Real-Time Visual Tracking of Complex Structures

Tom Drummond, *Member*, *IEEE Computer Society*, and Roberto Cipolla, *Member*, *IEEE*

Abstract—This paper presents a novel framework for three-dimensional model-based tracking. Graphical rendering technology is combined with constrained active contour tracking to create a robust wire-frame tracking system. It operates in real time at video frame rate (25 Hz) on standard hardware. It is based on an internal CAD model of the object to be tracked which is rendered using a binary space partition tree to perform hidden line removal. The visible edge features are thus identified online at each frame and correspondences are found in the video feed. A Lie group formalism is used to cast the motion computation problem into simple geometric terms so that tracking becomes a simple optimization problem solved by means of iterative reweighted least squares. A visual servoing system constructed using this framework is presented together with results showing the accuracy of the tracker. The system also incorporates real-time online calibration of internal camera parameters. The paper then describes how this tracking system has been extended to provide a general framework for tracking in complex configurations, including the use of multiple cameras, the tracking of structures with articulated components, or of multiple structures with constraints. The methodology used to achieve this exploits the simple geometric nature of the Lie group formalism which renders the constraints linear and homogeneous. The adjoint representation of the group is used to transform measurements into common coordinate frames. The constraints are then imposed by means of Lagrange multipliers. Results from a number of experiments performed using this framework are presented and discussed.

Index Terms—Visual tracking, real-time, 3D, Lie groups, articulated motion.

1 Introduction

The tracking of known three-dimensional objects is useful for numerous applications, including motion analysis, surveillance, and robotic control tasks. This paper describes a powerful framework for tracking such structures.

Section 2 describes the framework used to accurately track a known rigid three-dimensional object moving in the field of view of a single camera. This framework uses a formalism based on the use of Lie groups to simplify representation and computation of aspects of the tracking problem. The output of this tracker is a continuously updated estimate of the pose of the object relative to the camera. A visual servoing system constructed using this framework is presented in Section 2.4. This system closes the robot control loop to guide a robotic arm to a previously taught target location relative to a workpiece. An extension to the basic framework is then described in Section 2.6, which allows tracking of the internal camera parameters in addition to the pose, thus providing a mechanism for online calibration of internal camera parameters.

Section 3 then shows how this framework can be exploited within complex systems which are designed to operate within environments containing multiple cameras, multiple target structures, and articulated structures. A common approach to handling all of these factors is presented, which again exploits the geometric properties of the Lie group formalism. This expresses the multibody problem in simple and intuitive geometric terms so that the constraints which exist also have a simple form and are both linear and homogeneous. The adjoint representation of the group is used to transform image measurements into a common coordinate frame where the constraints can be imposed by means of Lagrange multipliers which can computed explicitly.

This paper makes extensive use of the mathematics of Lie groups and their algebras [30]. Approaches based on this formalism have been used previously (e.g., [32], [5]) and it is worth noting the benefits that follow from this:

- It provides efficient parameterizations via the exponential map. In particular, 3D pose motions are represented using a set of six parameters which is minimal unlike direct use of the matrix (12 elements) or translation plus quaternion (seven elements).
- It provides a canonical method of linearizing which is easy and fast to compute. Because the parameterization is minimal, there are no additional constraints to consider, unlike directly paramterizing matrix entries.

When more complex configurations such as those containing articulated structures are considered, the representation yields further benefits:

- The adjoint representation of the group can be used to transform motion parameters computed in one coordinate frame into another.
- Articulation constraints (from a hinge or slide, etc.) can be imposed on the motion parameters rather than on the projection matrices. This simplifies the expression of articulation constraints (which are nontrivial when expressed in terms of the projection matrices). In this form they are linear, homogeneous, and independent of the current pose of the articulated structure.

1.1 Model-Based Tracking

Because a video feed contains a very large amount of data, it is important to extract only a small amount of perceptually

• *The authors are with the Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK. E-mail: {twd20, cipolla}@eng.cam.ac.uk.*

Manuscript received 30 Oct. 2000; revised 10 Sept. 2001; accepted 11 Sept. 2001.
Recommended for acceptance by R. Nelson.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113078.

important information, if real-time frame (or field) rate performance is to be achieved [18]. This observation leads to the notion of *feature-based* tracking [17] in which processing is restricted to locating strong image features such as contours [33], [8].

A number of successful systems have been based on tracking the image contours of a known model. Lowe [24] used the Marr-Hildreth edge detector to extract edges from the image which were then chained together to form lines. These lines were matched and fitted to those in the model. A similar approach using the Hough transform has also been used [36]. The use of dense two-dimensional image processing incurs a significant computational cost and both of these systems make use of special purpose hardware in order to achieve frame rate processing.

An alternative approach is to render the model first and then use sparse one-dimensional search to find and measure the distance to matching (nearby) edges in the image. This approach has been used in RAPID [19], CONDENSATION [22], and other systems [9], [35], [26]. The efficiency yielded allows all these systems to run in real-time on standard workstations. The method is also used here and discussed in more detail in Section 2.1.

Using either of these approaches, most systems (except CONDENSATION) then compute the pose parameters by linearizing with respect to image motion. This process is expressed here in terms of the Lie group SE(3) and its Lie algebra. This is convenient because the group SE(3) exactly represents the space of poses that form the output of the system, while the Lie algebra is the tangent space to the group at the identity and is therefore the natural space in which to represent differential quantities such as velocities and small motions. Thus, the representation provides a canonical method for linearizing the relationship between image motion and pose parameters. This approach can also be generalized to other transformation groups (e.g., planar contour using the groups GA(2) and P(2) [12]).

Outliers are a key problem that must be addressed by systems which measure and fit edges. They frequently occur in the measurement process since additional edges may be present in the scene in close proximity to the model edges. These may be caused by shadows, for example, or strong background scene elements. Such outliers are a particular problem for the traditional least-squares fitting method used by many of the algorithms. Methods of improving robustness to these sorts of outliers include the use of RANSAC [2], factored sampling [22], or regularization, for example, the Levenberg-Marquadt scheme used in [24]. The approach used here employs iterative reweighted least squares which provides a robust M-estimator. Saliency criteria have often been to improve the performance of visual trackers [31], [27]. Here, the reweighting scheme is extended to incorporate a number of additional saliency measures, discussed in more detail in Section 2.3.

There is a trade-off to be made between robustness and precision. The CONDENSATION system, for example, obtains a high degree of robustness by taking a large number of sample hypotheses of the position of the tracked structure with a comparatively small number of edge measurements per sample. By contrast, the system presented here uses a large number of measurements for a single position hypothesis and is thus able to obtain very high precision in its positional estimates. This is particularly relevant in tasks such as visual servoing since the dynamics and environmental conditions can be controlled so as to constrain the robustness problems, while high precision is needed in real-time in order for the system to be useful.

Occlusion is also a significant cause of instabilities and may occur when the object occludes parts of itself (self occlusion) or where another object lies between the camera and the target (external occlusion). RAPID handles the first of these problems by use of a precomputed table of visible features indexed by what is essentially a view sphere. By contrast, the system presented here uses graphical rendering techniques to dynamically determine the visible features and is thus able to handle more complex situations (such as objects with holes) than can be tabulated on a view sphere. External occlusion can be treated by using outlier rejection, for example, in [2] which discards primitives for which insufficient support is found or by modifying statistical descriptions of the observation model (as in [25]). If a model is available for the intervening object, then it is possible to use this to reestimate the visible features [16], [36]. Both of these methods are used within the system presented here.

An important application domain for real-time visual tracking is that of visual servoing in which the output from such tracking systems is used for control of robotic manipulators. A distinction is often made in visual servoing [21] between an *image-based* approach in which the control loop is closed in the image by driving image features home to desired target positions [13], [29] and a *position-based* approach in which the control loop operates in an explicitly three-dimensional space [3], [34]. The approach presented here is position-based but closes the control loop by projecting the action of three-dimensional camera motion into the image where it is fitted to image measurements in a manner similar to [36]. Since the eye-in-hand approach is used, this generates a motion-to-image Jacobian (also known as the interaction screw [13]) which can be used to generate robot control commands to minimize the image error.

1.2 Articulated Structures

This paper then addresses the issue of tracking articulated structures which are characterized as comprising rigid components connected by simple constraints such as hinges, slides, etc. [11]. In the taxonomy of [1], this is classified as "articulated motion" as opposed to "elastic motion" which includes more general deformation models.

Lowe [23] considered articulated motion for a general class of structures, which are represented by means of internal model parameters which are stored in a tree structure giving a kinematic chain in which the full pose is represented only for the rigid component corresponding to the root node of the tree. This is also a representation commonly used for tracking human motion in activities such as walking, running, and waving. Gavrila and Davis [15] achieve this by matching edges in the image with those of an appearance model using distance transforms, Delamarre and Faugeras [10] force the appearance model to fit within silhouettes computed from multiple images, while Bregler and Malik [5] compute adaptive support maps to directly match pixels between images. By contrast, the approach presented in Section 3 uses a redundant symmetric representation in which the full pose of each rigid component is stored independently. Constraints are then imposed on the relationships between component pose estimates.

Other approaches do not make use of explicit 3D models such as [6] which tracks humans in two dimensions using scaled prismatic models. This approach tracks multiple hypotheses (like CONDENSATION) but represents the posterior more compactly as piecewise Gaussian (rather than as a set of samples). Alternatively, eigen images can be used to parameterize the motions purely in terms of their appearance in the image [4].

2 Rigid Body Tracking

The approach used here for tracking known three-dimensional structures is based upon maintaining an estimate of the camera projection matrix, P, in the coordinate system of the structure. This projection matrix is represented as the product of a matrix of internal camera parameters:

$$K = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

and a Euclidean projection matrix representing the position and orientation of the camera relative to the target structure:

$$E = [\, R \quad t \,] \qquad \text{with } RR^T = I \text{ and } |R| = 1. \tag{2}$$

The projective coordinates of an image feature are then given by

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = P \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3}$$

with the actual image coordinates given by

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} u/w \\ v/w \end{pmatrix}. \tag{4}$$

Rigid motions of the camera relative to the target structure between consecutive video frames can then be represented by right multiplication of the projection matrix by a Euclidean transformation of the form:

$$M = \begin{bmatrix} R & t \\ 0\,0\,0 & 1 \end{bmatrix}. \tag{5}$$

These M, form a 4×4 matrix representation of the group SE(3) of rigid body motions in three-dimensional space, which is a six-dimensional Lie Group. The generators of this group are typically taken to be translations in the x, y, and z directions and rotations about the x, y, and z axes, represented by the following matrices:

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$G_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

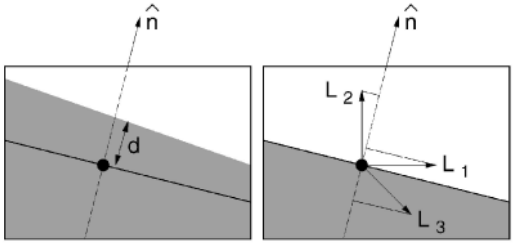


Fig. 1. Computing the normal component of the motion and generator vector fields.

These generators form a basis for the vector space (the Lie algebra) of derivatives of SE(3) at the identity. Group elements can be obtained from the generators via the exponential map:

$$M = \exp(\alpha_i G_i) \tag{7}$$

(with Einstein summation convention over Latin indices used throughout this paper). Thus, if M represents the transformation of the structure between two adjacent video frames, then the task of the tracking system becomes that of finding the α_i that describe the interframe transformation. Since the motion will be small, M can be approximated by the linear term:

$$M \approx I + \alpha_i G_i. \tag{8}$$

Consequently, the motion is approximately a linear sum of that produced by each of the generators. The partial derivative of projective image coordinates with respect the ith generating motion can be computed as:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = PG_i \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{9}$$

with

$$L_i = \begin{pmatrix} \tilde{u}' \\ \tilde{v}' \end{pmatrix} = \begin{pmatrix} \frac{u'}{w} - \frac{uw'}{w^2} \\ \frac{v'}{w} - \frac{vw'}{w^2} \end{pmatrix} \tag{10}$$

giving the motion in true image coordinates. A least-squares approach can then be used to fit the observed motion of image features between adjacent frames (see Section 2.2).

2.1 Tracking Edges

The features used in this work for tracking are the visible edges of a CAD model of the part to be tracked. These are strong features that can be reliably found in the image because they have a significant spatial extent. Furthermore, this means that a number of measurements can be made along each edge and, thus, they may be accurately localized within an image. This approach also takes advantage of the aperture problem (that the component of motion of an edge, tangent to itself, is not observable locally). This yields a significant benefit since the search for intensity discontinuities in the video image can be limited to a one dimensional path that lies along the edge normal, $\hat{n}$ (see Fig. 1) and, thus, has linear complexity in the search range, rather than quadratic [7]. This reduction in complexity makes it possible to track complex

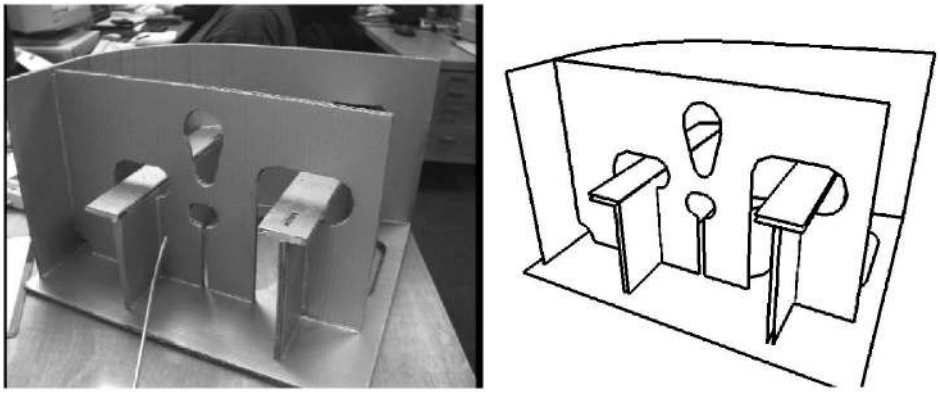


Fig. 2. Image and CAD model of ship part.

structures in real time on a standard workstation without special hardware. The normal component of the motion fields, L_i are then also computed (as $f_i = L_i \cdot \hat{n}$) and d can be fitted as a linear combination of the f_i to give a linearized estimate of the 3D motion.

In order to track the edges of the model as lines in the image, it is necessary to determine which (parts of) lines are visible at each frame and where they are located relative to the camera. This work uses binary space partition trees [28] to dynamically determine the visible features of the model in real-time. This technique allows accurate frame rate tracking of complex structures such as the ship part shown in Fig. 2. As rendering takes place, a stencil buffer is used to locate the visible parts of each edge by querying the buffer at a series of points along the edge prior to drawing the edge. Where the line is visible, sample points are assigned to search for the nearest intensity discontinuity in the video feed along the edge normal (see Fig. 3).

Fig. 4 shows system operation. At each cycle, the system renders the expected view of the object (Step a) using its current estimate of the projection matrix, P. The visible edges are identified and sample points are assigned at regular intervals in image coordinates along these edges (Step b). The edge normal is then searched in the video image for a nearby edge (Step c). Typically, $m \approx 400$ samples are assigned and measurements made in this way. The system then projects this m-dimensional measurement vector onto the six-dimensional subspace corresponding to Euclidean transformations (Step d) giving the least-squares estimate of the motion, M. The Euclidean part of the projection matrix, E is then updated by right multiplication with this transformation (Step e). Finally, the new projection matrix P is obtained by multiplying the camera parameters K with the updated Euclidean matrix to give a new current estimate of the local position (Step f). The system then loops back to Step a.

2.2 Computing the Motion

Step (d) in the process involves the projection of the measurement vector onto the subspace defined by the Euclidean transformation group. This subspace is given by the f_i^ξ which describe the magnitude of the edge normal motion that would be observed in the image at the ξth sample point for the ith group generator. These can be considered as a set of six m-dimensional vectors which describe the motion in the image for each of the six modes of Euclidean transformation. The system then projects the m-vector corresponding to the measured distances (to the observed edges) onto the six-dimensional subspace spanned by the transformation vectors. This corresponds to the geometric transformation of the part which best fits the observed edge positions and is found by minimising the square error between the transformed edge position and the actual edge position (in pixels). This process is performed using the standard least-squares algorithm as follows:

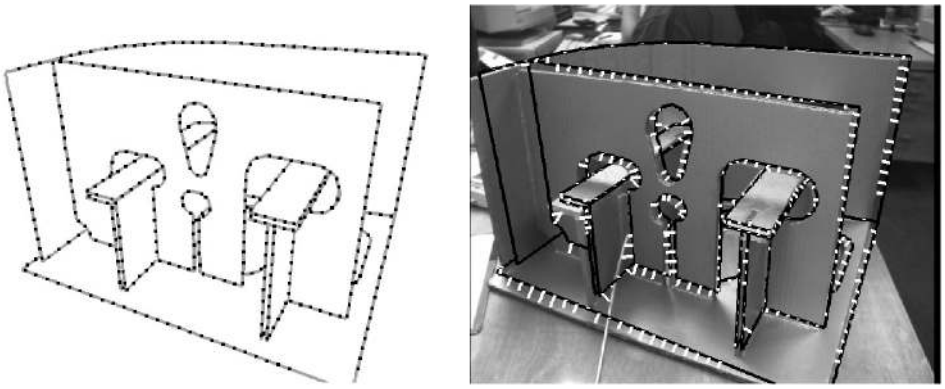


Fig. 3. Sample points are assigned and distances measured.

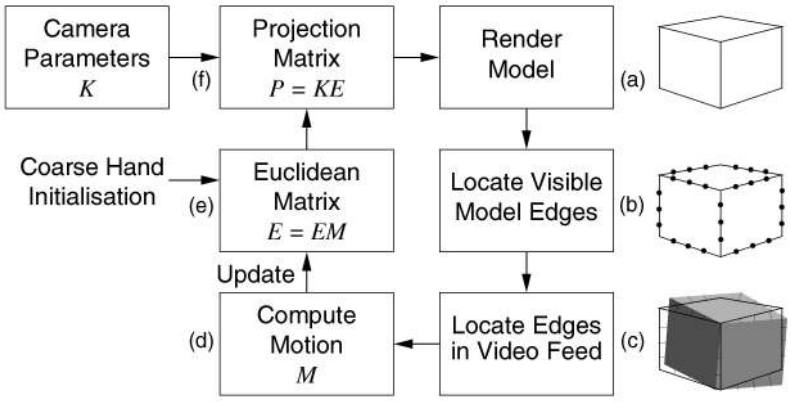


Fig. 4. Tracking system operation.

$$v_i = \sum_{\xi} d^{\xi} f_i^{\xi}, \tag{11}$$

$$C_{ij} = \sum_{\xi} f_i^{\xi} f_j^{\xi}, \tag{12}$$

$$\alpha_i = C_{ij}^{-1} v_j. \tag{13}$$

Giving α_i which are the six coefficients of the projected vector. It can be seen that setting $\beta_i = \alpha_i$ gives the minimum (least-squares) solution to

$$S = \sum_{\xi} \left(d^{\xi} - \beta_i f_i^{\xi} \right)^2 \tag{14}$$

$$\text{since} \qquad \frac{\partial S}{\partial \beta_i} = -2 \sum_{\xi} f_i^{\xi} \left(d^{\xi} - \beta_j f_j^{\xi} \right) \tag{15}$$

and setting $\beta_i = \alpha_i$ and substituting (13) gives

$$\frac{\partial S}{\partial \beta_i} = -2 \sum_{\xi} f_i^{\xi} d^{\xi} - f_i^{\xi} f_j^{\xi} C_{jk}^{-1} \sum_{\xi'} f_k^{\xi'} d^{\xi'} \tag{16}$$

$$= -2 \sum_{\xi} (f_i^{\xi} d^{\xi}) + 2 C_{ij} C_{jk}^{-1} \sum_{\xi'} f_k^{\xi'} d^{\xi'} \quad = 0. \tag{17}$$

The α_i are thus the coefficients of a linear approximation to the Euclidean motion which minimizes the sum squared error between the model and the observed lines. When more complex configurations are examined, it will become important to consider how the sum squared error varies when $\beta_i \neq \alpha_i$. Setting $\beta_i = \alpha_i + \varepsilon_i$, (15) gives

$$\frac{\partial S}{\partial \beta_i} = 0 + 2 \sum_{\xi} f_i^{\xi} f_j^{\xi} \varepsilon_j \tag{18}$$

$$= 2 C_{ij} \varepsilon_j \tag{19}$$

$$\text{and integrating gives} \;\; S = S_0 + \varepsilon_i C_{ij} \varepsilon_j \;\; \text{where } S_0 = S|_{\varepsilon=0}. \tag{20}$$

All that remains for the rigid body tracker is to compute the matrix for the motion of the model represented by the α_i and apply it to the matrix E in (2) which is done by using the exponential map:

$$E_{t+1} = E_t \; \exp\left(\sum_i \alpha_i G_i \right). \tag{10}$$

The system is therefore able to maintain an estimate of E (and, hence, P) by continually computing the coefficients α_i of interframe motions (see Fig. 5). One key advantage of this approach is that it is possible to extend it to include more complex situations such as tracking camera parameters in addition to object motion.

2.3 Robustness

The naïve least-squares algorithm presented in Section 2.2 is vulnerable to instabilities caused by the presence of outliers. This is because the sum-of-squares objective function can be significantly affected by a few measurements with large

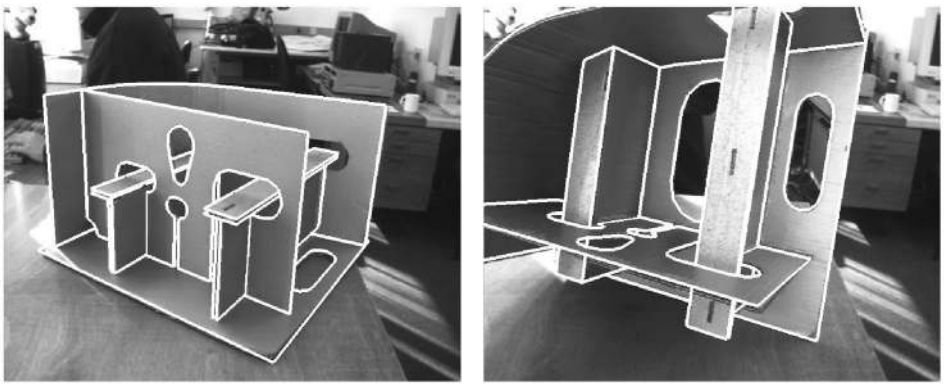


Fig. 5. Frames from video of tracking sequence: The CAD model of the ship part is superimposed on the video image using the estimate of the projection matrix.

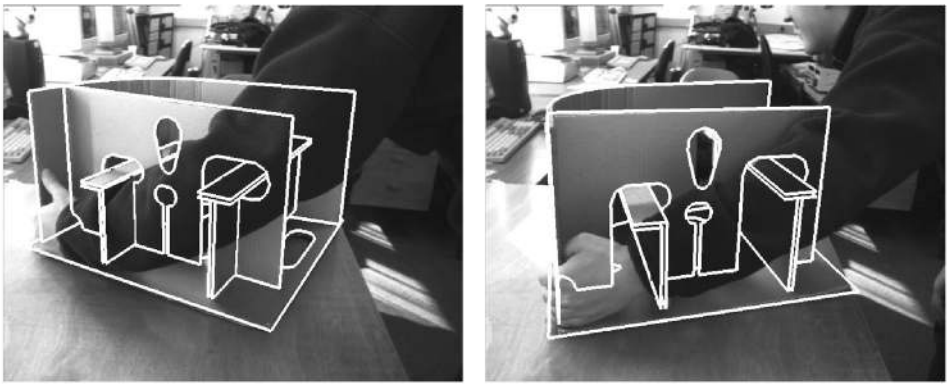


Fig. 6. Frames from tracking sequence with occlusion. The tracking system can provide reasonable estimates of the moving pose despite occlusion of up to approximately half the visible features.

errors. Equivalently, the corresponding Gaussian distribution dies off far too quickly to admit many sample measurements at a large number of standard deviations.

Two standard techniques for handling this problem are to use RANSAC (as in [2]) or to substitute a robust M-estimator for the least-squares estimator by replacing the objective function with one that applies less weighting to outlying measurements [20]. The latter approach is used here for speed and is achieved by modifying the least-squares algorithm and replacing (11) and (12) with:

$$v_i = \sum_{\xi} s(d^{\xi}) d^{\xi} (L_i^{\xi} \cdot \hat{n}^{\xi}), \tag{22}$$

$$C_{ij} = \sum_{\xi} s(d^{\xi}) (L_i^{\xi} \cdot \hat{n}^{\xi}) (L_j^{\xi} \cdot \hat{n}^{\xi}). \tag{23}$$

A common choice for the weighting function, s is:

$$s(d^{\xi}) = \frac{1}{c + |d^{\xi}|}, \tag{24}$$

which corresponds to replacing the Gaussian distribution with one of the form:

$$P(d) = e^{-|d|} \left(1 + \frac{d}{c}\right)^{c}, \tag{25}$$

which behaves like a Gaussian for $d \ll c$ and a Laplacian for $d \gg c$. The parameter c is chosen here to be approximately one standard deviation of the inlying data. This approach is known as iterative reweighted least squares (IRLS) since s depends on d, which changes with each iteration. In the current implementation, only a single iteration is performed for each frame of the video sequence and convergence occurs rapidly over sequential frames. Incorporating IRLS into the system improves its robustness to occlusion (see Fig. 6). The function s controls the confidence with which each measurement is fitted in the least-squares procedure and, thus, can be viewed as representing the *saliency* of the measurement.

2.3.1 Extending IRLS

This can be further exploited by extending IRLS by incorporating a number of additional criteria into the reweighting function. The measures presented here represent a heuristic method for applying saliency criteria to improve tracking performance and have been selected in order to deal with particular problems that have been observed during the operation of this system. The general approach of modifying the reweighting function, s, provides a powerful method of incorporating domain knowledge within the least-squares framework in a conceptually intuitive manner. This could be applied to other saliency criteria that have been developed, for example, based on the shape of the Sum-of-Squared-Differences surface around the proposed feature match [31], [27].

The criteria presented below are chosen to improve the robustness of the system when it is exposed to critical configurations which have been identified as causing instabilities. The saliency or reweighting of each measurement is modified to include four additional terms. The first three of these terms address statistical saliency (*can a feature be detected reliably?*), while the fourth is concerned with analytical saliency (*does the feature constrain the pose estimate?*).

1. **Multiple edges.** When the tracker sees multiple edges within its search range, it is possible for the wrong one to be chosen. Typically, many trackers on the same edge will do this, compounding the problem. To reduce this problem, the saliency is inversely proportional to the number of edge strength maxima visible within the search path.
2. **Many trackers disappear simultaneously.** If an edge of the CAD model runs parallel and near to a boundary of the image, it is possible for a small motion to take the entire edge out of the field of view. This entails a sudden change in the set of trackers used and may cause a sudden apparent motion of the model. This sudden change in the behavior of the tracker can be removed by constructing a border at the edge of the image. The saliency of nodes within this border is weakened linearly to zero as the pixel approaches the edge. A border of 40 pixels has been found to be sufficiently large for this purpose.
3. **Poor visibility.** Generally, the best measurements come from the strongest edges in the image since weak edges may be difficult to locate precisely. This is taken into account by examining the edge strengths found in the search path. If the edge strength along a search path is below a threshold, no measurement is made for that node. Between this threshold and a higher threshold (equal to double the lower one), the saliency of the node is varied

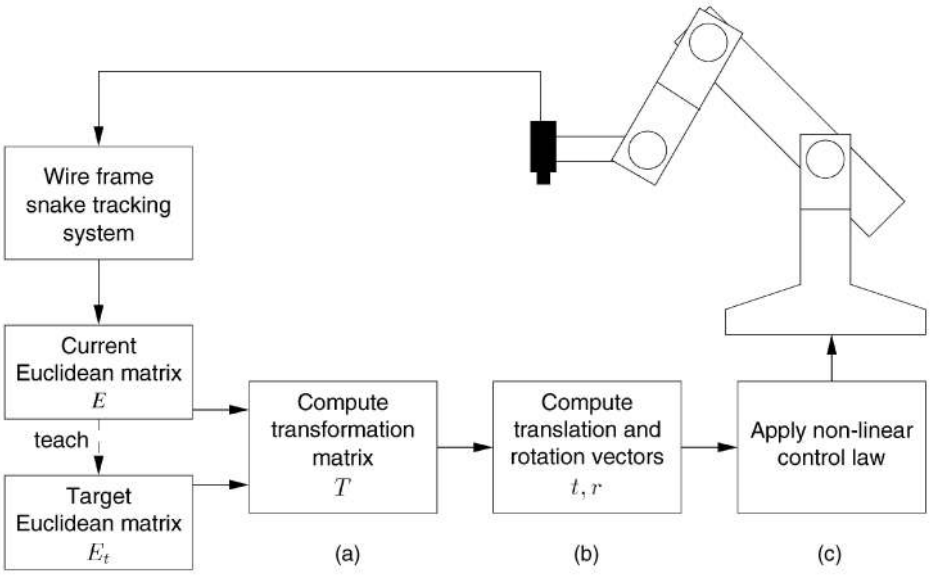


Fig. 7. Visual servoing system operation.

linearly. Above the higher threshold, the visibility does not affect the saliency. These thresholds are chosen manually so that at the upper threshold, matches are dominated by features present in the CAD model rather than by noise.

4. **Weak conditioning.** If the majority of the trackers belong to a single plane of the model (for example the feature rich front plane of the ship part) which is front on to the camera, then the least-squares matrix generated by these nodes becomes more weakly conditioned than in the general configuration. This can be improved by increasing the saliency of measurements that help to condition the least-squares matrix. If the vector comprising the six image motions at node i lies in the subspace spanned by the eigen vectors of C_{ij} corresponding to the smallest eigen values, then that node is particularly important in constraining the estimated motion. This is implemented by the simple expedient of doubling the saliency when $(L_i^\xi \cdot \hat{n}^\xi)(L_j^\xi \cdot \hat{n}^\xi)C_{ij}^{-1}$ is greater than the geometric mean of that quantity, computed over the visible features in the image.

A series of 10 experiments were performed in which the target structure was moved through configurations which exhibit the characteristics described above. Two trackers (one with and one without modified reweighting criteria) were run concurrently on this data. On five of the experiments, the unmodified tracker lost track of the target, while the modified version was able to successfully maintain track on all 10 occasions.

2.4 Visual Servoing System

A visual servoing system has been developed using the visual tracking system described in the previous sections. This system (shown in Fig. 7) takes the Euclidean matrix, E, output from the tracking system and uses this within a nonlinear control law to provide feedback to servo the robot to a stored target pose. These poses are learned using the principle of teaching-by-showing in which the robot is placed into the target pose by the supervisor and records the observed pose given by the tracker, E_t. The inverse of this target matrix, E_t^{-1}, is easily computed and the product of this with the current position matrix yields the transformation from the target position to the current position (Fig. 7a)

$$T = EE_t^{-1}. \tag{26}$$

The translation and rotation vectors that must be applied to the robot are then easily extracted from this representation (Fig. 7b). (here, $i, j, k = 1, 2, 3$):

$$t_i = T_{i4}, \tag{27}$$

$$r'_i = \frac{1}{2}\epsilon_{ijk}T_{jk},$$

$$r_i = \frac{r'_i \sin^{-1}(|r'|)}{|r'|}. \tag{28}$$

The vectors t and r are then multiplied by a gain factor and sent to the robot as end effector translation and rotation velocities (Fig. 7c). The gain is dependent on the magnitudes of t and r so that small velocities are damped to obtain higher precision, while large errors in position may be responded to quickly. A maximum velocity clamp is also applied for safety reasons and to prevent possible instabilities due to latency. Figs. 8 and 9 show the visual servoing system in action, performing closed loop control tracking a moving part and tracing a path between recorded waypoints.

2.5 Results

The tracking system and visual servoing system have been tested in a number of experiments to assess their performance both quantitatively and qualitatively. These experiments were conducted with an SGI O2 workstation (225 MHz) controlling a Mitsubishi RV-E2 robot.

2.5.1 Stability of the Tracker with Respect to Image Noise

The stability of the tracker with a stationary structure was measured to assess the effect of image noise on the tracker. The standard deviation of position and rotation as measured

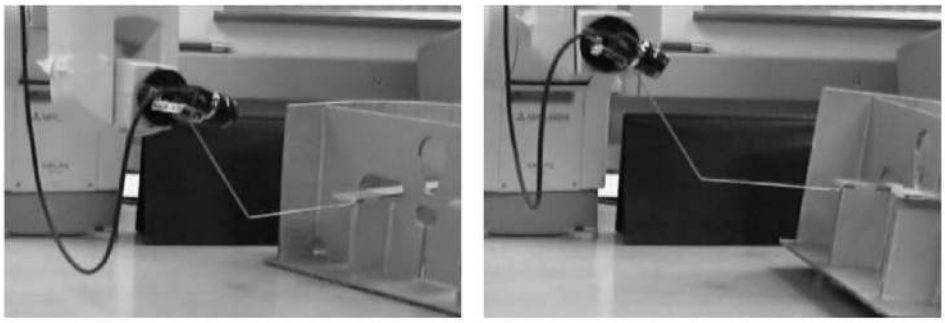


Fig. 8. Closed-loop visual servoing. The task is to maintain a fixed spatial position relative to the workpiece. This can be seen from the presence of the wire which is rigidly mounted relative to the camera and, hence, also maintains a fixed pose relative to the workpiece.

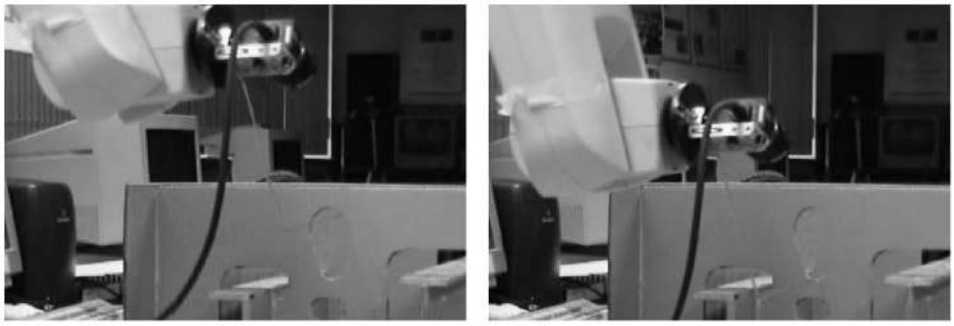


Fig. 9. Visual servoing. The task is to trace out a trajectory relative to the workpiece.

from the Euclidean matrix were measured over a run of 100 frames. From a viewing distance of 30 cm, the apparent r.m.s. translational motion was found to be 0.03 mm with the r.m.s. rotation being 0.015 degrees.

2.5.2 Accuracy of Positioning

The accuracy of positioning the robot was measured with two experiments. First, the ship part was held fixed and the robot asked to home to a given position from a number of different starting points, each of which was a long way from the target position (a few tens of centimeters and a few tens of degrees). When the robot had ceased to move, the program was terminated and the robot's position queried. The standard deviation of these positions was computed and the r.m.s. translational error was found to be 0.08 mm with the r.m.s. rotation being 0.06 degrees. These runs were performed consecutively with the tracker running continuously.

The second experiment was performed by positioning the ship part on an accurate turntable. The part was turned through 15 degrees in one degree rotations and the robot asked to move to the same relative target position each time. Again, the position of the robot was queried and a circle was fitted to the data. The residual error was computed and found to give an r.m.s. positional error of 0.12 mm per measurement (allowing for the three degrees of freedom absorbed into fitting the circle). Again, the tracker was run continuously throughout the experiment and servoing for each stage was performed from the pose attained at the end of the previous stage.

2.6 Online Camera Calibration

The system presented thus far requires that the internal camera parameters (the matrix K in (1)) be known. This section presents a method for extending the tracking system to incorporate estimation and tracking of these parameters online.

The internal characteristics of a pinhole camera can be described with five parameters. These are the focal length, aspect ratio, u and v coordinates of the principal point, and the skew [14]. The matrix of camera parameters is shown in (1). In practice, the skew of the camera is known to be zero and we enforce that condition here. Thus, there are just four parameters to be modeled.

For each of these parameters, there is an associated vector field, just as for motion in space. The vector fields corresponding to the camera parameters can be easily described in terms of the $\binom{u}{v}$ coordinates in the image plane. This creates four new vector fields, $L_i, (7 \leq i \leq 10)$. These are added to the vector fields already used for tracking in the system which then fits a least-squares solution in 10 dimensions instead of six. The resulting system is then able to dynamically track the camera parameters in addition to the motion of the target and is able, for example, to distinguish between motion towards the target and a zoom where there is substantial three-dimensional structure present in the view. Fig. 10 shows model reprojection using (deliberately poor) initial estimates of the camera parameters, together with reprojection using the parameters computed using this method. The initial estimate has a 10 percent error in the aspect ratio and a focal length double that of the lens. Convergence is fast, taking only 5-10 iterations.

One difficulty that can be introduced when tracking of internal camera parameters is activated is that the problem is much more likely to become ill-posed, in which case the matrix C in (12) becomes ill-conditioned. This can occur, for example, when the camera sees only a single plane of image features parallel to the image plane which causes an ambiguity between translation and focal length. Because these critical situations can often occur, internal camera

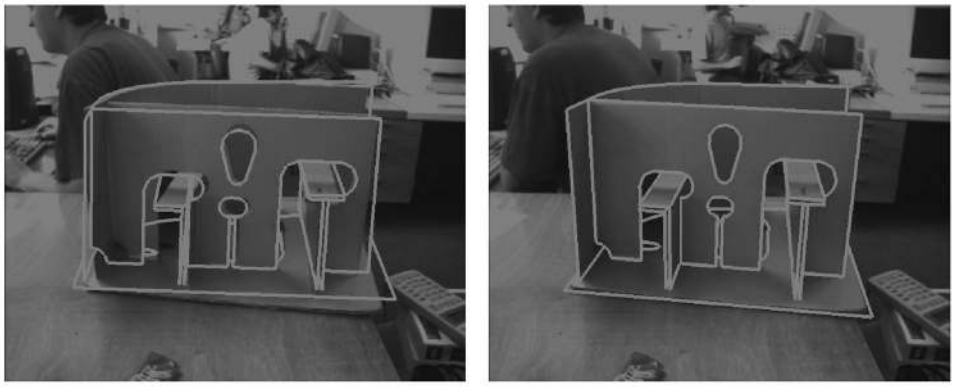


Fig. 10. When camera calibration is activated, the tracking system can adjust the internal camera parameters in order to improve the fit between the model and the image.

calibration is performed in an environment with rich three-dimensional structure and the parameters are then held fixed for the remainder of the task.

2.7 Results

In order to assess the performance of this method of camera calibration, two experiments were conducted.

2.7.1 Stability with Respect to Image Noise

First, the configuration of the camera and a calibration grid were kept fixed and the calibration calculated on a series of runs in order to assess the impact of image noise on the calibration. An 8.5 mm lens was used for this experiment. The mean and standard deviation of the focal length ($f = \sqrt{f_u f_v}$), aspect ratio($a = f_u/f_v$) and principal point (u_0, v_0) over these runs were computed. The results are shown in Table 1. In all cases, the standard deviation was $O(10^{-4})$ times the characteristic scale (for the principal point, this is the focal length).

2.7.2 Variation with Respect to Configuration

Second, a series of runs were performed in which the configuration was varied in order to provide an estimate of the true accuracy of the calibration measurements. A 16 mm lens was used for this experiment and the results are shown in the last column of Table 1. The standard deviation values obtained in this experiment were all less than 1 percent of the characteristic scale (with the exception of the y component of the principal point which was slightly larger). This compares well with a standard calibration technique [14] which was tested with images captured from the sequences and also generated errors of O(1 percent).

3 Complex Configurations

The rigid body tracking system presented in the previous sections is now used as the basis of an approach which is designed to operate in more complex configurations. A unified framework for constructing tracking systems within these configurations is now presented, which takes advantage of the formulation and computational operation of the rigid body tracker. Such configurations arise in a number of ways:

Multiple cameras. It is often desirable to use more than one camera to obtain information about a scene since multiple view configurations can provide higher pose precision (especially when a large baseline is used) and also increase the robustness of the tracker.

Multiple targets. There are many situations in which knowing the relationship between the camera and a single target is insufficient. This occurs particularly when the position of the camera is not of direct interest. In these situations, it is often desirable to measure the relationship between two or more targets that are present in the scene, for example, between two vehicles and the road, or between a robot tool and its workpiece.

Articulated targets. Many targets of interest are not simple rigid bodies, but contain internal degrees of freedom. This paper only considers targets which comprise a number of rigid components connected by hinges or slides, etc.

All of these configurations can be handled using a common approach in which multiple instances of the rigid body tracker are executed concurrently, one per component per camera. Because this naïve approach introduces more degrees of freedom into the system than are really present, it is necessary to couple the rigid body trackers together in

TABLE 1
Results of Camera Calibration Experiments Showing the Stability with Respect to Image Noise and the Variation in Estimate Across Different Configurations

	fixed configuration	varying configuration
lens	8.5 mm	16 mm
focal length	785.55 ± 0.05	1442.2 ± 7.1
aspect ratio	0.950055 ± 0.00004	0.94995 ± 0.00067
principal point	$(336.4 \pm 0.2, 278.8 \pm 0.5)$	$(355 \pm 10, 282 \pm 24)$

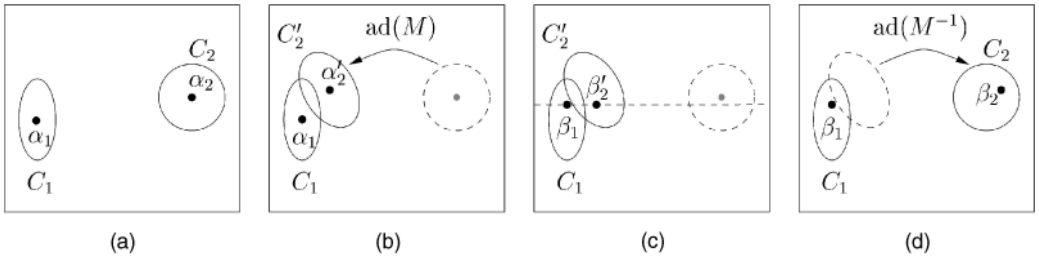


Fig. 11. Applying the constraints: Estimates and errors are computed for motions 1 and 2 (a), the estimate and error of motion 2 are mapped into 1's coordinate frame (b), the constraint is applied there (c), and then the new estimate of motion 2 is mapped back into its own frame (d).

order to condition the problem and also ensure that the solution corresponds to a physically correct configuration.

For example, three cameras viewing two structures would require six concurrent trackers. Even if the cameras and structures can move independently, there are only 24 degrees of freedom in the world, whereas the system of six trackers contains 36. The natural approach to this problem is to impose all of the constraints that are known about the world upon the tracking system.

3.1 Constrained Tracking

Multiple Cameras. In the case in which multiple cameras are used to view a scene, it may be that the cameras are known to be rigid relative to one another in space. In this case, there are six constraints that can be imposed on the system for every camera additional to the first.

Multiple structures. Where the system is being used to track multiple structures, it is often the case that other constraints apply between the structures. For example, two cars will share a common ground-plane and, thus, a system in which two vehicles observed from an airborne camera will have three constraints that apply to the raw 12 dimensions present in the two trackers, reflecting the nine degrees of freedom present in the world.

Articulated structures. This is really a special case of constrained multiple structures, except that there are usually more constraints. A hinged structure, for example, has seven degrees of freedom (six for position in the world and one for the angle of the hinge). When the two components of the structure are independently tracked, there are five hinge constraints which apply to the system.

Because these constraints exist in the world, it is highly desirable to impose them on the system of trackers. Each of the trackers generates an estimate for the motion of one rigid component in a given view, α_i in (13) as well as a matrix C_{ij} in (12), which describes how the error varies around that estimate. Thus, the goal is to use both of these pieces of information from each tracker to obtain a global maximum a posteriori estimate of the motion subject to satisfying the known constraints. This raises three issues which must be addressed:

1. Measurements from different trackers are made in different coordinate frames.
2. How can the constraints be expressed?
3. How can they then be imposed?

3.1.1 Coordinate Frames

The first difficulty is that the α_i and the C_{ij} are quantities in the Lie algebra deriving from the coordinate frame of the object being tracked. Since these are not the same, in general, for distinct trackers, a method for transforming the α_i and C_{ij} from one coordinate frame to another is needed. Specifically, this requires knowing what happens to the Lie algebra of SE(3) under $\mathbb{R}^3$ coordinate frame changes. Since these frame changes correspond to elements of the Lie group SE(3), this reduces to knowing what happens to the Lie algebra of the group under conjugation by elements of the group. This is (by definition) the adjoint representation of the group which is a $n \times n$ matrix representation, where n is the dimensionality of the group (six in the case of SE(3)). The adjoint representation, $\mathrm{ad}(M)$, for a matrix element of SE(3), M, can easily be computed by considering the action of M on the group generators, G_i, by conjugation:

$$MG_iM^{-1} = \sum_j \mathrm{ad}(M)_{ij} G_j. \tag{29}$$

If (with a slight abuse of notation) $M = [R|t]$, this is given by

$$\mathrm{ad}(M) = \begin{bmatrix} R & [t_\wedge]R \\ 0 & R \end{bmatrix} \qquad \text{where} [t_\wedge]_{ij} = \varepsilon_{ijk} t_k. \tag{30}$$

To see that these 6×6 matrices do form a representation of SE(3), it is only necessary to ensure that multiplication is preserved under the mapping into the adjoint space (that $\mathrm{ad}(M_1)\mathrm{ad}(M_2) = \mathrm{ad}(M_1 M_2)$) which can easily be checked using the identity $R_1[t_{2\wedge}]R_1^{-1} = [R_1 t_{2\wedge}]$. Thus, if M transforms points from coordinate frame 1 into frame 2, then $\mathrm{ad}(M)$ transforms a vector in the Lie algebra of frame 1 into the Lie algebra of frame 2. Using this, the quantities in (11), (12), and (13) can be transformed as follows (see Figs. 11a and 11b):

$$v' = \mathrm{ad}(M)^{-T}, v \tag{31}$$

$$\alpha' = \mathrm{ad}(M)\, \alpha, \tag{32}$$

$$C' = \mathrm{ad}(M)\, C\, \mathrm{ad}(M)^T. \tag{33}$$

3.1.2 Expressing Constraints

It is useful to have a generic method for expressing the constraints that are present on the given world configuration since this increases the speed with which models for new situations may be constructed. In the Lie algebra formalism, it is very easy to express the constraints that describe a hinge, a slide, or the existence of a common ground plane since the relationship between velocities in the algebra and the constraints is a simple one.

The presence of a hinge or common ground plane are holonomic constraints which reduce the dimensionality of the configuration space by five and three, respectively. This

results in a seven- or nine-dimensional submanifold representing legal configurations embedded within the raw 12-dimensional configuration manifold of the two rigid components. The tangent space to this submanifold corresponds to the vector space of velocities which locally respect the constraint. This means that at each legal configuration there is a linear subspace of legal velocities, which implies that the constraints on the velocities must be linear (and, also, homogeneous since zero velocity results in a legal configuration). Thus, if β_1 and β_2 correspond to the motions of the two rigid components (in their Lie algebras), then the constraints must take the form

$$\beta_1 \cdot c_1^i + \beta_2 \cdot c_2^i = 0. \tag{34}$$

There must be five such c_1 and c_2 for the hinge or three for the common ground plane. As a simple example, consider the case of a hinge in which the axis of rotation passes through the origin of component 1's coordinate frame and lies along its z axis. When the motions of the two parts are considered in 1's frame, then their translations along all three axes must be the same as must their rotations about the x and y axes; only their rotations about the z axis can differ. Since component 2's motion can be transformed into 1's coordinate frame using the adjoint representation of the coordinate transformation, the constraints now take the form

$$\beta_1 \cdot c_1^i + \beta_2' \cdot c_2^i = 0, \tag{35}$$

where $\beta_2' = \mathrm{ad}(E_1^{-1}E_2)\beta_2$ is the motion of component 2 in 1's frame. In this example, the c_1 and c_2 vectors for the five constraints become particularly simple:

$$c_1^i = \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix} \qquad (1 \le i \le 5) \tag{36}$$

with $c_2^i = -c_1^i$. Constraints 1, 2, and 3 say that the x, y, and z axis translations of the two components must be the same when measured in the coordinate frame of component 1. Further, constraints 4 and 5 say that the rotations about the x and y axes of this coordinate frame must also be the same. Thus, the only thing that is permitted to differ is the rotation about the z axis in this coordinate frame which corresponds to articulation of the hinge.

In the case of a common ground plane in 1's x-y plane, only constraints 3, 4, and 5 are needed. If the hinge or ground plane are placed elsewhere, then the adjoint representation can be used to transform the constraints by considering the Euclidean transformation which takes that situation back to the simple one.

3.1.3 Imposing Constraints

Since the constraints have a particularly simple form, finding the optimal β_1 and β_2' is also an easy matter. This is done by modifying the least-squares fitting procedure used for the single tracker which is adapted so that the motion which gives the least-square error *subject to satisfying the constraints* is found. Given the α and C computed in (11), (12), and (13), then (20) gives the increase in sum squared error if the motion β is used in place of α as $(\beta-\alpha)C(\beta-\alpha)$. Thus, given the independent solutions for the two motions (α_1, C_1) and (α_2', C_2'), the aim is to find β_1 and β_2' such that

$$(\beta_1-\alpha_1)C_1(\beta_1-\alpha_1) + (\beta_2'-\alpha_2')C_2'(\beta_2'-\alpha_2') \tag{37}$$

$$\text{is minimized subject to} \qquad \beta_1 \cdot c_1^i + \beta_2' \cdot c_2^i = 0. \tag{38}$$

This is a constrained optimization problem and ideal for solving by means of Lagrange multipliers. Thus, the solution is given by the constraints in (38) and

$$\begin{aligned}&\nabla\big((\beta_1-\alpha_1)^T C_1(\beta_1-\alpha_1) + (\beta_2'-\alpha_2')^T C_2'(\beta_2'-\alpha_2')\big)\\ &\quad + \lambda_i \nabla\big(\beta_1^T c_1^i + \beta_2'^T c_2^i\big) = 0\end{aligned} \tag{39}$$

with ∇ running over the 12 dimensions of $\begin{pmatrix}\beta_1\\ \beta_2'\end{pmatrix}$. This evaluates to

$$\begin{pmatrix}2C_1(\beta_1-\alpha_1)\\ 2C_2(\beta_2'-\alpha_2')\end{pmatrix} + \lambda_i \begin{pmatrix}c_1^i\\ c_2^i\end{pmatrix} = 0. \tag{40}$$

$$\begin{aligned}\text{Thus,} \qquad \beta_1 &= \alpha_1 - \frac{1}{2}C_1^{-1}\lambda_i c_1^i\\ \text{and} \qquad \beta_2' &= \alpha_2' - \frac{1}{2}C_2'^{-1}\lambda_i c_2^i.\end{aligned} \tag{41}$$

Substituting (38) back into (41) gives

$$c_1^i \cdot \alpha_1 + c_2^i \cdot \alpha_2' - \frac{1}{2}\lambda_j\Big(c_1^i \cdot C_1^{-1}c_1^j + c_2^i \cdot C_2'^{-1}c_2^j\Big) = 0. \tag{42}$$

So, the λ_i are given by

$$A_{ij} = c_1^i \cdot C_1^{-1}c_1^j + c_2^i \cdot C_2'^{-1}c_2^j, \tag{43}$$

$$l_i = 2\big(c_1^i \cdot \alpha_1 + c_2^i \cdot \alpha_2'\big), \tag{44}$$

$$\lambda_i = A_{ij}^{-1}l_j. \tag{45}$$

The λ_i can then be substituted back into (41) to obtain β_1 and β_2' (see Fig. 11c), from which β_2 can also be obtained by $\beta_2 = \mathrm{ad}(E_2^{-1}E_1)\beta_2'$ (see Fig. 11d). The β can then be used to update the configurations of the two rigid parts of the hinged structure giving the configuration with the least-square error that also satisfies the constraints.

4 Results

4.1 Multiple Cameras

A multicamera system was developed using up to three cameras multiplexed using the red, green, and blue components of a 4:2:2 digital signal to track the pose of a rigid structure. A number of experiments were conducted using this multiple camera configuration. In which the cameras are known to be fixed relative to each other. These are the only experiments in which PAL frame rate (25 Hz) was not achieved, with typical performance being around 60-75 percent of frame rate (14-19 Hz) when all three cameras were in use.

4.1.1 Accuracy

An experiment was conducted in order to assess how the use of multiple cameras can improve accuracy. In a single view, it is usually the case that the estimate of the position of a structure is much worse along the camera's optical axis than in orthogonal directions because the position of features in the image varies more slowly with motion along this axis. In this experiment, the rms error in estimated position using a single camera was 0.97 mm with an rms error in orientation of

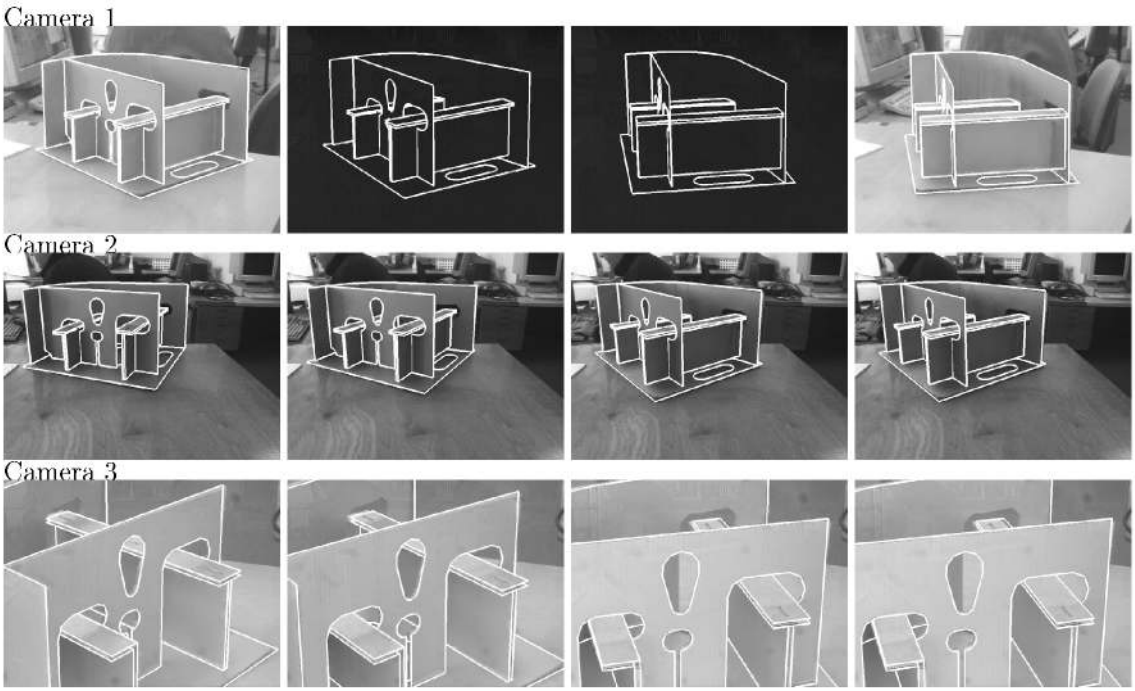


Fig. 12. Experiment in which Camera 1 is completely occluded for a time in the middle of the sequence. The system maintains an estimate of the pose of the model in this view and when the camera is disoccluded, the estimate is found to match the image.

0.0028 radians. By placing a second camera so that it views the structure from an orthogonal direction, the position of the part can be known to much higher precision. Combining the information from the two cameras reduced the rms estimate of position to 0.29 mm and the rms error in orientation to 0.0018 radians. This large improvement is due to the fact that the error distributions from the two cameras do not have axes of large variation in common.

4.1.2 Robustness

Another experiment was established to test robustness of the system to occlusion of an entire camera (see Fig. 12). During tracking, one of the cameras was covered and the model moved. An estimate for the position of the target structure in that view is maintained and when the camera was uncovered, this estimate was found to be accurate and tracking in that view resumed automatically.

4.1.3 Coverage

Because the system is robust to complete occlusion of any given view, this capability can be exploited in order to use multiple cameras to give increased coverage. Fig. 13 shows the view from each of three cameras at four times during a tracking sequence. In the second of these, the structure has left the field of view of the first camera. The system still has an estimate of the position because it is still visible to the second and third cameras. When the structure reenters the field of view of the first camera, tracking in that view re-commences automatically. In the fourth set of views, the structure is no longer visible in camera three. Thus, the system is able to use the constraints to effect hand-over between cameras covering different (but overlapping) regions of space.

4.2 Multiple and Articulated Structures

4.2.1 Hinge

A system was developed to test the tracking of a simple articulated structure (shown in Fig. 14a). This structure consists of two components, each 15 cm square, joined along one edge by a hinge. This structure is a difficult one to track since there are barely enough degrees of freedom in the image of the structure to constrain the parameters of the model. An experiment was conducted to examine the precision with which the system can estimate the angle between parts of the model with and without the hinge constraints imposed. The hinge of the part was moved through a series of known angles by hand and tracked throughout this process. At each angle, the tracker was queried to obtain its estimate of the angle between the two components. This experiment was performed twice, once with the hinge constraint imposed and once without. On the second run, the translational error at the midpoint of the hinge was also measured. Several measurements were taken at each known angle during both runs and the results are shown in Table 2.

In all cases, the estimate produced by the constrained tracker was within 1^o of the ground truth. The unconstrained (12 DoF) tracker was much less accurate, in general, and also reported substantial errors in violation of the known constraints. The variance in the angle estimate gives an indication of the stability of the tracker and it can be seen that the use of constraints improves this significantly. Fig. 14b shows the behavior of the unconstrained tracker. Because of the difficulty in finding the central crease, this tracker becomes weakly conditioned and noise fitting can introduce large errors.

4.2.2 Double Hinge

This system was then extended to track the structure with an additional square component and hinge (see Fig. 15a). The system is able to track the full configuration of the structure, even when the central component is fully hidden from view (see Fig. 15b). In this case, the observed positions of the two visible components are sufficient to determine the location of

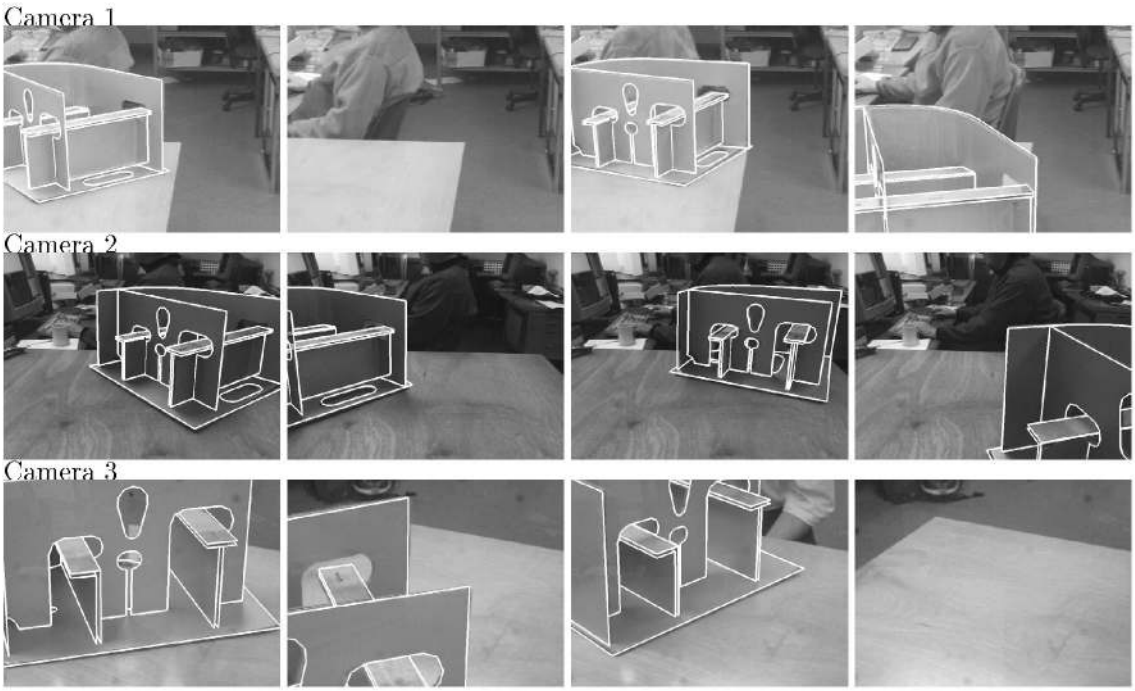


Fig. 13. Tracking sequence showing three views at four difference times during the sequence. The model leaves and reenters the view of camera 1 which continues tracking without need for reinitialization.

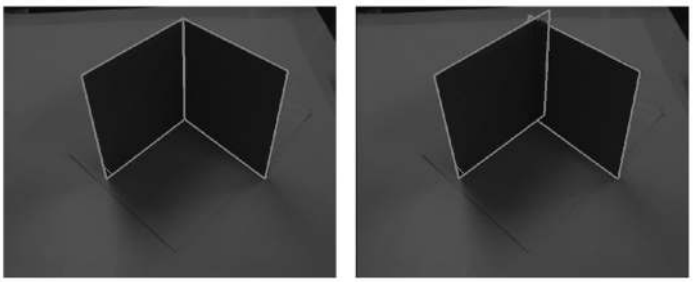


Fig. 14. Tracking a hinge. When the constraints are relaxed, the tracker falls into an erroneous minimum due to the lack of visibility of the central crease.

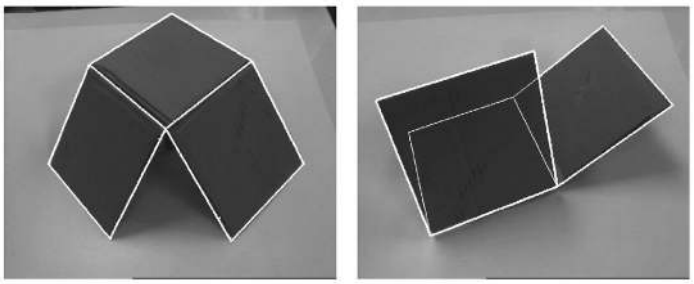
the hidden part. Further, the propagated constraints between the two end parts of the structure serve to improve the conditioning of the estimation of their positions.

4.2.3 Filing Cabinet

In order to show the system operating with more complex objects and also to illustrate operation with sliding rather than rotating joints, the system was applied to tracking a filing cabinet. The drawer is tracked separately but is constrained to a single axis of translation relative to the cabinet. Nonholonomic constraints corresponding to the limits of motion of the drawer have also been imposed. The first two frames in Fig. 16 show a tracking sequence with this configuration. The constraint is important for stable tracking of the drawer; when this is turned off, the estimate

TABLE 2
Results of Experiment Measuring Accuracy of Constrained and Unconstrained Trackers

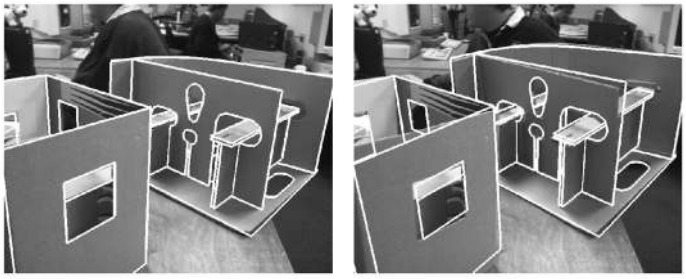
Ground truth ($\pm 1^o$)	Constrained	Unconstrained	R error	T error
80^o	$79.2^o \pm 0.12^o$	— Tracking Failed —		
90^o	$90.29^o \pm 0.14^o$	$94.46^o \pm 0.53^o$	2.97^o	2.32cm
100^o	$99.3^o \pm 0.11^o$	$102.56^o \pm 0.32^o$	4.55^o	2.76cm
110^o	$110.07^o \pm 0.11^o$	$111.34^o \pm 0.34^o$	5.75^o	3.23cm
120^o	$119.31^o \pm 0.05^o$	$119.09^o \pm 0.2^o$	3.95^o	1.43cm
130^o	$130.15^o \pm 0.08^o$	$128.77^o \pm 0.18^o$	1.38^o	1.35cm



(a) (b)

Fig. 15. Double hinge structure: The tracker can infer the position of a hidden component from the constraints.

Fig. 16. Two frames from sequence tracking seven DoF filing cabinet. The third frame shows the tracker in a local minimum after relaxation of the articulation constraints.



(a) (b)

Fig. 17. Two structures with common ground plane constraint: When the world violates, the constraint the system attempts to find a solution. In this case, by fitting parts of both structures.

of drawer position is likely to fall into a local minimum (as is seen in the final frame).

4.2.4 Common Ground Plane

A system was also developed to show that constraints of intermediate complexity such as the existence of a common ground plane can be implemented within this framework. The system can dynamically impose or relax the common ground plane constraint between the two structures shown in Fig. 17a. The presence of this constraint reduces the dimensionality of the tracker from 12 to 9. The two frames in Fig. 17 come from a sequence in which the constraint is deliberately violated by rotating one of the two components out of the ground plane. The use of a robust method means that the error is not simply shared between all visible features, but, in this case, a minimum has been found in which some features from each of the two components are fitted as the constraint is enforced.

5 Conclusion and Future Directions

This paper has presented an general framework for real-time three-dimensional tracking of complex structures. The system has been implemented and been shown to exhibit sufficient accuracy for many useful tasks, such as robot control. The formulation used is extensible, as has been demonstrated by the incorporation of real-time online camera calibration which yields accuracy comparable to existing techniques.

The use of Lie algebras for representing differential quantities within a rigid body tracker has facilitated the construction of systems which operate in more complex and constrained configurations. Within this representation, it is easy to transform rigid body tracking information between coordinate frames using the adjoint representation and also to express and impose the constraints corresponding to the presence of hinges or a common ground plane. This yields benefits in terms of ease of programming and implementation, which, in turn, make it readily possible to achieve real-time frame rate performance using standard hardware.

The system currently has two main limitations. It depends on coarse hand localization to begin tracking and it can only handle piecewise rigid polyhedral structures. Future work will be aimed at addressing these points.

ACKNOWLEDGMENTS

This work was supported by an EC (ESPRIT) grant no. LTR26247 (VIGOR) and by an EPSRC grant no. K84202.

REFERENCES

[1] J.K. Aggarwal, Q. Cai, W. Liao, and B. Sabata, "Nonrigid Motion Analysis: Articulated and Elastic Motion," *Computer Vision and Image Understanding,* vol. 70, no. 2, pp. 142-156, 1998.
[2] M. Armstrong and A. Zisserman, "Robust Object Tracking," *Proc. Second Asian Conf. Computer Vision,* pp. 58-62, 1995.
[3] R. Basri, E. Rivlin, and I. Shimshoni, "Visual Homing: Surfing on the Epipoles," *Proc. Int'l Conf. Computer Vision (ICCV '98),* pp. 863-869, 1998.
[4] M.J. Black and A.D. Jepson, "Eigen Tracking: Robust Matching and Tracking of Articulated Objects Using a View Based Representation," *Proc. European Conf. Computer Vision '96,* vol. 1, pp. 329-342, 1996.
[5] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," *Proc. Computer Vision and Pattern Recognition '98,* pp. 8-15, 1998.
[6] T.-J. Cham and J.M. Rehg, "A Multiple Hypothesis Approach to Figure Tracking," *Proc. Computer Vision and Pattern Recognition '99,* vol. 2, pp. 239-245, July 1999.
[7] R. Cipolla and A. Blake, "The Dynamic Analysis of Apparent Contours," *Proc. IEEE Third Int'l Conf. Computer Vision,* pp. 616-623, Dec. 1990.
[8] R. Cipolla and A. Blake, "Image Divergence and Deformation from Closed Curves," *Int'l J. Robotics Research,* vol. 16, no. 1, pp. 77-96, 1997.
[9] N. Daucher, M. Dhome, J. T. Lapresté, and G. Rives, "Modelled Object Pose Estimation and Tracking by Monocular Vision," *Proc. British Machine Vision Conf.,* pp. 249-258, 1993.
[10] Q. Delamarre and O. Faugeras, "3D Articulated Models and Multi-View Tracking with Silhouttes," *Proc. Int'l Conf. Computer Vision '99,* vol. 2, pp. 716-721, Sept. 1999.
[11] T. Drummond and R. Cipolla, "Real-Time Tracking of Multiple Articulated Structures in Multiple Views," *Proc. Sixth European Conf. Computer Vision,* vol. 2, pp. 20-36, June 2000.
[12] T. Drummond and R. Cipolla, "Application of Lie Algebras to Visual Servoing," *Int'l J. Computer Vision,* vol. 37, no. 1, pp. 21-41, 2000.
[13] B. Espiau, F. Chaumette, and P. Rives, "A New Approach to Visual Servoing in Robotics," *IEEE T-Robotics and Automation,* vol. 8, no. 3, 1992.
[14] O.D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, 1993.
[15] D.M. Gavrila and L.S. Davis, "3-D Model-Based Tracking of Humans in Action: A Multi-View Approach," *Proc. Computer Vision and Pattern Recognition '96,* pp. 73-80, 1996.
[16] M. Haag and H.-H. Nagel, "Tracking of Complex Driving Manoeuvres in Traffic Image Sequences," *Image and Vision Computing,* vol. 16, pp. 517-527, 1998.
[17] G. Hager, G. Grunwald, and K. Toyama, "Feature-Based Visual Servoing and Its Application to Telerobotics," *Intelligent Robotic Systems,* V. Graefe, ed., Elsevier, 1995.
[18] C. Harris, "Geometry from Visual Motion," *Active Vision,* A. Blake and A. Yuille, eds., chapter 16, pp. 263-284, MIT Press, 1992.
[19] C. Harris, "Tracking with Rigid Models," *Active Vision,* A. Blake and A. Yuille, eds., chapter 4, pp. 59-73, MIT Press, 1992.
[20] P.J. Huber, *Robust Statistics.* Wiley, 1981.
[21] S. Hutchinson, G.D. Hager, and P.I. Corke, "A Tutorial on Visual Servo Control," *IEEE T-Robotics and Automation,* vol. 12, no. 5, pp. 651-670, 1996.
[22] M. Isard and A. Blake, CONDENSATION—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.
[23] D.G. Lowe, "Fitting Parameterised 3-D Models to Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 5, pp. 441-450, 1991.
[24] D.G. Lowe, "Robust Model-Based Motion Tracking through the Integration of Search and Estimation," *Int'l J. Computer Vision,* vol. 8, no. 2, pp. 113-122, 1992.
[25] J. MacCormick and A. Blake, "Spatial Dependence in the Observation of Visual Contours," *Proc. Fifth European Conf. Computer Vision (ECCV '98),* pp. 765-781, 1998.
[26] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust Real-Time Visual Tracking Using a 2D-3D Model-Based Approach," *Proc. Int'l Conf. Computer Vision ,'99,* vol. 1, pp. 262-268, Sept. 1999.
[27] N. Papanikolopoulos, "Selection of Features and Evaluation of Visual Measurements During Robotic Visual Servoing Tasks," *J. Intelligent Robotic Systems,* vol. 13, pp. 279-304, 1995.
[28] M. Paterson and F. Yao, "Efficient Binary Space Partitions for Hidden Surface Removal and Solid Modeling," *Discrete and Computational Geometry,* vol. 5, no. 5, pp. 485-503, 1990.
[29] A.C. Sanderson, L.E. Weiss, and C.P. Neumann, "Dynamic Sensor Based Control of Robots with Visual Feedback," *IEEE J. Robotics and Automation,* vol. 3, pp. 404-417, 1987.
[30] D.H. Sattinger and O.L. Weaver, *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics.* Springer-Verlag, 1986.
[31] J. Shi and C. Tomasi, "Good Features to Track," *Proc. Conf. Computer Vision and Pattern Recognition (CVPR '94),* pp. 593-600, June 1994.
[32] C. Taylor and D. Kriegman, "Minimization on the Lie Group SO(3) and Related Manifolds," Technical Report 9405, Yale Univ., Apr. 1994.
[33] D. Terzopoulos and R. Szeliski, "Tracking with Kalman Snakes," *Active Vision,* A. Blake and A. Yuille, eds., chapter 1, pp. 3-20, MIT Press, 1992.
[34] W.J. Wilson, C.C. Williams Hulls, and G.S. Bell, "Relative End-Effector Control Using Cartesian Position Based Visual Servoing," *IEEE T-Robotics and Automation,* vol. 12, no. 5, pp. 684-696, 1996.
[35] A.D. Worrall, G.D. Sullivan, and K.D. Baker, "Pose Refinement of Active Models Using Forces in 3D," *Proc. Third European Conf. Computer Vision (ECCV '94),* J. Eklundh, ed., vol. 2, pp. 341-352, May 1994.
[36] P. Wunsch and G. Hirzinger, "Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion," *Proc. 1997 Int'l Conf. Robotics and Automation,* pp. 2868-2873, 1997.

Tom Drummond received the BA degree in mathematics from the University of Cambridge in 1988. From 1989 to 1998, he studied and worked in Australia and in 1998 received the PhD degree from Curtin University in Perth, Western Australia. In 1994, he joined the Department of Engineering at the University of Cambridge as a research associate. In 2001, he was appointed as a university lecturer. His research interests are in computer vision and robotics and include real-time visual tracking, visual servoing, and augmented reality. He is a member of the IEEE Computer Society.

Roberto Cipolla received the BA degree in engineering from the University of Cambridge in 1984 and the MSE degree in electrical engineering from the University of Pennsylvania in 1985. From 1985 to 1988, he studied and worked in Japan at the Osaka University of Foreign Studies (Japanese Language) and Electrotechnical Laboratory, Tsukuba (visiting scientist), and he received the MEng degree in robotics from the University of Electro-Communications in Tokyo in 1988. In 1991, he was received the DPhil degree in computer vision from the University of Oxford. From 1991 to 1992, he was a Toshiba Fellow and engineer at the Toshiba Corporation Research and Development Centre in Kawasaki, Japan. He joined the Department of Engineering, University of Cambridge in 1992 as a lecturer and a fellow of Jesus College. He became a reader in information engineering in 1997 and a professor in 2000. His research interests are in computer vision and robotics and include the recovery of motion and 3D shape of visible surfaces from image sequences, visual tracking and navigation, robot hand-eye coordination, algebraic and geometric invariants for object recognition and perceptual grouping, novel man-machine interfaces using visual gestures, and visual inspection. He has authored three books, edited five volumes, and coauthored more than 150 papers. He is a member of the IEEE and the IEEE Computer Society.