Robotica (2003) volume 21, pp. 511–519. © 2003 Cambridge University Press
DOI: 10.1017/S0263574703005137 Printed in the United Kingdom

Managing redundant visual measurements for accurate pose tracking

Vincenzo Lippiello and Luigi Villani

PRISMA Lab., Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli (Italy).
E-mail: {vincenzo.lippiello,luigi.villiani}@unina.it

(Received in Final Form: March 19, 2003)

SUMMARY

This paper proposes an algorithm for managing redundant measurements provided by a stereo multi-camera system to achieve accurate visual tracking of a moving object. Self-occlusion, different visual resolution zones and optimal selection of redundant measurements are some of the problems addressed. The algorithm uses the extended Kalman filter embedded in a computational efficient pose estimation procedure based on Binary Space Partitioning Tree geometric modelling of 3D objects. Experimental results are presented for the case of an object moving in the visual space of two fixed cameras.

KEYWORDS: Pose tracking; Visual data; Redundant measurements; Kalman filter.

1. INTRODUCTION

The progressive cost reduction of visual devices, e.g. cameras, frame-grabbers, video-data processing units, etc., have created an increasing interest in machine vision in the research and industrial communities.

The use of visual sensors may have a 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. Typical industrial applications are assembling of mechanical parts, edge following, object grasping; non industrial applications are automotive guidance, spatial and underwater robotics.

A vision system for robotic applications is based on eye-in-hand cameras, i.e. one or two cameras mounted on the robot end-effector, and/or a system of multiple fixed cameras. In some cases the use of eye-in-hand cameras may become problematic and inefficient, e.g. in the presence of occlusions, for the execution of assembling/disassembling tasks in restricted space, or when a very long tool is used. In this case the use of a system of two or more fixed cameras, suitably located with respect to the robot workspace, may guarantee an optimal and robust extraction of visual information.

On the other hand, the use of a multiple camera system requires the adoption of intelligent and efficient strategies for the management of highly redundant information. This task has to be realized in real-time and thus the extraction and interpretation of all the available visual information is not possible. Hence efficient algorithms must be devised which are able to improve the accuracy and robustness of the visual system by exploiting a minimal set of significant information suitably selected from the initial redundant set.

Another important problem to consider is that the visual measurements are usually affected by significant noise and disturbances due to temporal and spatial sampling and quantization of the image signal, lens distortion, etc. Hence, the use of visual measurements requires the adoption of suitable algorithms with good disturbance rejection capability.

In robotic applications visual systems may be used to estimate in real-time the position and orientation of a moving object. A feasible strategy is based on the recognition of some geometric features of the object, such edges and corners, from camera images. In particular, the reconstruction of the position of a suitable number of corners (feature points) allows the object pose to be computed using a simple point CAD model of the object.[1,2] A possible implementation of this strategy is based on the adoption of the extended Kalman filter, which represents a good trade-off between computational load and estimation accuracy.[3–6]

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 used to solve the photogrammetric equations, 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.[4] Automatic selection algorithms have been developed to find the optimal feature points.[7,8] It should be pointed out, however, that the complexity of the selection algorithms grows at a 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, e.g. by eliminating all the points that are occluded with respect to the camera.[9,10]

In this paper, an approach to pose estimation using a video-system composed by a generic number of cameras is presented. An estimation algorithm based on the extended Kalman filter is adopted, together with an optimal technique for managing redundant visual information. In particular, in order to reduce the computational time, a pre-selection algorithm of the feature points is proposed, based on the selection of all the points that are visible to each camera and

https://doi.org/10.1017/S0263574703005137 Published online by Cambridge University Press

well localizable in the corresponding image plane at a given sample time. This algorithm exhibits a complexity which grows linearly, thanks to the use of a Binary Space Partitioning (BSP) tree for object geometric representation.[11] In detail, the prediction of the object pose provided by the Kalman filter is used to drive a visit algorithm of the BSP tree which allows identifying all the feature points that are visible at the next sample time for each camera. After the preselection, 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. In particular, the selection algorithm exploits the redundancy of the vision system to achieve the best tracking accuracy.

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 estimated using two fixed cameras.

2. MODELLING

In the following, the geometric model of a system of n video cameras and a moving object is provided; then a system of nonlinear equations is derived, to compute the position and orientation of the object from the visual measurements; finally, the recursive solution of the above equations based on the extended Kalman filter is formulated.

2.1. Cameras system model

The geometry of the cameras system is described in Figure 1. 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.

A point P with coordinates $\boldsymbol{p}^{ci}=[x^{ci}\ y^{ci}\ z^{ci}]^{\mathrm{T}}$ in the camera frame is projected onto the point of the image plane whose coordinates can be computed by the equation

$$\begin{bmatrix} u^{ci} \\ v^{ci} \end{bmatrix} = \frac{f_e^{ci}}{z^{ci}} \begin{bmatrix} x^{ci} \\ y^{ci} \end{bmatrix} \tag{1}$$

which is known as perspective transformation. A spatial sampling can be applied to the image plane by expressing the coordinates in terms of number of pixels as

$$\begin{bmatrix} r^{ci} \\ c^{ci} \end{bmatrix} = \begin{bmatrix} r_0^{ci} \\ c_0^{ci} \end{bmatrix} + \begin{bmatrix} s_u^{ci} & 0 \\ 0 & s_v^{ci} \end{bmatrix} \begin{bmatrix} u^{ci} \\ v^{ci} \end{bmatrix} \tag{2}$$

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

2.2. Object pose equations

The position of the origin and the rotation matrix of the i-th camera frame with respect to the base frame are denoted by $\boldsymbol{o}_{ci}$ and $\boldsymbol{R}_{ci}$, 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.[12]

The position and orientation of the object with respect to the base frame can be specified by defining a frame $O_o-x_oy_oz_o$ attached to the object and considering the coordinate vector of the origin $\boldsymbol{o}_o=[x_o\ y_o\ z_o]^{\mathrm{T}}$ and the rotation matrix $\boldsymbol{R}_o(\boldsymbol{\varphi}_o)$, where $\boldsymbol{\varphi}_o=[\phi_o\ \alpha_o\ \psi_o]^{\mathrm{T}}$ is the vector of the Roll, Pitch and Yaw angles. The components of the vectors $\boldsymbol{o}_o$ and $\boldsymbol{\varphi}_o$ are the six quantities to be estimated.

Consider *m feature points* on the object. It can be shown[6] that the position of the j-th feature point in the i-th camera frame can be computed as

$$\boldsymbol{p}_j^{ci}=\boldsymbol{R}_{ci}^{\mathrm{T}}(\boldsymbol{o}_o-\boldsymbol{o}_{ci}+\boldsymbol{R}_o(\boldsymbol{\varphi}_o)\boldsymbol{p}_j^o), \tag{3}$$

where $\boldsymbol{p}_j^o$ 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 a known form of the object CAD model.

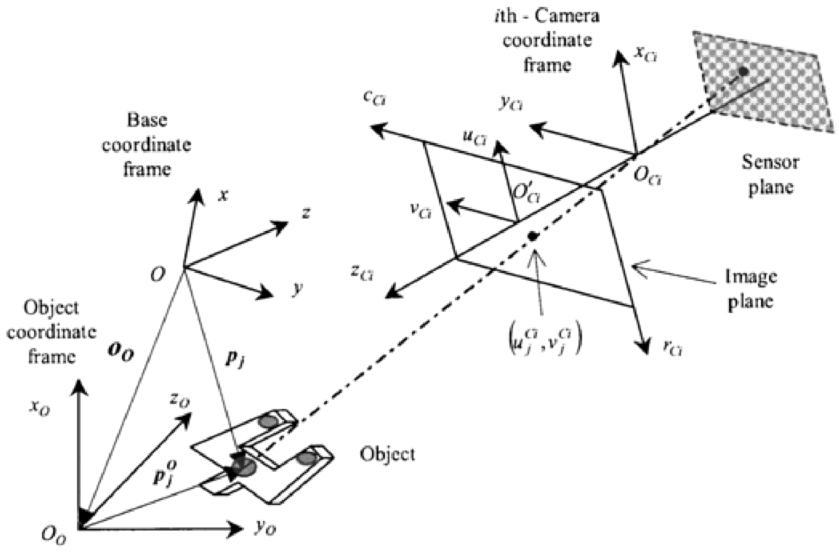

Fig. 1. Reference frames for the i-th camera and the object using the pinhole model.

By folding the m equations (3) into the perspective transformation of the n cameras, a system of 2 mn 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 $\boldsymbol{o}_o$ and $\boldsymbol{\varphi}_o$ are the unknown variables.

2.3. Extended Kalman filter

To solve the above system of nonlincar equations in real-time, an appropriate formulation of 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

$$\boldsymbol{w}_k = \boldsymbol{A}\boldsymbol{w}_{k-1} + \boldsymbol{\gamma}_k \tag{4}$$

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

$$\boldsymbol{A} = \mathrm{diag}\left\{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\right\}.$$

The output equation of Kalman filter is chosen as

$$\boldsymbol{\zeta}_k = \boldsymbol{g}(\boldsymbol{w}_k) + \boldsymbol{\nu}_k \tag{5}$$

with

$$\boldsymbol{g}(\boldsymbol{w}_k) = \left[\frac{x_1^{c1}}{z_1^{c1}}\ \frac{y_1^{c1}}{z_1^{c1}} \cdots \frac{x_1^{cn}}{z_1^{cn}}\ \frac{y_1^{cn}}{z_1^{cn}} \cdots \frac{x_m^{c1}}{z_m^{c1}}\ \frac{y_m^{c1}}{z_m^{c1}} \cdots \frac{x_m^{cn}}{z_m^{cn}}\ \frac{y_m^{cn}}{z_m^{cn}}\right]_k^{\mathrm{T}}, \tag{6}$$

where the coordinates of the feature points $\boldsymbol{p}_j^{ci}$ are computed from the state vector $\boldsymbol{w}_k$ via (3). 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 necessary to linearize the output equations about the current state estimate at each sample time, considering the so-called extended Kalman filter. The first step of the extended Kalman algorithm provides an optimal estimate of the state at the next sample time according to the recursive equations

$$\hat{\boldsymbol{w}}_{k,k-1} = \boldsymbol{A}\hat{\boldsymbol{w}}_{k-1,k-1} \tag{7}$$

$$\boldsymbol{P}_{k,k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1,k-1}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}, \tag{8}$$

where $\boldsymbol{P}_{k,k-1}$ is the (12×12) covariance matrix of the estimate state error. The second step improves the previous estimate by using the input measurements according to the equations

$$\hat{\boldsymbol{w}}_{k,k} = \hat{\boldsymbol{w}}_{k,k-1} + \boldsymbol{K}_k(\boldsymbol{\zeta}_k - \boldsymbol{g}(\hat{\boldsymbol{w}}_{k,k-1})) \tag{9}$$

$$\boldsymbol{P}_{k,k} = \boldsymbol{P}_{k,k-1} - K_k\boldsymbol{C}_k\boldsymbol{P}_{k,k-1}, \tag{10}$$

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

$$\boldsymbol{K}_k = \boldsymbol{P}_{k,k-1}\boldsymbol{C}_k^{\mathrm{T}}(\boldsymbol{R}_k + \boldsymbol{C}_k\boldsymbol{P}_{k,k-1}\boldsymbol{C}_k^{\mathrm{T}})^{-1}, \tag{11}$$

being $\boldsymbol{C}_k$ the $(2m \times 12)$ Jacobian matrix of the output function

$$\boldsymbol{C}_k = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\hat{\boldsymbol{w}}_{k,k-1}}. \tag{12}$$

The analytic expression of $\boldsymbol{C}_k$ can be found in another paper.[13]

3. REDUNDANCY MANAGEMENT

The accuracy of the estimate provided by the Kalman filter depends on the number of the available feature points. The inclusion of extra points will improve the estimation accuracy but will increase the computational cost of the feature extraction and of the Kalman filtering. On the other hand, it has been shown that a number of five or six feature points, with suitable properties depending on the object geometry, can ensure the best accuracy achievable for a given resolution of the camera system.[1,4] To guarantee the existence of such points for any object pose, the feature points of the object should be numerous and uniformly distributed. Moreover, to save computational time, efficient selection techniques of the optimal points should be devised.[8] In this work, an efficient selection method is prosed, which is based on a preselection algorithm and an optimal selection algorithm. In particular, the pre-selection algorithm (with linear complexity) is in charge of reducing the number of points on which the optimal selection algorithm (with factorial complexity) operates. The two-steps procedure allows a sensible reduction of the time spent for the whole selection process.

3.1. Pre-selection algorithm

The pre-selection technique is aimed at selecting all the feature points that are visible from each camera, using a geometric model of the object based on Binary Space Partitioning (BSP) tree structures. Moreover, a windowing test is performed in order to recognize all the feature points that are "well" localizable, i.e. their positions can be effectively measured with a given accuracy.

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.[14]

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.

https://doi.org/10.1017/S0263574703005137 Published online by Cambridge University Press

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 another paper.[15]

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, as shown in Figure 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 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.

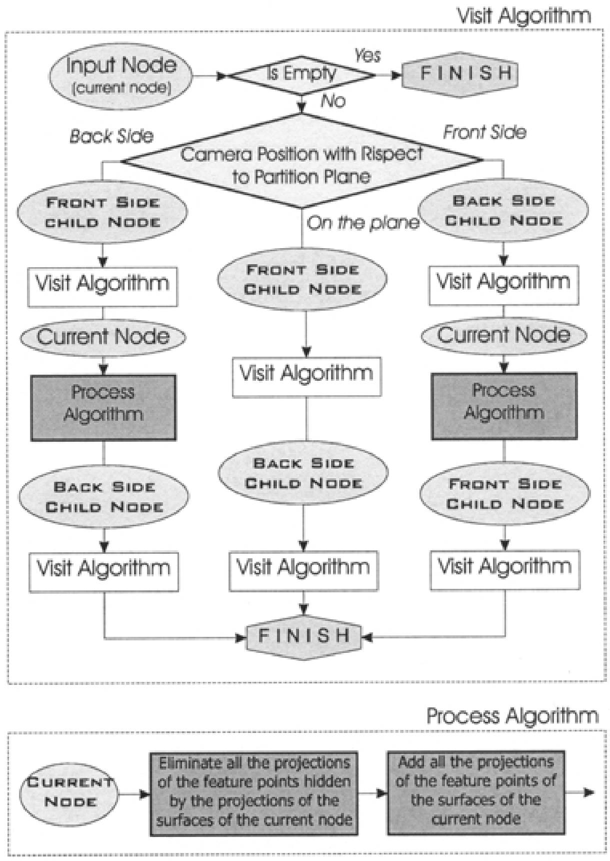

Fig. 2. Recursive visit algorithm of the BSP tree for the selection of visible feature points.

Finally, 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 to each other or to the boundaries of the image plane, are discarded.

3.2. *Selection algorithm*

The number of remaining feature points after pre-selection 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 a higher number of feature points is considered.

The optimality of a set Γ of feature points is valued through the composition of suitably selected quality indices into an optimal cost function. The quality indices 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, a 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, $i=1, 2$:

$$Q_{si}=\frac{1}{q_i}\sum_{k=1}^{q_i}\min_{\substack{j\in\{1,\ldots,q_i\}\\ j\neq k}}||\boldsymbol{p}_j-\boldsymbol{p}_k||.$$

Notice that $q=q_1+q_2$ is chosen between 6 and 8 to handle 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 $\boldsymbol{p}_{k+1}-\boldsymbol{p}_{C_i}$ and the vector $\boldsymbol{p}_k-\boldsymbol{p}_{Ci}$, being $\boldsymbol{p}_{Ci}$ 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 $\boldsymbol{p}_{Ci}$, with $\boldsymbol{p}_{q_i+1}=\boldsymbol{p}_1$.

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, $i=1, 2$:

$$Q_h=\begin{cases} 1+\epsilon & \text{if } \Gamma=\Gamma_{opt} \\ 1 & \text{otherwise} \end{cases}$$

where ϵ is a positive constant and Γ_{opt} is the optimal set of feature points at the previous sample time.

In order to distribute the points among the two cameras, the following indices 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. The second index takes into account the distance of the cameras from the object, and thus allows managing different resolution zones of different cameras.

The proposed quality indices 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 indices have been proposed,[8] and some of them can be added to the indices adopted in this paper.

The cost function is chosen as

$$\boldsymbol{Q}=Q_h\,\frac{Q_eQ_d}{q}\,(q_1Q_{s1}Q_{a1}+q_2Q_{s2}Q_{a2})$$

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.

It should be pointed out that, in some cases, the number of points resulting at the end of the pre-selection step may be too high to perform the optimal selection in a reasonable time. In such a case, a computational cheaper solution, based on the optimal set at the previous time-step, can be adopted to find a sub-optimal set. For sufficiently small sampling time the sub-optimal solution is very close or coincides with the optimal one.

4. ESTIMATION PROCEDURE

A functional chart of the estimation procedure is reported in Figure 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:

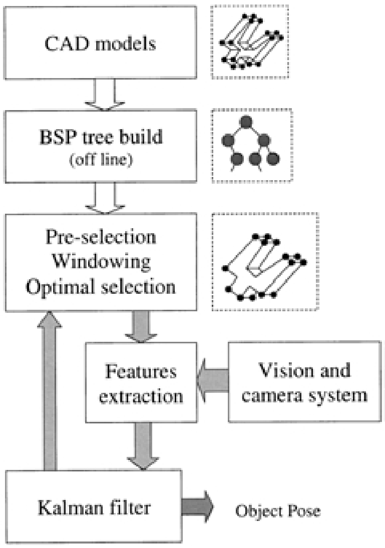

Fig. 3. Functional chart of the estimation procedure.

- **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 in the pose predicted by the Kalman filter.
- **Step 2:** The resulting set of visible points is input to the algorithm for the selection of the optimal feature points.
- **Step 3:** The location of the optimal feature points in the image plane 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 the 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 **Steps 1, 3**.

Notice that the procedure described above can be extended to the case of multiple objects moving among obstacles of known geometry;[15] if the obstacles are moving with respect to the base frame, the corresponding motion variables can be estimated using Kalman filters.

5. EXPERIMENTS

The proposed algorithm has been tested on a number of case studies on an industrial set-up.

5.1. Experimental set-up

The experimental set-up is composed of a PC with Pentium IV 1.7 Ghz 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 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 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 a suitable calibration procedure,[12] 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 $\pi/6$ 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[16] and on a simple custom implementation of a corner detector. 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 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 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. Figure 4 shows the stereo vision system and the robot carrying the object.

5.2. *Case studies*

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

The time history of the trajectory used for the first experiment is represented in Figure 5. The maximum linear velocity and angular velocity are about 3 cm/s and 3 deg/s, respectively. The time history of the estimation errors is shown in Figure 6. Noticeably, the accuracy of the system

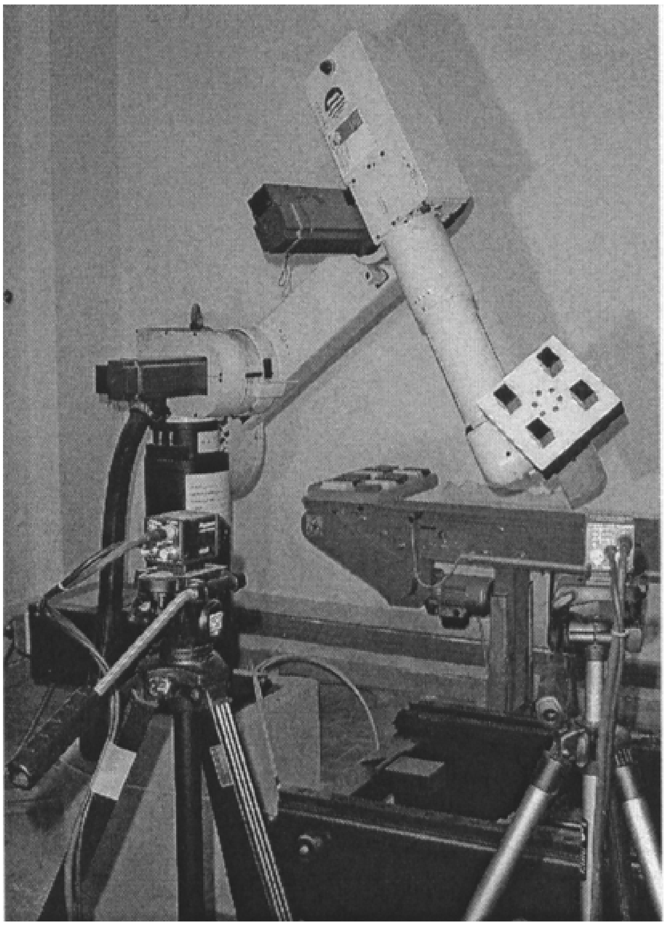

Fig. 4. COMAU Smart3-S robot and SONY 8500CE cameras.

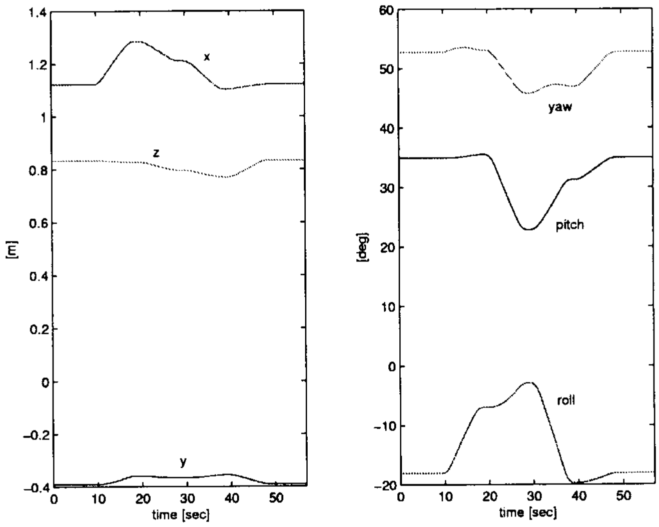

Fig. 5. Object trajectory in the first case study. Left: Position trajectory. Right: Orientation trajectory.

reaches the limit allowed by cameras calibration, for all the components of the motion, when the object does not move. 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 Figure 7 the output of the whole selection algorithm, for the two cameras, is reported. For each of the 40 feature

https://doi.org/10.1017/S0263574703005137 Published online by Cambridge University Press

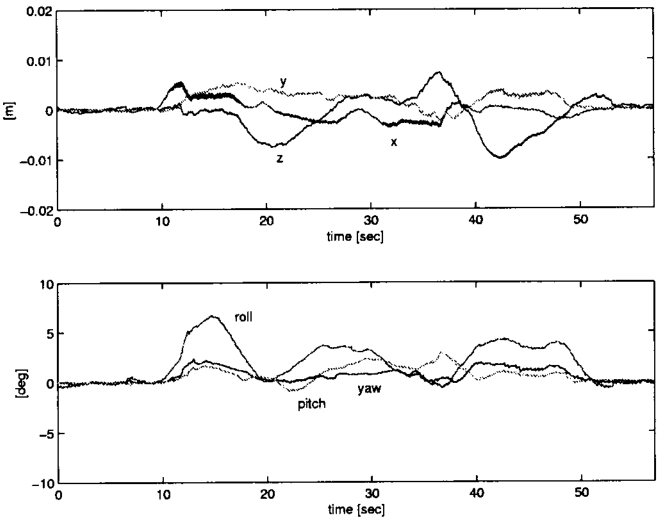

Fig. 6. Time history of the estimation errors in the first case study. Top: Position errors. Bottom: Orientation errors.

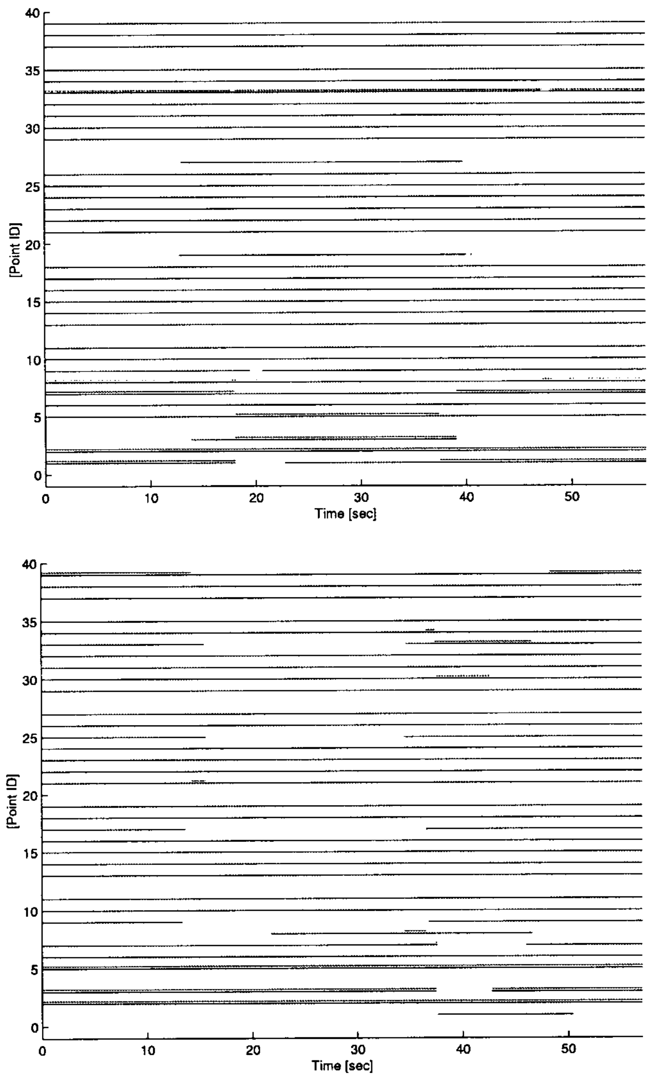

Fig. 7. Visible and selected points for camera 1 (top) and camera 2 (bottom) in the first case study. For each point, the bottom line indicates when it is visible, the top line indicates when it is selected for feature extraction.

points, two horizontal lines are considered: a point of the bottom line indicates that the feature point was classified as visible and localizable by the pre-selection algorithm at a particular sample time; a point of the top line indicates that the visible feature point was automatically 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 often chosen (but not always localized) at each sampling time, coherently with the symmetric disposition of the cameras with respect to the object.

The time history of the trajectory used for the second experiment is represented in Figure 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 Figure 9. It can be observed that the error remains low and is a bit greater than the estimation error of the previous experiment. This is due to the fact that the trajectories of the visible points are faster. Notice that the error remains low also when the object moves partially out of the visible space of the second camera, i.e. from t=45 s to t=65 s. This confirms the capability of the proposed algorithm of selecting the feature points that guarantee the best tracking accuracy for any configuration, by suitably exploiting the redundancy of the system. The corresponding output of the pre-selection and selection algorithms are reported in Figure 10. Notice that the set of well localizable points is more variable than in the previous experiment; moreover, from t=45 s to t=65 s, the number of well localizable points for the second camera is considerably reduced.

6. CONCLUSION

An algorithm for managing redundant measurements provided by a stereo multi-camera system has been proposed in

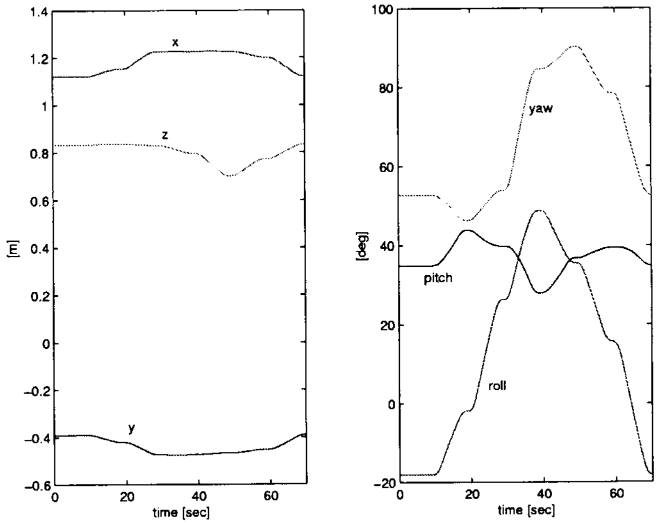

Fig. 8. Object trajectory in the second case study. Left: Position trajectory. Right: Orientation trajectory.

https://doi.org/10.1017/S0263574703005137 Published online by Cambridge University Press

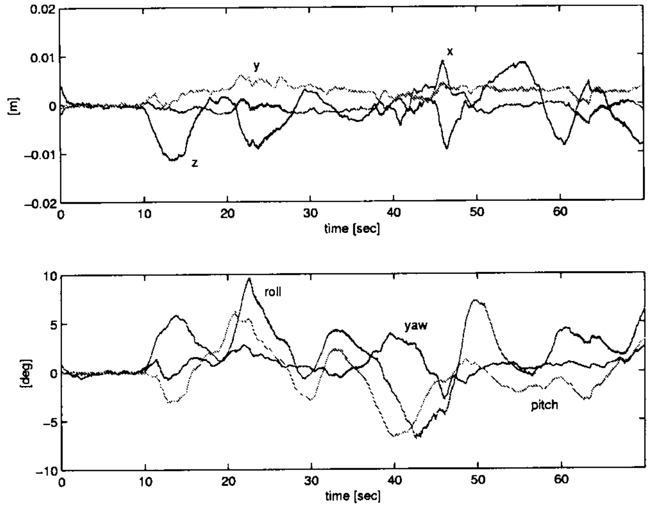

Fig. 9. Time history of the estimation errors in the second case study. Top: Position errors. Bottom: Orientation errors.

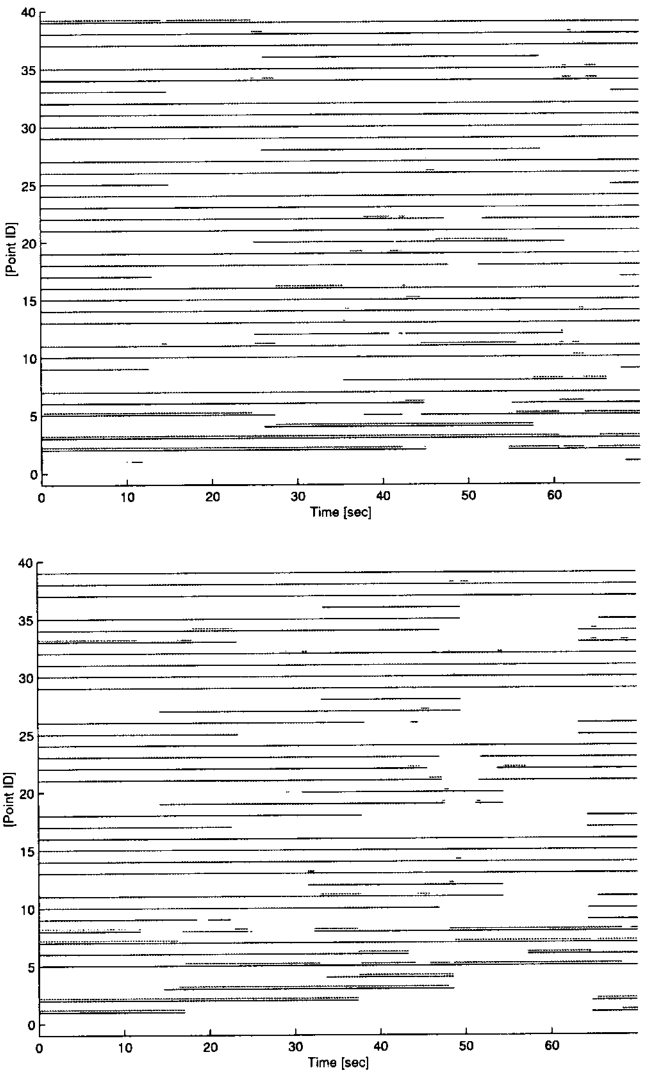

Fig. 10. Visible and selected points for camera 1 (top) and camera 2 (bottom) in the second case study. For each point, the bottom line indicates when it is visible, the top line indicates when it is selected for feature extraction.

this paper. The problem of tracking the position and orientation of a moving object has been considered. 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 redundancy provided by the visual system is suitably exploited to find the feature points that guarantee the best tracking accuracy for any configuration. 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, ASI and CNR.

References

1. W. J. Wilson, “Relative end-effector control using cartesian position based visual servoing,” *IEEE Trans. on Robotics and Automation*, **12**, 684–696 (1996).
2. F. Janabi-Sharifi and W. J. Wilson, “Automatic grasp planning for visual servo controlled robotic manipulators,” *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, **28**, 693–711 (1998).
3. 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).
4. J. Wang and W. J. 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).
5. J. N. Pan, Y. Q. Shi and C.Q. Shu, “A Kalman filter in motion analysis from stereo image sequence,” *1994 IEEE Int. Conf. on Image Processing*, pp. 63–67 (1994).
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. J. T. Feddema, C. S. G. Lee and O. R. Mitchell, “Weighted selection of image features for resolved rate visual feedback control,” *IEEE Trans. on Robotics and Automation*, **7**, 31–47 (1991).
8. F. Janabi-Sharifi and W. J. Wilson, “Automatic selection of image features for visual servoing,” *IEEE Trans. on Robotics and Automation*, **13**, 890–903 (1997).
9. K. Tarabanis, R. Y. Tsai and A. Kaul, “Computing occlusion-free viewpoints,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **18**, 279–292 (1996).
10. P. C. Ho and W. Wang, “Occlusion culling using minimum occluder set and opacity map,” *1999 IEEE Int. Conf. on Information Visualization*, pp. 292–300 (1999).
11. 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).
12. J. Weng, P. Cohen and M. Herniou, “Camera calibration with distortion models ad accuracy evaluation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **14**, 965–980 (1992).
13. 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*, **12**, 71–88 (2002).

https://doi.org/10.1017/S0263574703005137 Published online by Cambridge University Press

14. M. Paterson and F. Yao, “Efficient binary space partitions for hidden-surface removal and solid modeling,” *Discrete and Computational Geometry*, **5**, 485–503 (1990).
15. V. Lippiello, B. Siciliano and L. Villiani, “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).
16. J. Canny, “A computational approach to edge detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **8**, 679–698 (1986).

https://doi.org/10.1017/S0263574703005137 Published online by Cambridge University Press