Object-based visual 3D tracking of articulated objects via kinematic sets

Andrew Comport, E. Marchand, François Chaumette

Object-based visual 3D tracking of articulated objects via kinematic sets

Andrew I. Comport, Éric Marchand, François Chaumette
IRISA - INRIA Rennes
Campus de Beaulieu, 35042 Rennes, France
E-Mail : Firstname.Lastname@irisa.fr

Abstract

A theoretical framework based on robotics techniques is introduced for visual tracking of parametric non-rigid multi-body objects. It is based on an a-priori model of the object including a general mechanical link description. The objective equation is defined in the object-based coordinate system and non-linear minimization relates to the movement of the object and not the camera. This results in simultaneously estimating all degrees of freedom between the object's last known position relative to its previous position as well as internal articulated parameters. 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 derived for defining any type of mechanical link and experimental results show prismatic, rotational and helicoidal 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.

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**. Consequently, components of an object also have some degree of freedom between them. 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.

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, 3]. Another important issue is the 2D-3D registration problem. *Purely geometric* (eg, [5]), or *numerical and iterative* [4] 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, 3]) 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) 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. Recently, more complex features

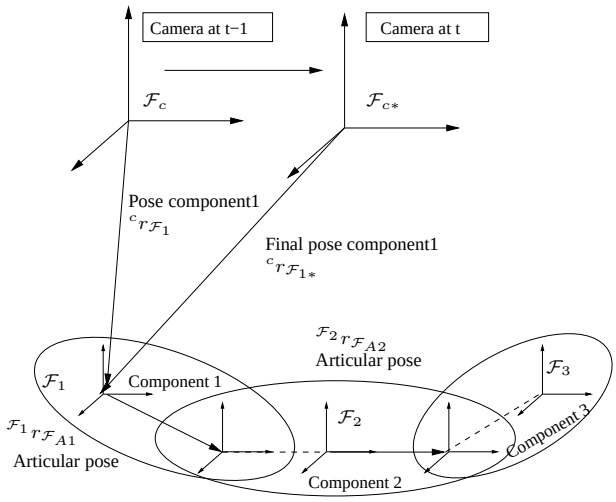

Figure 1. Kinematic chain method: The pose of an articulated object is determined via a kinematic chain of rigid bodies extending to sub components [12, 16, 17].

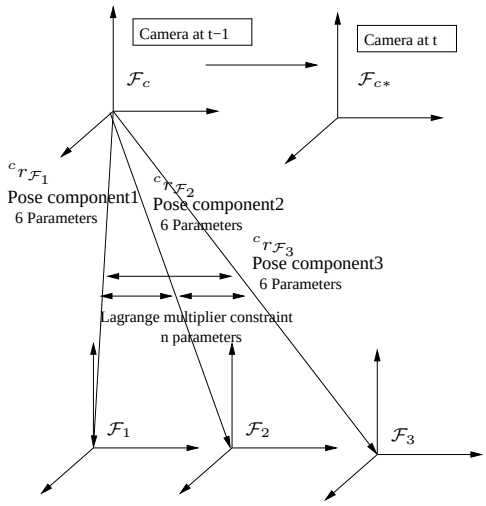

Figure 2. Lagrange Multiplier method: The pose between the camera and each part of the object is calculated directly. Constraints between the components are enforced via Lagrange multipliers [6]

have been used for non-rigid object tracking in [16]. They make use of deformable super-quadric models combined with a kinematic chain approach. 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. The minimization is determined in projective space and is therefore invariant to camera calibration parameters.

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 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* to the joint 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.

It is important, however, to correctly model the behavior of the system so as to obtain this decoupling of joint parameters in the minimization of the objective function. Most vision based tracking methods have been modeled in the camera frame. Indeed the motion of a rigid object can be equally represented as the inverse motion of a moving camera. 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. Therefore it is also desirable to represent these sets in the most general manner. In order to achieve this goal a novel object-based approach is used instead of a camera based approach for tracking, whereby the minimization is carried out in object coordinate space. This models movement as the object's relative velocity with respect to its initial position at each step of the minimization. As will be shown, 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. These sets are decoupled when the 6 parameter velocities, known as twists, are expressed with respect to the axe of a joint. The principal advantages of this approach are that it:

- is more efficient in terms of computation than previous methods.
- it eliminates the propagation of errors between free parameters.
- models more closely the real behavior of the system than a camera based approch and is easy to define.

This method integrates a mechanical link formulation for simple definition of articulations. Using a kinematic set representation eliminates the need to use Lagrange Multipliers and decouples the tracking of the different parts of the object. Recent multi-body dynamics literature [2] has also highlighted the advantage of eliminating Lagrange multipliers from the equation. With regards to the modeling of a non-linear system, this paper addresses the entire class of articulated transformations.

In the remainder of this paper, Section 2 presents the principle of the approach. In Section 3 a robust object-based visual servoing control law is derived. In Section 4 the case of rigid tracking is generalized for the entire set of 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 the object parameters. The set of object parameters are defined by a vector of n parameters $\mathbf{q}$. This vector is composed of the pose between the object and its last known position(not the camera) plus any additional degrees of freedom between components(joint parameters).

In order to maintain an estimate of the object parameters, the underlying idea is to move the object's position virtually, along with its joints, 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 moving the object 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 [3, 15].

To perform the alignment, an error Δ is defined in the image between the projected features $\mathbf{s}\big(\mathbf{q}\big)$ 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 dependent on a *subset* of the object parameters $\mathbf{q}$.

This alignment error is therefore defined as:

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

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

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

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

where $\rho(u)$ is a robust function [9] that grows subquadratically 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. Object-based Control Law

In this first section the simple case of tracking rigid object is described. A new tracking control law is derived in object space and the parameter vector q corresponds to the position of the obect relative to its last known position. The benefits of doing this will become apparent in the following section. The aim of the control scheme is to minimize the objective function given in equation (2). Thus, the error function is given as:

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{q}) - \mathbf{s_d}), \qquad (3)$$

where $\mathbf{D}$ is a diagonal weighting matrix corresponding to the likelihood of a particular error within the robust distribution:

$$\mathbf{D} = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{pmatrix},$$

and where the computation of weights w_i are described in [3].

If $\mathbf{D}$ were constant, the derivative of equation (3) would be given by:

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}}\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\frac{\partial \mathbf{r}}{\partial \mathbf{q}}\frac{d\mathbf{q}}{dt} = \mathbf{D}\mathbf{L_s}\mathbf{V}\dot{\mathbf{q}}, \qquad (4)$$

where $\mathbf{L_s}$ is called the image Jacobian [10] or interaction matrix [7] related to $\mathbf{s}$. $\dot{\mathbf{q}}$ is the object velocity or kinematic twist. $\mathbf{V}(\mathbf{r})$ is a kinematic twist transformation from the camera coordinate system to the object coordinate system given as:

$$\mathbf{V} = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times\mathbf{R} \\ 0_3 & \mathbf{R} \end{bmatrix}, \qquad (5)$$

where $\mathbf{R}$ is a 3x3 rotation matrix and $\mathbf{t}$ a translation vector which are obtained from $\mathbf{r}$. As described in Section 1 the movement seen in the image is initially considered to be object movement. Thus the pose $\mathbf{r}$ between the camera and the object's previously known position at time $t-1$ is assumed to be constant.

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

$$\dot{\mathbf{e}} = -\lambda\mathbf{e}, \qquad (6)$$

where λ is a positive scalar, the following control law is obtained from equation (4):

$$\dot{\mathbf{q}} = -\lambda(\widehat{\mathbf{D}\mathbf{L}_s}\widehat{\mathbf{V}})^+\mathbf{D}\big(\mathbf{s}(\mathbf{q}) - \mathbf{s_d}\big), \qquad (7)$$

where $\widehat{\mathbf{L}_s}$ is a model or an approximation of the real matrix $\mathbf{L_s}$. $\widehat{\mathbf{D}}$ a chosen model for $\mathbf{D}$ and $\widehat{\mathbf{V}}$ depends on the initial pose. In the rigid case $\dot{\mathbf{q}}$ represents the object's velocity with respect to its last position.

The new camera pose can then be determined by applying an instantaneous camera velocity:

$$\mathbf{v} = \widehat{\mathbf{V}}\dot{\mathbf{q}}. \tag{8}$$

where $\mathbf{v}$ is the velocity between the camera and the object.

4. Articular Tracking Control Law

This section addresses the objective of tracking objects with *articulations* between different components. A mechanical 'link' is fully defined by a pair 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:

$$\mathbf{S}^{\perp} = \begin{pmatrix} s^{\perp}_{1,1} & \cdots & s^{\perp}_{1,k} \\ \vdots & \ddots & \\ s^{\perp}_{6,1} & & s^{\perp}_{6,k} \end{pmatrix}, \tag{9}$$

$${}^{c}\mathbf{r}_{\mathcal{A}} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z),$$

where c indicates the camera frame and $_{\mathcal{A}}$ represents the joint frame. 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* k 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 three different types of class 1 links are considered:
A prismatic link along the x axis:

$$S^{\perp} = (1, 0, 0, 0, 0, 0)^T, \tag{10}$$

A rotational link around the x axis:

$$S^{\perp} = (0, 0, 0, 1, 0, 0)^T, \tag{11}$$

A helicoidal link around and along the z axis:

$$S^{\perp} = (0, 0, a, 0, 0, 1)^T, \tag{12}$$

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

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 be contained in se(3), however, joint movement could be considered by sub-algebras of se(3).

The set of velocities that a first component can undertake which leaves a second component invariant is defined by $S^{\perp} \in se(3)$. This is the orthogonal compliment of the subspace $S \in se(3)$ 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):

$$\mathbf{S} = Ker\big((\mathbf{S}^{\perp})^T\big), \tag{13}$$

Using these definitions, the case of rigid object-based tracking can now be generalized to include articulated components. To consider the three different links mentioned, it is necessary to consider an object with two components and one articulation. In this case equation (3) is rewritten as:

$$\begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} = \mathbf{D} \begin{pmatrix} \mathbf{s_1}(\mathbf{q}) - \mathbf{s_{d}}_1 \\ \mathbf{s_1}(\mathbf{q}) - \mathbf{s_{d}}_2 \end{pmatrix}, \tag{14}$$

where $\mathbf{q}$ is a vector composed of the minimal set of velocities corresponding to the object's motion.

Differentiating equation (14) as in equation (4) gives:

$$\begin{aligned} \begin{pmatrix} \dot{\mathbf{e}}_1 \\ \dot{\mathbf{e}}_2 \end{pmatrix} &= \begin{pmatrix} \frac{\partial \mathbf{e}_1}{\partial \mathbf{s_1}} \frac{\partial \mathbf{s_1}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{e}_2}{\partial \mathbf{s_2}} \frac{\partial \mathbf{s_2}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \end{pmatrix} \dot{\mathbf{q}}, \\ &= \mathbf{D}\mathbf{H}\dot{\mathbf{q}}, \end{aligned} \tag{15}$$

where $\dot{\mathbf{q}}$ is a minimal parameter vector of dimension $n = (6 + class)$ which represents the velocity of the object and its joints. The $class$ of the link is defined by the number of degrees of freedom between two components or the number of columns in $\mathbf{S}^{\perp}$. The stacked interaction matrix $\mathbf{H}$ is given by:

$$\mathbf{H} = \begin{pmatrix} \frac{\partial \mathbf{s_1}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{s_2}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{\mathbf{s}1} & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{L}_{\mathbf{s}2} \end{pmatrix} \mathbf{A}, \tag{16}$$

where $\mathbf{A}$ represents the mapping from a vector composed of each component's twist of dimension $m = (ncomponents \times 6)$ to the minimal parameter subspace n. This mapping is called an articulation matrix. This matrix can be found using the image and the kernel given previously:

$$\begin{aligned} \dot{\mathbf{q}}_{s1} &= \begin{pmatrix} \mathbf{S}\dot{\mathbf{q}}_{\cap} & \mathbf{S}^{\perp}\dot{\mathbf{q}}_1 \end{pmatrix}^T, \\ \dot{\mathbf{q}}_{s2} &= \begin{pmatrix} \mathbf{S}\dot{\mathbf{q}}_{\cap} & \mathbf{S}^{\perp}\dot{\mathbf{q}}_2 \end{pmatrix}^T, \end{aligned} \tag{17}$$

where $\dot{\mathbf{q}}_{s1}$ and $\dot{\mathbf{q}}_{s2}$ are 6 parameter twists representing the rigid body velocities for each component and $\dot{\mathbf{q}}_{\cap}$, $\dot{\mathbf{q}}_1$, $\dot{\mathbf{q}}_2$ are vectors representing the intersecting velocities and each components free parameters respectively. The velocities of each component which are not in the intersection belong to the degrees of freedom of the articulation and are denoted

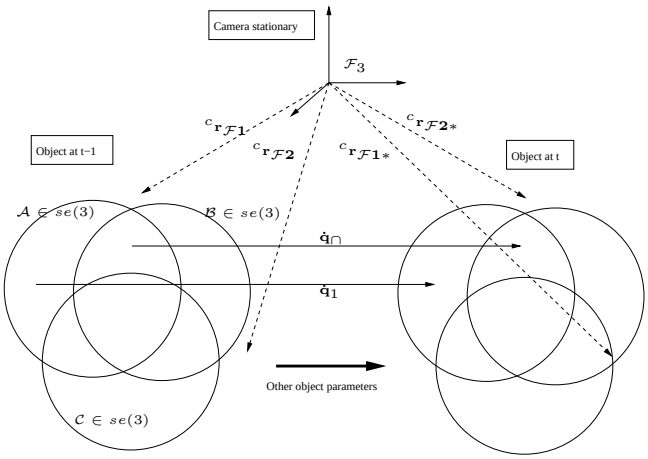

Figure 3. Kinematic set method: The joint parameters are minimized in object space and kinematic set are used to decouple the system. Decoupling occurs at the intersection of parameter sets

$\dot{\mathbf{q}}_1$ and $\dot{\mathbf{q}}_2$. These sets are easily identified when referring to 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. Therefore the twist transform matrix used in the previous subsection can be used to transform the velocities belonging to different components and different frames of reference to a common frame.

The articulation matrix is given by:

$$\begin{aligned}\mathbf{A} &= \begin{pmatrix} \frac{\partial \mathbf{r_1}}{\partial \mathbf{q}_\cap} & \frac{\partial \mathbf{r_1}}{\partial \mathbf{q_1}} & \mathbf{0} \\ \frac{\partial \mathbf{r_2}}{\partial \mathbf{q}_\cap} & \mathbf{0} & \frac{\partial \mathbf{r_2}}{\partial \mathbf{q_2}} \end{pmatrix}, \\ &= \begin{pmatrix} \widehat{\mathbf{V}}\mathbf{S} & \widehat{\mathbf{V}}\mathbf{S}^\perp & \mathbf{0} \\ \widehat{\mathbf{V}}\mathbf{S} & \mathbf{0} & \widehat{\mathbf{V}}\mathbf{S}^\perp \end{pmatrix}, \end{aligned} \tag{18}$$

where $\widehat{\mathbf{V}}$ is the twist transform from the camera frame of reference to the joint frame of reference.

It is important to note that this method introduces decoupling of the minimization problem. This is apparent in equation (18) where extra zeros appear in the Jacobain compared to the traditional case of a kinematic chain. In the particular case of two components and one articulation a kinematic chain has only one zero.

With this decoupling, the velocities which are estimated are represented in a different reference frame as given by:

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{\mathbf{q}}_\cap & \dot{\mathbf{q}}_1 & \dot{\mathbf{q}}_2 \end{pmatrix}^T, \tag{19}$$

Thus, as in equation (7) the control law for tracking articulated objects can be derived:

$$\dot{\mathbf{q}} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{H}})^+ \mathbf{D}\big(\mathbf{s}(\mathbf{q}) - \mathbf{s_d}\big), \tag{20}$$

Once these velocities are obtained they can be related back to the camera frame as in equation (8):

$$\begin{pmatrix} {}^c\mathbf{v}_1 \\ {}^c\mathbf{v}_2 \end{pmatrix} = \mathbf{A} \begin{pmatrix} \dot{\mathbf{q}}_\cap \\ \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{pmatrix}. \tag{21}$$

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 experiment. 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 the 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) [3].

To begin tracking 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.

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

---

1. Obtain initial pose.
2. 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 $(\widehat{\mathbf{D}}\widehat{\mathbf{H}})$.
6. Determine joint velocities as in equation (20).
7. Repeat to 5 until the error converges.
8. Update the pose parameters and repeat to 3.

5.1. Rotational Link

This first experiment was carried out for a class 1 type link with a single degree of rotation on the x axis. The constraint vector was defined as in equation (11) and the object frame was chosen to coincide with the joint frame.

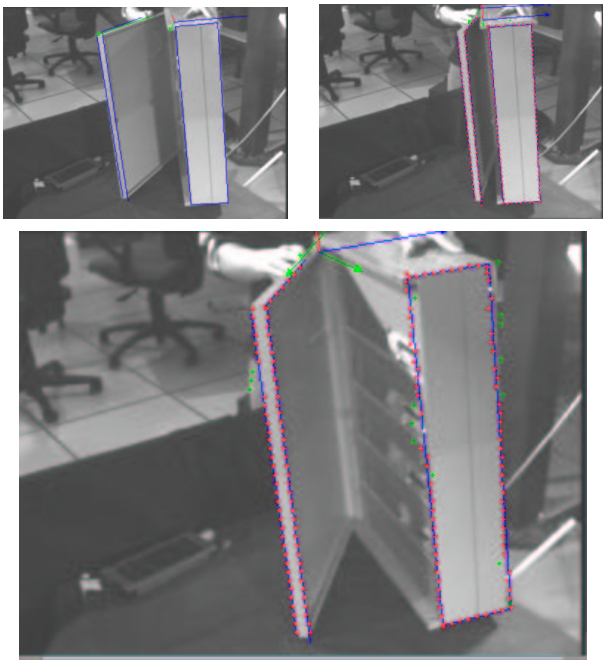

Figure 4. Rotation of a hinged door whilst undergoing movement. In this and all the following figures the reference frames for each component are shown as well as a projection of the CAD model onto the image. The axes of the frames are drawn in yellow, blue and red. The contour of the object is shown in blue. The points rejected by the M-estimator are shown in in green.

This sequence, along with others done with the same object model, have shown real time efficiency with the tracking computation taking on average 25ms per image. Both the hinge and the object were displaced during tracking without failure.

5.2. Prismatic Link

This second experiment was carried out for class one link with a single degree of translation on the x axis. The constraint vector was defined as in equation (10) and the object frame was chosen to coincide with the joint frame.

This sequence also displays real time efficiency with the tracking computation taking on average 20ms per image.

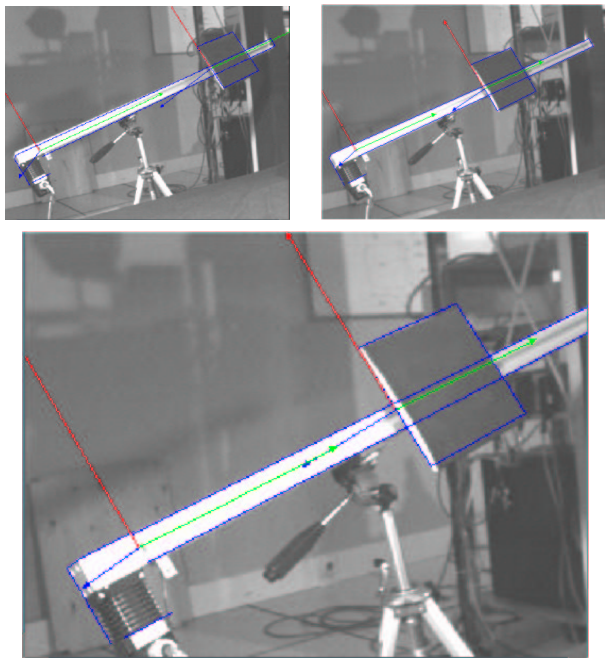

Figure 5. Translation along a sliding mechanism whilst the camera is moving. The first component is the rail and the second is the square which slides along the rail.

Both the camera and the slide were displaced simultaneously during tracking without failure. It can be seen in the image that the alignment of the drawn contour with the object is visually acceptable.

5.3. Helicoidal Link

This third experiment was carried out for class one link with helicoidal movement simultaneously along and around the z axis. The constraint vector was defined as in equation (12) 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 rotations of the screw it translated 4.5cm along the z axis.

This sequence also displays real time efficiency with the tracking computation taking on average 25ms per image. It should be noted that tracking of the screw alone as a rigid object fails completely. 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.

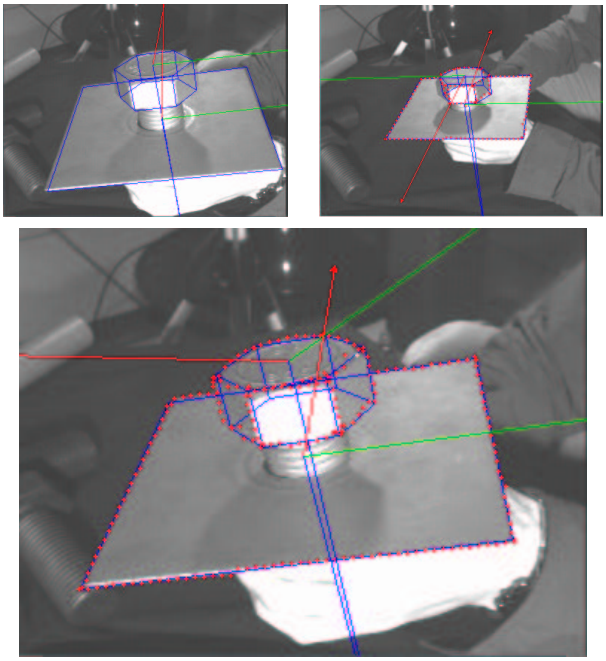

Figure 6. Helicoidal movement of a screw whilst the screw and the platform are simultaneously in movement.

6. Conclusion

The method presented here demonstrates a new approach to tracking 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 derived. 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 be obtained using a kinematic chain formulation. In the future more different classes of links will be considered along with a generalization of the formalism for n components.

References

[1] J. 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] F. Aghili and J. Piedboeuf. Simulation of motion of constrained multibody systems based on projection operator,. *Multibody System Dynamics,*, 10:3–16, 2003.

[3] A. 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.

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

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

[13] C. Lu, G. 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.