Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Eye-in-Hand/Eye-to-Hand Multi-Camera Visual Servoing

Vincenzo Lippiello, Bruno Siciliano, Luigi Villani

Abstract— **A position-based visual servoing algorithm using an hybrid eye-in-hand/eye-to-hand multi-camera configuration is presented in this paper. Based on an extended Kalman filter, this approach exploits the data provided by all the cameras without "a priori" discrimination, allowing real-time object pose estimation. A suitable algorithm is in charge of selecting an optimal subset of image features on the basis of the desired task and of the current configuration of the workspace. Only this subset is considered for feature extraction, thus ensuring a computational cost independent of the number of cameras. Experimental results are reported to demonstrate the feasibility and the effectiveness of the proposed technique.**

I. INTRODUCTION

The adoption of visual feedback for closed-loop control of robot manipulators is becoming a common practice both in research and in industrial areas. This approach is known as visual servoing. Moreover, the increase in the performance/cost ratio of machine vision is opening new scenarios where multi-camera systems are employed (see [1] and [2]).

The two most adopted camera configurations are known as eye-in-hand, where one or more cameras are rigidly attached to the robot end effector, and eye-to-hand, where the cameras are fixed in the workspace [3]. The first one guarantees good accuracy and the ability to explore the workspace although with a limited sight; the second one ensures a panoramic sight of the workspace, but a lower accuracy. Hence, the use of both configurations at the same time makes the execution of complex tasks easier and offers higher flexibility in the presence of a dynamic scenario.

Recently, some effort has been made to design visual servoing systems based on hybrid eye-in-hand/eye-to-hand camera configurations. In [4] an eye-to-hand camera is in charge of the robot tool positioning while an eye-in-hand camera is in charge of the robot tool orientation. A similar approach is used in [5], where an eye-to-hand camera is employed to estimate the robot tool pose with respect to the workspace and an eye-in-hand camera is employed as data source for object pose estimation. Further, in [6], a camera mounted on the end effector of a robot has been adopted as an eye-to-hand camera for another robot to benefit of the advantages of a mobile camera.

All the above approaches do not fully exploit the potentialities of hybrid camera configurations. In fact, the information provided by different types of cameras (fixed or mobile) is employed for different goals. Hence, a complete integration is not really achieved. Moreover, the possibility to adopt a multi-camera visual system for both camera configurations is not considered.

In this work, a new approach based on the Extended Kalman Filter (EKF) is proposed to achieve a complete data fusion in a multi-camera eye-in-hand/eye-to-hand visual system. This approach allows the data provided by all the cameras to be used at the same time, without any kind of "a priori" discrimination. A suitable image-feature selection algorithm is in charge of dynamically selecting the data required for the execution of a specific task depending on the current configuration of the workspace. Only the selected features are grabbed and elaborated to achieve the measurements, and thus the computational time spent for image processing is independent of the number of cameras.

The Kalman filter computes the estimate of the pose of an object in motion in the visible workspace, which is fed back to a position-based visual servoing algorithm. Since the frequency of the pose estimation algorithm is limited by the camera frame rate (25–60 Hz), while a higher control bandwidth (more than 100 Hz) is required to guarantee stability and disturbance rejection for position control of a robot manipulator, an "indirect" visual servoing algorithm is implemented [3]. This scheme is based on an inner/outer feedback loop where the inner position feedback loop runs at a frequency higher than the outer visual feedback loop.

This paper is organized as follows. In Section II the model of the visual system and of the workspace is presented. The formulation of the EKF is illustrated in Section III. In Section IV the pose estimation algorithm is described, and the position-based visual servoing control scheme is briefly outlined. Experimental results for the case of two robots performing a vision-guided master/slave trajectory following task are presented in Section V.

II. MODELING

Consider a system of n_f video cameras fixed in the workspace (eye-to-hand cameras) and n_m video cameras mounted on the end effector of one or more robots (eye-in-hand cameras), with $n = n_f + n_m$. The geometry of the system with respect to a generic camera can be described using the classical pinhole model (see Fig. 1). In the following, the symbols $\mathcal{F}$ and $\mathcal{M}$ will denote the set of eye-to-hand cameras and eye-in-hand cameras respectively; moreover, the index ci will be used to denote the quantities referred to the camera frame ci. For each camera, a frame O_{ci}–$x_{ci}y_{ci}z_{ci}$ attached to the camera ci is considered, with the z_{ci}-axis aligned to the optical axis and the origin in the optical center. The sensor plane is parallel to the $x_{ci}y_{ci}$-plane at a distance $-\lambda_e^{ci}$ along the z_{ci}-axis, where λ_e^{ci} is the effective focal length of the

The authors are with PRISMA Lab, Dipartimento di Informatica e Sistemistica Università degli Studi di Napoli Federico II Via Claudio 21, 80125 Napoli, Italy {lippiell,siciliano,lvillani}@unina.it

0-7803-9568-9/05/$20.00 ©2005 IEEE

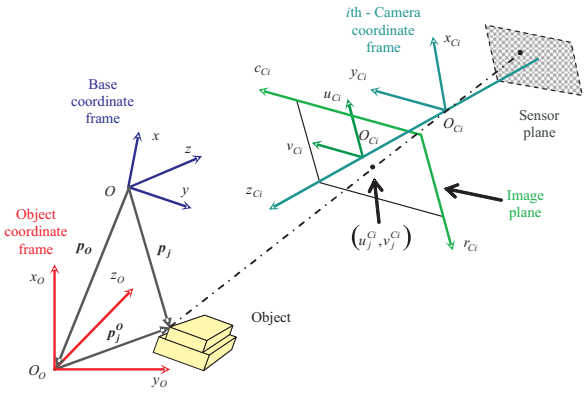

Fig. 1. Reference frames for the camera ci and for the object according to the pinhole model.

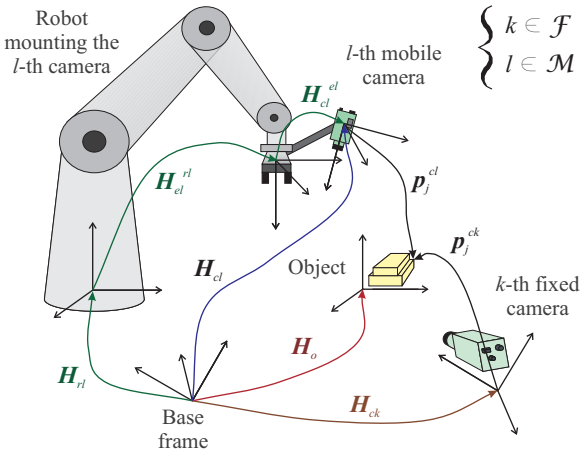

Fig. 2. Eye-in-hand/eye-to-hand cameras.

camera lens. The image plane is parallel to the $x_{ci}y_{ci}$-plane at a distance λ_e^{ci} along the z_{ci}-axis. The intersection of the optical axis with the image plane defines the principal optical point O'_{ci}, which is the origin of the image frame O'_{ci}–$u_{ci}v_{ci}$ whose axes u_{ci} and v_{ci} are taken parallel to the axes x_{ci} and y_{ci} respectively.

A point P with coordinates $\boldsymbol{p}^{ci} = \begin{bmatrix} x^{ci} & y^{ci} & z^{ci} \end{bmatrix}^{\mathrm{T}}$ in the camera frame is projected onto the point of the image plane whose coordinates can be computed by the equation

$$\begin{bmatrix} u^{ci} \\ v^{ci} \end{bmatrix} = \frac{\lambda_e^{ci}}{z^{ci}} \begin{bmatrix} x^{ci} \\ y^{ci} \end{bmatrix} \tag{1}$$

which is known as perspective transformation. A spatial sampling can be applied to the image plane by expressing the coordinates in terms of number of pixels $\begin{bmatrix} r_0^{ci} & c_0^{ci} \end{bmatrix}^{\mathrm{T}}$.

Without loss of generality, the case of a single moving object is considered. The position and orientation of a frame attached to the object O_o–$x_oy_oz_o$ with respect to a base coordinate frame O–xyz can be expressed in terms of the coordinate vector of the origin $\boldsymbol{o}_o = \begin{bmatrix} x_o & y_o & z_o \end{bmatrix}^{\mathrm{T}}$ and of the rotation matrix $\boldsymbol{R}_o(\boldsymbol{\varphi}_o)$, where $\boldsymbol{\varphi}_o = \begin{bmatrix} \phi_o & \alpha_o & \psi_o \end{bmatrix}^{\mathrm{T}}$ is the vector of the roll, pitch and yaw angles. The vector $\boldsymbol{x}_o = \begin{bmatrix} \boldsymbol{o}_o^{\mathrm{T}} & \boldsymbol{\varphi}_o^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ defines a minimal representation of the object pose with respect to the base frame.

Consider s feature points of the object. The homogeneous coordinate vector $\tilde{\boldsymbol{p}}_j = \begin{bmatrix} x_j & y_j & z_j & 1 \end{bmatrix}^{\mathrm{T}}$ of the feature point P_j $(j = 1, \ldots, s)$ can be expressed in the base frame as

$$\tilde{\boldsymbol{p}}_j(\boldsymbol{x}_o) = \boldsymbol{H}_o(\boldsymbol{x}_o)\tilde{\boldsymbol{p}}_j^o, \tag{2}$$

where $\tilde{\boldsymbol{p}}_j^o$ is the homogeneous coordinate vector of P_j expressed in the object frame and $\boldsymbol{H}_o$ is the homogeneous transformation matrix that represents the pose of the object frame referred to the base frame:

$$\boldsymbol{H}_o(\boldsymbol{x}_o) = \begin{bmatrix} \boldsymbol{R}_o(\boldsymbol{\varphi}_o) & \boldsymbol{o}_o \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix},$$

where $\boldsymbol{0}$ is the (3×1) null vector. The constant vector $\tilde{\boldsymbol{p}}_j^o$ is assumed to be known, and can be computed from a CAD model of the object or via a suitable calibration procedure.

Analogously, the homogeneous coordinate vector of P_j can be expressed in the camera ci frame as

$$\tilde{\boldsymbol{p}}_j^{ci} = \boldsymbol{H}_{ci}^{-1}\tilde{\boldsymbol{p}}_j, \tag{3}$$

where $\boldsymbol{H}_{ci}$ is the homogeneous transformation matrix that represents the pose of the frame of camera ci referred to the base frame.

For the eye-to-hand cameras, the matrix $\boldsymbol{H}_{ci}$ is constant, and can be computed through a suitable calibration procedure [7], while for the eye-in-hand cameras (see Fig. 2), this matrix depends on the robot current pose and can be expressed in the form:

$$\boldsymbol{H}_{ci} = \boldsymbol{H}_{ri}\boldsymbol{H}_{ei}^{ri}(\boldsymbol{x}_{ei}^{ri})\boldsymbol{H}_{ci}^{ei} \tag{4}$$

where $\boldsymbol{H}_{ri}$ is the homogeneous transformation matrix of the base frame of the robot ri carrying the camera ci with respect to the common base frame, $\boldsymbol{H}_{ei}^{ri}$ is the homogeneous transformation matrix of the end effector ei with respect to the base frame of the robot ri, and $\boldsymbol{H}_{ci}^{ei}$ is the homogeneous transformation matrix of the camera ci with respect to the frame ei of the end effector where the camera is mounted. Notice that $\boldsymbol{H}_{ri}$ and $\boldsymbol{H}_{ci}^{ei}$ are constant and can be estimated through suitable calibration procedures (see [8]), while $\boldsymbol{H}_{ei}^{ri}$ depends on the current end-effector pose $\boldsymbol{x}_{ei}^{ri}$ and may be computed using the robot kinematic model.

Considering (2), (3), and (4), the following equations are obtained

$$\tilde{\boldsymbol{p}}_j^{ci}(\boldsymbol{x}_o) = \boldsymbol{H}_{ci}^{-1}\boldsymbol{H}_o(\boldsymbol{x}_o)\tilde{\boldsymbol{p}}_j^o \tag{5}$$

if $i \in \mathcal{F}$, and

$$\tilde{\boldsymbol{p}}_j^{ci}(\boldsymbol{x}_o, \boldsymbol{x}_{ei}^{ri}) = (\boldsymbol{H}_{ri}\boldsymbol{H}_{el}^{ri}(\boldsymbol{x}_{ei}^{ri})\boldsymbol{H}_{ci}^{ei})^{-1}\boldsymbol{H}_o(\boldsymbol{x}_o)\tilde{\boldsymbol{p}}_j^o \tag{6}$$

if $i \in \mathcal{M}$. By plugging (5) and (6) in (1), a system of $2ns$ nonlinear equations is achieved, which depend on the measurements of the s feature points in the image plane of the n cameras and on the n_m current robot poses. Obviously,

the six components of the vector $\boldsymbol{x}_o$ are the unknown variables. To solve these equations at least six independent equations are required.

The computation of the solution is nontrivial and for visual servoing applications it has to be repeated at a high sampling rate. The recursive Kalman filter provides a computationally tractable solution, which can also incorporate redundant measurement information.

III. Extended Kalman Filter

In order to estimate the position and orientation of the object, a discrete-time state space dynamic model has to be considered, describing the object motion. The state vector of the dynamic model is chosen as $\boldsymbol{w} = \begin{bmatrix}\boldsymbol{x}_o^{\mathrm{T}} & \dot{\boldsymbol{x}}_o^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$. For simplicity, the object velocity is assumed to be constant over one sample period T. This approximation is reasonable in the hypothesis that T is sufficiently small. The corresponding dynamic modeling error can be considered as an input disturbance $\boldsymbol{\gamma}$ described by zero mean Gaussian noise with covariance $\boldsymbol{Q}$. The discrete-time dynamic model can be written as

$$\boldsymbol{w}_k = \boldsymbol{A}\boldsymbol{w}_{k-1} + \boldsymbol{\gamma}_k, \tag{7}$$

where $\boldsymbol{A}$ is the (12×12) block matrix

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{I}_6 & T\boldsymbol{I}_6 \\ \boldsymbol{0}_6 & \boldsymbol{I}_6\end{bmatrix},$$

with $\boldsymbol{I}_6$ and $\boldsymbol{0}_6$ denoting the identity and the null (6×6) matrices, respectively.

The output of the Kalman filter is the vector of the normalized coordinates of the s feature points in the image planes of the n cameras (n_f eye-to-hand cameras plus n_m eye-in-hand cameras)

$$\boldsymbol{\zeta}_k = \begin{bmatrix}\boldsymbol{\zeta}_k^{c1T} & \dots & \boldsymbol{\zeta}_k^{cnT}\end{bmatrix}_k^T, \tag{8}$$

where

$$\boldsymbol{\zeta}_k^{ci} = \begin{bmatrix}\frac{u_1^{ci}}{\lambda_e^{ci}} & \frac{v_1^{ci}}{\lambda_e^{ci}} & \dots & \frac{u_s^{ci}}{\lambda_e^{ci}} & \frac{v_s^{ci}}{\lambda_e^{ci}}\end{bmatrix}_k^{\mathrm{T}}.$$

By taking into account the equation (1), the output model of the Kalman filter can be written in the form:

$$\boldsymbol{\zeta}_k = \boldsymbol{g}(\boldsymbol{w}_k) + \boldsymbol{\nu}_k, \tag{9}$$

where $\boldsymbol{\nu}_k$ is the measurement noise, which is assumed to be zero mean Gaussian noise with covariance $\boldsymbol{R}$, and the function $\boldsymbol{g}(\boldsymbol{w}_k)$ is

$$\boldsymbol{g}(\boldsymbol{w}_k) = \begin{bmatrix}\boldsymbol{g}^{c1}(\boldsymbol{w}_k) \\ \vdots \\ \boldsymbol{g}^{cn}(\boldsymbol{w}_k)\end{bmatrix} \tag{10}$$

with

$$\boldsymbol{g}^{ci}(\boldsymbol{w}_k) = \begin{bmatrix}\frac{x_1^{ci}}{z_1^{ci}} & \frac{y_1^{ci}}{z_1^{ci}} & \dots & \frac{x_s^{ci}}{z_s^{ci}} & \frac{y_s^{ci}}{z_s^{ci}}\end{bmatrix}_k^{\mathrm{T}},$$

where the coordinates of the feature points $\boldsymbol{p}_j^{ci}$ are computed from the state vector $\boldsymbol{w}_k$ via equation (5) for the eye-to-hand cameras, and from the robots poses and the state vector via equation (6) for the eye-in-hand cameras. Matrix $\boldsymbol{R}$ can be evaluated during the calibration procedure of the cameras or by means of specific experiments.

Since the output model is nonlinear in the system state, the EKF must be adopted. The first step of the EKF algorithm provides an optimal estimate of the state at the next sample time according to the recursive equations

$$\hat{\boldsymbol{w}}_{k,k-1} = \boldsymbol{A}\hat{\boldsymbol{w}}_{k-1,k-1} \tag{11a}$$

$$\boldsymbol{P}_{k,k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1,k-1}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}, \tag{11b}$$

where $\boldsymbol{P}_{k,k-1}$ is the (12×12) covariance matrix of the estimate state error. The second step improves the previous estimate by using the input measurements according to the equations

$$\hat{\boldsymbol{w}}_{k,k} = \hat{\boldsymbol{w}}_{k,k-1} + \boldsymbol{K}_k(\boldsymbol{\zeta}_k - \boldsymbol{g}(\hat{\boldsymbol{w}}_{k,k-1})) \tag{12a}$$

$$\boldsymbol{P}_{k,k} = \boldsymbol{P}_{k,k-1} - \boldsymbol{K}\boldsymbol{C}_k\boldsymbol{P}_{k,k-1}, \tag{12b}$$

where $\boldsymbol{K}_k$ is the $(12 \times 2ns)$ Kalman matrix gain

$$\boldsymbol{K}_k = \boldsymbol{P}_{k,k-1}\boldsymbol{C}_k^{\mathrm{T}}(\boldsymbol{R}_k + \boldsymbol{C}_k\boldsymbol{P}_{k,k-1}\boldsymbol{C}_k^{\mathrm{T}})^{-1}, \tag{13}$$

being $\boldsymbol{C}_k$ the $(2ns \times 12)$ Jacobian matrix of the output function

$$\begin{aligned}\boldsymbol{C}_k &= \left.\frac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\hat{\boldsymbol{w}}_{k,k-1}} \\ &= \left[\frac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{x}_o} \quad \boldsymbol{0}_{2ns\times 12}\right]_{\boldsymbol{w}=\hat{\boldsymbol{w}}_{k,k-1}},\end{aligned} \tag{14}$$

where $\boldsymbol{0}_{2ns\times 12}$ is a null $(2ns \times 12)$ matrix corresponding to the partial derivative of $\boldsymbol{g}$ with respect to the velocity variables, which is null because function $\boldsymbol{g}$ does not depend on the velocity. The partial derivative of $\boldsymbol{g}$ with respect to the pose variables is

$$\frac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{x}_o} = \left[\frac{\partial \boldsymbol{g}}{\partial x_o} \quad \frac{\partial \boldsymbol{g}}{\partial y_o} \quad \frac{\partial \boldsymbol{g}}{\partial z_o} \quad \frac{\partial \boldsymbol{g}}{\partial \phi_o} \quad \frac{\partial \boldsymbol{g}}{\partial \vartheta_o} \quad \frac{\partial \boldsymbol{g}}{\partial \psi_o}\right],$$

and taking into account the expression of $\boldsymbol{g}$ in (10), the elements of this the matrix have the form:

$$\frac{\partial}{\partial \alpha}\left(\frac{x_j^{ci}}{z_j^{ci}}\right) = \left(\frac{\partial x_j^{ci}}{\partial \alpha} z_j^{ci} - x_j^{ci}\frac{\partial z_j^{ci}}{\partial \alpha}\right)(z_j^{ci})^{-2} \tag{15a}$$

$$\frac{\partial}{\partial \alpha}\left(\frac{y_j^{ci}}{z_j^{ci}}\right) = \left(\frac{\partial y_j^{ci}}{\partial \alpha} z_j^{ci} - y_j^{ci}\frac{\partial z_j^{ci}}{\partial \alpha}\right)(z_j^{ci})^{-2} \tag{15b}$$

where $\alpha = x_o, y_o, z_o, \phi_o, \vartheta_o, \psi_o$, $i = 1, \dots, n$, and $j = 1, \dots, s$.

The partial derivatives on the right-hand side of (15a) and (15b) can be computed as follows.

In view of (5) and (6), the partial derivatives with respect to the components of vector $\boldsymbol{o}_o = \begin{bmatrix}x_o & y_o & z_o\end{bmatrix}^{\mathrm{T}}$ are the elements of the Jacobian matrix

$$\frac{\partial \boldsymbol{p}_j^{ci}}{\partial \boldsymbol{o}_o} = \begin{cases}\boldsymbol{R}_{ci}^T = \text{const} & \text{if } i \in \mathcal{F} \\ (\boldsymbol{R}_{ri}\boldsymbol{R}_{el}^{ri}(\boldsymbol{x}_{ei}^{ri})\boldsymbol{R}_{ci}^{ei})^T & \text{if } i \in \mathcal{M}.\end{cases} \tag{16}$$

In order to express in compact form the partial derivatives with respect to the components of the vector $\boldsymbol{\varphi}_o =$

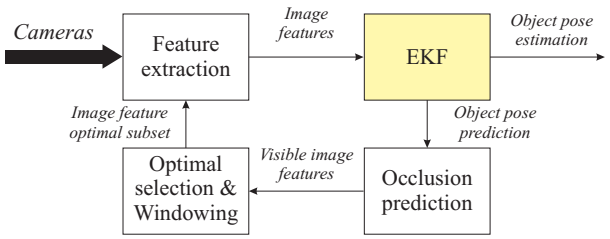

Fig. 3. Block scheme of the pose estimation algorithm.

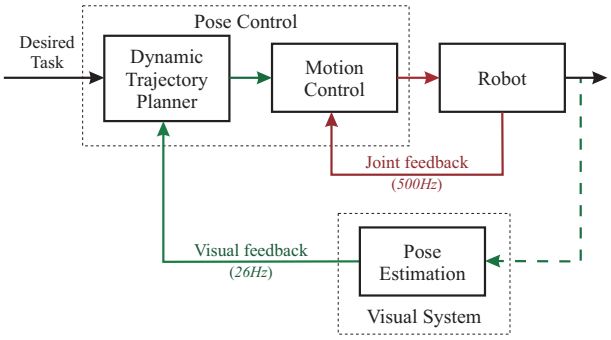

Fig. 4. Indirect position-based control scheme.

$\begin{bmatrix}\phi_o & \vartheta_o & \psi_o\end{bmatrix}^{\mathrm{T}}$, it is worth considering the following equalities

$$\begin{aligned}\mathrm{d}\boldsymbol{R}_o(\boldsymbol{\varphi}_o) &= \boldsymbol{S}(\mathrm{d}\boldsymbol{\omega}_o)\boldsymbol{R}_o(\boldsymbol{\varphi}_o)\\ &= \boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{S}(\boldsymbol{R}_o^T(\boldsymbol{\varphi}_o)\mathrm{d}\boldsymbol{\omega}_o) \qquad (17a)\\ \mathrm{d}\boldsymbol{\omega}_o &= \boldsymbol{T}_o(\boldsymbol{\varphi}_o)\mathrm{d}\boldsymbol{\varphi}_o \qquad (17b)\end{aligned}$$

where $\boldsymbol{S}(\cdot)$ is the skew-symmetric matrix operator, $\boldsymbol{\omega}_o$ is the angular velocity of the object frame with respect to the base frame, and the matrices $\boldsymbol{R}_o$ and $\boldsymbol{T}_o$, in the case of roll, pitch and yaw angles, have the form

$$\boldsymbol{R}_o(\boldsymbol{\varphi}_o) = \begin{bmatrix} c_{\phi_o}c_{\vartheta_o} & c_{\phi_o}s_{\vartheta_o}s_{\psi_o} - s_{\phi_o}c_{\psi_o} & c_{\phi_o}s_{\vartheta_o}c_{\psi_o} + s_{\phi_o}s_{\psi_o}\\ s_{\phi_o}c_{\vartheta_o} & s_{\phi_o}s_{\vartheta_o}s_{\psi_o} + c_{\phi_o}c_{\psi_o} & s_{\phi_o}s_{\vartheta_o}c_{\psi_o} - c_{\phi_o}s_{\psi_o}\\ -s_{\vartheta_o} & c_{\vartheta_o}s_{\psi_o} & c_{\vartheta_o}c_{\psi_o}\end{bmatrix}$$

$$\boldsymbol{T}_o(\boldsymbol{\varphi}_o) = \begin{bmatrix} 0 & -s_{\phi_o} & c_{\phi_o}c_{\vartheta_o}\\ 0 & c_{\phi_o} & s_{\phi_o}c_{\vartheta_o}\\ 1 & 0 & -s_{\vartheta_o}\end{bmatrix},$$

with $c_\alpha = \cos\alpha$ and $s_\alpha = \sin(\alpha)$. By virtue of (17) and the properties of the skew-symmetric matrix operator, the following chain of equalities holds

$$\begin{aligned}\mathrm{d}(\boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{p}_j^o) &= \mathrm{d}(\boldsymbol{R}_o(\boldsymbol{\varphi}_o))\boldsymbol{p}_j^o\\ &= \boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{S}(\boldsymbol{R}_o^T(\boldsymbol{\varphi}_o)\boldsymbol{T}_o(\boldsymbol{\varphi}_o)\mathrm{d}\boldsymbol{\varphi}_o)\boldsymbol{p}_j^o\\ &= \boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{S}^T(\boldsymbol{p}_j^o)\boldsymbol{R}_o^T(\boldsymbol{\varphi}_o)\boldsymbol{T}_o(\boldsymbol{\varphi}_o)\mathrm{d}\boldsymbol{\varphi}_o\\ &= \boldsymbol{S}^T(\boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{p}_j^o)\boldsymbol{T}_o(\boldsymbol{\varphi}_o)\mathrm{d}\boldsymbol{\varphi}_o,\end{aligned}$$

hence

$$\frac{\partial \boldsymbol{R}_o(\boldsymbol{\varphi}_o)}{\partial \boldsymbol{\varphi}_o}\boldsymbol{p}_j^o = \boldsymbol{S}^T(\boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{p}_j^o)\boldsymbol{T}_o(\boldsymbol{\varphi}_o). \qquad (18)$$

At this point, by virtue of (5), (6) and (18), the following equality holds

$$\frac{\partial \boldsymbol{p}_j^{ci}}{\partial \boldsymbol{\varphi}_o} = \begin{cases} \boldsymbol{R}_{ci}^T \dfrac{\partial \boldsymbol{R}_o(\boldsymbol{\varphi}_o)}{\partial \boldsymbol{\varphi}_o}\boldsymbol{p}_j^o & \text{if } i \in \mathcal{F}\\ (\boldsymbol{R}_{ri}\boldsymbol{R}_{el}^{ri}(\boldsymbol{x}_{ei}^{ri})\boldsymbol{R}_{ci}^{ei})^T \dfrac{\partial \boldsymbol{R}_o(\boldsymbol{\varphi}_o)}{\partial \boldsymbol{\varphi}_o}\boldsymbol{p}_j^o & \text{if } i \in \mathcal{M}.\end{cases}$$

IV. Position-Based Visual Servoing

The proposed algorithm based on the EKF is the core of the pose estimation algorithm shown in Fig. 3. Four main blocks are indicated, corresponding to the main elaboration steps needed to the estimation of the pose of an observed object.

The prediction of the object pose and velocity at the next sampling time, computed by the EKF, is input to an occlusion prediction algorithm that evaluates, for each camera, the set of visible object image features as well as an estimate of their projection on the image plane [9].

From this set, an optimal subset of image features is evaluated at each sample time via a feature selection algorithm [10]. Suitable quality indexes are used to perform this selection, aimed at maximizing the tracking performance using the minimum number of image features in the current configuration of the workspace. Notice that the number of selected image features is limited. Therefore, the computational cost of the feature extraction process is independent of the number of cameras. Moreover, a windowing technique is used to compute the size and location of the windows of the image plane to be grabbed for image processing. This considerably reduces the computational charge of the frame grabbing operations.

Finally, a feature extraction algorithm is applied to the selected windows to achieve the measurements used by the EKF algorithm. Notice that the set of input measurements is variable, therefore the output equations of the EKF is dynamically built on the basis of those image features that are effectively available.

The pose estimated by the algorithm described above is used to realize position-based visual servoing. Since the estimation algorithm runs at a frequency too low for the pose control loop, an "indirect" visual servoing scheme [3] is adopted, represented in Fig. 4. The presence of the high frequency inner feedback loop guarantees stability and disturbance rejection.

In detail, pose control is performed through an inner-outer control loop running at different frequency. The inner loop, running at the higher frequency (500 Hz in the experiments), implements motion control (independent joint control or any kind of joint space or task space control). In the outer loop, the block named dynamic trajectory planner computes the trajectory for the end effector on the basis of the current object pose and on the desired task. The input of this block is updated at the lower frequency (26 Hz in the experiments, corresponding to the frame rate of the employed cameras),

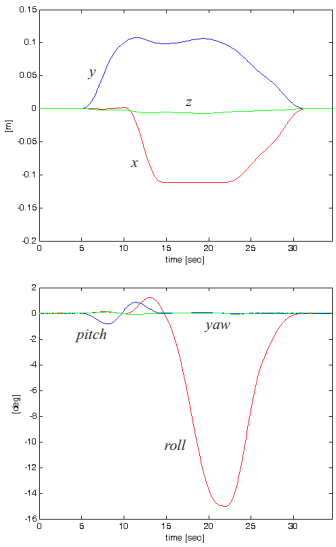

Fig. 5. Object trajectory. Top: position trajectory; bottom: orientation trajectory.

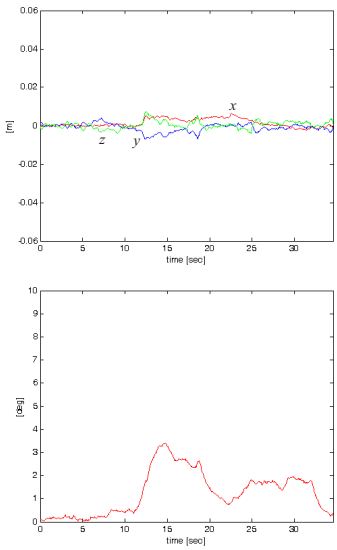

Fig. 6. Time history of the pose estimation error for the first case study. Top: position error. Bottom: orientation error.

while the output is available at the higher frequency, thanks to a second order interpolating filter. The pose estimation algorithm provides the measurements of the target object pose at the lower frequency.

V. Experiments

The experimental setup consists of two industrial robots Comau SMART-3 S. One arm is mounted on a sliding track which provides an additional degree of freedom (DOF) with respect to the standard six-DOF of the other arm. Both robots are controlled by a single PC with RTAI-Linux operating system. The experimental setup is completed with a stereo visual system composed by a PC equipped with two Matrox GENESIS boards and two Sony 8500CE B/W cameras (576×763 pixels). The Matrox boards are used both as frame grabber and for partial image processing (e.g., image windows extraction), while the PC host is in charge of executing vision-based algorithms (e.g., occlusion prediction and object motion estimation) and guarantees communication with the PC performing robot control via a standard serial connection. The first camera (with focal length $f_e = 16$ mm) is fixed and observes the entire workspace from a distance of about 1.5 m , while the second camera (with focal length $f_e = 8$ mm) is mounted on the end effector of the seven-DOF robot.

To test the effectiveness of the proposed algorithm, a visual synchronization task has been realized. The six-DOF robot (*master robot*) is used to move an object in the visual space of the fixed camera; thus the object position and orientation with respect to the base frame can be computed both from the joint position measurements, via the direct kinematic equation, and from visual measurements. The other robot (*slave robot*) is visually guided to preserve a fixed mutual pose with respect to the object. Notice that the measurements of the object pose computed from joint position measurements are used here only to evaluate the pose estimation error of the visual system.

An important issue is the calibration of the camera setup. In detail, the eye-to-hand camera was calibrated with respect to the base frame using the calibration algorithm proposed in [7], while the eye-in-hand camera was calibrated with respect to the end-effector frame of the slave robot using the calibration algorithm proposed in [8].

The value of the matrix $\boldsymbol{P}_{1,0}$ has been set to zero; moreover, the initial value of the state vector $\boldsymbol{w}_{1,0}$ have been set null for the velocity components, while the pose components has been roughly estimated through direct measurements. Finally, the covariance matrixes $\boldsymbol{Q}$ and $\boldsymbol{R}$ have been chosen as: $\boldsymbol{Q} = \text{diag}\{0,0,0,0,0,0,5,5,5,20,20,20\} \cdot 10^{-6}$, $\boldsymbol{R} = \text{diag}\{9,9,\cdots,9,9\}$, where $\boldsymbol{R}$ is a $(2s \times 2s)$ matrix being s the number of selected features. The values of the observation noise covariances have been evaluated during the camera calibration procedure while the values of the state noise covariances have been set on the basis of the velocity range of the object trajectories.

Two different case studies are considered. In the first case study both cameras are employed to estimate the pose of the objects, while in the second case study only the fixed camera is used. This allows comparing the pose tracking performance of the hybrid configuration with respect to that of the eye-to-hand configuration.

The trajectory of the object is represented in Fig. 5. The pose is referred to the initial object pose. The orientation is represented using roll, pitch and yaw angles.

The time history of the pose estimation error for the first case study is shown in Fig. 6. The orientation error is evaluated as the angle of an axis/angle representation corresponding to the rotation matrix of the slave end-effector frame with respect to the estimated object frame. Notice that

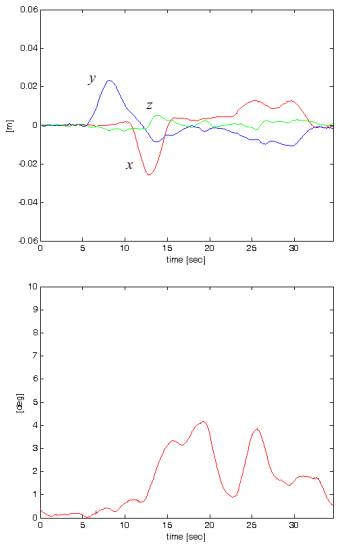

Fig. 7. Time history of the pose error for the first case study. Top: position error. Bottom: orientation error.

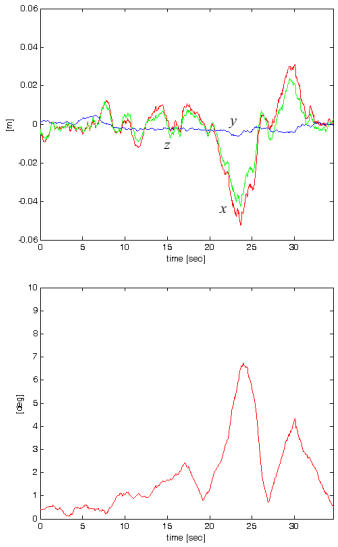

Fig. 8. Time history of the pose estimation error for the second case study. Top: position error. Bottom: orientation error.

the position error for the three components has the same magnitude and is lower than 1 cm; moreover, the orientation error is lower than 3 deg. Notice also that the errors are not significantly influenced by the velocity of the object; this is due to the adoption of a camera moving with the slave robot according to the object motion, in addition to the fixed camera.

In Fig. 7 the time history of the pose error of the end-effector frame of the slave robot with respect to the estimated object frame. The errors are higher with respect to the estimation errors of Fig. 6 because of the time delay caused by the low frame-rate of the visual system.

The time history of the pose estimation error for the second case study is shown in Fig. 8. Due to the adoption of only one fixed camera, the estimation error is sensibly higher than in the first case study. Moreover, the pose estimation error is quite sensible to the velocity of the object, especially for the orientation. The pose error of the end-effector frame of the slave robot with respect to the estimated object frame is not affected by the camera configuration and is not reported here for brevity.

VI. Conclusion

A position-based visual servoing algorithm based on an hybrid eye-in-hand/eye-to-hand multi-camera configuration is presented in this paper. The data provided by all the cameras are fully exploited, so that the benefits of both eye-in-hand and of the eye-to-hand configurations are preserved. Moreover, the adoption of a selection algorithm in charge of choosing an optimal subset of image features ensures a low computational cost for image processing, independent of the number of cameras. The experimental results have confirmed the feasibility of the proposed approach and have shown the superior performance of the hybrid configurations with respect to the eye-to-hand configuration in terms of pose tracking accuracy. A similar comparison with respect to the eye-in-hand configuration is not significant because the main advantage offered by the hybrid configuration is the wider sight.

References

[1] V. Gengenbach, H. H. Nagel, M. Tonko, and K. Schafer, "Automatic dismantling integrating optical flow into a machine vision-controlled robot system", in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, April 1996, vol. 2, pp. 1320–1325.

[2] C. Scheering and B. Kersting, "Uncalibrated hand-eye coordination with a redundant camera system", in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, May 1998, vol. 4, pp. 2953–2958.

[3] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[4] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing", in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, April 2000, vol. 3, pp. 2741–2746.

[5] M. Elena, M. Cristiano, F. Damiano, and M. Bonfe, "Variable structure pid controller for cooperative eye-in-hand/eye-to-hand visual servoing", in *Proceedings of the 2003 IEEE International Conference on Control Applications*, June 2003, vol. 2, pp. 989–994.

[6] A. Muis and K. Ohnishi, "Eye-to-hand approach on eye-in-hand configuration within real-time visual servoing", in *Proceedings of the 2004 IEEE International Workshop on Advanced Motion Control*, March 2004, pp. 647–652.

[7] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

[8] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lens", *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[9] V. Lippiello, B. Siciliano, and L. Villani, "An occlusion prediction algorithm for visual servoing tasks in a multi-arm robotic cell", in *Proceedings of the 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, June 2005.

[10] V. Lippiello and L. Villani, "Managing redundant visual measurements for accurate pose tracking", *Robotica*, vol. 21, pp. 511–519, 2003.