

A Systematic Approach to 3D Pose Estimation for Robotic Applications Based on a Multi-Camera Hybrid Visual System

Abstract—An algorithm for the estimation of the position and orientation of a moving object using an hybrid eye-in-hand/eye-to-hand multi-camera system is presented in this paper. Based on the extended Kalman filter, this approach exploits the data provided by all the cameras without “a priori” discrimination, allowing real-time object pose estimation. The proposed formulation can be used with different kinds of image features. Moreover, different kinds of representation of the object orientation can be adopted. A simulation case study is reported to test the feasibility and the effectiveness of the proposed technique.

I. INTRODUCTION

The interest on visual servoing, both in industrial and in service robotics, is growing in the last years. The two most adopted camera configurations for robotics applications are known as eye-in-hand configuration, where one or more cameras are rigidly attached to the robot end effector, and eye-to-hand configuration, where the cameras are fixed in the workspace [1]. 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 [2] 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 [3], 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 [4], 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.

Recently, a new approach based on the Extended Kalman Filter (EKF) has been proposed to achieve a complete data fusion in a multi-camera eye-in-hand/eye-to-hand visual system [5]. This approach allows the data provided by all the

cameras to be used at the same time, without any kind of “a priori” discrimination. Moreover, a suitable image-feature selection algorithm is adopted for the dynamic selection of 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 above approach is based on image features represented by points, i.e., the projections on the image plane of the corners of objects of polyhedral shape. This paper is aimed at generalizing the above formulation to the case of different kinds of features (edges, areas, etc.) usually adopted in visual servoing applications [6]. To this purpose, a systematic formulation of the extended Kalman filter for a multi-camera hybrid system is presented. One interesting feature of the proposed algorithm is that it allows to use any kind of image feature, as well as different kinds of features for different cameras, with a simple modification of the linearized output equation of the Kalman filter based on the computation of suitable image Jacobian matrices. The equations relating these image Jacobians to those used in image-based visual servoing (see, e.g., [6], [1]) are evidenced. Moreover, with the presented formulation, any parametrization of the object orientation, including the unit quaternion, can be adopted.

Simulation results are presented to test the effectiveness of the pose estimation algorithm for the simple case of a mono-camera system. Experimental tests are under development to fully exploit the capability of the proposed approach using an hybrid eye-in-hand/eye-to-hand camera configuration.

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 $n = n_f + n_m$. In the following, the index c will be used to denote the quantities referred to a frame $O_c-x_c y_c z_c$ attached to the camera c (eye-to-hand or eye-in-hand), with $c \in \{1, \dots, n\}$. The camera frame c is chosen with the z_c -axis aligned to the optical axis and the origin in the optical center of the lens. The sensor plane is parallel to the $x_c y_c$ -plane at a distance $-\lambda_c$ along the z_c -axis, where λ_c is the effective focal length of the lens. The image plane is parallel to the $x_c y_c$ -plane at a distance λ_c along the z_c -axis. The intersection of the optical axis with the image plane

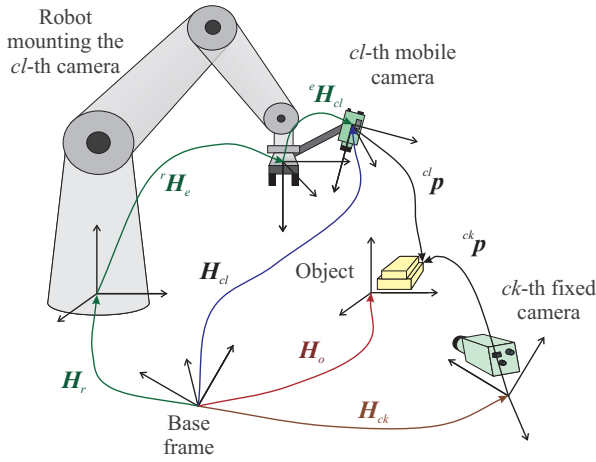


Fig. 1. Eye-in-hand/eye-to-hand cameras.

defines the principal optical point O'_c , which is the origin of the image frame $O'_c-X_cY_c$ whose axes X_c and Y_c are taken parallel to the axes x_c and y_c respectively.

Assuming that the projective geometry of the camera is modeled by perspective projection, a point P of the object with coordinates ${}^c\mathbf{p} = [x \ y \ z]^T$ with respect to the camera frame is projected onto the point of the image plane with coordinates

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{\lambda_c}{z} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (1)$$

Without loss of generality, the case of a single moving object is considered. The position and orientation of a frame attached to the object $O_o-x_o y_o z_o$ with respect to a base coordinate frame $O-xyz$ can be expressed in terms of the coordinate vector of the origin $\mathbf{o}_o = [x_o \ y_o \ z_o]^T$ and of the rotation matrix $\mathbf{R}_o(\boldsymbol{\varphi}_o)$, where $\boldsymbol{\varphi}_o$ is a $(p \times 1)$ vector corresponding to a suitable parametrization of the orientation. In the case that a minimal representation of the orientation is adopted, e.g., Euler angles, it is $p = 3$, while it is $p = 4$ if unit quaternions are used. Hence, the $(m \times 1)$ vector $\mathbf{x}_o = [\mathbf{o}_o^T \ \boldsymbol{\varphi}_o^T]^T$ defines a representation of the object pose with respect to the base frame in terms of $m = 3 + p$ parameters.

The homogeneous coordinate vector $\tilde{\mathbf{p}} = [\mathbf{p}^T \ 1]^T$ of the point P with respect to the base frame can be computed as

$$\tilde{\mathbf{p}} = \mathbf{H}_o(\mathbf{x}_o) {}^o\tilde{\mathbf{p}},$$

where ${}^o\tilde{\mathbf{p}}$ is the homogeneous coordinate vector of P with respect to the object frame and $\mathbf{H}_o$ is the homogeneous transformation matrix representing the pose of the object frame referred to the base frame:

$$\mathbf{H}_o(\mathbf{x}_o) = \begin{bmatrix} \mathbf{R}_o(\boldsymbol{\varphi}_o) & \mathbf{o}_o \\ \mathbf{0}_3^T & 1 \end{bmatrix},$$

where $\mathbf{0}_3$ is the (3×1) null vector. Notice that, if the object is rigid, the vector ${}^o\tilde{\mathbf{p}}$ is constant and can be computed from a CAD model of the object.

Let $\mathbf{H}_c$ denote the homogeneous transformation matrix representing the pose of the camera frame c referred to the base frame. For the eye-to-hand cameras, the matrix $\mathbf{H}_c$ is constant, and can be computed through a suitable calibration procedure [7], while for the eye-in-hand cameras (see Fig. 1) this matrix depends on the camera current pose $\mathbf{x}_c$, i.e., $\mathbf{H}_c = \mathbf{H}_c(\mathbf{x}_c)$, and can be computed as:

$$\mathbf{H}_c(\mathbf{x}_c) = \mathbf{H}_r {}^r\mathbf{H}_e({}^r\mathbf{x}_e) {}^e\mathbf{H}_c$$

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

Therefore, the homogeneous coordinate vector of P with respect to the camera frame c can be expressed as

$${}^c\tilde{\mathbf{p}} = {}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c) {}^o\tilde{\mathbf{p}} \quad (2)$$

where ${}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c) = {}^c\mathbf{H}^{-1}(\mathbf{x}_c) \mathbf{H}_o(\mathbf{x}_o)$. Notice that $\mathbf{x}_c$ is constant for eye-to-hand cameras; moreover, the matrix ${}^c\mathbf{H}_o$ does not depend on $\mathbf{x}_c$ and $\mathbf{x}_o$ separately but on the relative pose of the object frame with respect to the camera frame.

The (3×1) vector ${}^c\mathbf{p}$ corresponding to ${}^c\tilde{\mathbf{p}}$ in (2) is the coordinate vector of the point P with respect to the camera frame c . Hence, its time derivative ${}^c\dot{\mathbf{p}}$ has the meaning of relative velocity of the point P with respect to the camera frame c , expressed in the frame c . If the camera is fixed, this velocity represents also the absolute velocity of the point P with respect to the base frame, expressed in the camera frame, denoted by ${}^c\mathbf{v}_P$. When the camera frame is moving, the contribution to ${}^c\mathbf{v}_P$ due to the motion of the camera must be taken into account.

The velocity of the camera frame c with respect to the base frame can be characterized in terms of the translational velocity of the origin $\dot{\mathbf{v}}_{O_c}$ and of angular velocity $\boldsymbol{\omega}_c$. These vectors, expressed in the camera frame c , define the velocity screw ${}^c\boldsymbol{\nu}_c = [{}^c\dot{\mathbf{v}}_{O_c}^T \ {}^c\boldsymbol{\omega}_c^T]^T$. Then, the absolute velocity of the point P can be computed as

$${}^c\mathbf{v}_P = {}^c\dot{\mathbf{p}} + \boldsymbol{\Lambda}({}^c\mathbf{p}) {}^c\boldsymbol{\nu}_c \quad (3)$$

with $\boldsymbol{\Lambda}(\cdot) = [\mathbf{I}_3 \ -\mathbf{S}(\cdot)]$, where $\mathbf{I}_3$ is the (3×3) identity matrix and $\mathbf{S}(\cdot)$ denotes the (3×3) skew-symmetric matrix operator. Notice that the second term in the sum (3) represents the velocity of the point P due to the motion of the camera.

Equation (3) holds for any point P of the object, hence it can be applied to the origin O_o of the object frame. This yields

$${}^c\mathbf{v}_{O_o} = {}^c\dot{\mathbf{o}}_o + \boldsymbol{\Lambda}({}^c\mathbf{o}_o) {}^c\boldsymbol{\nu}_c, \quad (4)$$

where ${}^c\mathbf{o}_o$ is the vector of the coordinates of O_o with respect to the camera frame c , ${}^c\dot{\mathbf{o}}_o$ is the relative velocity of O_o with

respect to the camera frame c , while ${}^c\mathbf{v}_{O_o}$ is its absolute velocity; all the quantities are expressed in the camera frame c . On the other hand, the absolute angular velocity ${}^c\boldsymbol{\omega}_o$ of the object frame expressed in the camera frame c can be computed as

$${}^c\boldsymbol{\omega}_o = {}^c\boldsymbol{\omega}_{o,c} + {}^c\boldsymbol{\omega}_c \quad (5)$$

where ${}^c\boldsymbol{\omega}_{o,c}$ represents the relative angular velocity of the object frame with respect to the camera frame. The two equations (4) and (5) can be rewritten in the compact form

$${}^c\boldsymbol{\nu}_o = {}^c\boldsymbol{\nu}_{o,c} + \boldsymbol{\Gamma}({}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_c \quad (6)$$

where ${}^c\boldsymbol{\nu}_o = [{}^c\dot{\mathbf{v}}_{O_o}^T \quad {}^c\boldsymbol{\omega}_o^T]^T$ is the velocity screw corresponding to the absolute motion of the object frame, ${}^c\boldsymbol{\nu}_{o,c} = [{}^c\dot{\mathbf{o}}_o^T \quad {}^c\boldsymbol{\omega}_{o,c}^T]^T$ is the velocity screw corresponding to the relative motion of the object frame with respect to the camera frame c , and the matrix $\boldsymbol{\Gamma}(\cdot)$ is defined as

$$\boldsymbol{\Gamma}(\cdot) = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}(\cdot) \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix},$$

where $\mathbf{O}_3$ denotes the (3×3) null matrix. An useful expression of the relative velocity ${}^c\dot{\mathbf{p}}$ can be obtained observing that P is a point fixed in the object frame which is characterized by a relative velocity screw ${}^c\boldsymbol{\nu}_{o,c}$ with respect to the camera frame c . Hence, the following equality holds for the rigid body motion

$${}^c\dot{\mathbf{p}} = \boldsymbol{\Lambda}({}^c\mathbf{p} - {}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_{o,c}. \quad (7)$$

III. IMAGE FEATURES

An image feature is any structural feature that can be extracted from an image, corresponding to the projection of a physical feature of the object onto the camera image plane. An image feature can be characterized by a set of parameters that can be calculated from the image. Examples are, e.g., the image plane coordinates of points, the distance between two points in the image plane and the orientation of the line connecting those two points, the area of the projected surface, the parameters of lines in the image plane. The image feature parameters for the camera c can be grouped in a vector $\mathbf{f}_c = [f_{c,1} \cdots f_{c,k}]^T$, where $f_{c,j}$ is a real value and k is the dimension of the image feature parameter space. The mapping from the position and orientation of the object to the corresponding image feature parameters can be computed using the projective geometry of the camera and can be written in the form

$$\mathbf{f}_c = \mathbf{h}_c({}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c)), \quad (8)$$

where only the dependence from the relative pose of the object frame with respect to the camera frame has been explicitly evidenced. Usually, for visual tracking and visual servoing applications, it is required the computation of the differential mapping

$$\dot{\mathbf{f}}_c = \mathbf{J}_{o,c} {}^c\boldsymbol{\nu}_{o,c} \quad (9)$$

where the matrix $\mathbf{J}_{o,c}$ is the Jacobian mapping the relative velocity screw of the object frame with respect to the camera frame into the variation of the image feature parameters.

Taking into account the velocity composition (6), the equation (9) can be rewritten in the form

$$\dot{\mathbf{f}}_c = \mathbf{J}_{o,c} {}^c\boldsymbol{\nu}_o - \mathbf{J}_c {}^c\boldsymbol{\nu}_c \quad (10)$$

where $\mathbf{J}_{o,c}$ is also the Jacobian corresponding to the contribution of the absolute velocity screw of the object frame while

$$\mathbf{J}_c = \mathbf{J}_{o,c} \boldsymbol{\Gamma}({}^c\mathbf{o}_o) \quad (11)$$

is the Jacobian corresponding to the contribution of the absolute velocity screw of the camera frame. The Jacobian (11) is known in the literature as interaction matrix [6].

IV. EXTENDED KALMAN FILTER

In this work, the problem of the estimation of the pose vector $\mathbf{x}_o$ of the object with respect to the base frame from the measurements of the image parameters $\mathbf{f}_c$ obtained using a system of n eye-in-hand/eye-to-hand cameras is considered. The proposed solution is based on the Extended Kalman filter. To this purpose, 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 $\mathbf{w} = [\mathbf{x}_o^T \quad \dot{\mathbf{x}}_o^T]^T$. For simplicity, the object velocity is assumed to be constant over one sample period T_s . This approximation is reasonable in the hypothesis that T_s is sufficiently small. The corresponding dynamic modeling error can be considered as an input disturbance γ described by zero mean Gaussian noise with covariance $\mathbf{Q}$. The discrete-time dynamic model can be written as

$$\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \gamma_k, \quad (12)$$

where $\mathbf{A}$ is the $(2m \times 2m)$ block matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_m & T_s \mathbf{I}_m \\ \mathbf{O}_m & \mathbf{I}_m \end{bmatrix}.$$

The output of the Kalman filter is the vector of the image feature parameters measured on the image planes of the n cameras at the time kT_s

$$\boldsymbol{\zeta}_k = [\boldsymbol{\zeta}_{1,k}^T \quad \cdots \quad \boldsymbol{\zeta}_{n,k}^T]^T,$$

where $\boldsymbol{\zeta}_{c,k} = \mathbf{f}_{c,k} + \boldsymbol{\nu}_{c,k}$, $c = 1, \dots, n$, being $\boldsymbol{\nu}_{c,k}$ the measurement noise of the camera c . The measurement noise is assumed to be zero mean Gaussian noise with covariance $\boldsymbol{\Pi}_c$. The covariance matrix can be evaluated during the calibration procedure of the cameras or by means of specific experiments.

Taking into account the equation (8), the output model of the Kalman filter can be written in the form:

$$\boldsymbol{\zeta}_k = \mathbf{g}(\mathbf{w}_k, \mathbf{c}_k) + \boldsymbol{\nu}_k,$$

where $\mathbf{c}_k = [\mathbf{x}_{1,k}^T \cdots \mathbf{x}_{n,k}^T]^T$, $\boldsymbol{\nu}_k = [\boldsymbol{\nu}_{1,k}^T \cdots \boldsymbol{\nu}_{n,k}^T]^T$ and

$$\mathbf{g}(\mathbf{w}_k, \mathbf{x}_{c,k}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{w}_k, \mathbf{x}_{1,k}) \\ \vdots \\ \mathbf{g}_n(\mathbf{w}_k, \mathbf{x}_{n,k}) \end{bmatrix} \quad (13)$$

with $\mathbf{g}_c(\mathbf{w}_k, \mathbf{x}_{c,k}) = \mathbf{h}_c({}^c\mathbf{H}_o(\mathbf{w}_k, \mathbf{x}_{c,k}))$, $c = 1, \dots, n$. Notice that $\mathbf{x}_{c,k}$ is the pose of the camera frame c at time kT_s ;

this quantity is known and constant for eye-to-hand cameras while it can be computed from the robot direct kinematics for eye-in-hand cameras.

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

$$\begin{aligned}\hat{\mathbf{w}}_{k,k-1} &= \mathbf{A}\hat{\mathbf{w}}_{k-1,k-1} \\ \mathbf{P}_{k,k-1} &= \mathbf{A}\mathbf{P}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{k-1},\end{aligned}$$

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

$$\begin{aligned}\hat{\mathbf{w}}_{k,k} &= \hat{\mathbf{w}}_{k,k-1} + \mathbf{K}_k(\boldsymbol{\zeta}_k - \mathbf{g}(\hat{\mathbf{w}}_{k,k-1})) \\ \mathbf{P}_{k,k} &= \mathbf{P}_{k,k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k,k-1},\end{aligned}$$

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

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

being $\mathbf{C}_k$ the Jacobian matrix of the output function

$$\mathbf{C}_k = \left. \frac{\partial \mathbf{g}(\mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\hat{\mathbf{w}}_{k,k-1}} = \begin{bmatrix} \frac{\partial \mathbf{g}(\mathbf{w})}{\partial \mathbf{x}_o} & \mathbf{O} \end{bmatrix}_{\mathbf{w}=\hat{\mathbf{w}}_{k,k-1}},$$

where $\mathbf{O}$ is a null matrix of proper dimension corresponding to the partial derivative of $\mathbf{g}$ with respect to the velocity variables, which is null because function $\mathbf{g}$ does not depend on the velocity.

In view of (13), the computation of $\mathbf{C}_k$ requires the computation of the Jacobian matrix of $\mathbf{g}_c$ with respect to $\mathbf{x}_o$. In the case of eye-in-hand cameras, it is

$$\dot{\mathbf{g}}_c = \dot{\mathbf{h}}_c = \frac{\partial \mathbf{h}_c}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{h}_c}{\partial \mathbf{x}_c} \dot{\mathbf{x}}_c. \quad (14)$$

On the other hand, the time derivative of $\mathbf{h}_c$ can be computed also according to (10) as a function of ${}^c\nu_o$ and ${}^c\nu_c$. The velocity screws can be expressed in the form

$${}^c\nu_o = {}^c\mathbf{L}(\mathbf{x}_o)\dot{\mathbf{x}}_c, \quad {}^c\nu_c = {}^c\mathbf{L}(\mathbf{x}_c)\dot{\mathbf{x}}_c$$

where ${}^c\mathbf{L}(\cdot)$ is a Jacobian matrix depending on the particular choice of coordinates for the orientation. Hence, comparing (10) with (14), the following noticeable equality can be found

$$\frac{\partial \mathbf{g}_c}{\partial \mathbf{x}_o} = \frac{\partial \mathbf{h}_c}{\partial \mathbf{x}_o} = \mathbf{J}_{o,c} {}^c\mathbf{L}(\mathbf{x}_o),$$

where the Jacobian $\mathbf{J}_{o,c}$ depends on the choice of image features for the camera c while ${}^c\mathbf{L}(\mathbf{x}_o)$ depends on the parametrization used to represent the orientation of the object frame. Obviously, the same result holds for eye-to-hand cameras (in this case $\dot{\mathbf{x}}_c = \mathbf{0}$ and ${}^c\nu_c = \mathbf{0}$).

The above equality allows to separate the computation of the Jacobian dependent on the choice of the image features (hereafter referred as *image Jacobian*) from the computation of the Jacobian dependent on the object pose representation (hereafter referred as *pose representation Jacobian*). By virtue of this property, the Kalman filter formulation becomes quite

flexible and can be applied with straightforward modifications to different kinds of image features, using both eye-in-hand and eye-to-hand cameras and any kind of parametrization of the object orientation.

In the following the expression of the Jacobian matrices $\mathbf{J}_{o,c}$ and ${}^c\mathbf{L}$ will be given for different kinds of image features and orientation representations.

V. IMAGE JACOBIAN

The computation of the image Jacobian $\mathbf{J}_{o,c}$ depends on the type of image feature parameters and can be performed using a procedure similar to that adopted in [6] to compute the interaction matrix $\mathbf{J}_c$. In the following, the case of points and segments and are considered as examples.

A. Points

Consider a point P of the object with coordinates ${}^c\mathbf{p}$ with respect to the camera frame c . The image feature parameters are chosen as the coordinates of the projection of the point P on the image plane, that can be computed using the equations 12, i.e.

$$\mathbf{f}_c = \begin{bmatrix} X \\ Y \end{bmatrix} = \mathbf{h}_c({}^c\mathbf{p}).$$

Differentiating these equations with respect to the time and taking into account the expression of ${}^c\dot{\mathbf{p}}$ given in (7), the following equality holds

$$\dot{\mathbf{f}}_c = \frac{\partial \mathbf{h}_c}{\partial {}^c\mathbf{p}} {}^c\dot{\mathbf{p}} = \frac{\partial \mathbf{h}_c}{\partial {}^c\mathbf{p}} \boldsymbol{\Lambda}({}^c\mathbf{p} - {}^c\mathbf{o}_o) {}^c\nu_{o,c}, \quad (15)$$

hence the image Jacobian has the expression:

$$\mathbf{J}_{o,c} = \frac{\partial \mathbf{h}_c}{\partial {}^c\mathbf{p}} \boldsymbol{\Lambda}({}^c\mathbf{p} - {}^c\mathbf{o}_o).$$

Being

$$\frac{\partial \mathbf{h}_c}{\partial {}^c\mathbf{p}} = \frac{\lambda_c}{z} \begin{bmatrix} 1 & 0 & -x/z \\ 0 & 1 & -y/z \end{bmatrix},$$

the following explicit expression can be found for $\mathbf{J}_{o,c}$:

$$\mathbf{J}_{o,c} = \frac{1}{z} \begin{bmatrix} \lambda_c & 0 & -X & -r_y X & r_z \lambda_c + r_x X & -r_y \lambda_c \\ 0 & \lambda_c & -Y & -r_z \lambda_c - r_y Y & r_x Y & r_x \lambda_c \end{bmatrix},$$

where ${}^c\mathbf{p} - {}^c\mathbf{o}_o = [r_x \ r_y \ r_z]^T$ is a vector depending on the orientation of the object frame with respect to the camera frame.

By virtue of (11), the same result can be obtained starting from the interaction matrix $\mathbf{J}_c$ for point features derived in [6] and using the equality:

$$\mathbf{J}_{o,c} = \boldsymbol{\Gamma}^{-1}({}^c\mathbf{o}_o) \mathbf{J}_c.$$

On the other hand, starting from (15) and computing ${}^c\dot{\mathbf{p}}$ from (3) with ${}^c\mathbf{v}_P = \mathbf{0}$, the following expression can be found for $\mathbf{J}_c$

$$\mathbf{J}_c = -\frac{\partial \mathbf{h}_c}{\partial {}^c\mathbf{p}} \boldsymbol{\Lambda}({}^c\mathbf{p})$$

which is the same used in [6].

B. Segments

Consider a segment connecting the points P_1 e P_2 of the object, with coordinates ${}^c\mathbf{p}_j = [x_j \ y_j \ z_j]^T$, with $j = 1, 2$, respectively. The image feature parameters are chosen as the two coordinates of the medium point of the projection of the segment on the image plane, its length and its angle with respect to the axis X_c , that can be computed as follows

$$\mathbf{f}_c = \begin{bmatrix} X_m \\ Y_m \\ L \\ \Theta \end{bmatrix} = \begin{bmatrix} (X_1 + X_2)/2 \\ (Y_1 + Y_2)/2 \\ \sqrt{\Delta X^2 + \Delta Y^2} \\ \tan^{-1}(\Delta Y/\Delta X) \end{bmatrix} = \mathbf{h}_c(\mathbf{f}_1, \mathbf{f}_2)$$

where $\Delta X = X_1 - X_2$, $\Delta Y = Y_1 - Y_2$, $\mathbf{f}_j = [X_j \ Y_j]^T$, with $j = 1, 2$.

Differentiating these equation with respect to the time, the following equality holds

$$\begin{aligned} \dot{\mathbf{f}}_c &= \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_1} \dot{\mathbf{f}}_1 + \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_2} \dot{\mathbf{f}}_2 \\ &= \left(\frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_1} \mathbf{J}_1 + \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_2} \mathbf{J}_2 \right) \boldsymbol{\nu}_{o,c} \end{aligned}$$

where $\mathbf{J}_j$ is the image Jacobian for the point feature P_j , computed as in the previous section. Therefore, the image Jacobian for the segment is

$$\mathbf{J}_{o,c} = \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_1} \mathbf{J}_1 + \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_2} \mathbf{J}_2,$$

where

$$\begin{aligned} \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_1} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \\ \Delta X/L & \Delta Y/L \\ -\Delta X/L^2 & \Delta Y/L^2 \end{bmatrix} \\ \frac{\partial \mathbf{h}_c}{\partial \mathbf{f}_2} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \\ -\Delta X/L & -\Delta Y/L \\ \Delta Y/L^2 & -\Delta X/L^2 \end{bmatrix}. \end{aligned}$$

VI. POSE REPRESENTATION JACOBIAN

The pose representation Jacobian ${}^c\mathbf{L}(\mathbf{x}_o)$ depends on the parametrization of the orientation. In fact, the following expression can be found

$${}^c\mathbf{L}(\mathbf{x}_o) = \begin{bmatrix} \mathbf{R}_c^T & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_c^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{T}(\boldsymbol{\varphi}_o) \end{bmatrix}$$

where $\mathbf{R}_c$ is the rotation matrix of the camera frame c with respect to the base frame and $\mathbf{T}(\boldsymbol{\varphi}_o)$ is the matrix defining the mapping

$$\dot{\boldsymbol{\omega}}_o = \mathbf{T}(\boldsymbol{\varphi}_o) \dot{\boldsymbol{\varphi}}_o \quad (16)$$

which depends on the coordinate vector $\boldsymbol{\varphi}_o$ chosen to represent the orientation. In the following, the case of the RPY (Roll-Pitch-Yaw) angles and of the unit quaternion are considered.

A. RPY angles

A minimal representation of orientation can be obtained by using a set of three Euler angles $\boldsymbol{\varphi} = [\phi \ \vartheta \ \psi]^T$. Among the 12 possible definitions of Euler angles, the RPY representation can be considered leading to the rotation matrix

$$\mathbf{R}(\boldsymbol{\varphi}) = \mathbf{R}_z(\phi) \mathbf{R}_y(\vartheta) \mathbf{R}_x(\psi)$$

where $\mathbf{R}_z$, $\mathbf{R}_y$, $\mathbf{R}_x$ are the matrices of the elementary rotations about three independent coordinate axes of successive frames. The RPY angles can be extracted from a given rotation matrix by using closed-form inversion formulæ [9]. The matrix $\mathbf{T}$ in (16) is

$$\mathbf{T}(\boldsymbol{\varphi}_o) = \begin{bmatrix} 0 & -s_\phi & c_\phi c_\vartheta \\ 0 & c_\phi & s_\phi c_\vartheta \\ 1 & 0 & -s_\vartheta \end{bmatrix}^{-1},$$

with $c_\alpha = \cos \alpha$ and $s_\alpha = \sin(\alpha)$. Notice that this matrix is singular whenever $\vartheta = \pm\pi/2$ (representation singularity).

B. Unit quaternion

An alternative description of the orientation can be obtained by resorting to a four-parameter singularity-free representation in terms of a unit quaternion $\boldsymbol{\varphi}_o = [\eta \ \boldsymbol{\epsilon}^T]^T$ with

$$\eta = \cos \frac{\theta}{2}, \quad \boldsymbol{\epsilon} = \sin \frac{\theta}{2} \mathbf{r}, \quad (17)$$

where θ and $\mathbf{r}$ are respectively the rotation and the (3×1) unit vector of an equivalent angle/axis description of orientation corresponding to $\mathbf{R}_o$. Notice that the quaternion components are constrained by

$$\|\boldsymbol{\varphi}_o\|^2 = \eta^2 + \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = 1. \quad (18)$$

Several algorithms exist to extract the quaternion from a given rotation matrix; an efficient one is reported in [10]. The matrix $\mathbf{T}$ in (16) is

$$\mathbf{T}(\boldsymbol{\varphi}_o) = 2 \begin{bmatrix} -\boldsymbol{\epsilon} & \eta \mathbf{I} - \mathbf{S}(\boldsymbol{\epsilon}) \end{bmatrix},$$

which is a (3×4) matrix.

VII. SIMULATION

The proposed estimation algorithm has been tested in a simulation case study. For the purpose of this work, the case of a fixed mono camera visual system has been considered, by using the quaternion representation for the object orientation and two kinds of image features: points and segments.

The effective focal length of the adopted camera is 8 mm, the pixel dimensions are $10 \times 10 \ \mu\text{m}$, the resolution of the sensor is 500×500 pixel, and the sample rate is 50 Hz.

The target object is a planar square with a dimension of 10×10 cm. A suitable feature extraction algorithm is employed to extract four image features (the four corners in the case of points and the four sides of the square in the case of segments) at each sampling time.

The object moves according to a spiral trajectory for the position that, starting from the background of the scene, comes towards the camera. The axis of the spiral is the optical axis

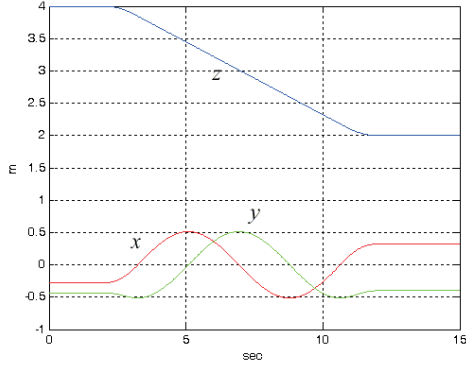


Fig. 2. Time history of the position components of the object frame trajectory.

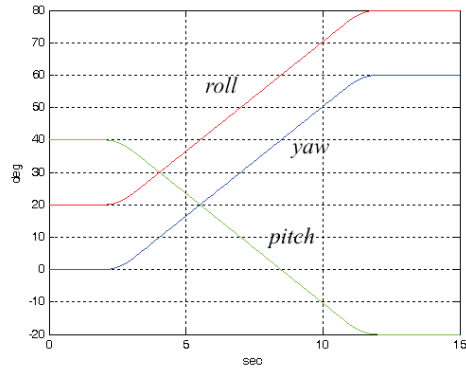


Fig. 3. Time history of the orientation components of object frame trajectory (RPY angles).

of the camera (Fig. 2). The trajectory for the orientation is shown in Fig. 3.

Notice that, although the orientation trajectory is assigned in terms of RPY angles, the model used by the Kalman filter is based on unit quaternions, to avoid any occurrence of representation singularities.

White independent Gaussian noise has been added to the true projections of the object on the cameras image plane. The variance of the noise has been set to 1 pixel, corresponding to spatial sampling and quantization errors.

The output measurement covariance matrix $\mathbf{\Pi}$ has been chosen according to the variance of the noise added to the measurements. The state noise covariance matrix has been selected so as to give a rough representation of the errors due to the simplification introduced on the model (constant velocity), by considering only velocity disturbance, i.e.

$$\mathbf{Q} = \text{diag}\{0, 0, 0, 0, 0, 0, 0, 500, 500, 50, 1, 1, 1, 1\} \times 10^{-6}.$$

The corresponding image trajectory of the object corners is represented in Fig. 4 (solid lines). The estimated image trajectory obtained using point features (the object corners) is shown in the same figure (dotted lines). Notice that the estimated trajectory cannot be distinguished from the real object

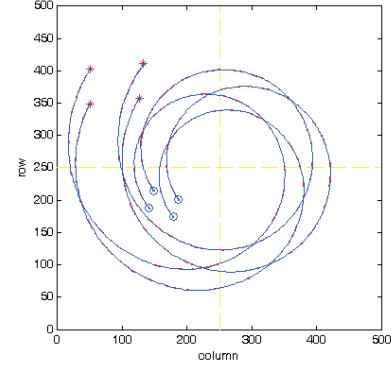


Fig. 4. Object trajectory (solid lines) and estimated trajectory (dotted lines) on the image plane.

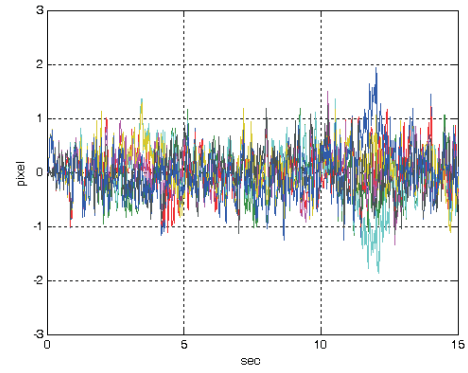


Fig. 5. Time history of the estimation error in the image plane in the case of point features.

trajectory. In fact, from the time history of the estimation error in the image plane, reported in Fig. 5, it can be seen that the maximum error is lower than ± 2 pixel, corresponding to the added noise range.

In order to appreciate the error due to the dynamic model approximation, in Fig. 6 the time history of the estimation error in the absence of noise added to the measurements is reported.

The simulation results obtained using segments (the sides of the square) as image features are very similar to those obtained using the points. However, the simulation does not take into account that, in real applications, the feature extraction process in the case of segments may be more accurate and robust than in the case of points. For brevity, only the estimation errors on the image plane have been reported in Figure 7.

VIII. CONCLUSION

An algorithm for the estimation of the position and orientation of a moving object using an hybrid eye-in-hand/eye-to-hand multi-camera system has been presented in this paper. Based on the extended Kalman filter, this approach exploits the data provided by all the cameras without “a priori” discrimination, allowing real-time object pose estimation. The

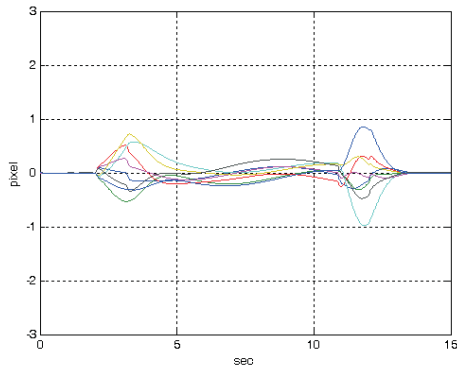


Fig. 6. Time history of the estimation error in the image plane in the case of point features, without measurement noise.

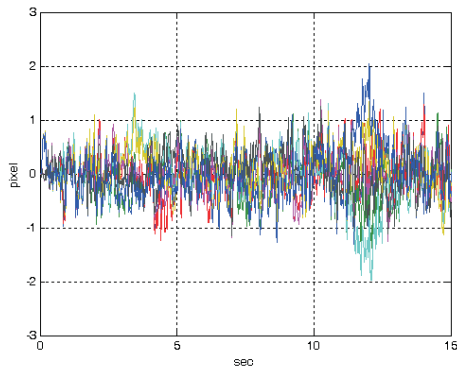


Fig. 7. Time history of the estimation error in the image plane in the case of segment features.

proposed formulation can be used with different kinds of image features. Moreover, different kinds of representation of the object orientation can be adopted. Experimental tests are under development to fully exploit the capability of the proposed approach using an hybrid eye-in-hand/eye-to-hand camera configuration.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] V. Lippiello, B. Siciliano, and L. Villani, "Eye-in-hand/eye-to-hand multi-camera visual servoing", in *Proceedings of the 2005 IEEE International Conference on Decision and Control*, December 2005.

- [6] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics", *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 313–326, 1992.
- [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] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, New York, NY, 1996.
- [10] S.W. Shepperd, "Quaternion from rotation matrix", *AIAA J. of Guidance and Control*, vol. 1, pp. 223–224, 1978.