

Visual Servoing when Visual Information is Missing: Experimental Comparison of Visual Feature Prediction Schemes

Nicolas Cazy, Pierre-Brice Wieber, Paolo Robuffo Giordano, and François Chaumette

Abstract—One way to deal with occlusions or loss of tracking of the visual features used for visual servoing tasks is to predict the feature behavior in the image plane when the measurements are missing. Different prediction and correction methods have already been proposed in the literature. The purpose of this paper is to compare and experimentally validate some of these methods for eye-in-hand and eye-to-hand configurations. In particular, we show that a correction based both on the image and the camera/target pose provides the best results.

I. INTRODUCTION

In the visual servoing literature (see [1] for a tutorial) there exist two main approaches for dealing with occlusions and/or loss of feature tracking due to the limited camera field of view. In the first approach, one tries to develop control schemes for avoiding that an occlusion or loss of tracking never occurs while the second approach proposes strategies for coping with the possibility that such events may occur.

On the control side, many techniques have been considered: occlusions avoidance is expressed as a secondary task combined with a classical visual task using the gradient projection method in [2]. This method has also been used in [3], combined with sequencing tasks to also consider obstacle avoidance. Avoiding that features leave the camera field of view is explicitly taken into account in the visual features design in [4], while switching control methods have been proposed in [5], [6]. Visual trajectory planning taking into account several constraints such as occlusions and feature loss has led to a series of works [7]–[10]. More recently, model predictive control has been applied to visual servoing since this approach allows an easy handling of hard constraints [11]–[16]. Finally, another method consisting in integrating the constraints in the feature set used as input of the control scheme has been proposed in [17].

On the other side, a visual servoing task can still be successful in case of occlusions or feature loss provided a sufficient number of features remains visible during motion (so that the rank of the corresponding Jacobian matrix can still match the number of controlled degrees of freedom). The most simple solution is thus to just remove from the feature set those no longer available. However, a change in

the feature set implies a change in the number of rows of the Jacobian matrix, and thus a discontinuity in the output of the control scheme. To solve this problem, a continuous weight can be associated to each feature as proposed in [18]: this weight varies between 1, when the feature is located near the center of the image, to 0 when the feature reaches the image limit. This then ensures continuity of the output of the control scheme. This strategy is of course no more efficient when the number of visible features becomes too small and cannot ensure full rankness of the Jacobian matrix. A general solution, explored in [19], is thus to predict the location of the features when they are no more visible, and to then use this prediction in the control scheme as well as for guiding the feature re-tracking when it becomes visible again. This is also the strategy considered in our previous work [20] and in this paper.

The purpose of this paper is to validate experimentally the conclusions of this previous work, which were validated only in simulation using a priori given camera motions while visual servoing is considered here. Section II begins with reviewing the prediction and correction methods previously discussed in [20]. Section III recalls the basic visual servoing equations for both eye-in-hand and eye-to-hand configurations. Section IV presents a number of experimental results aiming at comparing the different methods available. Finally, Section V concludes the paper.

II. VISUAL FEATURES MODEL ERROR CORRECTION

In this paper, we consider a monocular perspective camera and a target object made of N 3D points P_i with \mathcal{F}_C and \mathcal{F}_O two Cartesian frames attached respectively to the camera and target. We also denote with $\mathbf{x} = (x, y, z, \theta\mathbf{u}) \in \mathbb{R}^6$ the pose of \mathcal{F}_O with respect to \mathcal{F}_C , with $(\mathbf{u}, \theta) \in \mathbb{S}^2 \times \mathbb{R}$ the axis/angle parameterization of the orientation¹, and with $\mathbf{v}(t) = (\mathbf{v}(t), \boldsymbol{\omega}(t)) \in \mathbb{R}^6$ the linear and angular velocity of the camera, expressed in the camera frame.

Let ${}^O P_i = ({}^O X_i, {}^O Y_i, {}^O Z_i)$ and ${}^C P_i = ({}^C X_i, {}^C Y_i, {}^C Z_i)$ be the coordinates of the 3D points P_i in the object and camera frames. Assuming a calibrated camera, the normalized perspective 2-D projection of each point P_i on the image plane is

$$\mathbf{s}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} {}^C X_i / {}^C Z_i \\ {}^C Y_i / {}^C Z_i \end{bmatrix} = f_{s_i}(\mathbf{x}), \quad (1)$$

¹Among all the possible *minimal* representations for the orientation between \mathcal{F}_O and \mathcal{F}_C , we chose the vector $\theta\mathbf{u} \in \mathbb{R}^3$ because its singularities lie at $\theta = 2k\pi$, $k \in \mathbb{Z}^*$, *i.e.*, out of the possible workspace in any normal application (see [21]).

N. Cazy is with Inria at Inria Rennes-Bretagne Atlantique and Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France nicolas.cazy@inria.fr

P.-B. Wieber is with Inria at Inria Grenoble Rhône-Alpes, France pierre-brice.wieber@inria.fr

P. Robuffo Giordano is with the CNRS at Irisa and Inria Rennes-Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr.

F. Chaumette is with Inria at Inria Rennes-Bretagne Atlantique and Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France francois.chaumette@inria.fr

where the dependence on the camera/target relative pose \mathbf{x} has been emphasized. The set of all visual features is therefore

$$\mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix} = f_s(\mathbf{x}) \in \mathbb{R}^{2N}. \quad (2)$$

We finally let $\mathbf{x}_m(t)$ represent a *model/approximation* of the true target pose, with $\mathbf{s}_m(t) = f_s(\mathbf{x}_m(t))$ the corresponding *model* of $\mathbf{s}(t)$, derived from $\mathbf{x}_m(t)$ via (2).

In the following, we assume that the vector $\mathbf{s}(t)$ can be measured until some $t = t_{occ}$ after which some visual features or the full set $\mathbf{s}(t)$ become unavailable for a time period $T > 0$ because of some external ‘event’ such as, e.g., an occlusion, loss of tracking because of the limited camera field of view (fov), or a generic error of the image processing step. The following sections discuss then two possible strategies for (i) propagating the internal models $\mathbf{s}_m(t)$ and $\mathbf{x}_m(t)$ during $t \in [t_{occ}, t_{occ} + T]$, exploiting the known linear and angular velocity $\mathbf{v}(t) = (\mathbf{v}(t), \boldsymbol{\omega}(t))$ of the camera, and for (ii) correcting $\mathbf{s}_m(t)$ and $\mathbf{x}_m(t)$ as long as the measurements $\mathbf{s}(t)$ are available (thus for $t < t_{occ}$ and $t \geq t_{occ} + T$).

A. Model prediction

We assume the time interval $[t_{occ}, t_{occ} + T]$ is discretized in H uniform steps (corresponding to the number of camera frames without visual feature measurement) of duration $\tau = T/H$. We also set $\mathbf{v}(t) = \mathbf{v}(t_k) = \text{const}$ for $t \in [t_k, t_{k+1}]$, with $k = 1 \dots H$.

1) *Prediction in the image*: the prediction in the image proposed in [15], [20] is a simple forward integration of $\mathbf{s}_m(t)$ via the interaction matrix \mathbf{L}_s given by [1]

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{u}{Z} & uv & -(1+u^2) & v \\ 0 & -\frac{1}{Z} & \frac{v}{Z} & 1+v^2 & -uv & -u \end{bmatrix}. \quad (3)$$

More precisely, we have

$$\mathbf{s}_m(t_k) = \mathbf{s}_m(t_{k-1}) + \tau \mathbf{L}_s(\mathbf{s}_m(t_{k-1}), \mathbf{Z}_m(t_{k-1})) \mathbf{v}(t_{k-1}), \quad (4)$$

with $\mathbf{Z}_m = [Z_{m_1}, \dots, Z_{m_N}]$ a model of the depths associated to the N feature points. The vector $\mathbf{Z}_m(t)$ can also be updated from an initial $\mathbf{Z}_m(t_{occ})$ with

$$\mathbf{Z}_m(t_k) = \mathbf{Z}_m(t_{k-1}) + \tau \mathbf{L}_Z(\mathbf{s}_m(t_{k-1}), \mathbf{Z}_m(t_{k-1})) \mathbf{v}(t_{k-1}), \quad (5)$$

with

$$\mathbf{L}_Z(\mathbf{s}(t_{k-1}), \mathbf{Z}(t_{k-1})) = \begin{bmatrix} \mathbf{L}_{Z_1} \\ \vdots \\ \mathbf{L}_{Z_N} \end{bmatrix} \quad (6)$$

and

$$\mathbf{L}_{Z_i}(\mathbf{s}_i(t_{k-1}), Z_i(t_{k-1})) = \begin{bmatrix} 0 & 0 & -1 & -v_i Z_i & u_i Z_i & 0 \end{bmatrix}. \quad (7)$$

By exploiting (4) and (5), the behavior of the feature positions $\mathbf{s}_m(t)$ and depths $\mathbf{Z}_m(t)$ can be predicted during

the interval $[t_{occ}, t_{occ} + T]$ *without* requiring an explicit knowledge of the camera/target relative pose $\mathbf{x}(t)$ nor the 3D model of the target. Obviously the accuracy of this prediction scheme depends on the accuracy of the initial values $\mathbf{s}_m(t_{occ})$ and $\mathbf{Z}_m(t_{occ})$.

Note that if only camera rotational motions are involved in the control scheme, such as for gaze control where a pan-tilt camera is used to track a moving target, the depth does not come anymore into the interaction matrix (\mathbf{Z} only appears in the first three columns corresponding to the three translational motions), which makes useless the prediction $\mathbf{Z}_m(t)$.

2) *Prediction in the pose*: an alternative is to propagate instead the model $\mathbf{x}_m(t)$ of the target pose w.r.t. the camera, with the kinematic relationship

$$\mathbf{x}_m(t_k) = \mathbf{x}_m(t_{k-1}) + \tau \mathbf{L}_x \mathbf{v}(t_{k-1}), \quad (8)$$

where

$$\mathbf{L}_x = \begin{bmatrix} {}^O \mathbf{R}_C(\mathbf{u}, \theta) & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_\omega(\mathbf{u}, \theta) \end{bmatrix}$$

with

$${}^O \mathbf{R}_C(\mathbf{u}, \theta) = \mathbf{I}_3 + [\mathbf{u}]_\times \sin(\theta) + [\mathbf{u}]_\times^2 (1 - \cos(\theta))$$

is the rotation matrix from \mathcal{F}_C to \mathcal{F}_O , and

$$\mathbf{L}_\omega(\mathbf{u}, \theta) = \mathbf{I}_3 - \frac{\theta}{2} [\mathbf{u}]_\times + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\theta/2)} \right) [\mathbf{u}]_\times^2.$$

and $\text{sinc}(\theta) = \sin(\theta)/\theta$, see [21]. Since $\det(\mathbf{L}_\omega(\mathbf{u}, \theta)) = 1/\text{sinc}^2(\theta/2)$, matrix \mathbf{L}_x is invertible for $\theta \neq 2k\pi$, $k \in \mathbb{Z}^*$, i.e., within the workspace of any normal application.

Combined with (2), we obtain a model $\mathbf{s}_m(t) = f_s(\mathbf{x}_m(t))$ of the features in the image during the interval $[t_{occ}, t_{occ} + T]$. The difference with the previous prediction scheme is that the evolution of the camera/target pose $\mathbf{x}_m(t)$ is continuously tracked here while only $\mathbf{s}_m(t)$ and $\mathbf{Z}_m(t)$ are tracked with (4) and (5). As before, the accuracy of this prediction scheme depends on the accuracy of the initial value $\mathbf{x}_m(t_{occ})$.

B. Model correction

When measures $\mathbf{s}(t)$ are available, they can be exploited to correct the features and the pose models $\mathbf{s}_m(t)$ and $\mathbf{x}_m(t)$ in the following ways.

1) *Correction in the image*: this correction scheme, proposed in [15], [20], aims at correcting the model $\mathbf{s}_m(t_k)$ by exploiting the measures $\mathbf{s}(t_k)$ as follows:

$$\begin{cases} \mathbf{s}_m(t_k) \leftarrow \mathbf{s}(t_k), \\ {}^C \mathbf{Z}_m(t_k) \leftarrow {}^C \mathbf{Z}_m(t_k), \\ {}^C X_{m_i}(t_k) \leftarrow u_{m_i}(t_k) {}^C Z_{m_i}(t_k), \\ {}^C Y_{m_i}(t_k) \leftarrow v_{m_i}(t_k) {}^C Z_{m_i}(t_k). \end{cases} \quad (9)$$

This clearly ensures that $\mathbf{s}_m(t_{occ}) = \mathbf{s}(t_{occ})$ and $\mathbf{s}_m(t_{occ} + T) = \mathbf{s}(t_{occ} + T)$ when the occlusion begins and ends, but it fails in correcting any error in the depth model $\mathbf{Z}_m(t_{occ})$ and $\mathbf{Z}_m(t_{occ} + T)$ since it only accounts for the error in the image, without trying to correct any potential discrepancy

between the real and the model poses $\mathbf{x}(t)$ and $\mathbf{x}_m(t)$. As a result, the model $\mathbf{s}_m(t)$ will generally diverge from the real values $\mathbf{s}(t)$ for $t > t_{occ}$.

2) *Correction in the image and in the pose*: this improved correction scheme additionally exploits the mismatch between the model $\mathbf{s}_m(t_k) = f_s(\mathbf{x}_m(t_k))$ and the measures $\mathbf{s}(t_k)$ to compute a pose correction

$$\Delta \mathbf{x}(t_k) = \mathbf{L}_x \mathbf{L}_s^\dagger (\mathbf{s}(t_k) - \mathbf{s}_m(t_k)), \quad (10)$$

with \mathbf{L}_s^\dagger the Moore-Penrose inverse of $\mathbf{L}_s(\mathbf{s}_m(t), \mathbf{x}_m(t))$ (see [20] for more details):

$$\left\{ \begin{array}{l} \mathbf{x}_m(t_k) \leftarrow \mathbf{x}_m(t_k) \oplus \Delta \mathbf{x}(t_k), \\ \mathbf{s}_m(t_k) \leftarrow \mathbf{s}(t_k), \\ {}^C \mathbf{Z}_{m_i}(t_k) \leftarrow f_Z(\mathbf{x}_m(t_k)), \\ {}^C X_{m_i}(t_k) \leftarrow u_{m_i}(t_k) {}^C Z_{m_i}(t_k), \\ {}^C Y_{m_i}(t_k) \leftarrow v_{m_i}(t_k) {}^C Z_{m_i}(t_k). \end{array} \right. \quad (11)$$

Here, $f_Z(\cdot)$ provides depths \mathbf{Z} obtained directly from the camera/target pose model and point 3D coordinates.

Note that the pose correction scheme corresponds to a single iteration of a standard pose estimation by virtual visual servoing [22]. More iterations could be realized if necessary. Note also that because of the Moore-Penrose inverse property

$$\mathbf{L}_s^\dagger \mathbf{L}_s \mathbf{L}_s^\dagger = \mathbf{L}_s^\dagger, \quad (12)$$

one has

$$\mathbf{L}_s^\dagger \mathbf{s}_m(t) \approx \mathbf{L}_s^\dagger \mathbf{s}(t) \quad (13)$$

(see again [20] for more details). Therefore, a typical visual servoing control law (see Section III just below)

$$\mathbf{v} = -\lambda \mathbf{L}_s^\dagger (\mathbf{s}(t) - \mathbf{s}^*) \approx -\lambda \mathbf{L}_s^\dagger (\mathbf{s}_m(t) - \mathbf{s}^*)$$

will yield *similar* camera velocities when using either $\mathbf{s}(t)$ or $\mathbf{s}_m(t)$. Classical image-based visual servoing will thus not be perturbed by this pose correction method. This is of particular interest for $t > t_{occ}$ when $\mathbf{s}(t)$ is no more available.

III. IMAGE-BASED VISUAL SERVOING

An IBVS control strategy consists in finding the camera linear and angular velocity \mathbf{v} that will minimize the norm of an error function defined on the image plane

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^*,$$

where \mathbf{s}^* represents the desired location of the visual features [1]. By exploiting the relationship between the time variations of \mathbf{s} and \mathbf{v} , one has

$$\dot{\mathbf{s}} = \dot{\mathbf{e}} = \mathbf{L}_s \mathbf{v}.$$

A. Eye-In-Hand configuration

If we consider a camera mounted on a robot arm, one can implement the following control law

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_s^\dagger (\mathbf{s} - \mathbf{s}^*), \quad (14)$$

where $\widehat{\mathbf{L}}_s$ is an estimation of \mathbf{L}_s , for driving the error \mathbf{e} to zero.

B. Eye-To-Hand configuration

In order to illustrate how the proposed correction can be applied to different scenarios, we also consider an Eye-To-Hand system in which a static camera is observing a target fixed to the end effector of the robot arm, see [23]. The resulting control law is slightly different from (14) and results in

$$\mathbf{v}_e = \lambda {}^e \mathbf{V}_c \widehat{\mathbf{L}}_s^\dagger (\mathbf{s} - \mathbf{s}^*), \quad (15)$$

where \mathbf{v}_e is the end effector velocity expressed in the end effector frame and ${}^e \mathbf{V}_c$ is the velocity twist matrix given by

$${}^e \mathbf{V}_c = \begin{pmatrix} {}^e \mathbf{R}_c & [{}^e \mathbf{t}_c]_\times {}^e \mathbf{R}_c \\ \mathbf{0} & {}^e \mathbf{R}_c \end{pmatrix}.$$

Note that the measured robot velocity has to be expressed in the camera frame for using the prediction schemes (4), (5) and (8). This is simply obtained by:

$$\mathbf{v} = {}^c \mathbf{V}_e \mathbf{v}_e. \quad (16)$$

The main difficulty of the eye-to-hand configuration appears in the non-exact knowledge of the end effector pose in the camera frame, which is involved in ${}^c \mathbf{V}_e$ in (16). Indeed, even if the camera/target pose model \mathbf{x}_m is perfectly corrected, the target/effector pose has to be perfectly calibrated to avoid dissimilarities between real and predicted movements.

In the following experiments and for both configurations previously described, we have chosen $\widehat{\mathbf{L}}_s = \mathbf{L}_s(\mathbf{s}^*, \mathbf{Z}^*)$, that is, the value of the interaction matrix computed at the desired pose, to demonstrate that, even if the depth $\mathbf{Z}(t)$ is not involved in the control scheme, its use in the prediction step is crucial.

IV. EXPERIMENTAL RESULTS

This section presents results obtained when applying the corrections discussed in Section II to a standard Image-Based Visual Servoing (IBVS) task in presence of occlusions or loss of tracking of the controlled visual features.

The first two following experiments were realized with an eye-in-hand monocular camera mounted on a 6 dof robot arm. The target consists of four coplanar points forming a square of about 10 cm in length with coordinates in \mathcal{F}_0

$$\left\{ \begin{array}{l} {}^O \mathbf{P}_0 = (-0.05, -0.05, 0) \\ {}^O \mathbf{P}_1 = (0.05, -0.05, 0) \\ {}^O \mathbf{P}_2 = (0.05, 0.05, 0) \\ {}^O \mathbf{P}_3 = (-0.05, 0.05, 0) \end{array} \right.$$

The software ViSP [24] was used for image processing and visual tracking of 2D point coordinates.

The pose of \mathcal{F}_0 w.r.t. \mathcal{F}_C at the beginning of the task was

$$\mathbf{x}(t_0) = (-0.072, -0.035, 1.21, -0.35, 0.34, 0.050)^T$$

while the initial model of the pose was taken intentionally incorrect to highlight the value of the correction (11), such as

$$\mathbf{x}_m(t_0) = (0, 0, 2, 0, 0, 0.79)^T.$$

This was then used to compute the initial values of the model $\mathbf{s}_m(t_0) = f_s(\mathbf{x}_m(t_0))$ of visual features via (2) as well as ${}^C\mathbf{Z}_m(t_0) = f_Z(\mathbf{x}_m(t_0))$. During the servoing, we also simulated a camera field of view (fov) smaller than the real one for exacerbating the loss of tracking during motion.

A. Reference behavior

Figure 1a shows the initial camera image before the control law (14) is applied. Green crosses represent the desired locations \mathbf{s}^* for the visual features, and red crosses their current (measured) location $\mathbf{s}(t)$. This initial experiment does not include any occlusion or loss of tracking and is only meant to provide a *reference* baseline for comparison (see the first part of the video submitted as supplementary material). The solid red curves in Figure 1b represent the paths taken by each point in the image during the camera motion, while Figure 1c shows the evolution of the error $e_i(t) = \|\mathbf{s}_i(t) - \mathbf{s}_i^*(t)\|$ between the current and desired locations, eventually converging to zero.

B. Limited field of view

Let us now consider a case where the visual features leave the field of view of the camera, which is artificially limited to a small section of the real view obtained from the camera, represented in the following figures by a blue box. Figures 2a and 3a show the camera image at the end of the servoing loop, when using respectively the correction (9) and the correction (11). Green curves represent the paths taken by the features when measurements are available, while blue curves represent their predicted paths computed thanks to (8) when measurements are not available.

As expected, the non-corrected pose $\mathbf{x}_m(t)$ in (9) results in a significant drift of the prediction in the Fig. 2a, while the correction (11) leads to an almost perfect prediction in the Fig. 3a. As a result, the camera converges to an incorrect pose in the first case (see the second part of the video submitted as supplementary material), while converging to the desired pose in the second case (see the third part of the video). Note how the image processing is able to recover the tracking of the visual features when they re-enter the camera view, by exploiting their predicted location.

Figures 2b and 3b show the corresponding errors $e_i(t)$, and Figures 2c and 3c the error in estimating the depths $Z_i(t)$. In each plot, the blue parts of the curves correspond to occlusion phases. We can verify that the depth estimations converge to the real values when applying the correction (11), but naturally not when applying the correction (9).

As additional experimentation the fourth part presented in the video submitted as supplementary material leads to the same conclusion as the one with the pose and image correction but with a longer occlusion caused by visual features leaving the camera field of view and hidden by a sheet of paper.

C. Eye-To-Hand configuration

This last experiment was realized with an eye-to-hand monocular camera observing a target mounted on the end

effector of a 6 dof robot arm which has to grasp a cube of 4 cm for each side. The pose of the cube in the camera frame is computed at the beginning of the experimentation, and is used to generate desired visual features of the target in the camera image plane. The target is composed by 4 points with coordinates in \mathcal{F}_0

$$\begin{cases} {}^O\mathbf{P}_0 &= (0, 0.023, -0.134) \\ {}^O\mathbf{P}_1 &= (0, 0.023, -0.247) \\ {}^O\mathbf{P}_2 &= (-0.057, 0, -0.132) \\ {}^O\mathbf{P}_3 &= (-0.057, -0.113, -0.13) \end{cases}$$

The pose of the target in the camera frame at the beginning of the task was

$$\mathbf{x}(t_0) = (0.307, -0.522, 0.753, -1.155, -0.776, -0.977)^T$$

while the initial model of the pose was taken as

$$\mathbf{x}_m(t_0) = (0, 0, 1, 0, 0, 0)^T.$$

The visual servoing has to realize 2 steps using the control law (15). The first step consists in moving the grasp above the cube to avoid any collision, and the second step consists in moving the arm towards the cube for grasping it as shown on Figure 4a. Once again, the field of view of the camera is artificially limited to a small section of the real view obtained from the camera, which causes loss of measurements replaced by prediction computed thanks to (4), (5).

Figures 4b and 4c show respectively the camera image at the end of the servoing loops and the corresponding errors $e_i(t)$. This demonstrates the efficiency of (11) for an Eye-To-Hand system (see the last part of the video submitted as supplementary material). Indeed, the two steps required to grasp the cube are realized even when visual features leave the camera field of view during the first step. We can note a slight difference between real and predicted visual features at the end of the servoing loop because of the non-exact knowledge of the effector/target pose.

V. CONCLUSION

In this paper, we applied the prediction/correction scheme recently proposed in [20] to a classical image-based visual servoing task in order to cope with possible occlusions or loss of feature tracking due to the limited camera field of view. To this end, we reported several experimental results (for both an eye-in-hand and eye-to-hand configurations) that showed the effectiveness of the proposed approach. Indeed, thanks to the prediction/correction scheme [20], the servoing task is shown to gain a high level of robustness against occlusions/loss of tracking also during extended periods of time. Future works will investigate how to exploit the proposed machinery in the context of Model Predictive Control.

ACKNOWLEDGMENTS

This work was partly funded by the Oseo Romeo 2 Project.

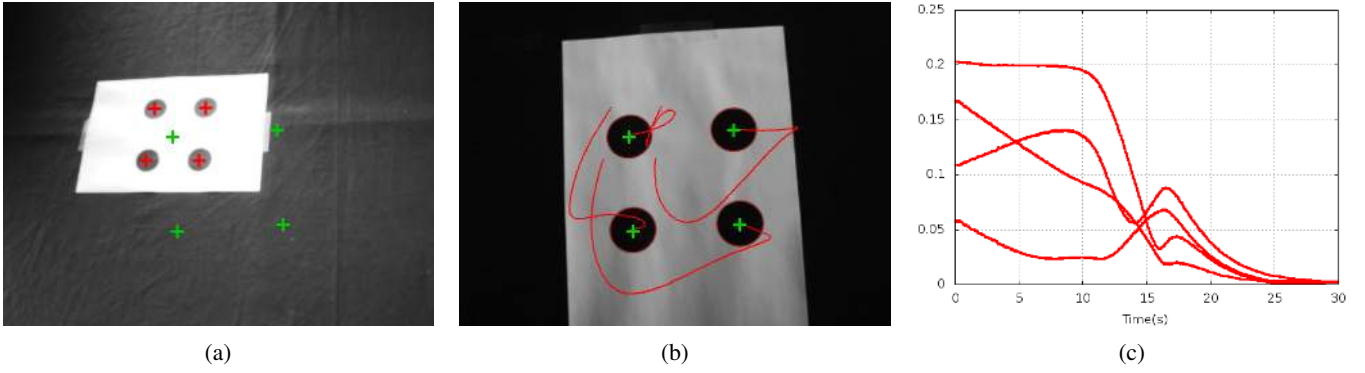


Fig. 1: Reference trajectory: (a) is the initial state where the locations of desired features s^* on the camera image plane are in green and the locations of measured features $s(t_0)$ are in red. (b) is the final state where the measured trajectory followed by each feature for reaching the green desired values are in red. (c) is the behavior of the error $e_i(t) = \|s_i(t) - s_i^*(t)\|$ for each visual feature error during the servoing.

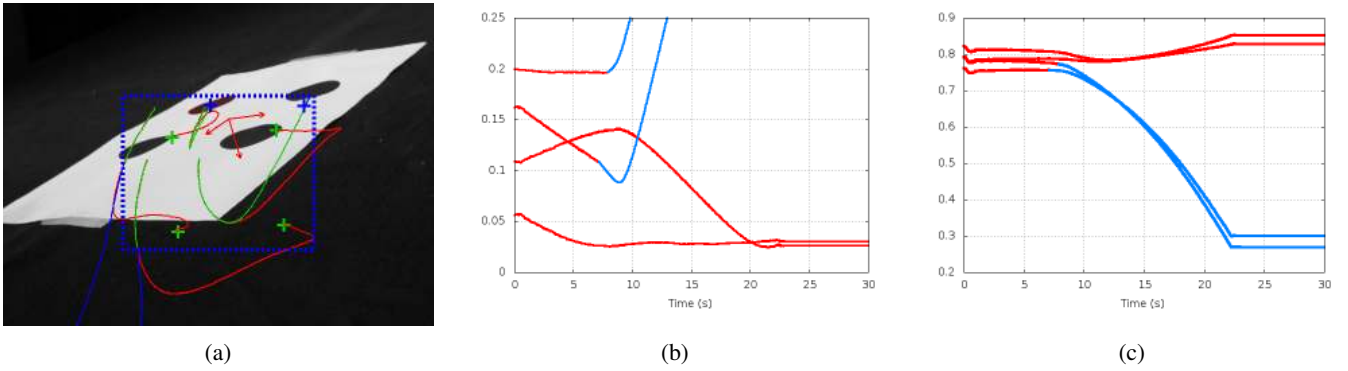


Fig. 2: Limited field of view with correction (9): (a) is the final state where the green curves represent the *measured* path of the visual features while the blue curves represent the *predicted* path (when no measurement is available). (b) is the evolution of the error $e_i(t) = \|s_i(t) - s_i^*\|$ (red parts) and of $e_i(t) = \|s_m(t) - s_i^*\|$ (blue parts). (c) is the evolution of $\|Z_{m_i}(t) - Z_i(t)\|$. Because of the poor performance of the correction (9), the predicted paths substantially diverge from the expected ones (in red), causing a complete failure of the servoing loop.

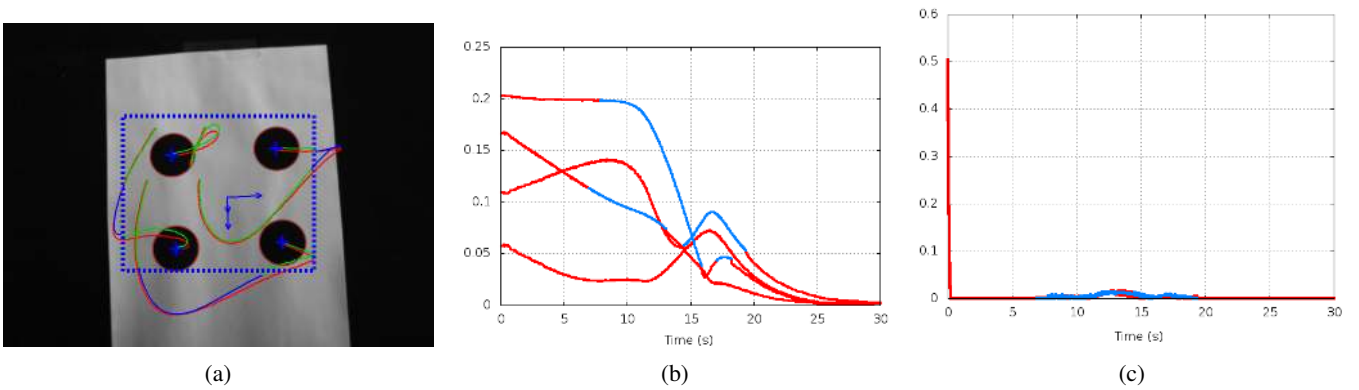


Fig. 3: Limited field of view with correction (11): (a) is the final state. (b) is the evolution of the error $e_i(t) = \|s_i(t) - s_i^*\|$ (red parts) and of $e_i(t) = \|s_m(t) - s_i^*\|$ (blue parts). (c) is the evolution of $\|Z_{m_i}(t) - Z_i(t)\|$. Note how, compared to Fig. 2, convergence towards the desired pose is now correctly achieved thanks to the superior performance of the correction (11), that allows for a more accurate prediction of $s_m(t)$ when no measurement is available.

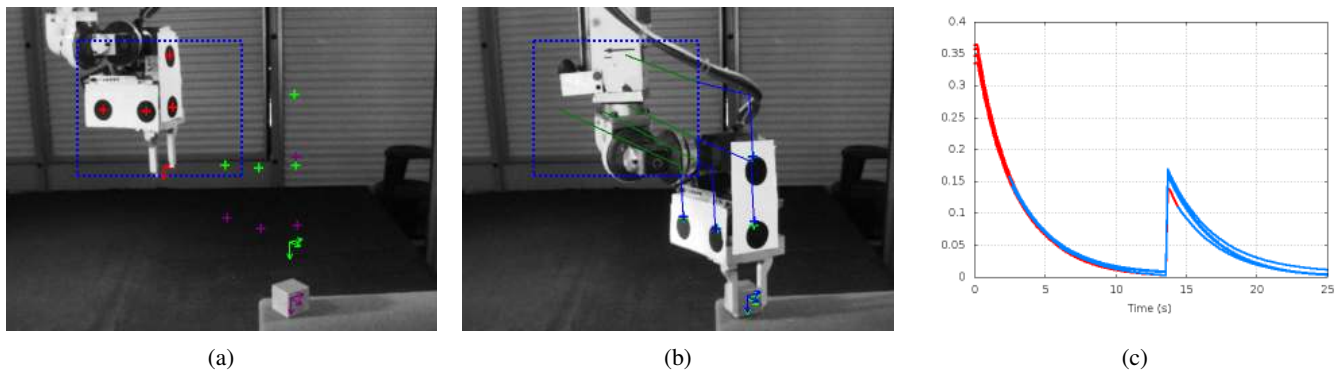


Fig. 4: Limited field of view with correction (11) applied to the Eye-To-Hand system. (a) is the initial state where desired visual features are in green for step 1 and in purple for step 2. (b) is the final state. (c) is the evolution of the error $e_i(t) = \|\mathbf{s}_i(t) - \mathbf{s}_i^*\|$ (red parts) and of $e_m(t) = \|\mathbf{s}_m(t) - \mathbf{s}_m^*\|$ (blue parts). The 2 servoing tasks are correctly realized. This again confirms the effectiveness of correction (11) what allows quickly recovering any initial error in the camera/target model pose for then allowing an accurate prediction $\mathbf{s}_m(t)$ during the occlusion phase.

REFERENCES

- [1] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.
- [2] E. Marchand and G. Hager, "Dynamic sensor planning in visual servoing," in *1998 IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 1988–1993.
- [3] D. Folio and V. Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, Dec 2005, pp. 3898–3903.
- [4] G. Morel, T. Liebezeit, J. Szwedczyk, S. Boudet, and J. Pot, "Dynamic sensor planning in visual servoing," in *6th Int. Symp. on Experimental Robotics*, 2000, pp. 99–108.
- [5] G. Chesi and K. Hashimoto, "Keeping features in the field of view in eye-in-hand visual servoing: a switching approach," *IEEE Trans. on Robotics*, vol. 20, no. 5, pp. 908–914, 2004.
- [6] N. Gans and S. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans. on Robotics*, vol. 23, no. 3, pp. 530–540, 2006.
- [7] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.
- [8] A. Hafez, A. Nelakanti, and C. Jawahar, "Path planning approach to visual servoing with feature visibility constraints: a convex optimization based solution," in *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2000.
- [9] T. Shen and G. Chesi, "Visual servoing path planning for cameras obeying the unified model," *Advanced Robotics*, vol. 26, no. 8–9, pp. 843–860, 2012.
- [10] M. Kazemi, K. Gupta, and M. Mehrendezh, "Randomized kinodynamic planning for robust visual servoing," *IEEE Trans. on Robotics*, vol. 29, no. 5, pp. 1197–1211, 2013.
- [11] T. Murao, T. Yamada, and M. Fujita, "Predictive visual feedback control with eye-in-hand system via stabilizing receding horizon approach," in *Decision and Control, 2006 45th IEEE Conference on*, dec. 2006, pp. 1758–1763.
- [12] M. Sauvée, P. Poignet, E. Dombre, and E. Courtial, "Image based visual servoing through nonlinear model predictive control," in *International Conference on Decision and Control*, San Diego, CA, USA, December 13–15 2006.
- [13] C. Lazar, A. Burlacu, and C. Copot, "Predictive control architecture for visual servoing of robot manipulators," in *IFAC World Congress*, Milano (Italy), August 28 - September 2 2011, pp. 9464–9469.
- [14] G. Allibert, E. Courtial, and Y. Tour, "Real-time visual predictive controller for image-based trajectory tracking of mobile robot," in *17th IFAC World Congr.*, Seoul, Korea, July 2008.
- [15] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transaction on Robotics*, vol. 26, no. 5, pp. 933–939, October 2010.
- [16] A. Assa and F. Janabi-Sharifi, "Robust model predictive control for visual servoing," in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.
- [17] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Trans. on Robotics*, vol. 30, no. 1, pp. 244–257, 2014.
- [18] N. García-Aracil, E. Malis, R. Aracil-Santonja, and C. Pérez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1214–1220, 2005.
- [19] D. Folio and V. Cadenat, "Dealing with visual features loss during a vision-based task for a mobile robot," *Int. Journal of Optomechatronics*, vol. 2, no. 3, pp. 185–204, 2008.
- [20] N. Cazy, C. Dune, P.-B. Wieber, P. Robuffo Giordano, and F. Chaumette, "Pose Error Correction For Visual Features Prediction," in *IROS 2014 - IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Chicago, United States, Sept. 2014. [Online]. Available: <http://hal.inria.fr/hal-01010772>
- [21] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D Visual Servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, pp. 238–250, 1999.
- [22] E. Marchand and F. Chaumette, "Virtual visual servoing: a framework for real-time augmented reality," in *EUROGRAPHICS'02 Conf. Proceeding*, ser. Computer Graphics Forum, G. Drettakis and H.-P. Seidel, Eds., vol. 21(3), no. 3, Saarbrücken, Germany, Sept. 2002, pp. 289–298.
- [23] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 4, pp. 525–532, Aug 1998.
- [24] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.