

# A NEW TECHNIQUE FOR REAL-TIME VISUAL TRACKING OF 3D OBJECTS

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,siciliano,lvillani}@unina.it*

**Abstract:** A new technique for real-time estimation of the pose of a moving object using a video camera is presented. A Binary Space Partition (BSP) tree is adopted to represent the object geometry and efficient selection techniques are developed to find an optimal set of feature points of the object to be used for feature extraction. The extended Kalman filter is used to estimate the pose of the object directly from the image measurements. The performance of the proposed approach is experimentally tested in the case of an object moving in the visible space of one fixed camera.

**Keywords:** Visual tracking; pose estimation; Kalman filter; feature selection.

## 1. INTRODUCTION

The use of visual sensors may have high impact in applications where it is required to measure the pose (position and orientation) and the visual features of objects moving in unstructured environments. In robotics, the measurements provided by video cameras can be directly used to perform closed-loop position control of the robot end effector, usually denoted as visual servoing control (Hutchinson *et al.*, 1996).

A strict requirement in visual servoing applications is that the information extracted from visual measurements must be available at control sampling rate. Therefore, computationally efficient visual tracking techniques must be adopted. Moreover, noise and disturbances affecting visual measurements may produce large estimation errors. The adoption of the extended Kalman filter for pose estimation represents a good trade-off between computational load and estimation accuracy (Lee and Kay, 1990; Lippiello *et al.*, 2001). The object pose can be estimated by recognizing some geometric features of the object, such edges and corners, from a camera image. In particular, the extraction of a suitable number of corners (feature points) allows the object pose to be computed using a simple point CAD model (Wilson,

1996). In principle, the accuracy of the estimate increases with the number of the available feature points, at the expense of the computation time. However, when Kalman filter is adopted, it has been shown that the best achievable accuracy obtained using all the available points is quite the same as that obtained using a number of five or six feature points, if properly chosen (Wang and Wilson, 1992). Selection algorithms have been developed to find the optimal feature points (Janabi-Sharifi and Wilson, 1997). It should be pointed out, however, that the complexity of the selection algorithms grows at factorial rate. Hence, when the number of feature points is large, it is crucial to perform a pre-selection of the points, e.g., by eliminating those that are occluded with respect to the camera.

In this paper, the extended Kalman filter is adopted for real-time pose estimation. In order to reduce computational time, a new pre-selection algorithm of the feature points is proposed, based on the selection of all the points that are visible to the camera at a given sample time. This algorithm exhibits a complexity which grows linearly, thanks to the use of Binary Space Partitioning (BSP) tree for object geometric representation (Drummond and Cipolla, 1999). After the pre-selection,

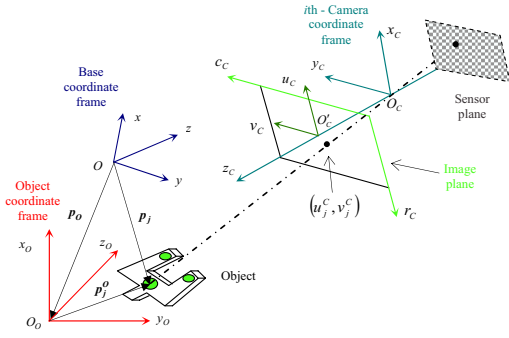


Fig. 1. Reference frames for the camera and the object using the pinhole model.

a dynamic windowing algorithm and an optimal point selection algorithm are adopted to find the windows of the image plane to be processed and input to the Kalman filter. The effectiveness of the proposed approach is tested in experimental case studies where the position and orientation of an object carried by a robot manipulator is tracked using one fixed camera.

## 2. EXTENDED KALMAN FILTER

Consider a video camera fixed with respect to a base coordinate frame  $O-xyz$  (see Fig. 1). Let  $O_c-x_c y_c z_c$  be the camera frame, with the  $z_c$ -axis aligned to the optical axis and the origin in the optical center. In the following, a superscript will be used to denote the reference frame of a variable, when different from the base frame.

The sensor plane is parallel to the  $x_c y_c$ -plane at a distance  $-f_c^e$  along the  $z_c$ -axis, where  $f_c^e$  is the effective focal length of the camera lens. The image plane is parallel to the  $x_c y_c$ -plane at a distance  $f_c^e$  along the  $z_c$ -axis. The intersection of the optical axis with the image plane defines the principal optic point  $O'_c$ , which is the origin of the image frame  $O'_c-u_c v_c$  whose axes  $u_c$  and  $v_c$  are taken parallel to the axes  $x_c$  and  $y_c$ , respectively. The position of the origin and the rotation matrix of the camera frame with respect to the base frame are denoted by  $\mathbf{o}_c$  and  $\mathbf{R}_c$  respectively. These quantities are constant because the camera is fixed to the workspace, and can be computed through a suitable calibration procedure.

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 unknown quantities to be estimated.

Consider  $m$  feature points on the object. It can be shown (see Lippiello *et al.*, 2001) that the position of the  $j$ -th feature point in the camera frame can be computed as

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

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 camera, a system of  $2m$  nonlinear equations is achieved. These equations depend on the measurements of the  $m$  feature points in the image plane of the camera, 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  $(12 \times 12)$  block diagonal matrix

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

The output equation of Kalman filter is chosen as

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

with

$$\mathbf{g}(\mathbf{w}_k) = \begin{bmatrix} \frac{x_1^c}{z_1^c} & \frac{y_1^c}{z_1^c} & \dots & \frac{x_m^c}{z_m^c} & \frac{y_m^c}{z_m^c} \end{bmatrix} \quad (4)$$

where the coordinates of the feature points  $\mathbf{p}_j^c$  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.

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 filter equations is reported in (Lippiello *et al.*, 2001).

## 3. 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. 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 adopted to represent the 3D geometry of an object, as reported in (Paterson and Yao, 1990).

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 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 (Lippiello *et al.*, 2002a).

Once a BSP tree representation of an object is available, it is possible to select the feature points of the objects 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.

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 points are not added to the set.

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

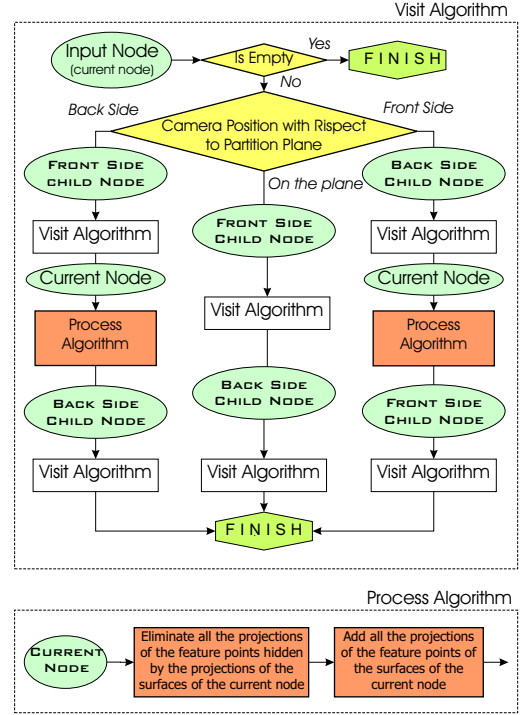


Fig. 2. Recursive visit algorithm of the BSP tree for the selection of visible feature points.

camera point of view from the background to the foreground.

#### 4. SELECTION ALGORITHM

The pre-selection technique recognizes all the feature points that are visible from the camera. 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.

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 the well localizable feature points may be larger than the optimal number of points (five or six) ensuring the best pose estimation accuracy (Wang and Wilson, 1992). The optimality of a given set of feature points can be 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.

A first quality index is the measure of spatial distribution of the predicted projections on the image planes of a subset of  $n$  selected points:

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

A second quality index is the measure of angular distribution of the predicted projections on the image plane of a subset of  $n$  selected points:

$$Q_a = 1 - \sum_{k=1}^n \left| \frac{\alpha_k}{2\pi} - \frac{1}{n} \right|$$

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

To avoid chattering phenomena, a quality index introducing hysteresis effects on the change of the optimal combination of points is considered:

$$Q_h = 1 + \epsilon$$

with  $\epsilon > 0$  if the actual combination is equal to the previous one,  $\epsilon = 0$  otherwise.

The proposed indexes are only some of the possible choices, but guarantee satisfactory performance when used with the pre-selection method and the windowing test presented in this paper. Other examples of quality indexes are proposed, e.g., in (Janabi-Sharifi and Wilson, 1997).

The cost function is a simple product of the quality indexes, but must be evaluated for all the possible combinations of the visible points on  $n$  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 with respect to the current optimal combination are tested on-line, thus achieving a considerable reduction of processing time.

## 5. 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.

- **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.

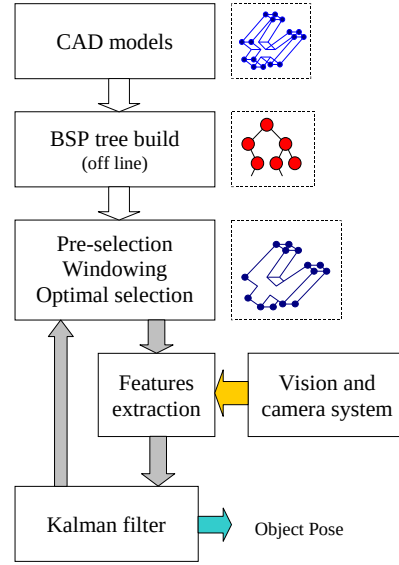


Fig. 3. Functional chart of the estimation procedure.

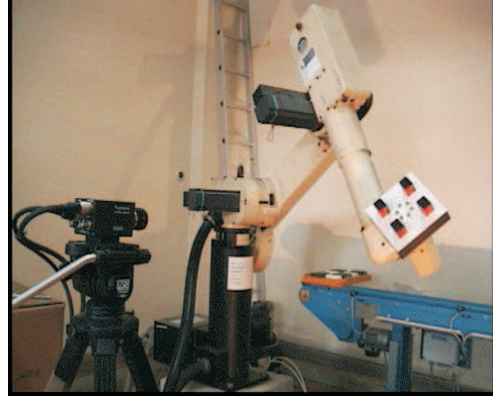


Fig. 4. Robot COMAU SMART3-S and camera SONY 8500CE.

- **Step 3:** The location of the optimal feature points in the image plane at the next sample time is computed from the object pose estimation provided by the Kalman filter.
- **Step 4:** A dynamic windowing algorithm selects 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 pose at the next sample time used into the steps 1 and 3.

The procedure described above can be extended to the case of multiple objects moving among obstacles of known geometry (Lippiello *et al.*, 2002b).

## 6. EXPERIMENTS

The experimental set-up is composed by a PC with Pentium IV 1.7GHz processor equipped with

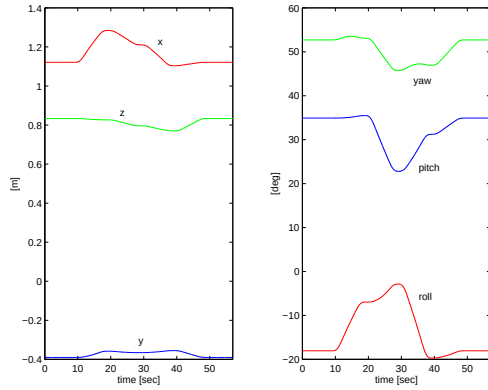


Fig. 5. Object trajectory in the first experiment. Left: Position trajectory. Right: Orientation trajectory.

a MATROX Genesis board, a SONY 8500CE B/W camera, and a COMAU SMART3-S robot. The MATROX board is used as frame grabber and for a partial image processing (e.g., windows extraction from the image). The PC host is also used to realize the whole BSP structure management, the pre-selection algorithm, the windows processing, the selection algorithm and the Kalman filtering. Some steps of image processing have been parallelized on the MATROX board and on the PC, so as to reduce computational time. The robot is used to move an object in the visual space of the camera; 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. The camera resolution is  $576 \times 763$  pixels and the nominal focal length of the lens is 16 mm. 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.

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. The object used in the experiment has 40 vertices, which are all used as feature point. In Fig. 4 is shown the robot carrying the object.

Two different experiments have been realized. The first experiment reflects a favorable situation where the object moves in the visible space of the camera and most of the feature points that are visible at the initial time remain visible during all the motion. The second experiment reflects an unfortunate situation where the set of the visible points is very variable, and a large part of the object goes out of the visible space of the camera during the motion.

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

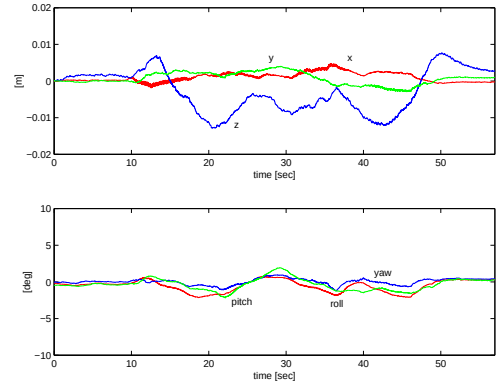


Fig. 6. Estimation errors in the first experiment. Top: Position errors. Bottom: Orientation errors.

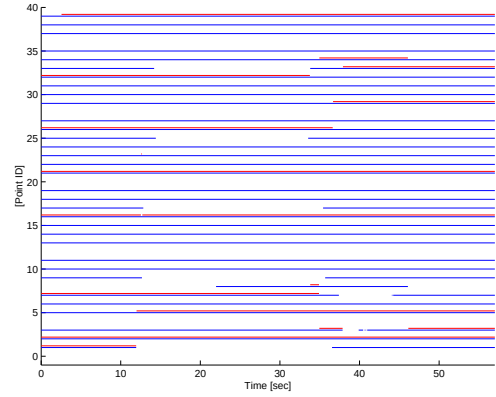


Fig. 7. Visible points and selected points in the first experiment.

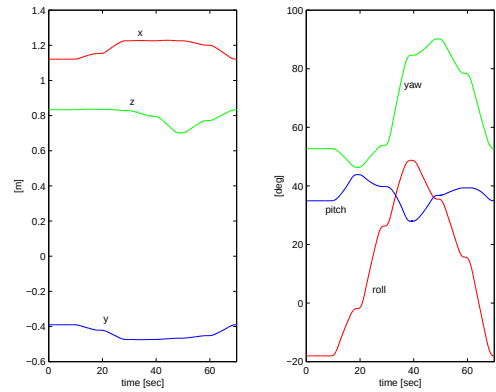


Fig. 8. Object trajectory in the second experiment. Left: Position trajectory. Right: Orientation trajectory.

linear velocity is about 3 cm/s and the maximum angular velocity is about 3 deg/s.

The time history of the estimation errors is shown in Fig. 6. As it was expected, the errors for some motion components are larger than others because only 2D information is available in a single camera system.

In Fig. 7 the output of the whole selection algorithm is reported. For each of the 40 feature points, two horizontal lines are considered: a point

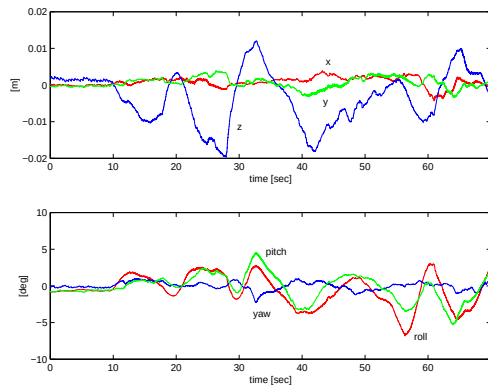


Fig. 9. Estimation errors in the second experiment. Top: Position errors. Bottom: Orientation errors.

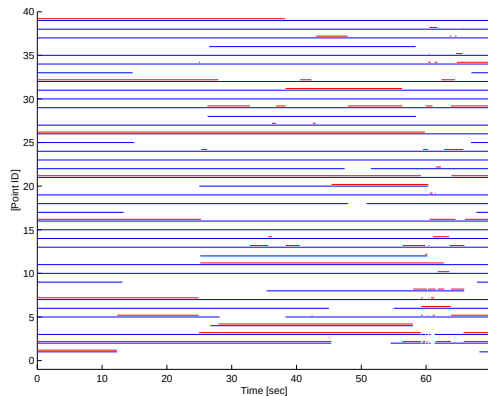


Fig. 10. Visible and selected points for the second experiment.

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. Also, some feature points are hidden during all the motion, while others are only visible over some time intervals. Finally, no chattering phenomena are present.

The time history of the trajectory used for the second experiment is represented in Fig. 8. The maximum linear velocity is about 2 cm/s and the maximum angular velocity is about 7 deg/s.

The time history of the estimation error is shown in Fig. 9. It can be observed that the error remains low but is greater than the estimation error of the previous experiment. This is due to the increased velocity of the feature points and to the fact that from about  $t = 30$  s to  $t = 60$  s the object moves so that it is partially out of the visible space of the camera; also, it rotates in such a way that a side remains almost parallel to the image plane. This condition penalizes the estimation accuracy and explains why the magnitude of the estimation

error components is greater than in the previous experiment. The corresponding output of the pre-selection and selection algorithms are reported in Fig. 10. It should be pointed out that the pre-selection and selection algorithm are able to provide the optimal set of points independently from the operating condition, although slight chattering phenomena appear in some situation where the elements in the set of localizable points is rapidly changing.

## 7. CONCLUSION

The estimation of the pose of a moving object from visual measurements was considered. The extended Kalman filter was 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 was improved by adopting a technique of pre-selection of the visible feature points at each sample time based on a Binary Space Partition tree representing the object geometry. The experiments have shown the effectiveness of the algorithm and have confirmed its practical feasibility.

## Acknowledgments

This work was supported by MIUR and ASI.

## References

- T.W. Drummond and R. Cipolla, "Real-time tracking of complex structures with on-line camera calibration," *British Machine Vision Conf.*, pp. 574–583, 1999.
- S. Hutchinson, G.D. Hager, and P.I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 651–670, 1996.
- F. Janabi-Sharifi and W.J. Wilson, "Automatic selection of image features for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 890–903, 1997.
- 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.
- V. Lippiello, B. Siciliano, and L. Villani, "Position and orientation estimation based on Kalman filtering of stereo images," *2001 IEEE Int. Conf. on Control Applications*, pp. 702–707, 2001.
- V. Lippiello, B. Siciliano, and L. Villani, "3-D Objects Motion Estimation Based on Kalman Filter and BSP Tree Models for Robot Stereo Vision," *Archives of Control Sciences*, in press, 2002a.
- 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. 2968–2973, 2002b.
- 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.
- 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.
- 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.