Real-time tracking of multiple articulated structures in multiple views

Tom Drummond and Roberto Cipolla

Department of Engineering, University of Cambridge Trumpington Street,
Cambridge, UK, CB2 1PZ
{twd20, cipolla}@eng.cam.ac.uk

Abstract. This paper describes a highly flexible approach to real-time frame-rate tracking in complex camera and structures configurations, including the use of multiple cameras and the tracking of multiple or articulated targets. A powerful and general method is presented for expressing and solving the constraints which exist in these configurations in a principled manner. This method exploits the geometric structure present in the Lie group and Lie algebra formalism to express the constraints that derive from structures such as hinges or a common ground plane. This method makes use of the adjoint representation to simplify the constraints which are then applied by means of Lagrange multipliers.

1 Introduction

The tracking of known three-dimensional objects is useful for numerous applications, including motion analysis, surveillance and robotic control tasks. This paper presents a novel approach to visual tracking in complex camera and structure configurations, including the use of multiple cameras and the tracking of multiple structures with constraints or of articulated structures. Earlier work in the tracking of rigid bodies [1] which employs a Lie group and Lie algebra formalism is exploited in order to simplify the difficulties that arise in these more complex situations and thus provide a real-time frame-rate tracking system.

The paper first reviews work on the tracking of rigid bodies and then describes the Lie group and Lie algebra formalism used within the rigid body tracking system which is used as the basis for more complex configurations. It then shows how this formalism provides a powerful means of managing complex multi-component configurations; the transformation of measurements made in differing co-ordinate frames is simplified as is the expression of constraints (e.g. hinge or slide) that are present in the system. These constraints can then be imposed by means of Lagrange multipliers. Results from experiments with real-time frame-rate systems using this framework are then presented and discussed.

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 salient information if real-time frame (or field) rate performance is to be achieved [2]. This observation leads to the notion of *feature based* tracking [3] in which processing is restricted to locating strong image features such as contours [4, 5].

A number of successful systems have been based on tracking the image contours of a known model. Lowe [6] 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 [7]. The use of 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 [8], CONDENSATION [9] and other systems [10, 11, 12]. The efficiency yielded by this approach allows all these systems to run in real-time on standard workstations. The approach is also used here.

Using either of these approaches, most systems (except CONDENSATION) then compute the pose parameters by linearising with respect to image motion. This process is reformulated here in terms of the Lie group SE(3) and its Lie algebra (see [13, 14] for a good introduction to Lie groups and their algebras). This formulation is a natural one to use since SE(3) exactly represents the space of poses that form the output of a system which tracks a rigid body. Differential quantities such as velocities and small motions in the group then correspond to the Lie algebra of the group (which is the tangent space to the identity). Thus the representation provides a canonical method for linearising the relationship between image motion and pose parameters. Further, this approach can be generalised to other transformation groups and has been successfully applied to deformations of a planar contour using the groups GA(2) and P(2) [15].

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 [16], factored sampling [9] or regularisation, for example the Levenberg-Marquadt scheme used in [6]. The approach used here employs iterative re-weighted least squares (a robust M-estimator).

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 pre-computed table of visible features indexed by what is essentially a view-sphere. By contrast, the system presented here uses graphical rendering techniques [17] 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 [16] which discards primitives for which insufficient support is found, or by modifying statistical descriptions of the observation model (as in [18]). If a model is available for the intervening object, then it is possible to use this to re-estimate the visible features [19, 7]. Both of these methods are used within the system presented here.

1.2 Articulated Structures

A taxonomy of non-rigid motion is given in [20]. This paper is only concerned with what is classified as *articulated motion*, which can be characterised as comprising rigid components connected by simple structures such as hinges, slides etc.

Lowe [21] also considered articulated motion, which was implemented by means of internal model parameters which are stored in a tree structure. By contrast, the approach presented here uses a 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. A similar approach has been taken for tracking people [22] which relies on prior extraction of accurate silhouettes in multiple synchronised views of each frame which are then used to apply forces on the components of the three dimensional model.

2 Tracking a Rigid Structure in a Single View

This section will review the rigid body tracking system which is used as a basis for the extensions which are presented in this paper. The approach used here for tracking a known 3-dimensional structure is based upon maintaining an estimate of the camera projection matrix, P, in the co-ordinate system of the structure. This projection matrix is represented as the product of a matrix of internal camera parameters:

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

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

$$E = \begin{bmatrix} R\, t \end{bmatrix} \qquad \text{with } RR^T = I \text{ and } |R| = 1 \tag{2}$$

The projective co-ordinates 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 co-ordinates 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 3-dimensional space, which is a 6-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:

$$\begin{gathered} G_1 = \begin{bmatrix} 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix}, G_2 = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix}, G_3 = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \end{bmatrix}, \\ G_4 = \begin{bmatrix} 0&0&0&0 \\ 0&0&-1&0 \\ 0&1&0&0 \\ 0&0&0&0 \end{bmatrix}, G_5 = \begin{bmatrix} 0&0&1&0 \\ 0&0&0&0 \\ -1&0&0&0 \\ 0&0&0&0 \end{bmatrix}, G_6 = \begin{bmatrix} 0&-1&0&0 \\ 1&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix} \end{gathered} \tag{6}$$

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:

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

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 inter-frame transformation. Since the motion will be small, M can be approximated by the linear terms:

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

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

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

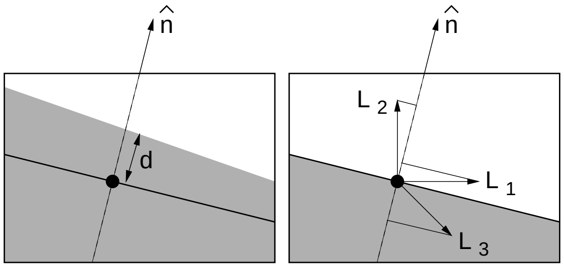


Fig. 1. Computing the normal component of the motion and generator vector fields

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 co-ordinates. A least-squares approach can then be used to fit the observed motion of image features between adjacent frames. This process is detailed in Section 2.1.

The features used in this work for tracking are the edges that are present in the model. 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 localised within an image. This choice also makes it possible to take advantage of the aperture problem (that the component of motion of an edge, tangent to itself, is not observable locally), since it allows the use of one-dimensional search along the edge normal (see Figure 1). 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 projections of the f_i.

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 [17] 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 Figure 2. As rendering takes place, the 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, tracking nodes are assigned to search for the nearest intensity discontinuity in the video feed along the edge normal (see Figure 4).

Figure 3 shows system operation. At each cycle, the system renders the expected view of the object (a) using its current estimate of the projection matrix, P. The visible edges are identified and tracking nodes are assigned at regular intervals in image co-ordinates along these edges (b). The edge normal is then searched in the video feed for a nearby edge (c). Typically $m \approx 400$ nodes are

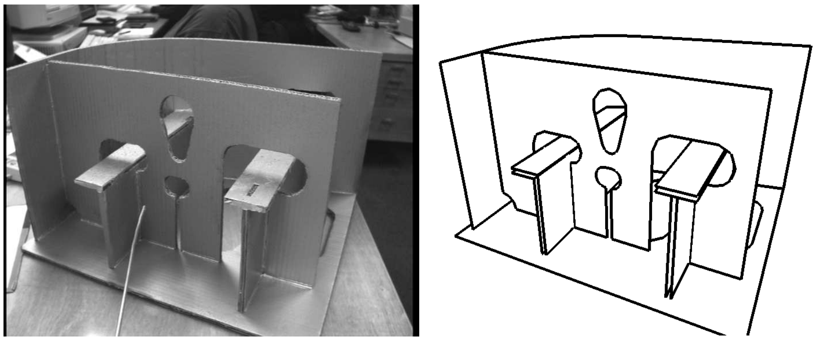


Fig. 2. Image and CAD model of ship part

assigned and measurements made in this way. The system then projects this m-dimensional measurement vector onto the 6-dimensional subspace corresponding to Euclidean transformations (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 (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 (f). The system then loops back to step (a).

2.1 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 ξ^{th} node for the i^{th} group generator. These can be considered as a set of m-dimensional vectors which describe the

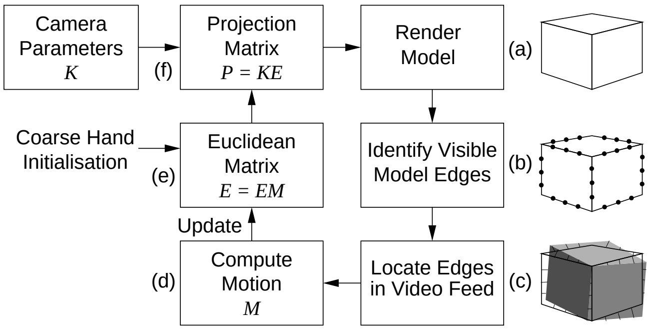


Fig. 3. Tracking system operation

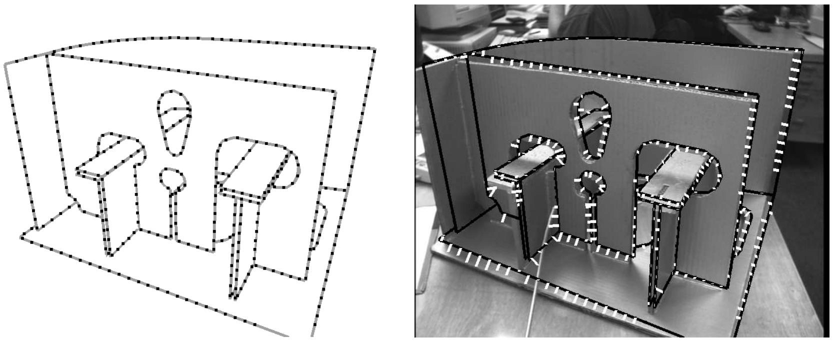


Fig. 4. Tracking nodes are assigned and distances measured

motion in the image for each mode of Euclidean transformation. The system then projects the m-vector corresponding to the measured distances (to the observed edges) onto the subspace spanned by the transformation vectors. The geometric transformation of the part which best fits the observed edge positions can be found by minimising the square error between the transformed edge position and the actual edge position (in pixels). This process is performed as follows:

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

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

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

(with Einstein summation convention over Latin indices used throughout this paper). It can be seen that setting $\beta_i = \alpha_i$ gives the minimum (least-squares) solution to

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

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

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

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

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

$$= 0 \tag{18}$$

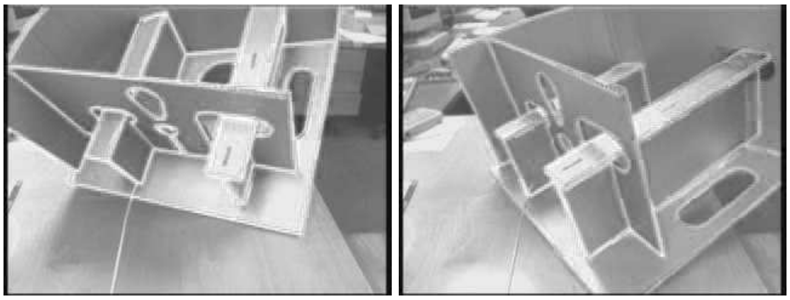


Fig. 5. Frames from video of tracking sequence: The CAD model of the ship part is superimposed on the video image using the estimate of the projection matrix.

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

$$\frac{\partial S}{\partial \beta_i} = 0 + 2\sum_{\xi} f_i^{\xi} f_j^{\xi} \varepsilon_j \tag{19}$$

$$= 2C_{ij}\varepsilon_j \tag{20}$$

and integrating gives

$$S = S_0 + \varepsilon_i C_{ij} \varepsilon_j \qquad \text{where } S_0 = S|_{\varepsilon=0} \tag{21}$$

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.

$$E_{t+1} = E_t \; \exp(\textstyle\sum_i \alpha_i G_i) \tag{22}$$

The system is therefore able to maintain an estimate of E (and hence P) by continually computing the coefficients α_i of inter-frame motions (see Figure 5). This method has also been extended to include the motion of image features due to the change in internal camera parameters and thus provide a method for on-line camera calibration [23]. In practice the simple least squares algorithm is not robust to outliers so the terms in (11) and (12) are reweighted by a decaying function of d^{ξ} to obtain a robust M-estimator. The reweighting causes the algorithm to become iterative (since d^{ξ} varies with each iteration) but convergence in all but extreme conditions is very fast and only one iteration is used per video frame/field.

3 Complex configurations

The rigid body tracking system presented in the previous section is now used as the basis of an approach which is designed to operate in more complex configurations. A novel 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 work is restricted to considering targets which comprise a number of rigid components connected by hinges or slides etc.

The simplest way to handle these configurations is merely to run multiple instances of the rigid body tracker concurrently, one per component per camera. Thus, for example three cameras viewing two structures would require six concurrent trackers. Unfortunately, this naïve approach can introduce many more degrees of freedom into the system than are really present. In this example, 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. In general, this is a bad thing since problems such as ill-conditioning and high search complexity are more prevalent in high dimensional systems and also because the solution thus generated can exhibit inconsistencies. The natural approach to this problem is to impose all of the constraints that are known about the world upon the tracking system.

4 Applying Constraints

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 6 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 twelve 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 co-ordinate frames.
2. How can the constraints be expressed?
3. How can they then be imposed?

4.1 Co-ordinate frames

The first difficulty is that the α_i and the C_{ij} are quantities in the Lie algebra deriving from the co-ordinate 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 co-ordinate frame to another is needed. Specifically, this requires knowing what happens to the Lie algebra of SE(3) under $\mathbb{R}^3$ co-ordinate 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:

$$MG_iM^{-1} = \sum_j \mathrm{ad}(M)_{ij} G_j \qquad (23)$$

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

$$\mathrm{ad}(M) = \begin{bmatrix} R & 0 \\ [t_\wedge]R & R \end{bmatrix} \qquad \text{where } [t_\wedge]_{ij} = \varepsilon_{ijk} t_k \qquad (24)$$

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 co-ordinate 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 equations (11) – (13) can be transformed as follows (see Figure 6(a–b)):

$$\alpha' = \mathrm{ad}(M)\,\alpha \tag{25}$$

$$C' = \mathrm{ad}(M)\,C\,\mathrm{ad}(M)^T \tag{26}$$

$$v' = \mathrm{ad}(M)^{-T} v \tag{27}$$

4.2 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 the motion 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 sub-manifold representing legal configurations embedded within the raw twelve dimensional configuration manifold. The tangent space to this submanifold corresponds to the space of velocities which 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 both linear and 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

$$\beta_1 \cdot c_1^i + \beta_2 \cdot c_2^i = 0 \tag{28}$$

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 co-ordinate 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 co-ordinate frame using the adjoint representation of the co-ordinate transformation, the constraints now take the form

$$\beta_1 \cdot c_1^i + \beta_2' \cdot c_2^i = 0 \tag{29}$$

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:

$$c_1^i = \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix} \qquad (1 \leq i \leq 5) \tag{30}$$

with $c_2^i = -c_1^i$. 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 a Euclidean transformation that takes this situation back to the simple one.

4.3 Imposing the 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)–(13), then (21) 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

$$(\beta_1 - \alpha_1)C_1(\beta_1 - \alpha_1) + (\beta_2' - \alpha_2')C_2'(\beta_2' - \alpha_2') \tag{31}$$

is minimised subject to

$$\beta_1 \cdot c_1^i + \beta_2' \cdot c_2^i = 0 \tag{32}$$

This is a constrained optimisation problem and ideal for solving by means of Lagrange multipliers. Thus the solution is given by the constraints in (32) and

$$\nabla\big((\beta_1 - \alpha_1)^T C_1(\beta_1 - \alpha_1) + (\beta_2' - \alpha_2')^T C_2'(\beta_2' - \alpha_2')\big) + \lambda_i \nabla\big(\beta_1^T c_1^i + \beta_2'^T c_2^i\big) = 0 \tag{33}$$

with ∇ running over the twelve dimensions of $\binom{\beta_1}{\beta_2'}$. This evaluates to

$$\begin{pmatrix}2C_1(\beta_1 - \alpha_1)\\ 2C_2(\beta_2' - \alpha_2')\end{pmatrix} + \lambda_i \begin{pmatrix}c_1^i\\ c_2^i\end{pmatrix} = 0 \tag{34}$$

$$\begin{aligned} \text{Thus} \qquad & \beta_1 = \alpha_1 - \tfrac{1}{2}C_1^{-1}\lambda_i c_1^i \\ \text{and} \qquad & \beta_2' = \alpha_2' - \tfrac{1}{2}C_2'^{-1}\lambda_i c_2^i \end{aligned} \tag{35}$$

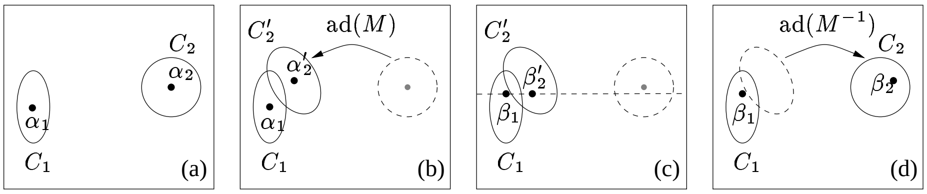


Fig. 6. Applying the constraints: Estimates and errors are computed for motions 1 and 2 (a), the estimate and error of motion 2 are mapped into 1's co-ordinate frame (b), the constraint is applied there (c) and then the new estimate of motion 2 is mapped back into its own frame (d).

Substituting (32) back into (35) gives

$$c_1^i \cdot \alpha_1 + c_2^i \cdot \alpha_2' - \tfrac{1}{2}\lambda_j \left(c_1^i \cdot C_1^{-1} c_1^j + c_2^i \cdot C_2'^{-1} c_2^j) \right) = 0 \tag{36}$$

So the λ_i are given by

$$A_{ij} = c_1^i \cdot C_1^{-1} c_1^j + c_2^i \cdot C_2'^{-1} c_2^j \tag{37}$$

$$l_i = 2 \left(c_1^i \cdot \alpha_1 + c_2^i \cdot \alpha_2' \right) \tag{38}$$

$$\lambda_i = A_{ij}^{-1} l_j \tag{39}$$

The λ_i can then be substituted back into (35) to obtain β_1 and β_2' (see Figure 6(c)), from which β_2 can also be obtained by $\beta_2 = \mathrm{ad}(E_2^{-1}E_1)\beta_2'$ (see Figure 6(d)). 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.

5 Results

A system was developed to test the tracking of a simple articulated structure (shown in Figure 7(a)). This system operates in real-time at PAL frame-rate (25Hz) on an SGI O2 (225 MHz R10K). The structure consists of two components, each 15cm 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. A series of experiments were 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 oriented at a series of known angles and for each angle a set of measurements were taken with and without the constraints imposed. The amount by which the rotational and translational constraints (measured at the hinge) are violated by the unconstrained tracker was also measured.

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

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. Figure 7(b) shows the behaviour 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.

This system was then extended to track the structure with an additional square component and hinge (see Figure 8(a)). The system is able to track the full configuration of the structure, even when the central component is fully hidden from view (see Figure 8(b)). In this case, the observed positions of the two visible components are sufficient to determine the location of the hidden part. Further, the indirect constraints between the two end parts of the structure serve to improve the conditioning of the estimation of their positions.

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. The objects to be tracked are shown in Figure 9(a) and Figure 9(b) shows how the tracker behaves when the constraint is deliberately violated; the output of the system still respects the constraint and is forced to find a compromise between the two components.

(a) (b)

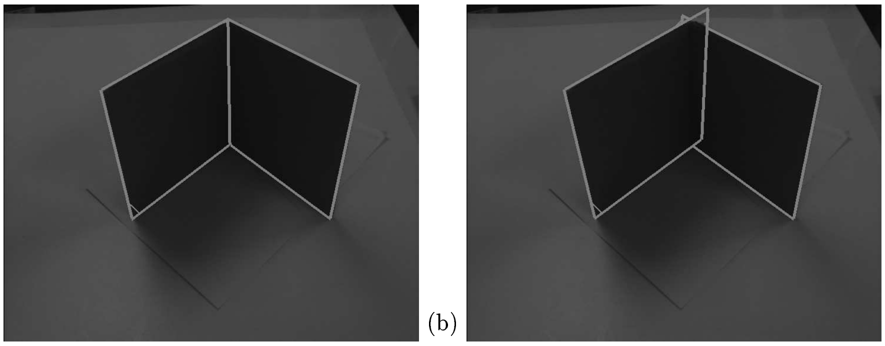


Fig. 7. Hinge tracking with and without constraints

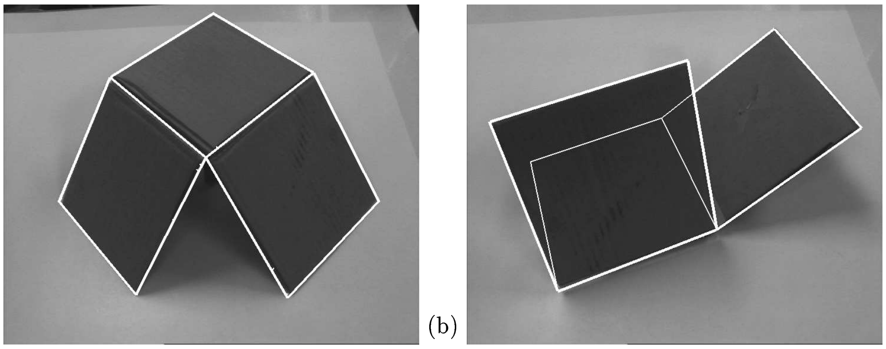
(a) (b)


Fig. 8. Double hinge structure: The tracker can infer the position of a hidden component from the constraints

Finally, a multi-camera system was developed using 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 (the ship part). With 3 cameras operating simultaneously (on a complex structure) the achieved frame rate dropped to 20Hz (this is believed to be due to speed limitations of the GL rendering hardware used in the tracking cycle. This 3 camera configuration is found to be much more stable and robust, maintaining a track over sequences that have been found to cause the single camera tracker to fall into a local minimum. These instabilities occur in a sparse set of configurations (e.g. when a feature rich plane passes through the camera and also in near-affine conditions when such a plane is fronto-parallel to the camera). By employing multiple cameras it becomes extremely difficult to contrive a situation that is critical in all camera views simultaneously.

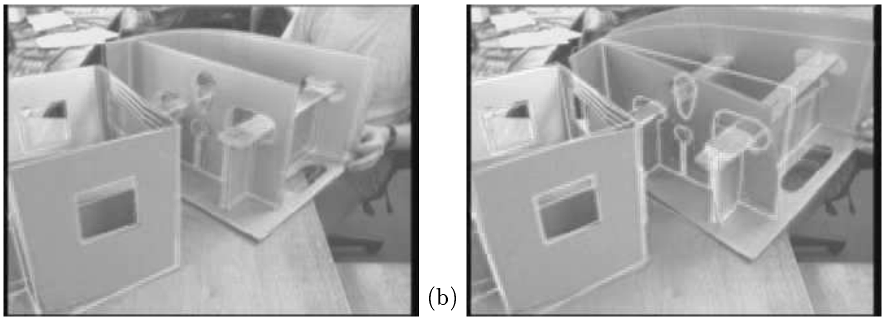
(a) (b)


Fig. 9. Two structures with common ground plane constraint: When the world violates the constraint, the tracker attempts to fit the constrained model. In this example, the tracker has fitted some parts of both models

6 Conclusion

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 co-ordinate 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.

References

[1] T. Drummond and R.Cipolla. Real-time tracking of complex structures for visual servoing. In *PreProceedings of Vision Algorithms: Theory and Practice*, pages 91–98, Corfu, Greece, 21–22 September 1999. Also to appear in Springer Lecture Notes in Computer Science.

[2] C. Harris. Geometry from visual motion. In A. Blake, editor, *Active Vision*, chapter 16, pages 263–284. MIT Press, 1992.

[3] G. Hager, G. Grunwald, and K. Toyama. Feature-based visual servoing and its application to telerobotics. In V. Graefe, editor, *Intelligent Robotic Systems*. Elsevier, 1995.

[4] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In A. Blake, editor, *Active Vision*, chapter 1, pages 3–20. MIT Press, 1992.

[5] R. Cipolla and A. Blake. *Active Vision*, chapter Geometry from Visual Motion, pages 189–202. 1992.

[6] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.

[7] P. Wunsch and G. Hirzinger. Real-time visual tracking of 3-Dobjects with dynamic handling of occlusion. In *Proceedings of the 1997 International Conference on Robotics and Automation*, pages 2868–2873, 1997.

[8] C. Harris. Tracking with rigid models. In A. Blake, editor, *Active Vision*, chapter 4, pages 59–73. MIT Press, 1992.

[9] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[10] N. Daucher, M. Dhome, J. T. Lapresté, and G. Rives. Modelled object pose estimation and tracking by monocular vision. In *Proceedings of the British Machine Vision Conference*, pages 249–258, 1993.

[11] A. D. Worrall, G. D. Sullivan, and K. D. Baker. Pose refinement of active models using forces in 3D. In J. Eklundh, editor, *Proceedings of the Third European Conference on Computer vision (ECCV'94)*, volume 2, pages 341–352, May 1994.

[12] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *Proceedings of ICCV'99*, volume 1, pages 262–268, Kerkyra, Greece, 20–27 September 1999.

[13] V.S. Varadarajan. *Lie Groups, Lie Algebras and Their Representations*. Number 102 in Graduate Texts in Mathematics. Springer-Verlag, 1974.

[14] D.H. Sattinger and O.L. Weaver. *Lie groups and algebras with applications to physics, geometry, and mechanics.* Number 61 in Applied Mathematical Sciences. Springer-Verlag, 1986.

[15] T. Drummond and R. Cipolla. Visual tracking and control using Lie algebras. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 652–657, Fort Collins, Colorado, 23–25 June 1999. IEEE.

[16] M. Armstrong and A. Zisserman. Robust object tracking. In *Proceedings of Second Asian Conference on Computer Vision*, pages 58–62, 1995.

[17] M. Paterson and F. Yao. Efficient binary space partitions for hidden surface removal and solid modeling. *Discrete and Computational Geometry*, 5(5):485–503, 1990.

[18] J. MacCormick and A. Blake. Spatial dependence in the observation of visual contours. In *Proceedings of the Fifth European Conference on Computer vision (ECCV'98)*, pages 765–781, 1998.

[19] M. Haag and H-H. Nagel. Tracking of complex driving manoeuvres in traffic image sequences. *Image and Vision Computing*, 16:517–527, 1998.

[20] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, 1998.

[21] D. G. Lowe. Fitting parameterised 3-D models to images. *IEEE T-PAMI*, 13(5):441–450, 1991.

[22] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouttes. In *Proceedings of ICCV'99*, volume 2, pages 716–721, Kerkyra, Greece, 20–27 September 1999.

[23] T. Drummond and R. Cipolla. Real-time tracking of complex structures with on-line camera calibration. In *Proceedings of British Machine Vision Conference 1999*, volume 2, pages 574–583, Nottingham, 13–16 September 1999. BMVA.