Complex articulated object tracking.

Andrew Comport, E. Marchand, François Chaumette

Electronic Letters on Computer Vision and Image Analysis 5(3):20-30, 2005

Complex articulated object tracking

Andrew I. Comport, Éric Marchand and François Chaumette

IRISA - INRIA Rennes, Campus de Beaulieu, 35042 Rennes, France

Received 20 December 2004; accepted 9 March 2005

Abstract

In this paper new results are presented for tracking complex multi-body objects. The theoretical framework is based on robotics techniques and uses an a-priori model of the object including a general mechanical link description. A new kinematic-set formulation takes into account that articulated degrees of freedom are directly observable from the camera and therefore their estimation does not need to pass via a kinematic-chain back to the root. By doing this the tracking techniques are efficient and precise leading to real-time performance and accurate measurements. The system is locally based upon an accurate modeling of a distance criteria. A general method is given for defining any type of mechanical link and experimental results show prismatic, rotational and helical type links. A statistical M-estimation technique is applied to improve robustness. A monocular camera system was used as a real-time sensor to verify the theory.

Key Words: Computer Vision, Image Registration, Non-rigid Motion, Multi-body Systems, Computer Vision, 3D Tracking, Articulated Objects, Kinematic Sets, Visual Servoing, Model-Based, Real-time.

1 Introduction

Previously, non-rigid motion has been classed into three categories describing different levels of constraints on the movement of a body: articulated, elastic and fluid [1]. In this paper the first class of non-rigid motion is considered and a link is made with the remaining classes. An ”articulated” object is defined as a multi-body system composed of at least two rigid **components** and at most six independent degrees of freedom between any two components. With articulated motion, a non-rigid but constrained dependence exists between the components of an **object**. Previous methods have attempted to describe articulated motion either with or without an a-priori model of the object. In this study a 3D model is used due to greater robustness and efficient computation. Knowing the object in advance helps to predict hidden movement, which is particularly interesting in the case of non-rigid motion because there is an increased amount of self-occlusion. Knowing the model also allows an analytic relation for the system dynamics to be more precisely derived.

1.0.1 State of the Art

In general, the methods which have been proposed in the past for articulated object tracking rely on a good rigid tracking method. In computer vision the geometric primitives considered for tracking have been numerous, however, amongst them distance based features have shown to be efficient and robust [11, 6, 14, 2]. Another important issue is the 2D-3D registration problem. *Purely geometric* (eg, [5]), or *numerical and iterative* [4]

Correspondence to: <Firstname.Lastname@irisa.fr>

Recommended for acceptance by <Perales F., Draper B.>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

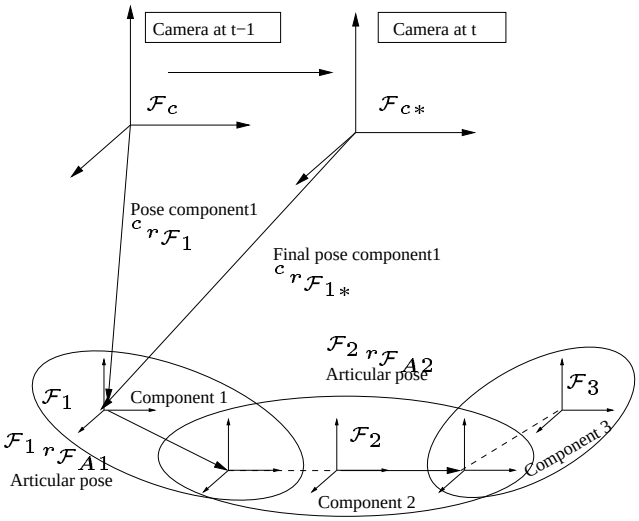

Figure 1: Kinematic chain method: The pose of an articulated object is determined via a kinematic chain of rigid bodies extending to sub components. The poses **r** describing the state of the system are intuitive.

approaches may be considered. *Linear approaches* use a least-squares method to estimate the pose and are considered to be more suitable for initialization procedures. *Full-scale non-linear optimization techniques* (e.g., [11, 13, 6, 2]) consists of minimizing the error between the observation and the forward-projection of the model. In this case, minimization is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquardt. The main advantage of these approaches are their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence. This approach is better suited to maintaining an already initialized estimation.

Within this context it is possible to envisage different ways to model the pose of an articulated object. The first method for tracking articulated objects using kinematic chains (see Figure 1.0.1) appears in well known work by Lowe [12]. He demonstrates a classical method using partial derivatives. In his paper the kinematic chain of articulations is represented as tree structure of internal rotation and translation parameters and the model points are stored in the leaves of this tree.The position and partial derivatives of each point in camera-centered coordinates is determined by the transformations along the path back to the root.

Recently, more complex features have been used for non-rigid object tracking in [16]. They make use of deformable super-quadric models combined with a kinematic chain approach. However, real-time performance is traded-off for more complex models. Furthermore, this method requires multiple viewpoints in order to minimize the system of equations. As Lowe points out, the tendencies in computer graphics have been toward local approximations via polyhedral models. Ruff and Horaud [17] give another kinematic-chain style method for the estimation of articulated motion with an un-calibrated stereo rig. They introduce the notion of projective kinematics which allows rigid and articulated motions to be represented within the transformation group of projective space. The authors link the inherent projective motions to the Lie-group structure of the displacement group. Minimization is performed in projective space making the parameters invariant to camera calibration.

A second approach has been proposed by Drummond and Cippola [6] which treats articulated objects as groups of rigid components with constraints between them directly in camera coordinates (see Figure 1.0.1). It appears that the full pose of each rigid component is initially computed independently requiring the estimation of a redundant number of parameters. Lagrange multipliers are then used to constrain these parameters according to simple link definitions. This method uses Lie Algebra to project the measurement vector (distances) onto the subspace defined by the Euclidean transformation group (kinematic screw). They also implement M-estimation to improve robustness.

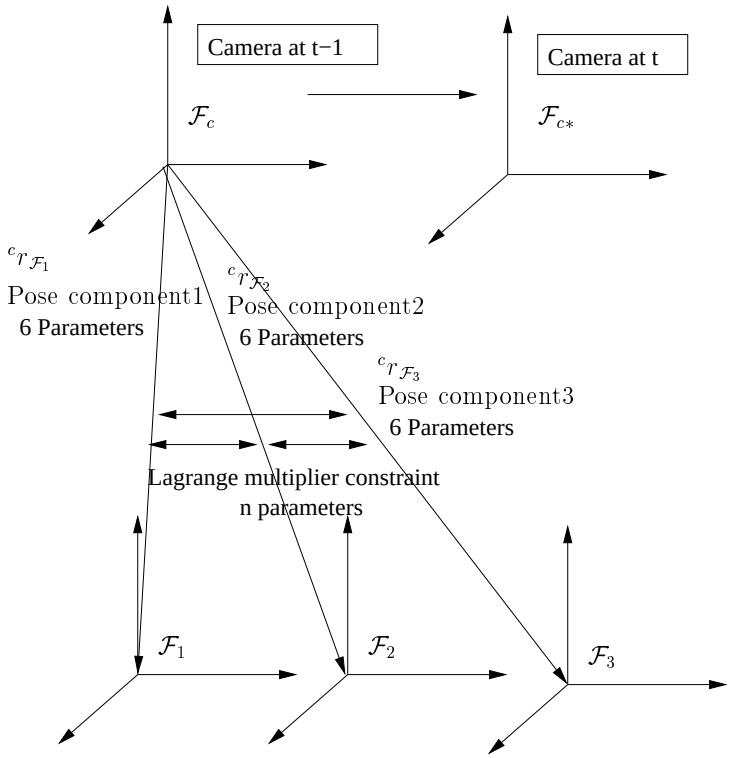

Figure 2: Lagrange Multiplier method: The pose between the camera and each component of the object is calculated individually in a first step. Constraints are then applied between the full pose of each component via Lagrange multipliers in a second step.

1.0.2 Contribution

A new model is proposed in this paper which is based on the observation that within a vision system one has *direct access* with a camera to the parameters of an articulated object. Thus, unlike traditional techniques using robotics based approaches, there is no need to sum partial derivatives along a kinematic chain back to the root. As will be shown, the joint reference frame plays an important role in modeling articulated objects. The method presented in this paper also integrates a mechanical link formulation for simple definition of articulations.

It is important to correctly model the behavior of the system to obtain maximum decoupling of joint parameters and therefore interesting minimization properties. In this paper a kinematic set approach is proposed. With articulated motion, unlike the case of rigid motion, the subsets of movement which may be attributed to either the object or camera are not unique. A novel subset approach is used whereby the minimization is carried out on decoupled subsets of parameters by defining subspace projectors from the joint definitions. This allows the error seen in the image to be partially decoupled from the velocities of the object by determining the independent sets of velocities present in object space. The principal advantages of this approach are that it:

- is more efficient in terms of computation than previous methods.
- eliminates the propagation of errors between free parameters.
- models more closely the real behavior of the system than a camera frame based approach.

In the remainder of this paper, Section 2 presents the principle of the approach. In Section 3 articulated object motion and velocity are defined. In Section 4 a non-linear control law is derived for tracking articulated objects. In Section 5, several experimental results are given for different virtual links.

2 Overview and Motivations

The objective of the proposed approach is to maintain an estimate of a set of minimal parameters describing the configuration of an articulated object in $SE(n)$. This set of parameters are defined by a vector of n parameters $\mathbf{q} \in \mathbb{R}^n$. This vector is composed of subsets which fully describe the velocity of each component.

In order to maintain an estimate of $\mathbf{q}$, the underlying idea is to minimize a non-linear system of equations so that the projected contour of the object model in the image is aligned with the actual position of the contours in the image. This can be seen as the dual problem of visual servoing whereby minimizing the parameters corresponds to moving an arm-to-eye robot so as to observe the arm at a given position in the image (note that an object is not necessarily fixed to the ground). This duality, known as Virtual Visual Servoing has been explained in depth in previous papers [2, 15].

To perform the alignment, an error Δ is defined in the image between the projected features $\mathbf{s}(\mathbf{q})$ of the model and their corresponding features in the image $\mathbf{s_d}$ (desired features). The features of each component are projected using their associated camera poses ${}^c\mathbf{r}_{\mathcal{F}_1}(\mathbf{q})$ and ${}^c\mathbf{r}_{\mathcal{F}_2}(\mathbf{q})$ where each component's camera pose is composed of a subset of object parameters $\mathbf{q}$. In this paper distance features are used. This error is therefore defined as:

$$\Delta = \Big(\mathbf{s}(\mathbf{q}) - \mathbf{s_d}\Big) = \Big[pr(\mathbf{q}, {}^o\mathbf{S}) - \mathbf{s_d}\Big], \tag{1}$$

where ${}^o\mathbf{S}$ are the 3D coordinates of the *sensor* features in the object frame of reference. $pr(\mathbf{q}, {}^o\mathbf{S})$ is the camera projection model according to the object parameters $\mathbf{q}$.

The parameters of the object are initially needed and they are computed using the algorithm of Dementhon and Davis [4]. This algorithm is used to calculate the component's poses in the camera frame and they are calculated separately. The parameters are projected into object space and variables in common between the components are averaged so that initialization errors are minimal.

In order to render the minimization of these errors more robust they are minimized using a robust approach based on M-estimation techniques.

$$\Delta_{\mathcal{R}} = \rho\Big(\mathbf{s}(\mathbf{q}) - \mathbf{s_d}\Big), \tag{2}$$

where $\rho(u)$ is a robust function [9] that grows sub-quadratically and is monotonically nondecreasing with increasing $|u|$. In this article Tukey's function is used because it allows complete rejection of outliers.

This is integrated into an iteratively re-weighted least squares(IRLS) minimization procedure so as to render those errors at the extremities of the distribution less likely.

3 Modeling

Articulated motion is defined as Euclidean transformations which preserve *subsets* of distances and orientation of object features.

The modeling of object motion is based on rigid body differential geometry. The set of rigid-body positions and orientations belongs to a Lie group, $SE(3)$ (Special Euclidean group). These vectors are known as screws. The tangent space is the vector space of all velocities and belongs to the Lie algebra, $se(3)$. This is the algebra of twists which is also inherent in the study of non-rigid motion. An articulated object, for example, must have a velocity contained in $se(3)$, however, joint movement can be considered by sub-algebras of $se(3)$.

The basic relation can be written which relates the movement of a sensor feature $\dot{\mathbf{s}}$ to the movement of the object parameters:

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{A}\dot{\mathbf{q}} \tag{3}$$

where

- $\mathbf{L_s}$ is called the feature Jacobian [10] or interaction matrix [7] between the camera and the sensor features $\mathbf{s}$.
- $\mathbf{A}$ is an Articulation matrix describing the differential relation between components.

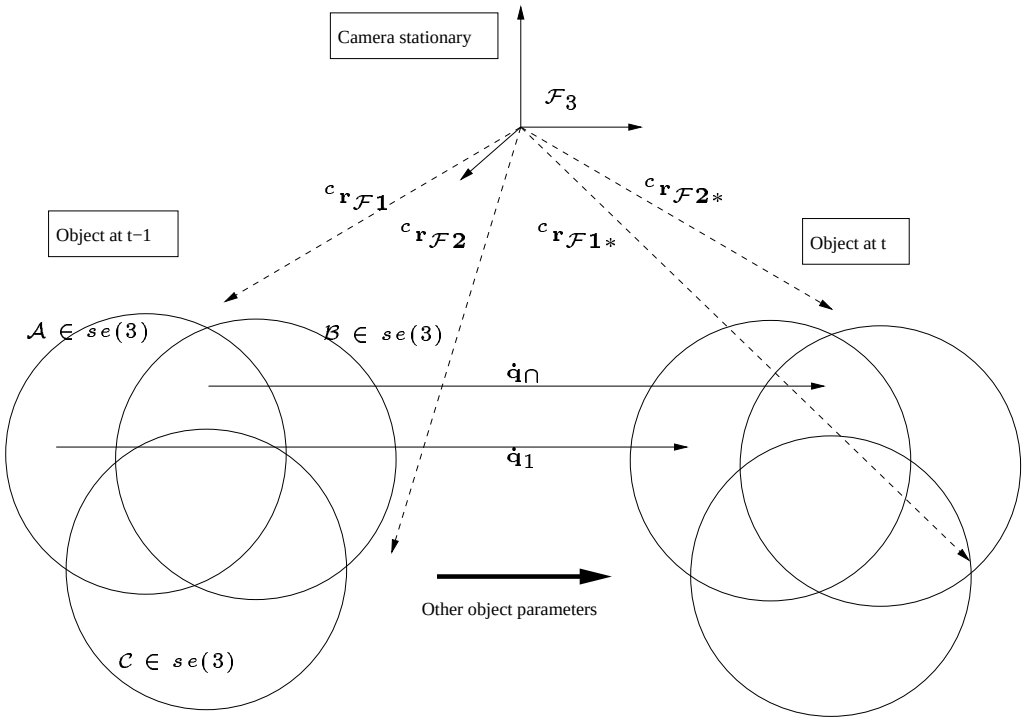

Figure 3: Kinematic set method: The joint parameters are minimized in object space and kinematic set are used to decouple the system. Decoupling occurs at the intersection of parameter sets. $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are the sets of velocity parameters corresponding to each rigid component. $\dot{\mathbf{q}}$ is a vector of generalized velocity parameters. The poses $\mathbf{r}$ correspond to the joint poses in 3D.

• $\mathbf{L_s A}$ being the Jacobian between the sensor and the entire object.

The Articulation matrix is the central issue in this paper. It corresponds to the mapping:

$$\mathbf{v} = \mathbf{A}\dot{\mathbf{q}} \tag{4}$$

where $\mathbf{v}$ is a vector of 'stacked' 6 dimensional twists each corresponding to the full motion in $se(3)$ of each component.

The subsets of parameters which make up the object parameters are illustrated by a Venn diagram in Figure 3. In order that these sets can be obtained independently it is necessary to decouple their interaction. The only case where this occurs is in the joint frame of reference. Thus the following section considers the definition of a joint.

3.1 Mechanical Joint Concept

A mechanical joint is fully defined by a matrix, vector pair which links two components. It is composed of a constraint matrix $\mathbf{S}^\perp$ which defines the type of the link and a pose vector $\mathbf{r}$ defining the position of the articulation. Thus the articulation matrix is:

$$\mathbf{A}_l(\mathbf{S}_l^\perp, \mathbf{r}_l), \tag{5}$$

where $\mathbf{S}_l^\perp$ corresponds to the configuration of joint l and $\mathbf{r}$ is a parameter vector describing the location of joint l.

The following two subsections explain the definition of these parameters.

3.2 Joint Configuration - $\mathbf{S}_l$

A joint configuration is fully defined by:

$$\mathbf{S}^{\perp}{}_l = \begin{pmatrix} s^{\perp}_{1,1} & \dots & s^{\perp}_{1,c} \\ \vdots & \ddots & \\ s^{\perp}_{6,1} & & s^{\perp}_{6,c} \end{pmatrix}, \tag{6}$$

The holonomic constraint matrix, $\mathbf{S}^{\perp}$, is defined such that each column vector defines one free degree of freedom at the corresponding link. The number of non-zero columns of $\mathbf{S}^{\perp}$ is referred to as the *class* c of the link. The rows of a column define the type of the link by defining which combination of translations and rotations are permitted as well as their proportions. In the experiments considered in Section 5 two different types of class 1 links are considered:
A rotational link around the x axis:

$$\mathbf{S}^{\perp} = (0, 0, 0, 1, 0, 0), \tag{7}$$

A helical link around and along the z axis:

$$\mathbf{S}^{\perp} = (0, 0, a, 0, 0, 1), \tag{8}$$

where the value of 'a' relates the translation along the z axis to a one rotation around the z axis.

The set of velocities that a first component can undertake which leaves a second component invariant is defined by $S^{\perp} \subset se(3)$. This is the orthogonal compliment of the sub-space $S \subset se(3)$ which constitutes the velocities which are in common between two components. Since a component, that is linked to another, is composed of these two subspaces it is possible to extract these subspaces by defining standard bases for the kernel and the image. The kernel is chosen to be $\mathbf{S}^{\perp}$ so that the image is given by (with abuse of notation):

$$\mathbf{S}_l = Ker((\mathbf{S}^{\perp}_l)^T), \tag{9}$$

The matrix $\mathbf{S}_l$ and its orthogonal compliment $\mathbf{S}^{\perp}_l$ can be used to project the kinematic twist (velocities) onto two orthogonal subspaces (For more than 1 joint it is necessary to project onto a common vector basis).

Thus a subspace projection matrix is given as:

$$\begin{aligned} \mathbf{P}_l &= \mathbf{S}_l \mathbf{S}^{+}_l, \\ \mathbf{P}^{\perp}_l &= \mathbf{S}^{\perp}_l \mathbf{S}^{\perp +}_l = \mathbb{I}_6 - \mathbf{P}_l, \end{aligned} \tag{10}$$

where $\mathbf{S}^{+} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T$ is the pseudo-inverse of $\mathbf{S}$.

This ensures that the resulting projected velocities are defined according to a common basis defined by the parameters of the pose vector in equation (6). This then allows the twist transformations, given in the following section, to be applied to these quantities.

3.3 Joint Location - $\mathbf{r}_l$

A joint location is fully defined by a pose vector:

$$\mathbf{r}_l = {}^{\mathcal{F}_c}\mathbf{r}_l = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z), \tag{11}$$

where $\mathcal{F}_c$ indicates the camera frame and l represents the joint frame.

A joint configuration is only valid in the joint reference frame. A kinematic twist transformation matrix is used to obtain the velocity of the joint frame w.r.t its previous position. The Lie algebra provides the transformation of vector quantities as $\mathbf{V}(\mathbf{r})$. This is a kinematic twist transformation from frame a to frame b given as:

$${}^{a}\mathbf{V}_b = \begin{bmatrix} {}^{a}\mathbf{R}_b & [{}^{a}\mathbf{t}_b]^{a}_{\times}\mathbf{R}_b \\ 0_3 & {}^{a}\mathbf{R}_b \end{bmatrix}, \tag{12}$$

where ${}^a\mathbf{R}_b$ is a rotation matrix between frames and ${}^a\mathbf{t}_b$ a translation vector between frames which are obtained from ${}^a\mathbf{r}_b$. $[\mathbf{t}]_x$ is the skew symmetric matrix related to $\mathbf{t}$.

The projector defined in (10) is applied in the joint reference frame. It is possible to choose the object frame as a common reference frame as in [3]. In this paper the camera frame is chosen as the common reference frame so that a generic subspace projection operators $\mathbf{J}_l$ and $\mathbf{J}_l^\perp$ can be defined as:

$$\begin{array}{l} \mathbf{J}_l = Im({}^c\mathbf{V}_l \, \mathbf{P}_l \, {}^l\mathbf{V}_c), \\ \mathbf{J}_l^\perp = Ker(\mathbf{J}) = Im({}^c\mathbf{V}_l \, \mathbf{P}_l^\perp \, {}^l\mathbf{V}_c), \end{array} \tag{13}$$

where Im represents the Image operator which reduces the column space to its mutually independent basis form. The first transformation $\mathbf{V}$ maps the velocities to the joint frame l and the second re-maps back to the camera reference frame.

3.4 Articulation Matrix

Using the previous joint definition is is possible to define the Articulation matrix according to equation (4) and taking into account the joint subspaces given by equation (13).

The derivation of the Articulation matrix corresponds to:

$$\mathbf{A} \;=\; \begin{pmatrix} \frac{\partial \mathbf{r_1}}{\partial \mathbf{q}} \\ \vdots \\ \frac{\partial \mathbf{r}_m}{\partial \mathbf{q}} \end{pmatrix}, \tag{14}$$

where m is the number of components.

For an object with two components and one jointand using the orthogonal subspace projectors given in equation (13), $\mathbf{A}$ is given by:

$$\mathbf{A} \;=\; \begin{pmatrix} \frac{\partial \mathbf{r_1}}{\partial \mathbf{q}_\cap} & \frac{\partial \mathbf{r_1}}{\partial \mathbf{q_1}} & \mathbf{0} \\ \frac{\partial \mathbf{r_2}}{\partial \mathbf{q}_\cap} & \mathbf{0} & \frac{\partial \mathbf{r_2}}{\partial \mathbf{q_2}} \end{pmatrix} \;=\; \begin{pmatrix} \mathbf{J}_1 & \mathbf{J}_1^\perp & \mathbf{0} \\ \mathbf{J}_1 & \mathbf{0} & \mathbf{J}_1^\perp \end{pmatrix}, \tag{15}$$

where $\mathbf{q}_\cap$, $\mathbf{q_1}$, $\mathbf{q_2}$ are vectors representing the sets of intersecting velocities and each components free parameters respectively. These sets are easily identified when referring to Figure 3. Given $dim(\mathbf{J}_1) = 6 - c$ and $dim(\mathbf{J}_1^\perp) = c$, the mapping $\mathbf{A}$ is indeed dimension $12 \times (6 + c)$, remembering that c is the class of the link. The derivation of objects with more than one joint follows in a similar manner and is left to the reader.

It is important to note that this method introduces decoupling of the minimization problem. This is apparent in equation (15) where extra zeros appear in the Jacobian compared to the traditional case of a kinematic chain. Indeed, in the particular case of two components and one articulation a kinematic chain has only one zero.

4 Registration

In this section a new tracking control law is derived. The aim of the control scheme is to minimize the objective function given in equation (2). Thus, the error function is given as:

$$\begin{pmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_m \end{pmatrix} = \mathbf{D} \begin{pmatrix} \mathbf{s}_1(\mathbf{q}) - \mathbf{s_{d}}_1 \\ \vdots \\ \mathbf{s}_m(\mathbf{q}) - \mathbf{s_{d}}_m \end{pmatrix}, \tag{16}$$

where $\mathbf{q}$ is a vector composed of the minimal set of velocities corresponding to the object's motion and each $\mathbf{e}_i$ corresponds to an error vector for component i. $\mathbf{D}$ is a diagonal weighting matrix corresponding to the

likelihood of a particular error within the robust distribution:

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{D}_m \end{pmatrix},$$

where each matrix $\mathbf{D}_i$ is a diagonal matrix corresponding to a component which has weights w_j along the diagonal. These weights correspond to the uncertainty of the measured visual feature j. The computation of the weights are described in [2].

If $\mathbf{D}$ were constant, the derivative of equation (16) would be given by:

$$\begin{pmatrix} \dot{\mathbf{e}}_1 \\ \vdots \\ \dot{\mathbf{e}}_i \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{e}_1}{\partial \mathbf{s}_1}\frac{\partial \mathbf{s}_1}{\partial \mathbf{r}_1}\frac{\partial \mathbf{r}_1}{\partial \mathbf{q}} \\ \vdots \\ \frac{\partial \mathbf{e}_m}{\partial \mathbf{s}_m}\frac{\partial \mathbf{s}_m}{\partial \mathbf{r}_m}\frac{\partial \mathbf{r}_m}{\partial \mathbf{q}} \end{pmatrix} \dot{\mathbf{q}} = \mathbf{D}\mathbf{L}_\mathbf{s}\mathbf{A}\dot{\mathbf{q}} \tag{17}$$

where $\dot{\mathbf{q}}$ is the minimal velocity vector, $\mathbf{A}$ is the articulation matrix describing the mapping in equation (4) and $\mathbf{L}_\mathbf{s}$ is the 'stacked' interaction matrix as in equation (3) given as:

$$\mathbf{L}_\mathbf{s} = \begin{pmatrix} \frac{\partial \mathbf{s}_1}{\partial \mathbf{r}_1} \\ \vdots \\ \frac{\partial \mathbf{s}_m}{\partial \mathbf{r}_m} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{\mathbf{s}1} & & \mathbf{0}_6 \\ & \ddots & \\ \mathbf{0}_6 & & \mathbf{L}_{\mathbf{s}m} \end{pmatrix}, \tag{18}$$

If an exponential decrease of the error $\mathbf{e}$ is specified:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}, \tag{19}$$

where λ is a positive scalar, the following control law is obtained by combining equation (19) and equation (17):

$$\dot{\mathbf{q}} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L}_s}\widehat{\mathbf{A}})^{+}\widehat{\mathbf{D}}(\mathbf{s}(\mathbf{q}) - \mathbf{s}_\mathbf{d}), \tag{20}$$

where $\widehat{\mathbf{L}_s}$ is a model or an approximation of the real matrix $\mathbf{L}_\mathbf{s}$. $\widehat{\mathbf{D}}$ a chosen model for $\mathbf{D}$ and $\widehat{\mathbf{A}}$ depends on the previous pose estimation.

For the example of one joint given in equation (15), the sets of velocities to be estimated are:

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{\mathbf{q}}_\cap, & \dot{\mathbf{q}}_1, & \dot{\mathbf{q}}_2 \end{pmatrix}, \tag{21}$$

Once these velocities are obtained they can be related back to the camera frame as in equation (4):

$$\begin{pmatrix} {}^c\mathbf{v}_1 \\ {}^c\mathbf{v}_2 \end{pmatrix} = \mathbf{A} \begin{pmatrix} \dot{\mathbf{q}}_\cap \\ \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{pmatrix} \tag{22}$$

5 Results

In this section three experiments are presented for tracking of articulated objects in *real* sequences. Both camera and object motion as well as articulated motion have been introduced into each experiment. The complex task of implementing this algorithm was a major part of the work. Indeed this required correct modeling of features of type distance to lines, correct modeling of feature sets and correct implementation of the interaction between these feature sets represented as a graph of feature sets.

The rigid tracking method used here is based on a monocular vision system. Local tracking is performed via a 1D oriented gradient search to the normal of parametric contours at a specified sampling distance. This 1D search provides real-time performance. Local tracking provides a redundant group of distance to contour based features which are used together in order to calculate the global pose of the object. The use of redundant measures allows the elimination of noise and leads to high estimation precision. These local measures form an objective function which is minimized via a non-linear minimization procedure using virtual visual servoing(VVS) [2]. These previous results demonstrate a general method for deriving interaction matrices for any type of distance to contour and also show the robustness of this approach with respect to occlusion and background clutter.

The basic implementation of the algorithm gives the following pseudo-code:

1. Obtain initial pose.
2. Acquire new image and project the model onto the image.
3. Search for corresponding points normal to the projected contours.
4. Determine the error $\mathbf{e}$ in the image.
5. Calculate $(\widehat{\mathbf{D}}\widehat{\mathbf{H}}\widehat{\mathbf{A}})$.
6. Determine set velocities as in equation (20) and then component positions.
7. Repeat to 4 until the error converges.
8. Update the pose parameters and repeat to 3.

5.1 Helical Link

This first experiment, reported in Figure 4 was carried out for class one link with helical movement simultaneously along and around the z axis. The constraint vector was defined as in equation (8) and the object frame was chosen to coincide with the joint frame. Note that the constraint vector was defined by taking into consideration that for $10 \times 2\pi$ rotations of the screw it translated 4.5cm along the z axis.

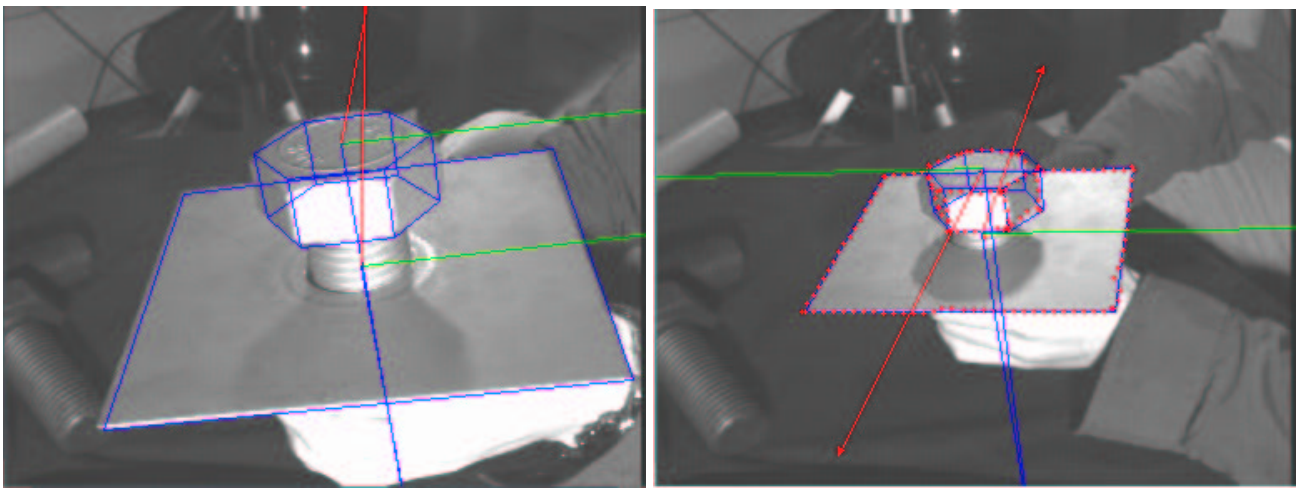

Figure 4: Helical movement of a screw whilst the screw and the platform are simultaneously in movement.In this and all the following figures the reference frames for each component are shown as well as a projection of the CAD model onto the image. The axes of the frames are drawn in yellow, blue and red. The contour of the object is shown in blue. The points found to the normal of the contour are in red and the points rejected by the M-estimator are shown in green.

Tracking of this object displayed real time efficiency with the main loop computation taking on average 25ms per image. It should be noted that tracking of the screw alone as a rigid object fails completely due to the limited contour information and difficult self occulsions. When tracked simultaneously with the plate as an articulated object the tracking of the screw is also based on the measurements of the plate making the tracking possible. M-estimation was carried out separately for each component.

5.2 Robotic Arm

A recent experiment was carried out for two class one links on a robotic arm. The articulations tracked were rotational links around the z and x axes. The constraint vectors were each defined by a pose and a constraint matrix as given in equation (7). Note that the constraints are no longer defined in the same coordinate system as in the previous case. This sequence also displays real time efficiency with the tracking computation taking

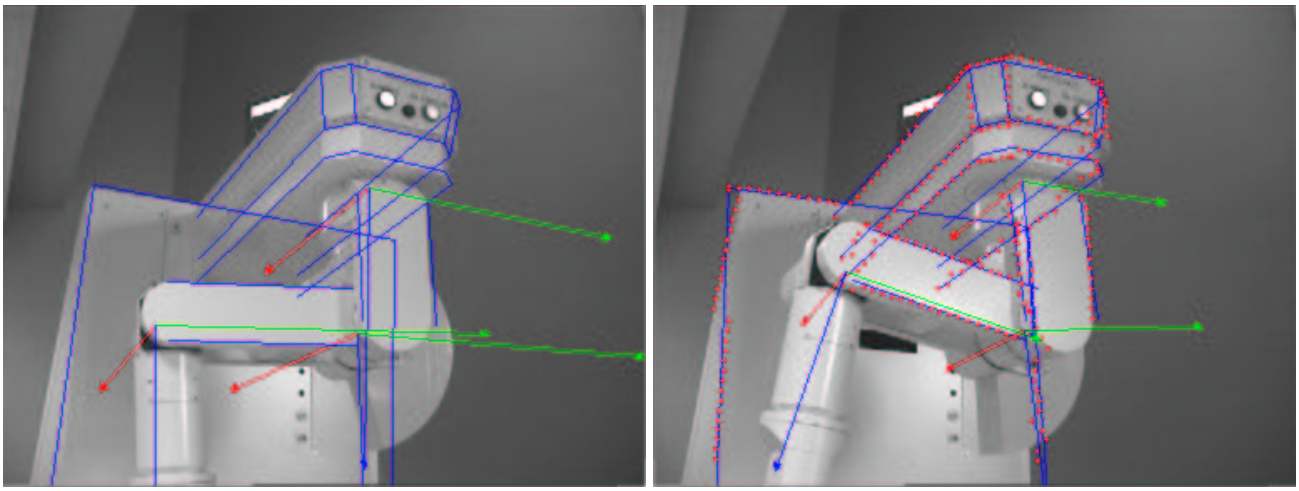

Figure 5: Movement of a robotic arm with two degrees of freedom.

on average 25ms per image. It should be noted that the features used on the components of the arm are not full rank and do not hold enough information to calculate their pose individually. As in the previous experiment, the articulated tracker overcomes this situation.

6 Conclusion

The method presented here demonstrates an efficient approach to tracking complex articulated objects. A framework is given for defining any type of mechanical link between components of a object. A method for object-based tracking has been derived and implemented. Furthermore, a kinematic set formulation for tracking articulated objects has been described. It has been shown that it is possible to decouple the interaction between articulated components using this approach. Subsequent computational efficiency and visual precision have been demonstrated.

In perspective, automatic initialization methods could be considered using partially exhaustive RANSAC [8] based techniques. A better initial estimate could also be obtained using a kinematic chain formulation.

Acknowledgements: This study has been supported by the French government within the RIAM national project SORA.

References

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

[2] A.-I. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pages 36–45, Tokyo, Japan, October 2003.

[3] A.-I. Comport, E. Marchand, and F. Chaumette. Object-based visual 3d tracking of articulated objects via kinematic sets. In *IEEE Workshop on Articulated and Non-Rigid Motion*, Washington, DC, June 2004.

[4] D. Dementhon and L. Davis. Model-based object pose in 25 lines of codes. *Int. J. of Computer Vision*, 15:123–141, 1995.

[5] M. Dhome, M. Richetin, J.-T. Lapresté, and G. Rives. Determination of the attitude of 3-d objects from a single perspective view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[6] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(7):932–946, July 2002.

[7] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[8] N. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395, June 1981.

[9] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[10] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[11] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–394, March 1987.

[12] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[13] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):610–622, June 2000.

[14] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *IEEE Int. Conf. on Computer Vision, ICCV'99*, volume 1, pages 262–268, Kerkira, Greece, September 1999.

[15] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS'02 Conference Proceeding*, volume 21(3) of *Computer Graphics Forum*, pages 289–298, Saarebrücken, Germany, September 2002.

[16] T. Nunomaki, S. Yonemoto, D. Arita, and R. Taniguchi. Multipart non-rigid object tracking based on time model-space gradients. *Articulated Motion and Deformable Objects First International Workshop*, pages 78–82, September 2000.

[17] A. Ruf and R. Horaud. Rigid and articulated motion seen with an uncalibrated stereo rig. In *IEEE Int. Conf. on Computer Vision*, pages 789–796, Corfu, Greece, September 1999.