depth, we propose to build a predictor/corrector based on a number n_{pc}[3] of visual data $X(t_j)$ and $Y(t_j)$, with $t_j = t - jT_e$, $j \in [1, ..., n_{pc}]$ and T_e the sampling rate of the visual servoing. Using this notation, equations (8), (9) and (10) can be rewritten as follows:

$$X(t) = \frac{z(t_j)}{z(t)}\left[H^{11}X(t_j) + H^{21}Y(t_j) + H^{31}f + \frac{fT_1}{z(t_j)}\right] \quad (11)$$

$$Y(t) = \frac{z(t_j)}{z(t)}\left[H^{12}X(t_j) + H^{22}Y(t_j) + H^{32}f + \frac{fT_2}{z(t_j)}\right] \quad (12)$$

$$z(t) = z(t_j)\left[H^{13}\frac{X(t_j)}{f} + H^{23}\frac{Y(t_j)}{f} + H^{33}\right] + T_3 \quad (13)$$

where H^{ij} is the element in the i^{th} row and j^{th} column of $H_{C(t_j)/C(t)}$. As our goal is to estimate $z(t)$, we have to make (11) and (12) independent of $z(t_j)$. To this aim, we first rewrite (13) as follows:

$$z(t_j) = \frac{z(t) - T_3}{\alpha_1} \quad (14)$$

with

$$\alpha_1 = H^{13}\frac{X(t_j)}{f} + H^{23}\frac{Y(t_j)}{f} + H^{33}$$

Using (14), and rewriting (11) and (12) leads to:

$$X(t) = \alpha_2\frac{z(t) - T_3}{\alpha_1 z(t)} + \frac{f}{z(t)}T_1 \quad (15)$$

$$Y(t) = \alpha_3\frac{z(t) - T_3}{\alpha_1 z(t)} + \frac{f}{z(t)}T_2 \quad (16)$$

with

$$\alpha_2 = H^{11}X(t_j) + H^{21}Y(t_j) + H^{31}f$$

and

$$\alpha_3 = H^{12}X(t_j) + H^{22}Y(t_j) + H^{32}f$$

We can now consider (15) and (16) as the predictors $\hat{X}(t|t_j)$ and $\hat{Y}(t|t_j)$ which depend on $\hat{z}(t|t_j)$:

$$\hat{X}(t|t_j) = \alpha_2\frac{\hat{z}(t|t_j) - T_3}{\alpha_1\hat{z}(t|t_j)} + \frac{f}{\hat{z}(t|t_j)}T_1 \quad (17)$$

$$\hat{Y}(t|t_j) = \alpha_3\frac{\hat{z}(t|t_j) - T_3}{\alpha_1\hat{z}(t|t_j)} + \frac{f}{\hat{z}(t|t_j)}T_2 \quad (18)$$

The predictors are computed and the correction step can now be performed. To do so we propose to define the criterion (19) which expresses the difference between the predicted visual features, $\hat{X}(t|t_j)$ and $\hat{Y}(t|t_j)$, and the measured ones, $\tilde{X}(t)$ and $\tilde{Y}(t)$.

$$C = \sum_{j=1}^{n_{pc}}\left(\hat{X}(t|t_j) - \tilde{X}(t)\right)^2 + \left(\hat{Y}(t|t_j) - \tilde{Y}(t)\right)^2 \quad (19)$$

To estimate the depth, we minimize criterion (19) with respect to $\hat{z}(t|t_j)$. Differentiating C with respect to $\hat{z}(t|t_j)$ leads to:

$$\begin{array}{rcl}\frac{\partial C}{\partial\hat{z}(t|t_j)} = \sum_{j=1}^{n_{pc}} & \Big\{2\Big(\hat{X}(t|t_j) - \tilde{X}(t)\Big)\frac{\partial\hat{X}(t|t_j)}{\partial\hat{z}(t|t_j)} \\ + & 2\Big(\hat{Y}(t|t_j) - \tilde{Y}(t)\Big)\frac{\partial\hat{Y}(t|t_j)}{\partial\hat{z}(t|t_j)}\Big\}\end{array} \quad (20)$$

where

$$\frac{\partial\hat{X}(t|t_j)}{\partial\hat{z}(t|t_j)} = \frac{\alpha_2 T_3}{\alpha_1\hat{z}(t|t_j)^2} - \frac{fT_1}{\hat{z}(t|t_j)^2} \quad (21)$$

$$\frac{\partial\hat{Y}(t|t_j)}{\partial\hat{z}(t|t_j)} = \frac{\alpha_3 T_3}{\alpha_1\hat{z}(t|t_j)^2} - \frac{fT_2}{\hat{z}(t|t_j)^2} \quad (22)$$

By replacing (17), (18), (21) and (22) in (20), we obtain:

$$\begin{array}{rcl}\frac{\partial C}{\partial\hat{z}(t|t_j)} & = & \sum_{j=1}^{n_{pc}} \Big[\frac{\alpha_2(\hat{z}(t|t_j)-T_3)}{\alpha_1\hat{z}(t|t_j)} + \frac{fT_1}{\hat{z}(t|t_j)} - \tilde{X}(t)\Big] \\ & & \times \Big[\frac{\alpha_2 T_3}{\alpha_1\hat{z}(t|t_j)^2} - \frac{fT_1}{\hat{z}(t|t_j)^2}\Big] \\ & & + \Big[\frac{\alpha_3(\hat{z}(t|t_j)-T_3)}{\alpha_1\hat{z}(t|t_j)} + \frac{fT_2}{\hat{z}(t|t_j)} - \tilde{Y}(t)\Big] \\ & & \times \Big[\frac{\alpha_3 T_3}{\alpha_1\hat{z}(t|t_j)^2} - \frac{fT_2}{\hat{z}(t|t_j)^2}\Big] \\ & = & 0\end{array} \quad (23)$$

Finally, thanks to (23), we obtain an estimator of the depth using n_{pc} images:

$$\hat{z}(t|t) = \frac{-\sum_{j=1}^{n_{pc}} Num_j}{\sum_{j=1}^{n_{pc}} Den_j} \quad (24)$$

with

$$\begin{array}{rcl}Num_j & = & \alpha_2 T_3\left(-\frac{\alpha_2 T_3}{\alpha_1} + 2fT_1\right) - \alpha_1 f^2 T_1^2 \\ & + & \alpha_3 T_3\left(-\frac{\alpha_3 T_3}{\alpha_1} + 2fT_2\right) - \alpha_1 f^2 T_2^2\end{array}$$

and

$$\begin{array}{rcl}Den_j & = & \alpha_2\left(\frac{\alpha_2 T_3}{\alpha_1} - fT_1 - T_3\tilde{X}(t)\right) + \alpha_1 fT_1\tilde{X}(t) \\ & + & \alpha_3\left(\frac{\alpha_2 T_3}{\alpha_1} - fT_2 - T_3\tilde{Y}(t)\right) + \alpha_1 fT_2\tilde{Y}(t)\end{array}$$

[2]Usually $z(t_1)$ is called depth by misuse of language. However it corresponds to the z-coordinate of the point, which is not the real depth.

[3]Note that n_{pc} images are used to improve the depth estimation even in presence of a small signal/noise ratio. This problem is presented in [19].

Thus, using estimator (24), we can provide an initial depth value for the visual data predictor. We then can manage a visual signal loss by estimating the visual data thanks to the developed tools.

IV. Simulations

To illustrate the efficiency of our approach, we have simulated several depth estimations and visual signal losses management using Matlab^{TM}software. For all the simulations, the camera is controlled thanks to an IBVS using the projection on the image plane of four points as visual features. The camera has to reach the state $\chi = [2, 0, 0, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}]^T$ (the green camera) from $\chi = [0, 1, 1, \frac{\pi}{4}, \frac{\pi}{2}, \frac{\pi}{4}]^T$ (the red camera) as shown in figure 3. The corresponding evolution of the visual features is given on the figure 4(a) when the data are noise-free. The red and green crosses respectively represent the initial and desired visual features. In figure 4(b), the real and estimated depths of one visual feature is presented. Without any perturbation during the acquisition process, the depth is immediately and perfectly estimated.

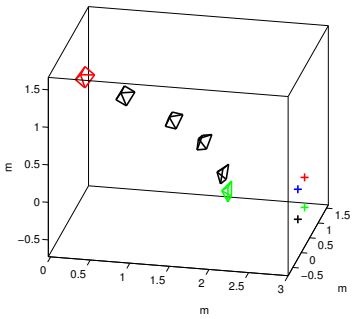
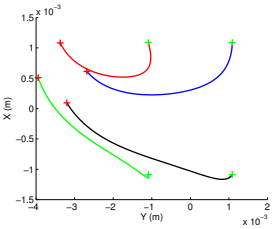


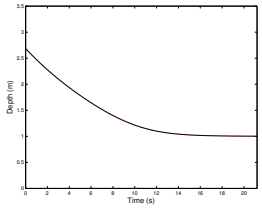
Fig. 3: Camera trajectory during the IBVS

In a second step, we have evaluated our methods in a noisy context. We have studied six cases. In the first three ones, a 5% odometry error and a 2 pixels random noise are introduced. Cases $\#1$, $\#2$, $\#3$ respectively use $n_{pc} = 10$, $n_{pc} = 25$ and $n_{pc} = 50$ images. For the last three ones, the errors are 10% and 5 pixels. Cases $\#4$, $\#5$, $\#6$ respectively use $n_{pc} = 10$, $n_{pc} = 25$ and $n_{pc} = 50$ images. For this simulation, the IBVS has been performed 100 times to obtain significant data sets. Figures 4(c) and 4(d) respectively represent the realizations mean and the variance. For all cases, our estimator converges towards the real depth value which shows the efficiency of our approach even in an unfavorable context. However, for a too small value of n_{pc}, the estimator is less efficient at the end of the IBVS, because of a too small signal/noise ratio. Indeed, in this particular case, due to the imposed exponential decrease, the control inputs nearly vanish inducing small displacements of the camera. The visual data are then too similar to provide a suitable depth estimation. This problem is overcome by choosing a large value for n_{pc}. Generally, the value of n_{pc} depends on the signal/noise ratio. This latter could be increased by using a larger value of the sampling time or by performing larger displacements between two images.

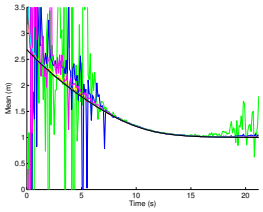
We have also analyzed the effect of coarse calibration by introducing successively an error of 2%, 5% and 10% on the focal value. The other data are perfectly acquired. As we can see in figure 4(e), the convergence rate is greater than in the previous case. However, the estimator converges towards the depth real value with negligible bias.



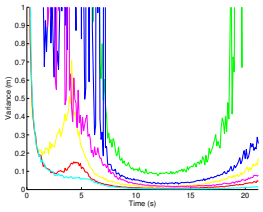
(a) Visual features evolution



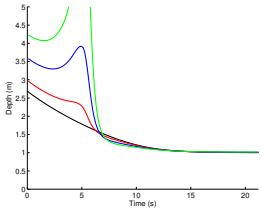
(b) Depth estimation without noise. Black line is the real depth and dotted red line is the estimated one.



(c) Realizations mean with noises. Yellow #1, Red #2 , Cyan #3, Green #4, Blue #5, Magenta #6.



(d) Variance with noises. Yellow #1, Red #2 , Cyan #3, Green #4, Blue #5, Magenta #6.

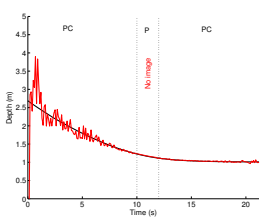


(e) Depth estimation with an error on focal. Red: 2%, Blue: 5%, Green: 10%

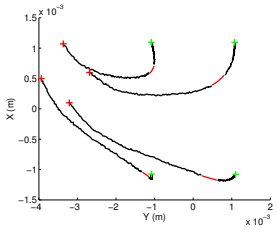
Fig. 4: Depth estimation

Finally, we have simulated visual signal losses during an IBVS in a noisy context (2 pixels and 5% error are respectively introduced on the visual data and odometry). Two cases have been considered: in the first one only a $1.5s$ signal loss occurs while in the second one two $1s$ occlusions have been generated. We have obtained the results presented on figure 5. In both cases, the depth is well estimated either by the predictor/corrector pair (24) or the predictor (10). Thanks to these suitable depth estimations, the visual signal losses are perfectly managed. Indeed figures 5(b) and 5(d) show the efficiency of the reconstruction. Thanks to these methods, our vision based control law can still be applied to the robot even when the image is no more available.

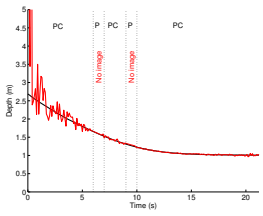
Remark: As shown in figures 5(a) and 5(c), the estimated depth value at the beginning of the IBVS, is not suitable to predict the visual features. To overcome this problem, an initialization phase can be realized as in [21].



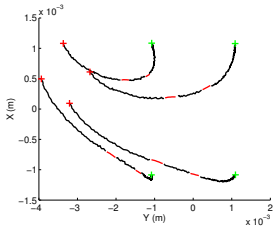
(a) Depth estimation with one occlusion. Black: real depth, Red: estimated depth. PC: predictor/corrector, P: predictor



(b) Features evolutions with one occlusion. Black: measured features, Red: estimated features



(c) Depth estimation with two occlusions. Black: real depth, Red: estimated depth. PC: predictor/corrector, P: predictor



(d) Features evolutions with two occlusions. Black: measured features, Red: estimated features

Fig. 5: IBVS with visual signal loss management

V. Conclusion

In this paper, we have dealt with the image features total loss during visual servoing. We have presented a novel approach which allows to reconstruct these features and to extend the results presented in [11] and [19]. The method addresses the case of a 6 DOF calibrated camera and of a static landmark characterized by point features. It is based on a predictor/corrector pair coupled with a depth estimation algorithm. The obtained simulation results have shown the proposed method efficiency. Thanks to this approach, it becomes possible to keep on controlling the robot and executing the task, even if the image is not provided. It then offers a general answer to all phenomenons leading to visual features loss (camera breakdown, image processing failure, landmark occlusion, etc.), increasing the robot autonomy and the range of realizable tasks. Furthermore, if this approach has been described in conjunction with an IBVS control scheme, it can be used with any kind of vision-based controllers and in other related applications. For example, it could be possible to benefit from this method to automatically build the reference visual features s^*, as it has already been done in [21] for a 3 DOF camera. It could also be interesting to use the estimated visual features to feed the control law with a higher sampling period. Another relevant application would be to exploit our reconstruction algorithm in the predictive control domain [22]. In addition to these first prospects, we plan to experiment the proposed approach on different robots from our laboratory. We also believe that refining the focal value by coupling our approach with the non-linear observer proposed in [17] could help improving the accuracy of the results. Finally, an extension of this method to a dynamic environment will be studied in the longer term.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control, part 1 : Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, 2006.

[2] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *Robotics, IEEE Transactions on*, vol. 20, no. 4, pp. 713 – 723, aug. 2004.

[3] E. Malis, F. Chaumette, and S. Boudet, "2d 1/2 visual servoing," *IEEE Trans. on Rob. and Automation*, vol. 15, no. 2, pp. 234–246, 1999.

[4] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Transactions on Robotics and Automation*, vol. 21, 2005.

[5] J. Jackson, A. Yezzi, and S. Soatto, "Tracking deformable objects under severe occlusions," in *IEEE International Conference on Decision and Control*, Atlanta, USA, Decembre 2004.

[6] A. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *IEEE/RSJ 2004 International Conference on Intelligent Robots and Systems*, Sendai, Japan, Octobre 2004.

[7] V. Lepetit and P. Fua, "Monocular based-model 3d tracking of rigid objects," *found. Trends. Comput. Graph. Vis.*, vol. 1, 2006.

[8] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and V. Van Gool, "Online multi-person tracking-by-detection from a single, uncalibrated camera," *IEEE Transaction on pattern analysis and machine intelligence*, vol. 33, 2011.

[9] R. Sharma, "A framework for robot motion planning with sensors constraints," *IEEE Transaction o Robotics and Automation*, vol. 13, 1997.

[10] M. Kazemi, K. Gupta, and M. Mehrandezh, "Path-planning for visual servoing : A review and issues," *G. Chesi et K. Hashimoto, editors, Visual Servoing via Advanced Numerical Methods, Springer Verlag*, 2010.

[11] D. Folio and V. Cadenat, *Computer Vision - Treating Image Loss by using the Vision/Motion Link: A Generic Framework*. IN-TECH, 2008, ch. 4.

[12] Matthies, Kanade, and Szeliski, "Kalman filter-based algorithms for estimating depth in image sequences," *Int, Journal of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.

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

[14] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipoles," *Int. J. Comput. Vision*, vol. 33, no. 2, pp. 117–137, 1999.

[15] E. Cervera, P. Martinet, and F. Berry, "Robotic manipulation with stereo visual servoing," *Robotics and Machine Perception, SPIE International Group Newsletter*, vol. 1, no. 1, p. 3, 2002.

[16] C. Jerian and R. Jain, "Structure from motion: a critical analysis of methods," *IEEE Transactions on systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 572–588, 1991.

[17] De Luca, Oriolo, and Giordano, "Features depth observation for image based visual servoing: theory and experiments," *Int. Journal of Robotics Research*, vol. 27, no. 10, 2008.

[18] F. Morbidi and D. Prattichizzo, "Range estimation from moving camera : an immersion and invariance approach," in *IEEE 2009 International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[19] A. Durand Petiteville, M. Courdesses, and V. Cadenat, "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, June 2010.

[20] A. Durand Petiteville, V. Cadenat, M. Courdesses, F. Delpech de Frayssinet, and A. Magassouba, "A comparison of several approaches to perform a vision-based long range navigation," in *10th IEEE International workshop on Electronics, Control, Modelling, Measurement and Signals 2011*, Liberec, Czech Republic, June 2011.

[21] A. Durand Petiteville, V. Cadenat, and M. Courdesses, "An unified initialization phase to improve visual servoing in an unknown environment," in *7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italia, September 2010.

[22] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Trans. on Robotics*, vol. 26, no. 5, pp. 933–939, October 2010.