

Robust Visual Tracking Using a Fixed Multi-camera System

Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani
 PRISMA Lab – Dipartimento di Informatica e Sistemistica
 Università degli Studi di Napoli Federico II
 Via Claudio 21, 80125 Napoli, Italy
 {vincenzo.lippiello,bruno.siciliano,luigi.villani}@unina.it

Abstract—The problem of tracking the position and orientation of a moving object using a stereo camera system is considered in this paper. A robust algorithm based on the extended Kalman filter is adopted, combined with an efficient selection technique of the object image features, based on Binary Space Partitioning tree geometric models. An experimental study is carried out using a vision system of two fixed cameras.

I. INTRODUCTION

A common problem in machine vision is that of tracking the 3D-pose (position and orientation) of moving objects in a wide workspace. Visual tracking applications require the adoption of motion estimation algorithms ensuring robustness in the presence of noise and stable tracking also in unstructured environments. Moreover, the information extracted from visual measurements must be available in real time.

A widely adopted approach to robust motion estimation from visual measurements is based on the extended Kalman filter, which represents a good trade-off between computational load and estimation accuracy [1], [2], [3]. When the object is moving in unstructured environments, good performance can be ensured using a multi-camera system, which guarantees information redundancy.

In this paper the extended Kalman filter is used to set up a visual tracking algorithm for a stereo camera system. The filter equations are derived according to the systematic procedure presented in [4], for the case of n video cameras fixed in the workspace. The algorithm requires the recognition of the object corners (feature points) from which the object pose can be computed using a simple point CAD model.

The estimation process is improved by adopting an efficient selection method of the feature points based on Binary Space Partitioning (BSP) trees [5]. The selection algorithm allows managing in real time a large amount of information provided by the stereo cameras system. Thus the information redundancy can be effectively exploited to improve accuracy and robustness of the visual tracking task.

Experimental tests are presented to show the computational feasibility and the effectiveness of the proposed approach.

II. EXTENDED KALMAN FILTER

Consider a system of n video cameras fixed in the workspace and a moving object. The geometry of the system is described in Fig. 1.

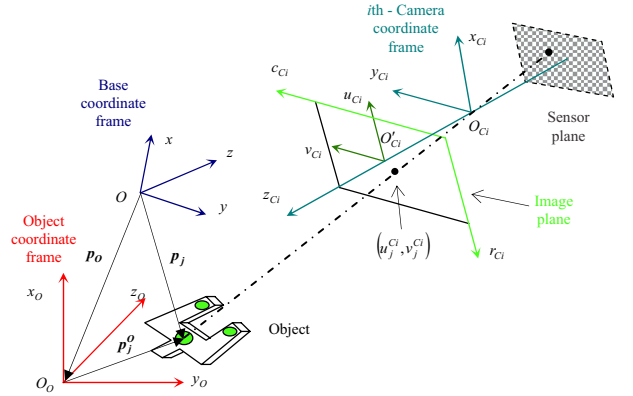


Fig. 1. Reference frames for the i -th camera and the object using the pinhole model.

A frame $O_{ci}-x_{ci}y_{ci}z_{ci}$ attached to the i -th camera (camera frame), with the z_{ci} -axis aligned to the optical axis and the origin in the optical center, is considered for each camera. The sensor plane is parallel to the $x_{ci}y_{ci}$ -plane at a distance $-f_e^{ci}$ along the z_{ci} -axis, where f_e^{ci} is the effective focal length of the camera lens. The image plane is parallel to the $x_{ci}y_{ci}$ -plane at a distance f_e^{ci} along the z_{ci} -axis. The intersection of the optical axis with the image plane defines the principal optic 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.

The position of the origin and the rotation matrix of the i -th camera frame with respect to the base frame are denoted by $\mathbf{o}_{ci}$ and $\mathbf{R}_{ci}$ respectively. These quantities are constant because the cameras are assumed to be fixed to the workspace, and can be computed through a suitable calibration procedure [6].

The position and orientation of the object with respect to the base frame can be specified by defining a frame $O_o-x_o y_o z_o$ attached to the object and considering the

coordinate vector of the origin $\mathbf{o}_o = [x_o \ y_o \ z_o]^T$ and the rotation matrix $\mathbf{R}_o(\boldsymbol{\varphi}_o)$, where $\boldsymbol{\varphi}_o = [\phi_o \ \alpha_o \ \psi_o]^T$ is the vector of the Roll, Pitch and Yaw angles. The components of the vectors $\mathbf{o}_o$ and $\boldsymbol{\varphi}_o$ are the six quantities to be estimated.

Consider m feature points on the object. It can be shown (see [4]) that the position of the j -th feature point in the i -th camera frame can be computed as

$$\mathbf{p}_j^{ci} = \mathbf{R}_{ci}^T(\mathbf{o}_o - \mathbf{o}_{ci} + \mathbf{R}_o(\boldsymbol{\varphi}_o)\mathbf{p}_j^o), \quad (1)$$

where $\mathbf{p}_j^o$ is the position vector of the j -th feature point with respect to the object frame. This vector is constant and is assumed to be known from the object CAD model.

By folding the m equations (1) into the perspective transformation of the n cameras, a system of $2mn$ nonlinear equations is achieved. These equations depend on the measurements of the m feature points in the image planes of the n cameras, while the six components of the vectors $\mathbf{o}_o$ and $\boldsymbol{\varphi}_o$ are the unknown variables.

To solve these equation in real-time, the extended Kalman filter is adopted, which provides a recursive solution.

In order to write the Kalman filter equations, a discrete-time dynamic model of the object motion has to be considered. Assuming that the object velocity is constant over one sample period T , the model can be written in the form

$$\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \boldsymbol{\gamma}_k \quad (2)$$

where $\mathbf{w} = [x_o \ \dot{x}_o \ y_o \ \dot{y}_o \ z_o \ \dot{z}_o \ \phi_o \ \dot{\phi}_o \ \alpha_o \ \dot{\alpha}_o \ \psi_o \ \dot{\psi}_o]^T$ is the state vector, $\boldsymbol{\gamma}$ is the dynamic modeling error, and $\mathbf{A}$ is a (12×12) block diagonal matrix of the form

$$\mathbf{A} = \text{diag} \left\{ \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right\}.$$

The output equation of Kalman filter is chosen as

$$\boldsymbol{\zeta}_k = \mathbf{g}(\mathbf{w}_k) + \boldsymbol{\nu}_k \quad (3)$$

with

$$\mathbf{g}(\mathbf{w}_k) = \begin{bmatrix} \frac{x_1^{c1}}{z_1^{c1}} & \frac{y_1^{c1}}{z_1^{c1}} & \dots & \frac{x_1^{cn}}{z_1^{cn}} & \frac{y_1^{cn}}{z_1^{cn}} \\ \dots & \frac{x_m^{c1}}{z_m^{c1}} & \frac{y_m^{c1}}{z_m^{c1}} & \dots & \frac{x_m^{cn}}{z_m^{cn}} & \frac{y_m^{cn}}{z_m^{cn}} \end{bmatrix}_k^T,$$

where the coordinates of the feature points $\mathbf{p}_j^{ci}$ are computed from the state vector $\mathbf{w}_k$ via (1). In the above equation, $\boldsymbol{\nu}_k$ is the measurement noise and the $\boldsymbol{\zeta}_k$ is the vector of the *normalized* coordinates of the m feature points in the image plane of the n cameras.

Since the output model is nonlinear in the system state, it is required to linearize the output equations about the current state estimate at each sample time, considering the so-called extended Kalman filter. The recursive form of the extended Kalman filter equations is reported in [4].

III. PRE-SELECTION ALGORITHM

The accuracy of the estimate provided by the Kalman filter depends on the number of the available feature points. Inclusion of extra points will improve the estimation accuracy but will increase the computational cost. It has been shown that a number of five or six feature points, if properly chosen, may represent a good trade-off [2]. Automatic selection algorithms have been developed to find the optimal feature points [7]. In order to increase the efficiency of the selection algorithms, it is advisable to perform a pre-selection of the points that are visible to the camera at a given sample time. The pre-selection technique proposed in this paper is based on Binary Space Partitioning (BSP) trees.

A BSP tree is a data structure representing a recursive and hierarchical partition of a n -dimensional space into convex subspaces. It can be effectively adopted to represent the 3D CAD geometry of an object [8].

In order to build the tree, each object has to be modelled as a set of planar *polygons*; this means that the curved surfaces have to be approximated. Each polygon is characterized by a set of *feature points* (the vertices of the polygon) and by the vector normal to the plane leaving from the object. For each node of the tree, a *partition plane*, characterized by its normal vector and by a point, is chosen according to a specific criterion; the node is defined as the set containing the partition plane and all the polygons lying on it.

Each node is the root of two subtrees: the *front* subtree corresponding to the subset of all the polygons lying entirely on the front side of the partition plane (i.e., the side corresponding to the half-space containing the normal vector), and the *back* subtree corresponding to the subset of all the polygons lying entirely on the back side of the partition plane.

The construction procedure can be applied recursively to the two subsets by choosing, for each node, a new partition plane among those corresponding to the polygons contained in that subtree. The construction ends when all the polygons are placed in a node of the tree.

Further details on BSP trees and an example of construction can be found in [9].

Once a BSP tree representation of an object is available, it is possible to select the feature points of the object visible from a given camera position and orientation, by implementing a suitable visit algorithm of the tree. The algorithm can be applied recursively to all the nodes of the tree, starting from the root node as showed in Fig. 2.

When the algorithm processes a node, the current set of projections of the visible feature points on the image plane is updated by adding all the projections of the feature points of the polygons of the current node and eliminating all the projections of the feature points that are hidden by the projections of the polygons of the current node.

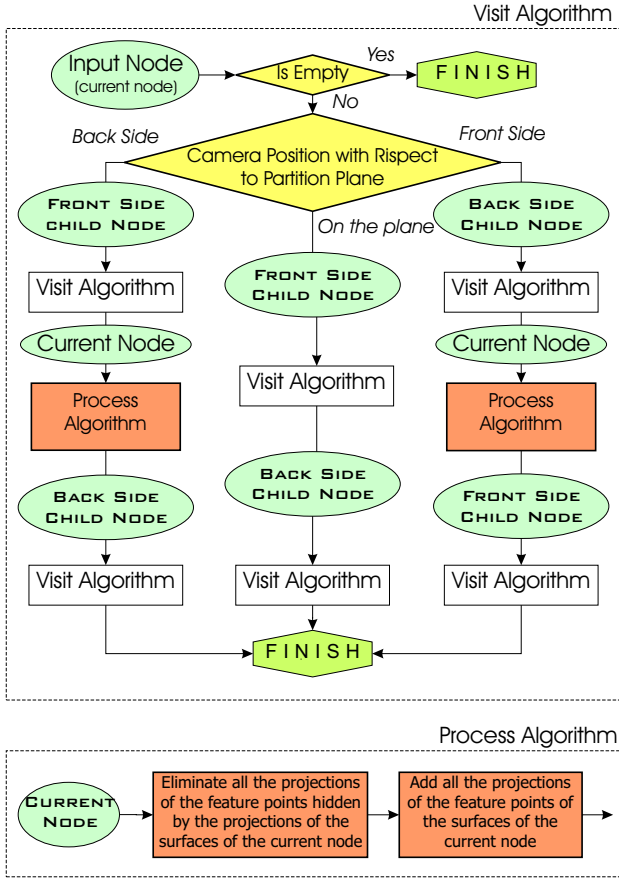


Fig. 2. Recursive visit algorithm of the BSP tree for the selection of visible feature points.

If a polygon is hidden from the camera (i.e., the angle between the normal vector to the polygon and the camera z -axis is not in the interval $]-\pi/2, \pi/2[$ or the polygon is behind the camera), the corresponding feature points are not added to the set.

At the end of the visit, the current set will contain all the projections of the feature points visible from the camera, while all the hidden feature points will be discarded. Notice that the visit algorithm updates the set by ordering the polygons with respect to the camera from the background to the foreground.

IV. SELECTION ALGORITHM

The pre-selection technique recognizes all the feature points that are visible from a camera view point. However, this does not ensure that all the visible points are “well” localizable, i.e., their positions can be effectively measured with a given accuracy. Moreover, the number of the well localizable feature points may be larger than the *optimal* number of points ensuring the best pose estimation accuracy.

A windowing test is adopted to select the projections of the feature points that can be well localized. In particular, only the points that can be centered into suitable rectangular windows of the image plane are considered for the next step of selection, while the points that are out of the field of view of the camera, and the points that are too close each other or to the boundaries of the image plane, are discarded.

The number of remaining feature points after pre-selection and windowing test is typically too high with respect to the minimum number sufficient to achieve the best Kalman filter precision. It has been demonstrated that an optimal set of five or six feature points guarantees about the same precision as that of the case when an higher number of feature points is considered [2], [3].

The optimality of a given set of feature points is valued through the composition of suitably selected quality indexes into an optimal cost function. The quality indexes must be able to provide accuracy, robustness and to minimize the oscillations in the pose estimation variables. To achieve this goal it is necessary to ensure an optimal spatial distribution of the projections of the feature points on the image plan and to avoid chattering events between different optimal subsets of feature points chosen during the object motion. Moreover, in order to exploit the potentialities of a multi-camera system, it is important to achieve an optimal distribution of the feature points among the different cameras.

Without loss of generality, the case of two identical cameras is considered.

A first quality index is the measure of spatial distribution of the predicted projections on the image planes of a subset of q_i selected points for the i -th camera, $i = 1, 2$:

$$Q_{si} = \frac{1}{q_i} \sum_{k=1}^{q_i} \min_{\substack{j \in \{1, \dots, q_i\} \\ j \neq k}} \|p_j - p_k\|.$$

Notice that $q = q_1 + q_2$ is chosen between 6 and 8 to prevent fault cases.

A second quality index is the measure of angular distribution of the predicted projections on the image planes of a subset of q_i selected points for the i -th camera, $i = 1, 2$:

$$Q_{ai} = 1 - \sum_{k=1}^{q_i} \left| \frac{\alpha_k}{2\pi} - \frac{1}{q_i} \right|$$

where α_k is the angle between the vector $p_{k+1} - p_{Ci}$ and the vector $p_k - p_{Ci}$, being p_{Ci} the central gravity point of the whole subset of feature points, and the q_i points of the subset are considered in a counter-clockwise ordered sequence with respect to p_{Ci} , with $p_{q_i+1} = p_1$.

In order to avoid chattering phenomena, the following quality index, which introduces hysteresis effects on the change of the optimal combination of points, is considered

for the i -th camera, $i = 1, 2$:

$$Q_{hi} = \begin{cases} 1 + \epsilon & \text{if actual combination = previous one} \\ 1 & \text{otherwise} \end{cases}$$

where ϵ is a positive constant.

In order to distribute the points among the two cameras, the following indexes are considered:

$$Q_e = 1 + \frac{2}{q} \left(\frac{2}{q} - 1 \right) \left| q_1 - \frac{q}{2} \right|$$

$$Q_d = \frac{q_1/d_1 + q_2/d_2}{q / \min\{d_1, d_2\}}$$

where q_i is the number of points assigned to the i -th camera, and d_i is the distance of the i -th camera from the object, $i = 1, 2$. The first index ensures an equal distribution of points among the cameras, while the second index takes into account the distance of the cameras from the object.

The proposed quality indexes represent only some of the possible choices, but guarantee satisfactory performance when used with the pre-selection method and the windowing test presented above, for the case of two fixed cameras. Other examples of quality indexes are presented, e.g., in [7], and some of them can be added to the indexes adopted in this paper.

The cost function is chosen as

$$Q = \frac{Q_e Q_d}{q} \left(q_1 Q_{s1} Q_{a1} Q_{h1} + q_2 Q_{s2} Q_{a2} Q_{h2} \right)$$

and must be evaluated for all the possible combinations of the visible points on q positions. In order to determine the optimal set at each sample time, the initial optimal combination of points is first evaluated off-line. Then, only the combinations that modify at most one point for camera with respect to the current optimal combination are tested on-line, thus achieving a considerable reduction of processing time.

V. ESTIMATION PROCEDURE

A functional chart of the estimation procedure is reported in Fig. 3. It is assumed that a BSP tree representation of the object is built off-line from the CAD model. A Kalman filter is used to estimate the corresponding pose with respect to the base frame at the next sample time. The feature points selection and windows placing operation can be detailed as follows. For each camera:

- **Step 1:** The visit algorithm described in the previous section is applied to the BSP tree of the object to find the set of all the feature points that are visible from the camera.
- **Step 2:** The resulting set of visible points is input to the algorithm for the selection of the optimal feature points.

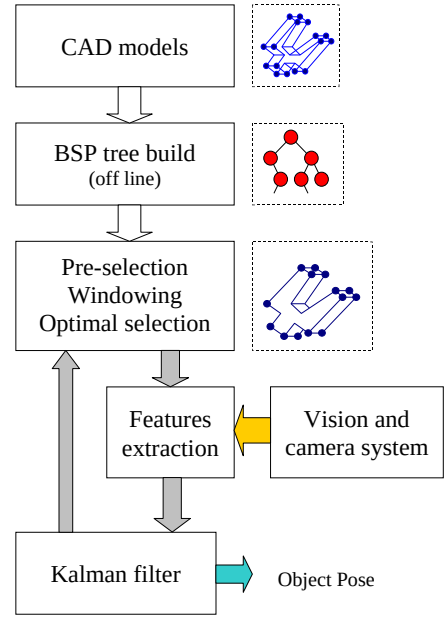


Fig. 3. Functional chart of the estimation procedure.

- **Step 3:** The location of the optimal feature points in the image plane at the next sample time is computed on the basis of the object pose estimation provided by the Kalman filter.
- **Step 4:** A dynamic windowing algorithm is executed to select the parts of the image plane to be input to the feature extraction algorithm.

At this point, all the image windows of the optimal selected points are elaborated using a feature extraction algorithm. The computed coordinates of the points in the image plane are input to the Kalman filter which provides the estimate of the actual object pose and the predicted pose at the next sample time used by the pre-selection algorithm.

Notice that the procedure described above can be extended to the case of multiple objects moving among obstacles of known geometry [9]; if the obstacles are moving with respect to the base frame, the corresponding motion variables can be estimated using Kalman filters.

VI. EXPERIMENTS

The experimental set-up is composed by a PC with Pentium IV 1.7GHz processor equipped with two MATROX Genesis boards, two SONY 8500CE B/W cameras, and a COMAU Smart3-S robot. The MATROX boards are used as frame grabbers and for a partial image processing (e.g., windows extraction from the image). The PC host is also used to realize the whole BSP structures management, the pre-selection algorithm, windows processing, the selection algorithm and the Kalman filtering. Some steps of image

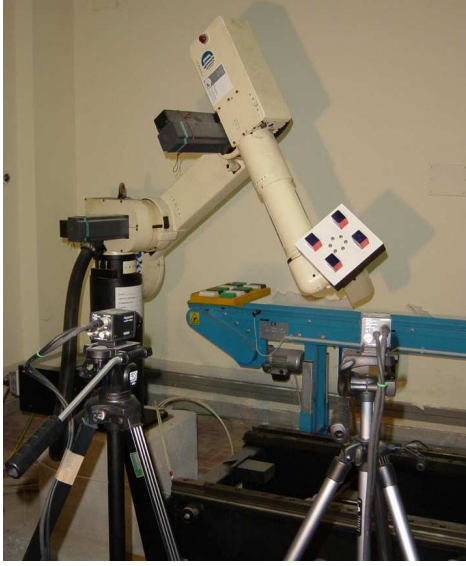


Fig. 4. COMAU Smart3-S robot and SONY 8500CE cameras.

processing have been parallelized on the MATROX boards and on the PC, so as to reduce computational time. The robot is used to move an object in the visual space of the cameras; thus the object position and orientation with respect to the base frame of the robot can be computed from joint position measurements via the direct kinematic equation. In order to test the accuracy of the estimation provided by the Kalman filter, the cameras were calibrated with respect to the base frame of the robot using the calibration procedure presented in [6], where the robot is exploited to place a calibration pattern in some known pose of the visible space of the camera. The cameras resolution is 576×763 pixels and the nominal focal length of the lenses is 16 mm. The two cameras are disposed at a distance of about 150 cm from the object with an angle of about $\pi/2$ between the z_c axes. The sampling time used for estimation is limited by the camera frame rate, which is about 26 fps. No particular illumination equipment has been used, in order to test the robustness of the setup in the case of noisy visual measurements.

All the algorithms for BSP structure management, image processing and pose estimation have been implemented in ANSI C. The image features are the corners of the object, which can be extracted with high robustness in various environmental conditions. The feature extraction algorithm is based on Canny's method for edge detection and on a simple custom implementation of a corner detector [10]. In particular, to locate the position of a corner in a small window, all the straight segments are searched first, using an LSQ interpolator algorithm; then all the intersection points of these segments into the window are evaluated. The intersection points closer than a given

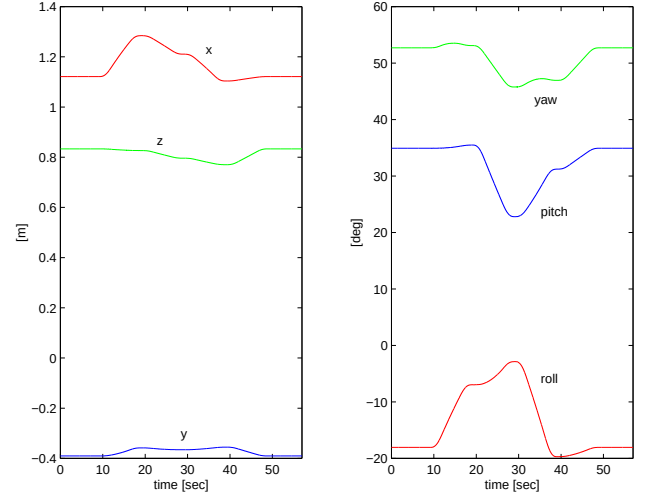


Fig. 5. Object trajectory. Left: Position trajectory. Right: Orientation trajectory

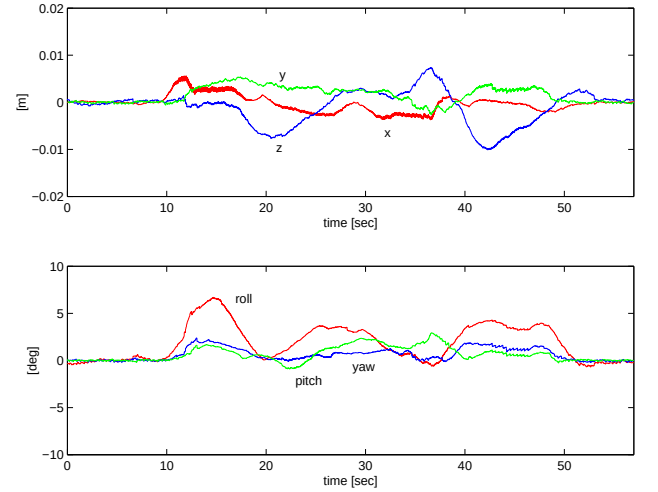


Fig. 6. Time history of the estimation errors. Top: Position errors. Bottom: Orientation errors

threshold are considered as a unique average corner, due to the image noise. All the corners that are at a distance from the center of the window (which corresponds to the position of the corner so as predicted by the Kalman filter) greater than a maximum distance, are considered as fault measurements and are discarded. The maximum distance corresponds to the variance of the distance between the measured corner positions and those predicted by the Kalman filter.

The object used in the experiment has 40 vertices, which are all used as feature points. Fig. 4 shows the stereo vision system and the robot carrying the object.

The time history of the trajectory used for the experiment is represented in Fig. 5. The maximum linear

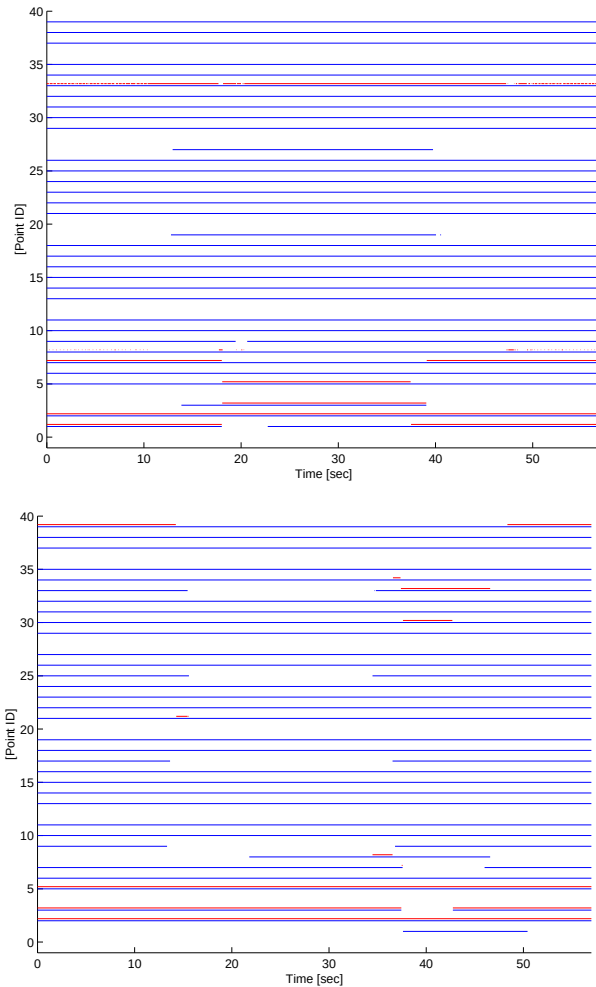


Fig. 7. Visible and selected points for camera 1 (top) and camera 2 (bottom). For each point: the bottom line indicates when it is visible; the top line indicates when it is selected for feature extraction.

velocity and angular velocity are about 3 cm/s and 3 deg/s respectively. The time history of the estimation errors is shown in Fig. 6. Noticeably, the accuracy of the system reaches the limit allowed by cameras calibration, for all the components of the motion. As it was expected, the errors for the motion components are of the same order of magnitude, thanks to the use of a stereo camera system.

In Fig. 7 the output of the whole selection algorithm, for the two cameras, is reported. For each of the 40 feature points, two horizontal lines are considered: a point of the bottom line indicates that the feature point was classified as visible by the pre-selection algorithm at a particular sample time; a point of the top line indicates that the visible feature point was chosen by the selection algorithm. Notice that 8 feature points are selected at each sample time in order to guarantee at least five or six measurements in the case of fault of the extraction algorithm for some of the points. Remarkably, 4 feature points for camera are chosen at each sampling time,

coherently with the symmetric disposition of the cameras with respect to the object.

VII. CONCLUSION

The problem of estimating the pose (position and orientation) of a moving object from visual measurements has been considered in this paper. The extended Kalman filter has been used to recursively compute an estimate of the motion variables from the measurements of the position of suitable feature points of the object. The efficiency of the algorithm has been improved by adopting a technique of selection of the visible feature points at each sample time based on a Binary Space Partition tree representation of the object geometry. The experiments on a stereo camera system have shown the effectiveness of the algorithm and have confirmed its practical feasibility.

Acknowledgments

This work was supported by MIUR and ASI.

VIII. REFERENCES

- [1] S. Lee and Y. Kay, "An accurate estimation of 3-D position and orientation of a moving object for robot stereo vision: Kalman filter approach," *1990 IEEE Int. Conf. on Robotics and Automation*, pp. 414–419, 1990.
- [2] J. Wang and J.W. Wilson, "3D relative position and orientation estimation using Kalman filter for robot control," *1992 IEEE Int. Conf. on Robotics and Automation*, pp. 2638–2645, 1992.
- [3] J.W. Wilson, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 684–696, 1996.
- [4] V. Lippiello, B. Siciliano, and L. Villani, "Position and orientation estimation using Kalman filtering of stereo images," *2001 IEEE Int. Conf. on Control Applications*, pp. 702–707, 2001.
- [5] V. Lippiello, B. Siciliano, and L. Villani, "Objects motion estimation via BSP tree modeling and Kalman filtering of stereo images," *2002 IEEE Int. Conf. on Robotics and Automation*, pp. 2969–2973, 2002.
- [6] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 965–980, 1992.
- [7] F. Janabi-Sharifi and J.W. Wilson, "Automatic selection of image features for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 890–903, 1997.
- [8] M. Paterson and F. Yao, "Efficient binary space partitions for hidden-surface removal and solid modeling," *Discrete and Computational Geometry*, vol. 5, pp. 485–503, 1990.
- [9] V. Lippiello, B. Siciliano, and L. Villani, "A new method of image features pre-selection for real-time pose estimation based on Kalman filter," *2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 372–377, 2002.
- [10] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.