Handling visual features losses during a coordinated vision-based task with a dual-arm robotic system

Renliw Fleurmond, Viviane Cadenat

Handling visual features losses during a coordinated vision-based task with a dual-arm robotic system

Renliw Fleurmond[1] and Viviane Cadenat[1]

***Abstract*— We address the problem of total visual features losses during a coordinated vision-based task. We present a new method allowing to reconstruct these features even if the image is completely unavailable. Contrary to previous works, the proposed technique allows to deal with moving objects characterized by both points and lines features. Simulation results show the relevance of the proposed approach.**

I. INTRODUCTION

Nowadays, robots are more and more expected to operate in human friendly environments. This trend is visible in both industrial and service settings. The considered applications are various and range from domestic tasks [1] to manufacturing missions [2]. To successfully realize these typical human tasks, bi-manual systems have been intensively developed. As a consequence, the interest for dual arm manipulation has increased [3], [4]. This problem can be tackled at different levels and through many approaches. See [4] for a survey.

In previous works [5], we have addressed this problem using low-level control techniques, and more precisely image-based visual servoing (IBVS). This technique consists in using visual features to control a robot [6]. We have proposed a multi-camera image-based control strategy allowing to truly coordinate both arms to recap a pen [5]. However, two problems still occur. First, in the considered task, it is not always possible to insure the visual features visibility, especially at the end when the pen is about to be recapped. It follows that this task cannot be performed if the visual features total loss is not tolerated. In addition, other phenomenons such as camera temporary breakdowns or image processing failures can also lead to the lack of visual features and to a task failure. The second problem is directly related to the IBVS controller. Indeed, this controller classically depends on the interaction matrix which relates the visual signal variation to the camera and objects motions. This matrix is function of both the measured visual features and some generally unknown 3D information [6]. Traditionally, the interaction matrix is expressed at the absolute pose to be reached [6]. However, this solution cannot be used in our case. Indeed, to guarantee a true coordination between the arms, it is necessary to reach a desired relative pose between both end-effectors, and not an absolute one [5], [7]. Thus, the interaction matrix must be estimated at each instant.

Our objective is to develop a method allowing to cope with these two problems: the image cues total losses and the interaction matrix estimation. Concerning the first one, we need methods able to tolerate the total occlusions and not to simply avoid them as [8], [9]. We have first considered tracking techniques. They allow to follow mobile or static objects despite total occlusions. See for example [10], [11]. However, they appear to be unsuitable in our context because they generally rely on measures extracted from the image. Other solutions consist in virtually projecting the visual features describing the object, given the relative pose between this object and the camera. This method requires the knowledge of the object 3D structure whereas in IBVS, this information is a priori unavailable. To recover this structure, it is possible to use methods based on nonlinear observers [12], [13]. However, the first one depends on gains which are difficult to tune a priori [14], while, in the second one, the camera motion is constrained to maximize the estimation quality (active vision). It is also possible to build a predictor/corrector to reconstruct the depth of pointwise features when images are available and then use this information to estimate the visual cues if an occlusion occurs [14]. However, these solutions cannot be used in our context. First of all, they all are restricted to the case of a static object characterized by only point-wise visual features and seen from only one camera. In our case, the cap and the pen are moving because they are manipulated by two arms. Furthermore, they are represented by two kinds of visual features (a point and a line [5]) and seen by two cameras.

In this paper, we propose a method adapted to the different constraints imposed by the considered task. This method will be able to reconstruct the 3D structure of the objects in the presence of images, and then to use this result to compute the visual features whenever needed. In addition, the estimated data will be also used to compute the necessary interaction matrices, showing that our method is able to cope with the two highlighted problems.

The paper is organized as follows: in section 2, we briefly recall the initial vision-based control strategy. Then, we describe the method allowing to reconstruct the objects 3D structure and the corresponding visual features whenever they become unavailable. Finally, we present simulation results showing the efficiency of our approach.

II. PRELIMINARIES

A. The robotic system

1) The robot modeling: Our robotic platform is the PR2 from Willow Garage. It consists of an omni-directional mobile base equipped with two 7-DOF arms. We suppose that only the arms are moving. The robot is equipped with

[1]Renliw Fleurmond and Viviane Cadenat are with CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France and Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France rfleurmo@laas.fr, cadenat@laas.fr

six cameras: four on the head and one on each forearm. Here, we only use one head camera and one in a forearm to get complementary points of view. We assume that the cap and the pen: (i) are respectively gripped by the right and left arms; (ii) can be seen initially by both cameras; (iii) are cylindrical and can be described by the same parameters set.

We denote by q_r and q_l (respectively $\dot{q}_r$ and $\dot{q}_l$) the joint coordinates (respectively the joint velocities) of the right and left arms. We introduce the different frames which are necessary to model our problem (see Fig. 1). The frames F_w, F_r and F_l are respectively linked to the world, the right and left end effectors, while frames F_f and F_m are attached to the fixed and mobile cameras. From them, the kinematic and differential kinematic models of both arms have been determined [5]. From these models, we can deduce the poses and the kinematic screws of the end-effectors and of the different cameras with respect to F_w.

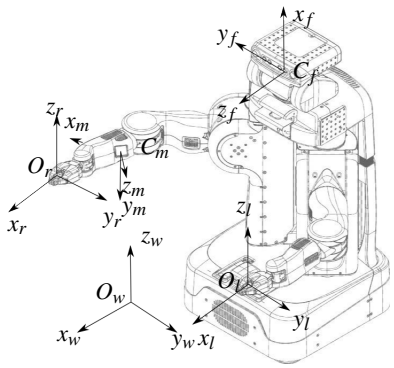

Fig. 1: The robotic platform and the different frames.

2) The camera modeling: All cameras are supposed to be calibrated. They are modeled using the pinhole model [15]. In this model, each camera consists of an image plane I attached to an orthonormal frame linked to the considered camera $F_C(C;\ x_C;\ y_C;\ z_C)$ as shown on Fig. 2. The image

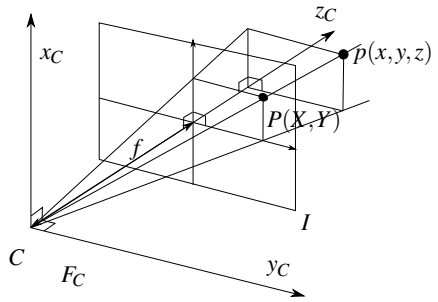

Fig. 2: Pinhole camera model

plane I is parallel to the $(x_C;\ y_C)$ plane and the distance between those two planes along z_C is known as the focal length f (see Fig. 2). When considering a pinhole camera model, a 3D point of the environment p whose coordinates in the camera frame are given by the triple (x,y,z), can be projected in the image plane. The coordinates of the so-obtained 2D point P are given by the perspective projection equations as follows: $X = fx/z$ and $Y = fy/z$.

B. Modeling the task

1) Choosing the visual features: The cap and the pen are supposed to be cylindrical objects. To perform the task, it is necessary to position them in the image and to monitor the translation of each cylinder along its axis. Three visual features suffice to respect these requirements [5]: the polar parameters $(\rho,\ \theta)$ of the cylinder inertia axis (the grey coloured part in Fig. 3) and parameter k which expresses the position of one end point r $(x_r;y_r)$ of the cylinder on the straight line (see Fig. 3). Thus our image cues vector S is defined by $S = [\rho\ \theta\ k]^T$. We denote by S_{fc} and S_{mc} the visual features corresponding to the cap seen by respectively the fixed and the moving cameras, by S_{fp} and S_{mp} the visual features corresponding to the pen seen by respectively the fixed and the moving cameras. More details about the visual features and their interaction matrices can be found in [5].

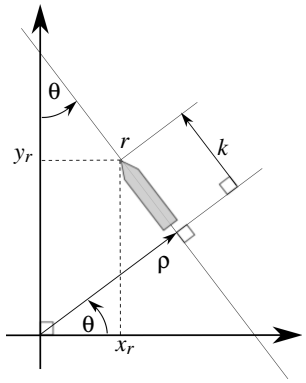

Fig. 3: Visual features used: ρ, θ and k.

2) Definition of the manipulation task: To recap the pen, the robot has to align both cylinders before connecting them. We then divide the task into three subtasks: (i) making both cylinders axes coplanar while ensuring a sufficient distance between the two objects ; (ii) aligning the latter axes ; and (iii) bringing the cap near the pen. Thus, it suffices to sequence these three substasks to recap the pen. Now, it remains to model them and to design a suitable control law. To do so, we have chosen to use the task function formalism [16]. In this framework, the task is modeled by a function $e(q,t)$ chosen so that its regulation will ensure the realization of the mission. We have shown in [5] that the sequence of the following task functions e_i (i=$\{1,2,3\}$) represent the above mentioned task and can be written as follows :

$$e_i = H_i \cdot \left(\begin{bmatrix} S_{mc} - S_{mp} \\ S_{fc} - S_{fp} \end{bmatrix} - \alpha_i \right) \tag{1}$$

where H_i is an activation matrix allowing to select only the necessary visual features for the current task function e_i, and α_i is a constant vector intended to monitor the distance between the cap and the pen. Note that the task functions e_i express as a relative error between the visual features representing the cap and the pen. In this way, the task can be executed in a coordinated manner. Note also that this task is based on visual data provided by several cameras. A multi-cameras visual servoing will then be designed.

C. The multi-cameras vision based control strategy

As classically done in the visual servoing area [6], we have chosen to impose an exponential decay to make e_i vanish. The corresponding controller is given by [5], [6] :

$$\dot{q} = -J_i^{+}\lambda_i e_i \tag{2}$$

where λ_i is a positive gain or a positive-definite matrix, J_i the Jacobian of the task function, J_i^{+} its Moore-Penrose inverse and $q = \begin{bmatrix} q_r^T & q_l^T \end{bmatrix}^T$. The expression of J_i and its computation are detailed in [5]. However, the above mentioned control law only makes the current task function e_i decrease to zero. From this, it follows that the different control laws must be sequenced to perform successively the three subtasks. To do so, we use the following control law [5]:

$$\dot{q}(t) = \dot{q}_N(t) - \exp\left(-\mu(t-t_s)\right)\left(\dot{q}_N(t_s) - \dot{q}_C(t_s)\right) \tag{3}$$

where $\dot{q}_C(t)$ and $\dot{q}_N(t)$ are respectively the current and next controllers to be applied to the robot, t_s is the switching instant and $\mu \in \mathbb{R}^{*+}$ regulates the transition delay. The switching between the controllers occurs when the norm of the corresponding tasks drops under a given threshold.

III. OCCLUSION HANDLING STRATEGY

As previously mentioned, we cope with two problems: the image cues total losses and the interaction matrix estimation. To do so, we have chosen to estimate the 3D structure during the execution of the task, while the images are available. We then use this result to compute the interaction matrices and the visual features whenever it is needed. The proposed method is summarized in the algorithm 1. At first, very rough values for the 3D structure are provided from which the interaction matrices are determined. These values are refined by our estimation process using the available images. Once this process has converged ($t > t_c$), the interaction matrices are computed with the reconstructed 3D structure. At this time, the system is ready to deal with the problem of occlusions. If such a problem occurs, the control law is computed using the estimated visual features instead of the measured ones. In this way, it is possible to keep on executing the task despite the image loss.

Now, we focus on our estimation process. To build it, we have made several choices. First of all, contrary to [14], we minimize a quadratic criterion expressing the distance between the bundles and the objects 3D structure. In this way, it is possible to use all the images provided by all the cameras before the occlusion to refine the estimation, without significantly increasing the computational cost. The visual features can then be reconstructed sufficiently rapidly with respect to the control law sampling period, which allows to reuse them to feed our IBVS control law. Furthermore, by considering the totality of the available images, our method remains accurate, even at the end of the task when the motion of the objects is reduced[1]. The second fundamental aspect

[1]The techniques proposed in [14] might suffer from this drawback if the constant number of images which is used for the reconstruction is not sufficient.

Algorithm 1: Extended visual servoing loop.

```
3DStructure model = initialvalues();
3DStructure estim = initialvalues();
dt > 0; // period of the control loop.
t_c > 0; // Time for convergence of estimation.
t = 0;
repeat
    if Features available then
        S = processLastImage();
        estim = estimation(S);
        if t > t_c then
            L = interactionMatrix(estim);
        else
            L = interactionMatrix(model);
        end
    else
        S = projection(estim);
        L = interactionMatrix(estim);
    end
    e = computeTask(S);
    J = computeTaskJacobian(L, q);
    q̇ = computeControlLaw(e, J);
    sendControlToRobot(q̇);
    t = t + dt;
until ||e|| < threshold;
```

concerns the frame in which we have chosen to reconstruct the 3D structure. This frame is defined by the frame attached to each object. In this way, the estimated data will converge towards a constant value, making the implementation easier.

Now, we focus on the reconstruction of the point and the line. We detail the method for one mobile object and we will consider n images provided by all the cameras during a time interval. As explained above, we introduce a frame F_t linked to the manipulated object (the cap or the pen). As the object has been already grasped, we have chosen F_t so that it is identical to the corresponding end-effector frame. Let us recall that the transformations between the world frame F_w and the end-effector frames on one hand and F_w and the fixed and mobile cameras on the other hand are known. We can then deduce the transformation between F_t and all the considered camera frames at any time.

A. Estimation of pointwise visual features

Our first goal is to estimate the 3D point R corresponding to point r in the image plane (see Fig. 3). We aim at estimating the constant coordinates $\bar{R}$ of R in F_t. To do so, we introduce the following definitions. We denote by $(x_{r_i},\ y_{r_i})$ the projections of this point in the i^{th} image. We denote by C_i a position of the optical center of one camera and by $\bar{C}_i$ its coordinates in F_t (see Fig. 4). These coordinates can be easily deduced from the above mentioned transformations. We also introduce Δ_i as the line of sight connecting C_i to the measured point r_i. Its unitary direction vector $\overrightarrow{V}_i$ can be expressed in the camera frame as follows:

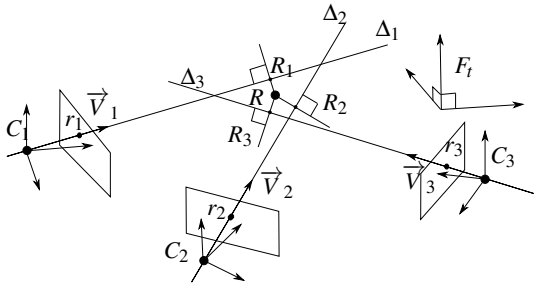

Fig. 4: Geometric error in point case.

$\frac{1}{\sqrt{x_{r_i}^2+y_{r_i}^2+f^2}}\begin{bmatrix} x_{r_i} & y_{r_i} & f \end{bmatrix}^T$. From this, we can compute its components $\bar{V}_i$ in frame F_t. Finally, we define R_i as the point given by the orthogonal projection of point R on Δ_i.

Now, let us state our problem. From Fig. 4, it follows that, for the 3D point R to be properly estimated, it is necessary to minimize the distances which separate each point R_i from point R. The geometric error D_n can then be defined for the n available images by:

$$D_n = \sum_{i=1}^{n} d_i \text{ with } d_i = (\bar{R}-\bar{R}_i)^T(\bar{R}-\bar{R}_i) \tag{4}$$

where $\bar{R}_i$ represents the coordinates of point R_i in F_t. They express as follows:

$$\bar{R}_i = \bar{V}_i\bar{V}_i^T \cdot (\bar{R}-\bar{C}_i)+\bar{C}_i \tag{5}$$

Using (5) into (4) leads to:

$$d_i = \bar{R}^T\Pi_i\bar{R} - 2\bar{R}^T\Pi_i\bar{C}_i + \bar{C}_i^T\Pi_i\bar{C}_i \tag{6}$$

where Π_i is the symmetrical projector $I_3 - \bar{V}_i\bar{V}_i^T$. From this and relation (4), we deduce:

$$D_n = \bar{R}^T\Phi_n\bar{R} - 2\bar{R}^T\beta_n + \sigma_n \tag{7}$$

$$with: \Phi_n = \sum_{i=1}^{n}\Pi_i \quad \beta_n = \sum_{i=1}^{n}[\Pi_i\bar{C}_i] \quad \sigma_n = \sum_{i=1}^{n}[\bar{C}_i^T\Pi_i\bar{C}_i]$$

D_n is then a quadratic function of $\bar{R}$. Minimizing it leads to the following estimator of $\bar{R}$: $\widehat{R}_n = [\Phi_n]^+\beta_n$.

Let us remark that Φ_n is a 3×3 hermitian matrix. Two cases may occur. Either all points C_i are aligned with R, then Φ_n is a rank 2 positive semi-definite matrix, and there exits an infinite number of solutions. The pseudo inverse provides the nearest point to the origin of F_t. Or $n>1$, Φ_n is an invertible positive definite matrix: $\Phi_n^+ = \Phi_n^{-1}$, and the obtained solution is unique.

Our method can also be implemented iteratively, making easier the handling of new images:

$$\begin{aligned} &\Phi_{n+1} = \Phi_n + \Pi_{n+1} \;; \quad \beta_{n+1} = \beta_n + \Pi_{n+1}\bar{C}_{n+1} \\ &\widehat{R}_{n+1} = [\Phi_{n+1}]^+\beta_{n+1} \end{aligned} \tag{8}$$

The necessary informations to properly estimate $\bar{R}$ are stored in low-dimensional elements (a 3×3 matrix and a 3×1 vector). The implementation complexity is significantly reduced, making easier the use of this algorithm in a control context.

Finally, it remains to compute the coordinates of point r in the image. To do so, knowing $\widehat{R}_n$, it suffices to use the perspective projection equations (see section II-A.2).

B. Estimation of the line visual features

We now focus on the estimation of the 3D cylinder axis denoted by Δ (see Fig. 5) from which we will deduce the visual cues ρ and θ. As previously, we estimate the necessary parameters in the frame F_t linked to the manipulated object. To do so we first introduce the following definitions. We

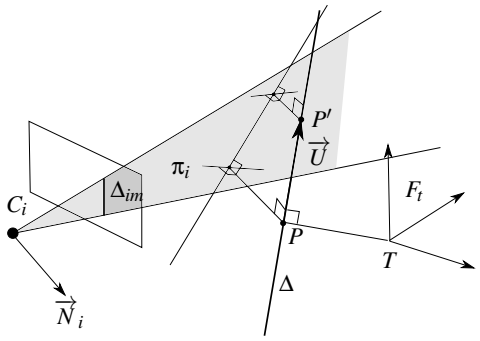

Fig. 5: Geometric error in straight line case.

denote by C_i a position of the optical center of one camera. We also define by Δ_{im} the perceived line in the image plane. This line is characterized by the visual features $(\rho_i,\ \theta_i)$. Finally, we introduce $\overrightarrow{N}_i$ the normal unitary vector to the plane π_i defined by Δ_{im} and the optical center C_i [15]. Its components in the camera frame can be deduced from the visual features (ρ_i, θ_i) as follows [15]:

$$\frac{1}{\sqrt{f^2+\rho_i^2}}\begin{bmatrix} f\cos\theta_i & f\sin\theta_i & -\rho_i \end{bmatrix}^T \tag{9}$$

As previously, our idea is to express a geometrical error. To do so, we first choose a suitable representation for the straight line materializing the cylinder axis: a 3D point $P\in\Delta$ which is the nearest to the origin of F_t and a unitary direction vector $\overrightarrow{U}$ (see Fig. 5). It then follows that:

$$\bar{P}^T\cdot\bar{U} = 0 \text{ and } \bar{U}^T\cdot\bar{U} = 1$$

where $\bar{P}$ and $\bar{U}$ represent the components of P and $\overrightarrow{U}$ in F_t. Thus, our goal is to estimate these constant parameters using the information $\rho_i,\ \theta_i$ extracted from n images. To this aim, we define the geometric error so that it vanishes when the projection error is zero. Fig. 5 shows that this condition is fulfilled when Δ and Δ_{im} belong to π_i, that is if two distinct points of Δ belong to this plane. To ease the problem modeling, the points have been chosen so that their coordinates in F_t can be easily expressed from the parameters describing the 3D line. P and P' both satisfy this property (see Fig. 5). Point P' is defined such that its coordinates in F_t are given by $\bar{P}' = \bar{P}+\bar{U}$. The geometric error at instant t_i can then be defined by:

$$g_i = (\bar{P}'-\bar{C}_i)^T\bar{N}_i\bar{N}_i^T(\bar{P}'-\bar{C}_i) + (\bar{P}-\bar{C}_i)^T\bar{N}_i\bar{N}_i^T(\bar{P}-\bar{C}_i) \tag{10}$$

where $\bar{N}_i$ is the components of $\overrightarrow{N}_i$ in F_t. Defining $\Theta_i = \bar{N}_i\bar{N}_i^T$, $\xi_i = \Theta_i\bar{C}_i$ and $\varepsilon_i = \bar{C}_i^T\xi_i$, g_i can be rewritten as:

$$g_i = 2\bar{P}^T\Theta_i\bar{P} + 2\bar{P}^T\Theta_i\bar{U} - 4\bar{P}^T\xi_i + \bar{U}^T\Theta_i\bar{U} - 2\bar{U}^T\xi_i + 2\varepsilon_i$$

Now, we consider the sum of these errors over n images:

$$G_n = \sum_{i=1}^{n} g_i = 2\bar{P}^T[\sum_{i=1}^{n}\Theta_i]\bar{P} + 2\bar{P}^T[\sum_{i=1}^{n}\Theta_i]\bar{U} - 4\bar{P}^T[\sum_{i=1}^{n}\xi_i] + \bar{U}^T[\sum_{i=1}^{n}\Theta_i]\bar{U} - 2\bar{U}^T[\sum_{i=1}^{n}\xi_i] + 2\sum_{i=1}^{n}\varepsilon_i \tag{11}$$

Introducing: $\Psi_n = \sum_{i=1}^{n}\Theta_i,\ \delta_n = \sum_{i=1}^{n}[\xi_i],\ \mu_n = \sum_{i=1}^{n}[\varepsilon_i]$, it is possible to obtain a simpler expression of the total error:

$$G_n = 2\bar{P}^T\Psi_n\bar{P} + \bar{U}^T\Psi_n\bar{U} + 2\bar{P}^T\Psi_n\bar{U} - 4\bar{P}^T\delta_n \quad -2\bar{U}^T\delta_n + 2\mu_n \tag{12}$$

Now, it remains to minimize this error with respect to $\bar{P}$ and $\bar{U}$. Derivating G_n with respect to these parameters leads to:

$$\Psi_n\bar{P} - \delta_n = 0 \tag{13}$$

$$\Psi_n\bar{U} = 0 \tag{14}$$

From equation (13), the estimator $\hat{P}$ of $\bar{P}$ is given by: $\hat{P}_n = [\Psi_n]^+\delta_n$. As Ψ_n is a 3×3 matrix obtained by adding rank 1 positive semi-definite hermitian matrices, its rank is necessarily greater than or equal to 1. Two cases may occur: either all π_i are identical (degenerate case), Ψ_n is rank 1 and it is impossible to determine the line [15] ; or, this matrix is rank 2 and the 3D line can be determined. Assuming Ψ_n is rank 2, we can deduce from equation (14) that $\bar{U} \in \ker(\Psi_n)$. The estimator $\hat{U}$ of $\bar{U}$ will then be defined by the eigenvector associated to the smallest eigenvalue of Ψ_n.

Our estimator can also be expressed using the following iterative formula:

$$\Psi_{n+1} = \Psi_n + \Theta_{n+1} \qquad \delta_{n+1} = \delta_n + \Theta_{n+1}\bar{C}_{n+1}$$
$$\widehat{P}_{n+1} = [\Psi_{n+1}]^+\delta_{n+1}$$
$$\widehat{U}_{n+1} = \text{associed vector to smallest eigenvalue of } \Psi_{n+1}$$

Now it remains to compute the visual features corresponding to the 3D line. To this aim, we first compute the equation of the plane defined by C_i, P and P'. The normal vector to this plane is given by : $\overrightarrow{\hat{N}_i} = \overrightarrow{C_iP} \times \overrightarrow{U}$. From this relation, we can deduce the following equation for the considered plane in the camera frame: $a_ix + b_iy + c_iz = 0$ where (a_i, b_i, c_i) correspond to the components of $\overrightarrow{\hat{N}_i}$ in the camera frame. The 2D projected line is then given by the intersection of this plane with the image plane defined by $z = f$. Its equation then expresses as: $a_ix + b_iy + c_if = 0$. From this, $(\widehat{\rho}_i, \widehat{\theta}_i)$ can be deduced as follows [17]:

$$\widehat{\rho}_i = -\frac{c_if}{\sqrt{a_i^2 + b_i^2}} \qquad \widehat{\theta}_i = \arctan(b_i/a_i)$$

IV. SIMULATION RESULTS

These results have been obtained using ROS and Gazebo. To get closer to experimental conditions, we have added a Gaussian noise on the visual features (mean = 0, standard deviation = 2 pixels)[2] and on the joint positions and velocities (mean = 0, covariance 7.6×10^{-9}). We consider the task consisting in recapping a pen. At the beginning, it is assumed that the cap and the pen have already been grasped by the robot, and that the visual features are fully available. After 3 seconds, we simulate an image processing failure on both cameras to evaluate our method in the worst case. At $t = 6$ seconds, the visual features are available anew. The evolution of the corresponding task functions and control inputs are shown on Fig. 9 and 10, while the estimated parameters are illustrated on pictures 6, 7, and 8. The time instants between which the image cues are totally lost are materialized by the dotted red and blue vertical arrows on each figure. These plots will allow to evaluate the efficiency of our approach.

We first focus on Fig. 6, 7 and 8. They respectively show the evolution of $\bar{P}$, $\bar{U}$ and $\bar{R}$ for the pen. As their real values have been initially set to (in meters): $\bar{P} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$, $\bar{U} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$, $\bar{R} = \begin{bmatrix}0 & 0 & 0.07\end{bmatrix}^T$.

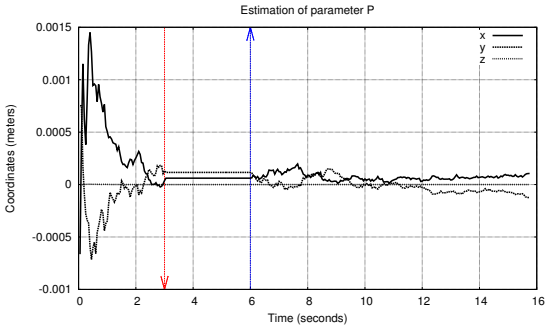

Fig. 6: Evolution of coordinates of parameter $\hat{P}$ in F_t.

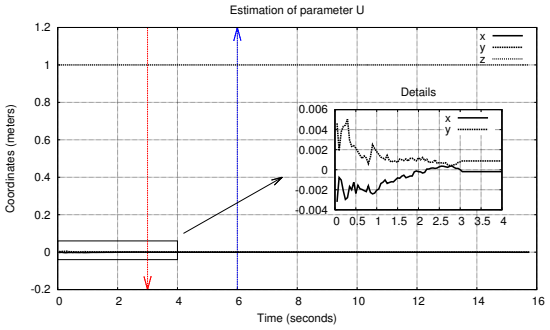

Fig. 7: Evolution of coordinates of parameter $\hat{U}$ in F_t.

The obtained curves show that the estimators of these parameters quickly converge towards their real values despite the different added noises. The residual error is very small (less than 1 millimeter), because the 3D parameters are estimated using the information provided by two cameras. When only one view point is available, the performances are lower but remain satisfactory.

When the image processing failure occurs, the algorithm has already converged and an estimation of the 3D point and of the 3D line have been obtained. From them, the visual features are computed and then used to feed the control law.

[2]The sensors provide 640 × 480 pixels images.

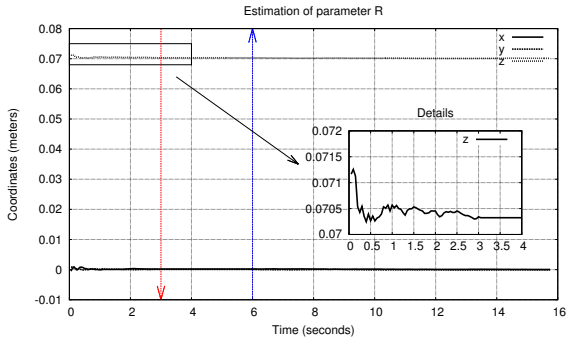

Fig. 8: Evolution of coordinates of parameter $\hat{R}$ in F_t.

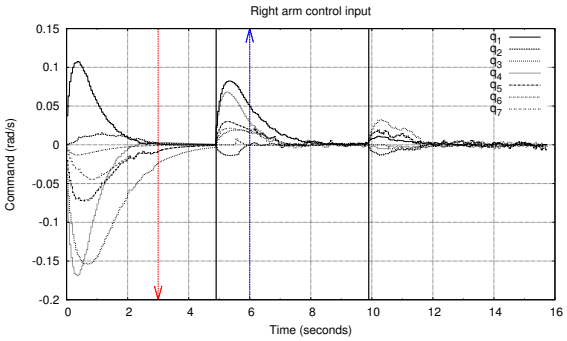

Fig. 10: Evolution of the 7 right arm control inputs.

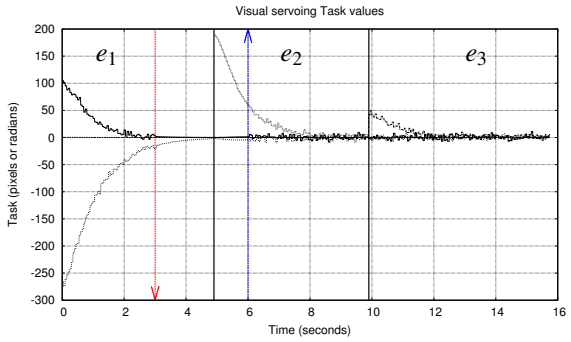

Fig. 9: Evolution of the task functions e_1, e_2 and e_3 (the switching instants are shown by the vertical black lines.).

Fig. 9 and 10 show that the evolution of the task functions and of the control law are not perturbed during the image processing failure, which demonstrates the method accuracy. Let us also remark that e_1, e_2 and e_3 vanish, which means that the pen is correctly recapped. Thus, the vision-based task is successfully performed despite the total loss of the visual features in both vision systems.

V. CONCLUSION

In this paper, we have dealt with the image features total loss during a coordinated vision-based task. The proposed solution allows to cope with the specificities of this task, namely: the manipulated objects are moving and they are perceived by several mobile cameras. Its main originality lies in the idea of reconstructing the 3D structure parameters directly in the frame linked to each object. This idea allows to reconstruct only constant data. In addition, the method benefits from a low computational cost, which allows to use the different data in the control loop.

We are now working experimentally to couple the proposed estimation approach to the existing vision-based control strategy. We also plan to evaluate the robustness of the estimation technique in the presence of calibration errors.

REFERENCES

[1] J. Maytin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *IEEE International Conference on Robotics and Automation*, pages 2308–2315, 2010.

[2] Jun Takamatsu, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. Recognizing assembly tasks through human demonstration. *The International Journal of Robotics Research*, 26(7):641–659, 2007.

[3] Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulators. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 701–718. Springer Berlin Heidelberg, 2008.

[4] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D.V. Dimarogonas, and D. Kragic. Dual arm manipulation - a survey. *Robotics and Autonomous Systems*, 60(10):1340 – 1353, 2012.

[5] Renliw Fleurmond and Viviane Cadenat. Multi-cameras visual servoing to perform a coordinated task using a dual arm robot. In *International Conference on Informatics in Control, Automation and Robotics*, September 2014.

[6] F. Chaumette and S. Hutchinson. Visual servo control part 1: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.

[7] B.V. Adorno, P. Fraisse, and S. Druon. Dual position control strategies using the cooperative dual task-space framework. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3955–3960, 2010.

[8] 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.

[9] O. Kermorgant and F. Chaumette. Dealing with constraints in sensor-based robot control. *Robotics, IEEE Transactions on*, 30(1):244–257, Feb 2014.

[10] 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.

[11] 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*, 33, 2011.

[12] A. De Luca, G. Oriolo, and P.R. Giordano. On-line estimation of feature depth for image-based visual servoing schemes. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2823–2828, April 2007.

[13] R. Spica, P. Robuffo Giordano, and F. Chaumette. Active structure from motion: Application to point, sphere and cylinder. *IEEE Trans. on Robotics*, 30(6):1499–1513, December 2014.

[14] A. Durand Petiteville, S. Durola, V. Cadenat, and M. Courdesses. Management of visual signal loss during image based visual servoing. In *Control Conference (ECC), 2013 European*, pages 2305–2310, July 2013.

[15] Yi Ma, Jana Kosecka, Stefano Soatto, and Shankar Sastry. *An Invitation to 3D Vision:From Images to Models*. Springer, 2001.

[16] C. Samson, M. Le Borgne, and B. Espiau. *Robot control: the task function approach*. Oxford engineering science series. Clarendon Press, 1991.

[17] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.