

A New Method of Image Features Pre-Selection for Real-Time Pose Estimation Based on Kalman Filter

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

The problem of real-time pose estimation of moving objects using a stereo video camera system is considered in this paper. A computationally efficient algorithm is proposed based on Kalman filtering of the position measurements of suitable feature points selected on the target objects. The efficiency of the algorithm is improved by adopting a new pre-selection technique of the feature points, based on Binary Space Partition (BSP) trees, which takes advantage of the Kalman filter prediction capability. Computer simulations are presented.

1 Introduction

The effectiveness and autonomy of a robotic system operating in unstructured environments can be enhanced if a vision system based on one or more cameras is used to achieve direct measurements of the state of the environment and of the task in progress. Visual measurements can be directly used to perform closed-loop position control of the robot end-effector, usually denoted *visual servoing* control.

One fundamental issue of position-based visual servoing is the real-time estimation of the pose, i.e., the position and orientation trajectory, of known target objects [1]. Typically, a visual system based on one or more cameras is used to measure the position of suitable feature points selected on the target objects.

In order to improve estimation accuracy in spite of noise and disturbances affecting visual measurements, the extended Kalman filter represents an effective solution [1, 2]. In fact, Kalman filter offers many advantages over other methods employed to solve the photogrammetric equations, e.g., temporal filtering, recursive implementation, ability to change the measurement set during the operation. Moreover, its prediction capability allows setting up a dynamic windowing technique of the image plane of

the cameras which may sensibly reduce image processing time. Thanks to these features, Kalman filter is adopted also for tracking objects with complex geometry and many internal degrees of freedom [3, 4].

It should be pointed out, however, that Kalman filter usually requires high computation time which increases with the number of feature points. The use of algorithms for real-time selection of an optimal subset of points is useful to reduce the computation burden but may be inadequate in case of target objects with a large number of feature points [5].

In this paper, the extended Kalman filter is adopted to estimate the pose of moving objects using a stereo camera system. The systematic procedure presented in [6] is used to compute the objects pose, combined with an efficient method of pre-selection of the visible feature points [7]. This work is mainly focused on the pre-selection algorithm, based on Binary Space Partition (BSP) trees for real-time rendering of CAD models of moving objects [8]. In detail, the prediction of the objects pose provided by the Kalman filter is used to guide a visit of the BSP tree which allows identifying all the feature points visible at the next sample time. Then a dynamic windowing algorithm and an optimal point selection algorithm are used to find the image plane windows to be processed. The proposed method can be applied also to the case of objects and obstacles with interposing parts; differently from other algorithms (see [5] and references therein), it allows recognizing all the points of the objects which are hidden to the camera or occluded by some other objects or obstacles of known geometry. Complex situations can be handled in real-time thanks to the computational low cost of BSP tree visit algorithms, whose complexity grows linearly with the number of feature points.

The effectiveness of the proposed approach is tested in a simulation case study. Ongoing experiments have shown the practical feasibility of the method.

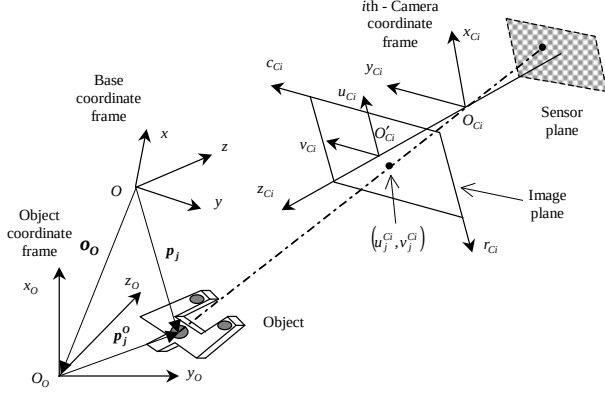


Figure 1: Camera and object frames.

2 Extended Kalman filter

Consider a system of n fixed video cameras. 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 (see Fig. 1). 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 parallel to the axes x_{ci} and y_{ci} respectively.

Without loss of generality, the case of a single moving target 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 $\mathbf{R}_o(\phi_o)$, where $\phi_o = [\varphi_o \ \vartheta_o \ \psi_o]^T$ is the vector of the Roll, Pitch and Yaw angles. The components of the vectors $\mathbf{o}_o$ and ϕ_o are the six unknown quantities to be estimated.

Consider m feature points of the object. The coordinate vector $\mathbf{p}_j = [x_j^{ci} \ y_j^{ci} \ z_j^{ci}]^T$ of the feature point P_j ($j = 1, \dots, m$) in the i -th camera frame ($i = 1, \dots, n$) can be computed as

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

where $\mathbf{p}_j^o$ is the coordinate vector of P_j expressed in the object frame, $\mathbf{o}_{ci}$ and $\mathbf{R}_{ci}$ are respectively the position vector and the rotation matrix of the i -th camera frame referred to the base frame. All those quantities are constant and are assumed to be known.

The point P_j is projected onto the point of the image

plane of coordinates

$$\begin{bmatrix} u_j^{ci} \\ v_j^{ci} \end{bmatrix} = \frac{f_e^{ci}}{z_j^{ci}} \begin{bmatrix} x_j^{ci} \\ y_j^{ci} \end{bmatrix}, \quad (2)$$

which can be expressed in terms of number of pixels as

$$\begin{bmatrix} r_j^{ci} \\ c_j^{ci} \end{bmatrix} = \begin{bmatrix} r_0^{ci} \\ c_0^{ci} \end{bmatrix} + \begin{bmatrix} s_u^{ci} & 0 \\ 0 & s_v^{ci} \end{bmatrix} \begin{bmatrix} u_j^{ci} \\ v_j^{ci} \end{bmatrix} \quad (3)$$

being $[r_0^{ci} \ c_0^{ci}]^T$ the coordinates of the point O'_{ci} , whereas s_u^{ci} (s_v^{ci}) is the row (column) scaling factor.

By folding the mn equations (1) into (2) and (3), a system of $2mn$ nonlinear equations is achieved, which depend on the measurements of the position of the m feature points in the image plane of the n cameras, whereas the six components of the vectors $\mathbf{o}_o$ and ϕ_o are the unknown variables.

An implicit solution to the system can be computed using the extended Kalman filter in discrete time, based on a suitable model of the object motion. A simplified kinematic model can be written in terms of the state vector $\mathbf{w} = [x_o \ \dot{x}_o \ y_o \ \dot{y}_o \ z_o \ \dot{z}_o \ \phi_o \ \dot{\phi}_o \ \alpha_o \ \dot{\alpha}_o \ \psi_o \ \dot{\psi}_o]^T$ in the form

$$\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \gamma_k \quad (4)$$

where $\mathbf{A}$ is a (12×12) block diagonal matrix

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

and γ_k is the modelling error. The model (4) has been derived in the reasonable hypothesis that the velocity is constant over one sample period T , for small T . The output of the Kalman filter is the vector of the *normalized* coordinates of the m points

$$\zeta_k = \begin{bmatrix} \frac{u_1^{c1}}{f_e^{c1}} \frac{v_1^{c1}}{f_e^{c1}} \cdots \frac{u_1^{cn}}{f_e^{cn}} \frac{v_1^{cn}}{f_e^{cn}} \\ \cdots \frac{u_m^{c1}}{f_e^{c1}} \frac{v_m^{c1}}{f_e^{c1}} \cdots \frac{u_m^{cn}}{f_e^{cn}} \frac{v_m^{cn}}{f_e^{cn}} \end{bmatrix}_k^T.$$

In view of (2), the corresponding output model can be written in the form

$$\zeta_k = \mathbf{g}(\mathbf{w}_k) + \nu_k \quad (5)$$

where ν_k is the measurement noise, and $\mathbf{g}(\mathbf{w}_k)$ can be computed from the state vector $\mathbf{w}_k$ via (1).

Since the output model is nonlinear, it is required to linearize the output equations about the current state estimate. This leads to the so-called extended Kalman filter, whose equations are not reported here for brevity and can be found in [6]. Notice that, in the case of multiple target objects, an extended Kalman filter for each object has to be considered.

3 Feature points selection

The accuracy of the estimate provided by the Kalman filter depends on the number of the available feature points. Inclusion of extra points improves the estimation accuracy but increases 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]; to this purpose, automatic selection algorithms have been developed [5]. It should be pointed out, however, that the complexity of the selection algorithms grows at factorial rate. Hence, in case of objects with a large number of feature points, it is crucial to perform a pre-selection of the points. In this paper a computationally efficient pre-selection method is proposed, which allows identifying all the points that are visible to the camera at a given sample time. The method, based on Binary Space Partition (BSP) trees, exhibits a computational complexity which grows linearly.

3.1 BSP tree geometric modelling

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 a set of objects as reported in [9].

To build the tree, each object has to be modelled as a set of planar *polygons*. 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* is chosen, characterized by a vector normal to the plane; the node is defined as the set containing the partition plane and all the polygons on it. The choice of the partition planes depends on how the tree will be used. For the purpose of removing the hidden surfaces, it is appropriate to choose the partition plane from the input set of polygons.

The node is the root of the *front* subtree, corresponding to the subset of all the polygons lying entirely to the front side of the partition plane (i.e., the side containing the normal vector), and of the *back* subtree, corresponding to the subset of all the polygons lying entirely to 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. If a polygon happens to span the partition plane, it can be split into two or more pieces and the resulting parts are added to the corresponding subsets. The construction ends when all the polygons and their parts are placed in a node of the tree.

As an example, consider the object and the BSP tree represented in Fig. 2. A partition plane is repre-

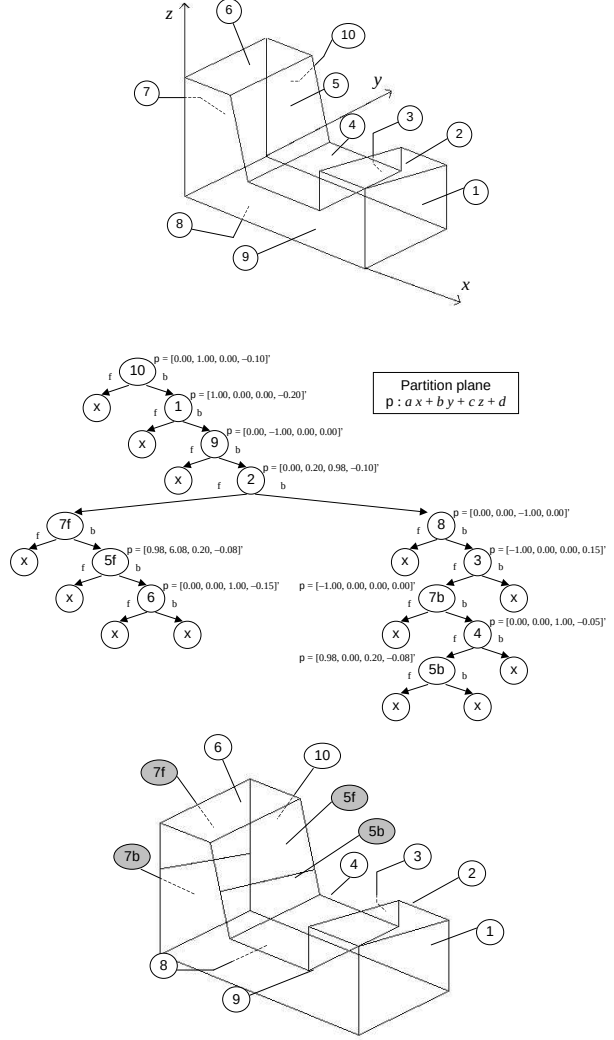


Figure 2: Object and BSP tree.

sented by the vector $\pi = [a \ b \ c \ d]^T$ of the coefficients of the equation of the plane with respect to a base reference frame, $ax + by + cz + d = 0$, where $\mathbf{n} = [a \ b \ c]^T$ is the unit vector normal to the plane. The root of the tree contains the polygon number 10, which is on the first partition plane; the front subtree is empty while the back subtree contains all the remaining polygons. The partition plane of the back subtree contains the polygon number 1; the front subtree is empty while the back subtree contains the polygons from number 2 to number 9. The construction ends when all the polygons are added to the nodes of the tree. Remarkably, the partition plane containing the polygon number 2 cuts polygons number 5 and 7 (notice that polygons number 9 and 10, which also span the partition plane, were already added to previous nodes of the tree), hence they have been split into two pieces each (see polygons number 5f, 5b, 7f, 7b).

3.2 Pre-selection algorithm

Once that a BSP tree representation of an object is available, it is possible to select the feature points of the object that can be 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, by updating a current set of visible feature points as follows.

For the current node, classify the camera position with respect to the current partition plane: **Front** side, **Back** side, **On** the plane. Hence:

- **Front:** Visit the back subtree; process the node; visit the front subtree.
- **Back:** Visit the front subtree; process the node; visit the back subtree.
- **On:** Visit the front subtree; visit the back subtree.

When the algorithm processes a node, the current set of visible feature points is updated by discarding the points of the set that are hidden by the polygons of the current node and by adding the points of the polygons of the current node. If a polygon is hidden to the camera (i.e., the angle between the polygon normal vector and the camera z -axis is not in the interval $]-\pi/2, \pi/2[$), the corresponding feature 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 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.

With reference to the BSP tree of Fig. 2, assuming that the camera is placed as the observer of the image, the sequence of the processed nodes is: 10, 8, 7b, 4, 5b, 3, 2, 7f, 6, 5f, 9, 1, where the polygons number 10, 8, 7b, 3, 7f result to be hidden.

The technique described above can be exploited to set up a real-time pre-selection algorithm of the feature points to be localized on the cameras image planes, using the prediction of the estimated pose of the target objects provided by the Kalman filter.

4 Estimation procedure

Two different situations must be considered: the case of objects whose parts cannot be interposed, and the case of objects with interposing parts (e.g., a gripper grasping an object).

In the first case (see Fig. 3 (top)), it is assumed that a BSP tree representation of each object is built off-line from the CAD model. A Kalman filter is used for

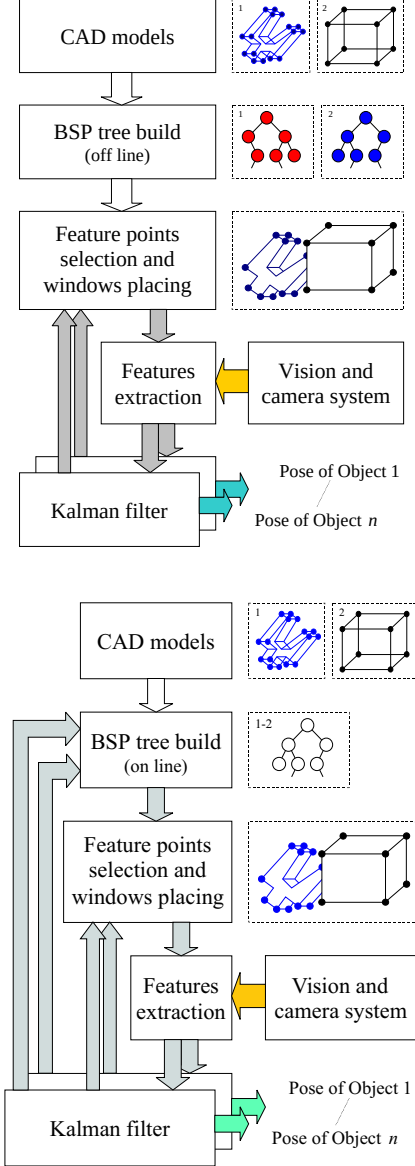


Figure 3: Functional chart of the estimation procedure. Top: non interposing objects. Bottom: interposing objects.

each object to estimate the corresponding pose with respect to the base frame at the next sample time. The feature points selection and windows placing operation, for each camera, can be detailed as follows.

- **Step 1:** An ordered sequence of BSP tree representations is built, according to the estimated distance of the objects from the camera, from the farthest to the nearest.
- **Step 2:** The visit algorithm described in the previous Section is applied to each BSP tree of the sequence to find the set of all the feature points that are visible from the camera. For each item of the ordered sequence, a current set of visible points is updated, by discarding the

feature points of the previous objects occluded by the current object and by adding the visible feature points of the current object.

- **Step 3:** The resulting set of visible points is input to an algorithm for the selection of the optimal feature points.
- **Step 4:** The location of the optimal feature points in the image plane at the next sample time is computed.
- **Step 5:** A dynamic windowing algorithm is applied to select the parts of the image plane to be input to the feature extraction algorithm.

In the case of multiple target objects with interposing parts, this procedure may fail because the objects cannot be correctly ordered with respect to the distance from the camera. This problem can be overcome at expense of the computation time by adopting the solution represented in the functional chart of Fig. 3 (bottom). As before, a Kalman filter is used for each object to estimate the corresponding pose with respect to the base frame at the next sample time. Differently from the previous case, a unique BSP tree representation of all the objects is built online, using the CAD model and the estimation provided by the Kalman filters. Hence, for each camera, the visit algorithm of the tree is applied once to find the set of all the visible points. Then, Steps 3, 4, and 5 are applied.

Notice that the procedures described above can be extended to the case of objects moving among obstacles of known geometry; in case of moving obstacles, the corresponding motion variables can be estimated using Kalman filters.

5 Simulation

The effectiveness of the estimation algorithm has been tested in simulation case studies for a stereo vision system composed by three cameras with a 50 Hz sampling rate. The cameras are fixed in the workspace and their position and orientation with respect to the base frame is reported in Fig. 4. The effective focal length of the three cameras is 16 mm, and the pixel dimensions are $8.3 \times 8.3 \mu\text{m}$. To reproduce the geometric distortion effects of real cameras, a polynomial model including radial distortion, decentering distortion and thin prism distortion has been adopted [10]. The numerical values of the distortion parameters have been chosen so as to achieve errors up to 1.4% of the sensor half dimension. The same model has been considered in the estimation algorithm, and the corresponding parameters have been chosen so as to achieve errors ranging from 1% up to 56%, reproducing a typical situation of partial distortion compensation. Errors up to 3.8% have

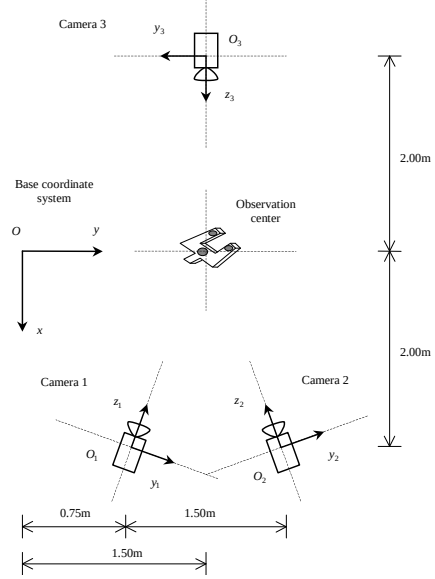


Figure 4: Position and orientation of the three cameras with respect to the base frame.

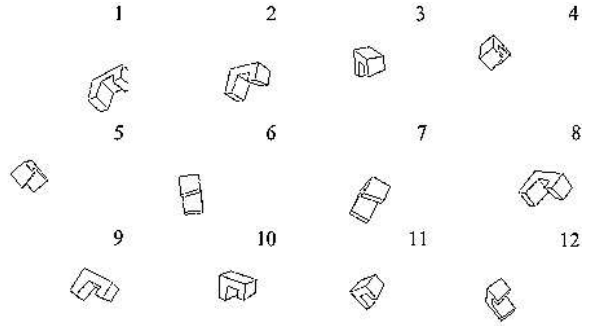


Figure 5: Sequence of projections of the target object on the image plane of camera 1.

been introduced in the parameters of the geometric model of the stereo vision system in order to reflect a typical situation of imperfect system calibration.

A target object characterized by 16 feature points, corresponding to the object corners, is considered (see Fig. 2). The selection procedure is used to find five optimal feature points at each sample time.

The object moves according to a sinusoidal trajectory of 8 s duration and amplitude 200 mm, 150 mm, 100 mm respectively for the x , y and z component of the position, and 30 deg, 25 deg, 45 deg respectively for the Roll, Pitch and Yaw angles. White independent Gaussian noise is added to the true projections on the cameras image plane of the feature points. In order to simulate spatial sampling and quantization errors, the variance of the noise has been chosen as $d^2/12$ where $d = 8.3 \mu\text{m}$ is the pixel dimension. In

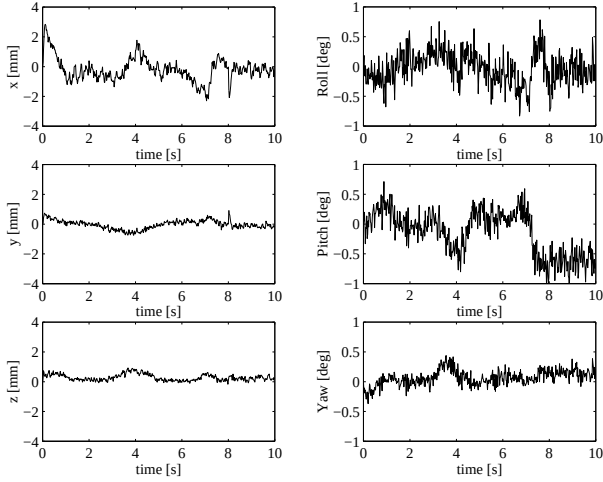


Figure 6: Position (left) and orientation(right) error.

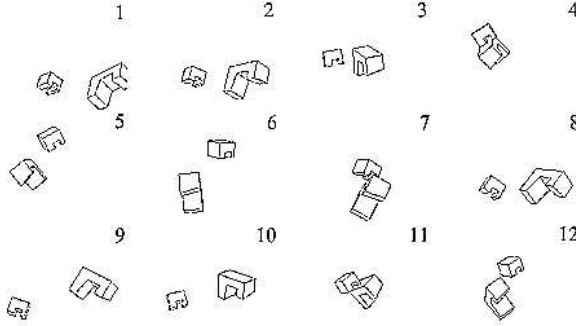


Figure 7: Sequence of projections of two target objects on the image plane of camera 1.

Fig. 5 a sequence of 12 different projections of the target object on the image plane of camera 1 is reported, as reconstructed by the proposed algorithm. It can be seen that the pre-selection algorithm allows for correct prediction of the positions of the feature points (object corners) that are visible from the camera at the next sample time. The corresponding time histories of the components of the position and orientation estimation error are reported in Fig. 6.

In order to test the effectiveness of the proposed approach in the case of multiple target objects, a second object moving according to a sinusoidal trajectory has been added to the previous one. The time histories of the tracking errors are not reported for brevity. In Fig. 7 the projections of the two target objects on the image plane of camera 1 are represented, as reconstructed using the Kalman filter estimates and the pre-selection algorithm. It can be seen that the visible feature points are correctly identified also when the objects are superimposed.

6 Conclusion

The estimation of the pose of moving objects from visual measurements was considered in this paper. 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 objects. 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 BSP tree representation of the objects geometry. The proposed algorithm can be applied also to the case of target objects and obstacles with interposing parts. Ongoing experiments on a visual system of two cameras are satisfactory and will be reported in a future paper.

Acknowledgments. This work was supported by MIUR and ASI.

References

- [1] 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.
- [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] K. Nickels and S. Hutchinson, "Weighting observations: The use of kinematic models in object tracking," *1998 IEEE Int. Conf. on Robotics and Automation*, pp. 1677–1682, 1998.
- [4] F. Keçeci and H.-H. Nagel, "Machine-vision-based estimation of pose and size parameters from a generic workpiece description," *2001 IEEE Int. Conf. on Robotics and Automation*, pp. 2159–2164, 2001.
- [5] 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.
- [6] 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.
- [7] 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, 2002.
- [8] 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.
- [9] 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.
- [10] 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.