

POSITION-BASED VISUAL SERVOING IN INDUSTRIAL MULTI-ARM ROBOTIC CELLS USING MULTIPLE CAMERAS

Vincenzo Lippiello, Bruno Siciliano and Luigi Villani

*Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125 Napoli, Italy
{lippie11,siciliano,lvillani}@unina.it*

Abstract: This paper deals with the problem of position-based visual servoing in a multi-arm robotic cell equipped with a hybrid eye-in-hand/eye-to-hand multi-camera system. The proposed approach is based on the real-time estimation of the pose of a target object by using the extended Kalman filter. The data provided by all the cameras are selected by a suitable algorithm on the basis of the prediction of the object self occlusions and of the mutual occlusions caused by the robot links and tools. Only an optimal subset of image features is considered for feature extraction, thus ensuring high estimation accuracy with a computational cost independent of the number of cameras. An experimental case study is presented for the case of two industrial robots performing a vision-guided grasping task.

Keywords: Visual servoing, pose estimation, multi-arm robots, Kalman filter, occlusion prediction

1. INTRODUCTION

One of the most important features of the new generation of industrial robots is the enhanced sensing capability, which is based on the use of heteroceptive sensors like force and vision.

In a multi-arm robotic cell, visual systems are usually composed by two or more cameras that can be rigidly attached to the robot end effectors (in the so called eye-in-hand configuration) or fixed in the workspace (in the so called eye-to-hand configuration) (Hutchinson *et al.*, 1996). The first configuration 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 (i.e., the so called hybrid configuration) makes the execution of complex tasks easier and offers higher flexibility in the presence of a dynamic scenario.

Another important issue is related to the occurrence of occlusions of the object features used as

visual data. Occlusions may happen when some parts of a workpiece are hidden with respect to the cameras by other parts of the object itself (self-occlusion), or when a workpiece is hidden by a robot link or tool, or by another object (mutual occlusion). They may cause failures to any kind of algorithm based on the extraction of image features. Hence, it is important to adopt suitable strategies able to cope with this problem.

The control scheme considered in this paper can be classified as position-based visual servoing, because it requires the computation of position and orientation errors defined in the Cartesian space (Wilson *et al.*, 1996). The main challenge of position-based algorithms is the real-time estimation of the pose of target objects from visual measurements. This problem has been largely investigated in the computer vision literature (see, e.g., (Chiuso *et al.*, 2002)) as well as in the robotic literature (see, e.g., (Wilson *et al.*, 1996)).

Thanks to its noise and disturbance rejection capabilities, the Extended Kalman Filter (EKF) is usually adopted to enhance the pose estimation

accuracy (Broida and Chellappa, 1986). Moreover, Kalman filtering offers many advantages over other estimation methods, e.g., temporal filtering, recursive implementation, possibility of realizing a proper statistical combination of redundant measurements, ability to change the measurement set during the operation. Also, the pose prediction computed by the filter allows setting up a dynamic windowing technique which may sensibly reduce the time required for feature extraction.

These features are suitably exploited in the algorithm presented here. In fact, differently from previous approaches based on hybrid configurations (see, e.g., (Flandin *et al.*, 2000; Muis and Ohnishi, 2004)) the pose estimation is achieved by using all the data provided by all the cameras, without any kind of “a priori” discrimination.

At each sampling time, a suitable selection algorithm is used for the dynamic selection of an optimal set of visual data available for pose estimation, on the basis of the specific task and of the current configuration of the workspace. Only the selected features are grabbed to achieve the measurements, and thus the computational time spent for image processing is independent of the number of cameras (Lippiello and Villani, 2003).

The computational efficiency of the selection algorithm is improved thanks to the adoption of a fast occlusion prediction algorithm purposely designed for multi-arm robotics cells. In this way, the occluded features can be eliminated from the set of features candidates to be extracted by the images. The algorithm is based on Binary Space Partitioning (BSP) tree structures to represent the 3D geometry of the cell. The BSP tree representation is updated in real time on the basis of the measurements of the robot joint positions and of the estimated poses of the workpieces. Hence the occlusion detection is realized through a visit algorithm of the tree (Lippiello and Villani, 2003).

In this paper, a complete scheme for position-based visual servoing on an industrial multi-arm robotic cell is described. The proposed formulation generalizes the results presented in the previous papers (Lippiello *et al.*, 2005b) (concerning hybrid camera configurations) and (Lippiello *et al.*, 2005a) (concerning occlusion detection in multi-arm robotic cells), where more details can be found. An experimental case study on a dual arm industrial robotic cell is described to show the feasibility of the proposed approach.

2. POSITION-BASED VISUAL SERVOING

A typical position-based visual servoing scheme for industrial robots is represented in Fig. 1. This algorithm requires the estimation of the pose of a target object with respect to a reference frame by using the vision system; the estimated pose is

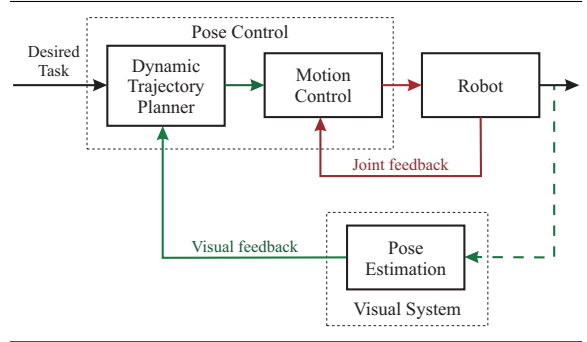


Fig. 1. Position-based visual servoing scheme

then fed back to a pose controller. Hence, the two main operations to be performed are *pose control* and *pose estimation*.

Notice that pose estimation is a computationally demanding task, because it requires processing of the measurements of some geometric features extracted from the images of one or more cameras. Hence, the frequency of the pose estimation algorithm is usually higher than the frequency of the pose control loop (greater or equal to 500 Hz in industrial robots, to guarantee tracking accuracy and disturbance rejection). In the best case, the pose estimation can be performed at camera frame rate (between 25 Hz and 60 Hz).

Pose control is performed through an inner-outer control loop. The inner loop, which usually runs at higher frequency, 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 of the desired task.

The pose estimation algorithm provides the measurements of the target object pose. The use of a multi-camera system requires the adoption of intelligent and computationally efficient strategies for the management of highly redundant information (a large number of object image features from multiple points of view). This task has to be realized with real-time constraints and thus the extraction and interpretation of all the available visual information is not possible.

To solve this problem, an efficient technique (described by the scheme represented in Fig. 2) has been developed which is able to improve the accuracy and robustness of the estimation by exploiting a minimal set of significant visual data suitably selected from the initial redundant set.

In detail, the EKF is used to compute an estimate of the object pose on the basis of the image features extracted from visual information. The filter provides also a prediction of the pose of the workpieces at the next sampling time, that is input to the occlusion prediction algorithm, in

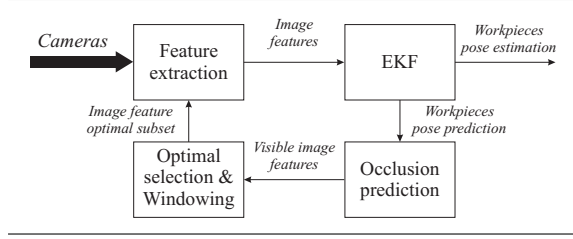


Fig. 2. Block scheme of the pose estimation algorithm

charge of filtering out all the feature points that are not visible at the next sampling time.

Notice that not all the visible feature points provided by the occlusion prediction algorithm are used for feature extraction. In fact, in a multi-camera system, the available feature points may be highly redundant and may increase the computational cost without a significant enhancement of the estimation accuracy (Wilson *et al.*, 1996). Therefore, a selection algorithm is adopted to dynamically select an optimal set of visible points. This algorithm is based on the minimization of an optimal cost function based on a combination of suitable quality indexes ensuring a balanced spatial distribution of the projections of the feature points on the image plane of each camera as well as a balanced distribution of the features among the different cameras, considering their different resolutions and focuses (Lippiello and Villani, 2003). After the optimal selection, 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.

In the following, the issues concerning the EKF and occlusion detection in a multi-arm robotic cell with an hybrid camera system will be considered.

3. HYBRID CAMERA CONFIGURATION

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 $c \in \{1, \dots, n\}$. Assuming that the projective geometry of the camera is modeled by perspective projection, a point P of the object with coordinates ${}^c p = [x \ y \ z]^T$ with respect to the camera frame is projected onto the point of the image plane with coordinates

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{\lambda_c}{z} \begin{bmatrix} x \\ y \end{bmatrix} = h_{p,c}({}^c p), \quad (1)$$

where λ_c denotes the focal length of the lens of the camera c . Without loss of generality, the case of

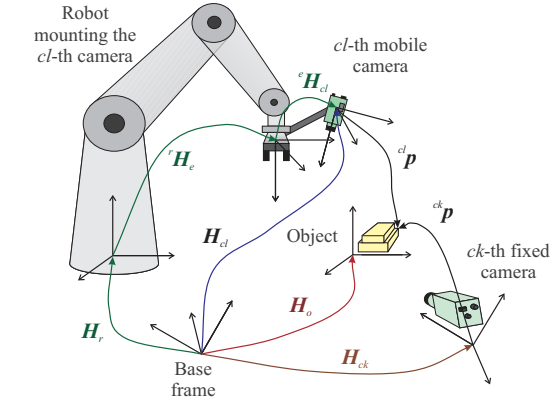


Fig. 3. Eye-in-hand/eye-to-hand cameras

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 $\mathbf{R}_o(\varphi_o)$, where φ_o is a $(q \times 1)$ 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 $q = 3$, while it is $q = 4$ if unit quaternions are used. Hence, the $(m \times 1)$ vector $\mathbf{x}_o = [\mathbf{o}_o^T \ \varphi_o^T]^T$ defines a representation of the object pose with respect to the base frame in terms of $m = 3 + q$ 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:

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

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, while for the eye-in-hand cameras (see Fig. 3) 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 transforma-

tion matrix of the camera c with respect to the frame e of the end effector where the camera is mounted. Notice that ${}^e\mathbf{H}_r$ and ${}^e\mathbf{H}_c$ are constant and can be estimated through suitable calibration procedures, 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 . This velocity can be computed as

$${}^c\dot{\mathbf{p}} = \mathbf{\Lambda}({}^c\mathbf{p} - {}^c\mathbf{o}_o) {}^c\boldsymbol{\nu}_{o,c}, \quad (3)$$

where ${}^c\mathbf{o}_o$ is the vector of the coordinates of the origin of the object frame O_o with respect to the camera frame c , ${}^c\boldsymbol{\nu}_{o,c} = [{}^c\dot{\mathbf{o}}_o^T {}^c\boldsymbol{\omega}_{o,c}^T]^T$ is the velocity screw characterizing the relative motion of the object frame with respect to the camera frame c in terms of the relative velocity of the origin ${}^c\dot{\mathbf{o}}_o$ and of the relative angular velocity ${}^c\boldsymbol{\omega}_{o,c}$. The matrix operator $\mathbf{\Lambda}(\cdot)$ is defined as $\mathbf{\Lambda}(\cdot) = [\mathbf{I}_3 \ -\mathbf{S}(\cdot)]$, 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 absolute velocity screw ${}^c\boldsymbol{\nu}_o = [{}^c\dot{\mathbf{v}}_{O_o}^T {}^c\boldsymbol{\omega}_o^T]^T$ corresponding to the absolute motion of the object frame with respect to the base frame can be expressed as

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

where ${}^c\boldsymbol{\nu}_c = [{}^c\dot{\mathbf{v}}_{O_c}^T {}^c\boldsymbol{\omega}_c^T]^T$ is the absolute velocity of the frame of the camera c expressed in the frame c . The matrix operator $\mathbf{\Gamma}(\cdot)$ is defined as

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

where $\mathbf{O}_3$ denotes the (3×3) null matrix.

To estimate the position and orientation of the object by using visual data, a set of image features must be selected, corresponding to the projection on the image plane of some physical features of the object. Also, the image features must be characterized by a set of parameters computed from the visual data. For the purpose of this work, point features are considered; hence, the image feature parameters are the coordinates of these point on the image plane.

The 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 parameter vector $\mathbf{f}_c$ can be computed using the projective geometry of the camera and can be written in the form

$$\mathbf{f}_c = \mathbf{h}_c({}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c)). \quad (5)$$

In the simple case of a single feature point, it is $k = 2$, $\mathbf{f}_c = [X \ Y]^T$ and the mapping (5) can be easily computed by replacing the equation (2) into the equation (1).

Usually, for visual tracking and visual servoing applications, it is required the computation of the differential mapping

$$\dot{\mathbf{f}}_c = \mathbf{J}_{o,c} {}^c\boldsymbol{\nu}_{o,c} \quad (6)$$

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 (4), the equation (6) can be rewritten in the form

$$\dot{\mathbf{f}}_c = \mathbf{J}_{o,c} {}^c\boldsymbol{\nu}_o - \mathbf{J}_c {}^c\boldsymbol{\nu}_c \quad (7)$$

where

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

is the Jacobian corresponding to the contribution of the absolute velocity screw of the camera frame, known as interaction matrix (Espiau *et al.*, 1996).

4. EKF

To compute an estimate of the pose vector $\mathbf{x}_o$ of the object with respect to the base frame using the EKF, 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 $\mathbf{w} = [\mathbf{x}_o^T \ \dot{\mathbf{x}}_o^T]^T$. The corresponding dynamic modeling error can be considered as an input disturbance $\boldsymbol{\gamma}$ described by zero mean Gaussian noise with covariance $\mathbf{Q}$. The discrete-time dynamic model can be written as

$$\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \boldsymbol{\gamma}_k, \quad (8)$$

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

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

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

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

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 $\mathbf{\Pi}_c$.

Taking into account the equation (5), the output model of the Kalman filter can be written in the form $\zeta_k = \mathbf{g}(\mathbf{w}_k, \mathbf{c}_k) + \nu_k$ where $\mathbf{c}_k = [\mathbf{x}_{1,k}^T \dots \mathbf{x}_{n,k}^T]^T$, $\nu_k = [\nu_{1,k}^T \dots \nu_{n,k}^T]^T$ and

$$\mathbf{g}(\mathbf{w}_k, \mathbf{x}_{c,k}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{w}_k, \mathbf{x}_{1,k}) \\ \vdots \\ \mathbf{g}_n(\mathbf{w}_k, \mathbf{x}_{n,k}) \end{bmatrix} \quad (9)$$

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.

The linearized output model can be written in the form $\zeta_k = \mathbf{C}_k \mathbf{w}_k + \nu_k$, with

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

where $\mathbf{O}$ is a null matrix. In view of (9), 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

$$\dot{\mathbf{g}}_c = \dot{\mathbf{h}}_c = \frac{\partial \mathbf{h}_c}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{h}_c}{\partial \mathbf{x}_c} \dot{\mathbf{x}}_c. \quad (10)$$

On the other hand, the time derivative of $\mathbf{h}_c$ can be computed also according to (7) as a function of ${}^c\nu_o$ and ${}^c\nu_c$. The velocity screws can be expressed in the form

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

where ${}^c\mathbf{L}(\cdot)$ is a Jacobian matrix depending on the particular choice of coordinates for the orientation (Sciavicco and Siciliano, 2000). Hence, comparing (7) with (10), 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 ${}^c\nu_c = \mathbf{0}$).

For the case of a single feature point, by virtue of Eq. (1), the following equality holds

$$\dot{\mathbf{f}}_c = \frac{\partial \mathbf{h}_{p,c}}{\partial {}^c\mathbf{p}} \dot{{}^c\mathbf{p}} = \frac{\partial \mathbf{h}_{p,c}}{\partial {}^c\mathbf{p}} \mathbf{\Lambda}({}^c\mathbf{p} - {}^c\mathbf{o}_o) {}^c\nu_{o,c}, \quad (11)$$

hence $\mathbf{J}_{o,c}$ has the expression

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

The derivation of the recursive equations of the EKF is standard and can be found, e.g., in (Lippiello and Villani, 2003).

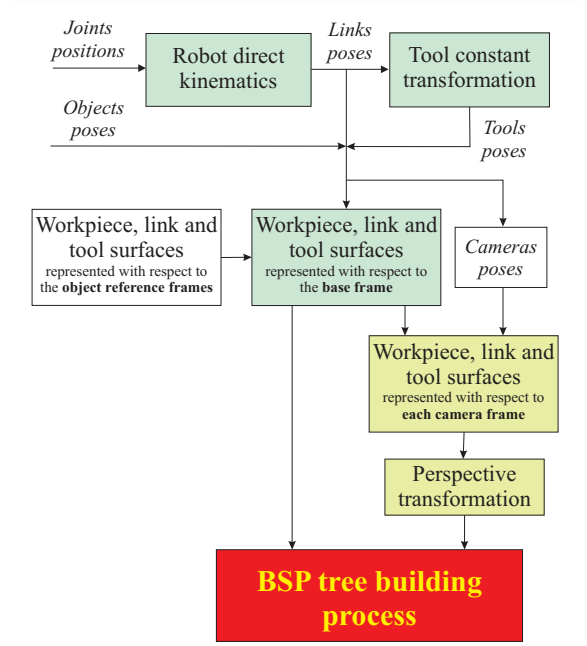


Fig. 4. Dynamic BSP tree building process

5. OCCLUSION PREDICTION

To the purpose of the occlusion prediction algorithm, the robotic cell is seen as a collection of objects which includes the workpieces (target objects), the robot links and tools, as well as all the possible obstacles that may occlude the workpieces with respect to the cameras.

At each sampling time, the algorithm provides the prediction of the positions on the camera image planes of all the visible feature points of the workpieces. The inputs of the algorithm are the robot joint measurements and the prediction of the pose of the workpieces provided by the EKF.

The occlusion prediction algorithm can be decomposed in two parts: geometric modelling and occlusion prediction.

The modelling part, described in Fig. 4, is aimed at generating on-line a BSP tree, representing the 3D geometrical model of the cell (Paterson and Yao, 1990). To this purpose, the CAD model of all the objects of the cell are assumed to be known. Moreover, each object has to be represented as a set of surfaces with respect to a reference frame fixed to the object itself. This type of representation can be generated once off-line to facilitate the on-line BSP tree construction.

The first step of the modelling procedure is the computation of the poses of all the robots links and tools using the joint position measurements and the direct kinematics. The second step is the computation of the pose of all the objects of the cell, estimated using the EKFs (one EKF for each object) with respect to the frame of the camera c . At this point, by using the camera perspective

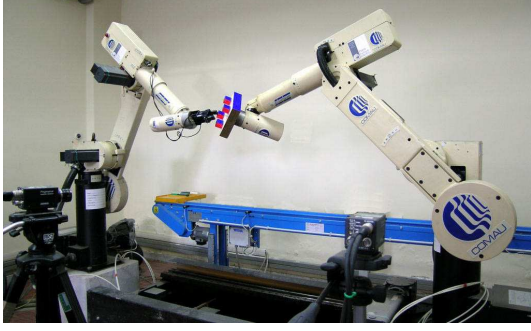


Fig. 5. Dual-arm industrial robotic cell.

transformation, which depends on the intrinsic camera parameters, it is possible to compute the projections of all the objects of the cell (each seen as a set of surfaces) on the image plane of all the cameras. On the basis of these data, the BSP tree structure representing the current geometric configuration of the cell, as it is seen by all the cameras, can be built.

The detection of the occluded parts of the workpieces with respect to a given camera can be achieved by implementing a suitable recursive visit of the corresponding BSP tree representation. This algorithm allows recognizing all the feature points lying on parts of the workpieces that are not occluded with respect to the camera. Notice that the whole occlusion detection algorithm exhibits a complexity $O(N)$, where N is the number of surfaces of the object (Paterson and Yao, 1990).

Details about the BSP tree construction and visit can be found in (Lippiello and Villani, 2003).

6. EXPERIMENT

An example of visual servoing task is considered for the dual-arm industrial robotic cell available at PRISMA Lab (see Fig. 5. A detailed description of the cell can be found, e.g., in (Lippiello *et al.*, 2005a). The two robots of the cell (named hereafter Robot 1 and Robot 2) execute independent tasks. Robot 1 performs a grasping task of a moving object localized using the visual system. For this experiment, an eye-to-hand configuration with two cameras is adopted. Robot 2 performs a different task and during motion it may occlude the object with respect to the cameras. The task assigned to Robot 1 includes the following phases:

- (1) *Approaching* – When the target object is localized, starting from the HOME pose, approach the grasp pose in two steps: first go over the target object (at 5 cm height), and then descend on it.
- (2) *Grasping* – Grasp the object and check the state of the gripper;
- (3) *Manipulating* – Return to the HOME pose carrying the object.

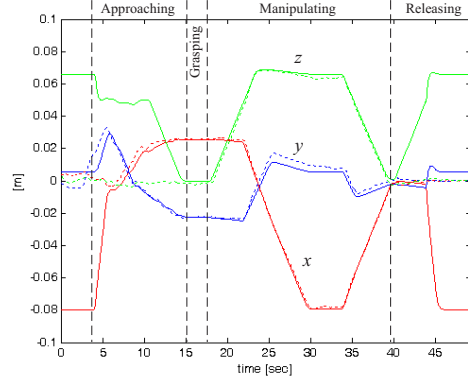


Fig. 6. Gripper (solid) and estimated object (dashed) trajectories.

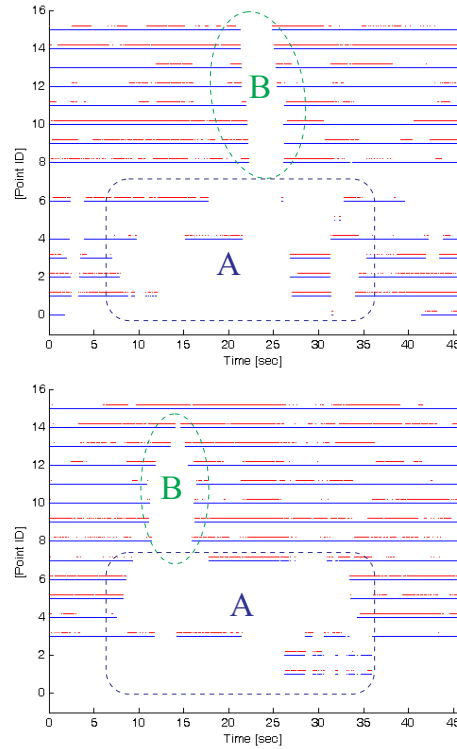


Fig. 7. Visible and selected object feature points for Camera 1 (top) and Camera 2 (bottom).

- (4) *Releasing* – Go to the FINAL pose and release the object.

The trajectory of the robot gripper and the estimated object position (the orientation is omitted for brevity) are shown in Fig. 6. The Approaching phase begins after about 4 sec and ends in about 11 sec. During this phase the robot recognizes the target object and moves over it, initially keeping a distance of about 5 cm along the vertical direction (z component); then the robot begins the descent to the grasping pose. Notice that the object is moving during this phase while the gripper motion in the horizontal plane (x and y components) matches the object motion. When the grasping

pose has been reached, the Grasping phase begins and ends after about 2.5 sec. During this time interval the pneumatic gripper is closed and a check of the state of the gripper is performed using the magnetic sensors. At about 17.5 sec the Manipulating phase begins and the robot return to the HOME pose carrying the object. At about 40 sec the robot reaches the FINAL pose and releases the object; then, it returns to HOME (Releasing phase).

In Fig. 7 the state of the visible and selected object feature points during the experiment are represented; the total number of image features (object corners) is 16. For each point, the bottom line indicates when it is visible, the top line indicates when it is selected for feature extraction. The maximum number of visible image features, for both cameras, is 32, but the selection algorithm, at each sampling time, chooses a subset of 11 optimal features to be extracted. The A-area corresponds to the occlusion caused by the gripper during the grasping, while the B-area corresponds to the occlusion caused by Robot 1. Notice that the motion of Robot 1 generates a partial occlusion (only on point remains visible) on the Camera 2 between the Approach and the Grasping phase. This event does not affect the accuracy of pose estimation and allows the successful execution of the Grasping phase. Moreover, during the Manipulating phase, Robot 1 occludes completely the object with respect to Camera 1. Again, the visual tracking algorithm maintains high accuracy, even though the estimated pose is not used after grasping (only the joint resolver measurements are used in the Manipulating and Releasing phases).

7. CONCLUSION

The problem of position-based visual servoing in a multi-arm robotic cell equipped with a hybrid eye-in-hand/eye-to-hand multi-camera system was considered in this paper. The estimation of the pose of the workpieces, based on the EKF, takes advantage of a computationally fast algorithm of the objects self occlusions and of the mutual occlusions caused by the robot links and tools. Experiments on a cell with two industrial robots performing a vision-guided grasping task have been presented, for the case of two eye-to-hand cameras. Further experiments with an hybrid eye-in-hand/eye-to-hand camera system are in progress. Some preliminary results can be found in (Lippiello *et al.*, 2005a).

REFERENCES

- Broida, T. and R. Chellappa (1986). Estimation of object motion parameters form noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(1), 90–99.
- Chiuso, A., P. Favaro, H. Jin and S. Soatto (2002). Structure from motion casually integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(4), 1–13.
- Espiau, B., F. Chaumette and P. Rives (1996). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation* **8**(3), 313–326.
- Flandin, G., F. Chaumette and E. Marchand (2000). Eye-in-hand/eye-to-hand cooperation for visual servoing. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*. Vol. 3. pp. 2741–2746.
- Hutchinson, S., G. D. Hager and P. I. Corke (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* **12**(5), 651–670.
- Lippiello, V. and L. Villani (2003). Managing redundant visual measurements for accurate pose tracking. *Robotica* **21**, 511–519.
- Lippiello, V., B. Siciliano and L. Villani (2005a). Eye-in-hand/eye-to-hand multi-camera visual servoing. In: *Proceedings of the 2005 IEEE International Conference on Decision and Control*. Seville, Spain.
- Lippiello, V., B. Siciliano and L. Villani (2005b). 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.
- Muis, A. and K. Ohnishi (2004). 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*. pp. 647–652.
- Paterson, M. and F. Yao (1990). Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete and Computational Geometry* **5**, 485–503.
- Sciavicco, L. and B. Siciliano (2000). *Modelling and Control of Robot Manipulators*. 2nd ed.. Springer-Verlag. London, UK.
- Wilson, W. J., C. C. W. Hulls and G. S. Bell (1996). Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation* **12**(5), 684–696.