Model-Based 3D Tracking of an Articulated Hand

B. Stenger P. R. S. Mendonça R. Cipolla
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ, UK
{bdrs2|prdsm2|cipolla}@eng.cam.ac.uk

Abstract

This paper presents a practical technique for model-based 3D hand tracking. An anatomically accurate hand model is built from truncated quadrics. This allows for the generation of 2D profiles of the model using elegant tools from projective geometry, and for an efficient method to handle self-occlusion. The pose of the hand model is estimated with an Unscented Kalman filter (UKF), which minimizes the geometric error between the profiles and edges extracted from the images. The use of the UKF permits higher frame rates than more sophisticated estimation methods such as particle filtering, whilst providing higher accuracy than the extended Kalman filter. The system is easily scalable from single to multiple views, and from rigid to articulated models. First experiments on real data using one and two cameras demonstrate the quality of the proposed method for tracking a 7 DOF hand model.

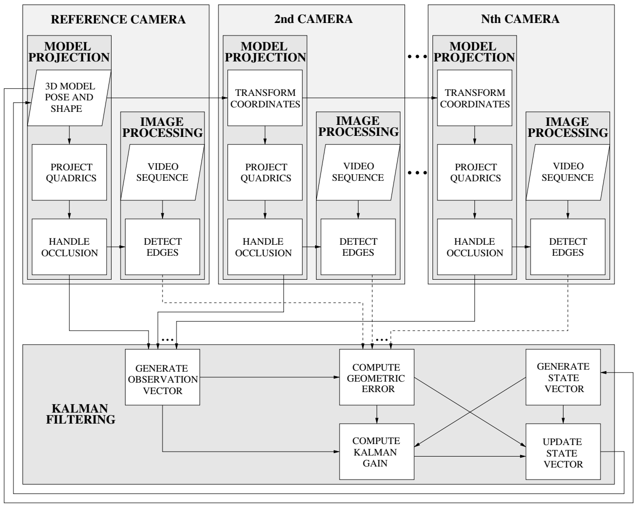

Figure 1. Flowchart of the tracking system. A detailed description of each stage of the process is given in section 3

1. Introduction

Hand tracking has great potential as a tool for better human-computer interaction. This paper presents a method for hand tracking that estimates the pose of a 3D hand model constructed from truncated quadrics by using an *Unscented Kalman filter* [18, 24]. The use of quadrics as building blocks permits the application of elegant techniques from projective geometry [23, 8, 6], as well as the handling of self-occlusion. The profiles of the model are computed for each camera view. These are then compared to images from a video sequence, and the filter is used to estimate the pose of the model (and its covariance matrix), minimizing the geometric error between the projections of the model and edges detected in the images. The flowchart in figure 1 gives an overview of the tracking system.

The next section presents a brief literature survey of hand tracking and human body tracking. Section 2 reviews some of the material on projective geometry of quadrics and conics and on the Unscented Kalman filter that are used in the remainder of the paper. The tracking system proposed here is detailed in section 3. Section 4 shows experimental results from real data, and conclusions are presented in section 5.

1.1. Previous Work

Different methods have been proposed to capture human hand motion. Rehg and Kanade [21] introduced the use of a highly articulated 3D hand model in their *DigitEyes* hand tracking system. For tracking, the axes of the truncated cylinders that are used to model phalanges are projected onto the image, and local edges are found. Finger tip positions are measured through a similar procedure. The error between the measured joint and tip locations is minimized and the locations predicted by the model. The system runs in real-time, but dealing with self-occlusions remains a problem and is treated separately [22]. In [13] a deformable 3D hand model is used. The model is defined by a surface mesh which is constructed via PCA from training examples. Real-time tracking is achieved by finding the closest possibly deformed model matching the image. In [7] a stereo hand tracking system using a 2D deformable model was presented. A two-step algorithm to estimate the hand pose is proposed in [25], first estimating the global pose and subsequently finding the configuration of the joints. However, the algorithm relies on the assumption that all fingertips are visible. Recently, a vision based drawing system was proposed in [16]. The 2D shape of a hand is modeled with B-splines and *partitioned sampling* is used to track contours in real-time. In [26] a model-based method for capturing articulated hand motion is presented. The constraints on the joint configurations are learned from natural hand motions, using a data glove as input device. A sequential Monte Carlo tracking algorithm, based on importance sampling, produces good results, but is view-dependent, and does not handle global hand motion.

Hand tracking is a special case of tracking articulated objects. Most of the work in this area has been done in human body tracking, for which there is an extensive literature. The first works on body tracking are due to O'Rourke and Badler [20] and Hogg [14]. O'Rourke and Badler used a constraint network, based on the structure and dynamics of a human model, to estimate the body pose that best agrees with the location of the head, feet and hands found in the images. Hogg developed a system in which the geometric error between the projection of a model and edges detected in the image is minimized using hierarchical search. A survey of early results in the field can be found in [1]. A direct search of the model parameters in order to match the model projection with the images was presented in [12]. In [19] three cameras are used for tracking in the presence of self-occlusion and an extended Kalman filter is employed for motion prediction. The technique of exponential maps and twist motions is successfully applied to body tracking in [4]. A novel formulation of multiple hypothesis tracking is introduced in [5]. In [9] body silhouettes are tracked with active contours, and a physical force model is used to estimate 3D pose. An annealed version of particle filtering is developed in [10], which, although producing very good results, has extremely high computational cost.

2. Theoretical Framework

2.1. Projective Geometry of Quadrics and Conics

A quadric Q is a second degree implicit surface in 3D space, and it can be represented in homogeneous coordinates as a symmetric 4×4 matrix $\mathbf{Q}$ such that $\mathbf{X}^{\mathrm{T}}\mathbf{Q}\mathbf{X} = 0 \ \forall \mathbf{x} \in Q$ [23]. Consider a quadric $\mathbf{Q}$ given by $\mathbf{Q} = \left[\begin{smallmatrix}\mathbf{A} & \mathbf{b}\\ \mathbf{b}^{\mathrm{T}} & c\end{smallmatrix}\right]$, seen from a normalized projective camera $\mathbf{P} = [\mathbb{I} \mid \mathbf{0}]$. The camera center and a point $\mathbf{x}$ in image coordinates define a ray $\mathbf{X}(s) = \left[\begin{smallmatrix}\mathbf{x}\\ s\end{smallmatrix}\right]$. The depth of the point in space is determined by the free parameter s (see figure 2). The point of intersection of the ray with the quadric $\mathbf{Q}$ can be found by solving the equation

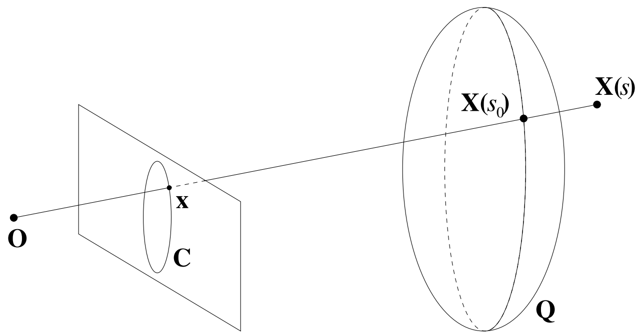

Figure 2. A quadric $\mathbf{Q}$ and its projection $\mathbf{C}$ on the image plane.

$$\mathbf{X}(s)^{\mathrm{T}}\mathbf{Q}\mathbf{X}(s) = 0 \ , \tag{1}$$

which can be written as

$$cs^2 + 2\mathbf{b}^{\mathrm{T}}\mathbf{x}s + \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} = 0 \ . \tag{2}$$

When the ray represented by $\mathbf{X}(s)$ is tangent to $\mathbf{Q}$, equation (2) will have a unique solution. Algebraically, this condition is satisfied when the discriminant is zero, i.e.

$$\mathbf{x}^{\mathrm{T}}(\mathbf{b}\mathbf{b}^{\mathrm{T}} - c\mathbf{A})\mathbf{x} = 0 \ . \tag{3}$$

Therefore, the profile of the quadric $\mathbf{Q}$ is a conic $\mathbf{C}$, given by $\mathbf{C} = c\mathbf{A} - \mathbf{b}\mathbf{b}^{\mathrm{T}}$ [6]. Note that this formula is valid only for a normalized camera $\mathbf{P} = [\mathbb{I} \mid \mathbf{0}]$. To obtain the image of a quadric $\mathbf{Q}$ in an arbitrary projective camera $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ it is necessary to compute the transformation $\mathbf{H}$ such that $\mathbf{P}\mathbf{H} = [\mathbb{I} \mid \mathbf{0}]$. In this new coordinate system $\mathbf{Q}$ is represented as $\tilde{\mathbf{Q}}$, where $\tilde{\mathbf{Q}} = \mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H}$.

For points $\mathbf{x} \in \mathbf{C}$, the unique solution of equation (2) is $s_0 = -\mathbf{b}^{\mathrm{T}}\mathbf{x}/c$, which gives the depth of the point on the contour generator of quadric $\mathbf{Q}$.

The tangent $\mathbf{l}$ to the conic at the point $\mathbf{x} \in \mathbf{C}$ is given by $\mathbf{l} = [l_1,\ l_2,\ l_3]^{\mathrm{T}} = \mathbf{C}\mathbf{x}$ [23]. Therefore, the homogeneous representation of the normal $\mathbf{n}$ of $\mathbf{C}$ at $\mathbf{x}$ is given by $\mathbf{n} = [l_1,\ l_2,\ 0]^{\mathrm{T}}$.

If the matrix $\mathbf{Q}$ of a quadric is singular, the quadric is said to be *degenerate*. Different families of quadrics are obtained from matrices $\mathbf{Q}$ of different ranks. Particular cases of interest are:

ellipsoids, represented by matrices $\mathbf{Q}$ with full rank,

cones and cylinders, represented by matrices $\mathbf{Q}$ with $\mathrm{rank}(\mathbf{Q}) = 3$,

a pair of planes π and π', represented as $\mathbf{Q} = \boldsymbol{\pi}\boldsymbol{\pi}'^{\mathrm{T}} + \boldsymbol{\pi}'\boldsymbol{\pi}^{\mathrm{T}}$ with $\mathrm{rank}(\mathbf{Q}) = 2$.

In order to employ quadrics for modeling more general shapes, it is necessary to truncate them. For any quadric $\mathbf{Q}$ the truncated quadric $\mathbf{Q_\Pi}$ can be obtained by finding points $\mathbf{X}$ satisfying:

$$\mathbf{X}^{\mathrm{T}}\mathbf{Q}\mathbf{X} = 0 \quad \text{and} \quad \mathbf{X}^{\mathrm{T}}\mathbf{\Pi}\mathbf{X} \geq 0, \tag{4}$$

where $\mathbf{\Pi}$ is a matrix representing a pair of clipping planes.

2.2. Nonlinear Filtering

The tracking of an object in 3D space from images can be formulated as a nonlinear estimation problem. This formulation allows the use of any nonlinear estimation technique, such as extended Kalman filtering (EKF) [17, 2], particle filtering [15], or other Monte Carlo methods [11]. The Unscented Kalman filter (UKF), an alternative to the EKF, has been proposed by Julier et al. [18]. It is provably superior to the EKF, producing better estimates of the covariance matrices of the parameters involved. It is also more efficient and simpler to implement, avoiding the computation of Jacobian matrices, necessary to propagate distributions in the EKF. Instead, a small number of carefully chosen sample points is propagated in each estimation step, which provide a compact parameterization of the underlying distribution. This is also in contrast to random sampling methods such as particle filtering which demand a larger number of sample points, and are therefore computationally expensive. It should be noted, however, that both EKF and UKF assume unimodal distributions.

Consider the nonlinear state transition equation

$$\mathcal{X}(k+1) = \mathbf{f}\big(\mathcal{X}(k), \mathbf{v}(k+1), k+1\big) \tag{5}$$

where $\mathbf{f}$ describes the system dynamics, $\mathcal{X}(k)$ is the n-dimensional state of the system at time step k and $\mathbf{v}(k+1)$ is the process noise. The covariance matrix of the state distribution is given by $\Sigma_{\mathcal{X}}$. A set of observations, related to the state vector, are obtained through the equation

$$\mathcal{Z}(k+1) = \mathbf{h}\Big(\mathcal{X}(k+1), \mathbf{w}(k+1), k+1\Big), \tag{6}$$

where $\mathcal{Z}(k+1)$ is the observation vector, $\mathbf{h}$ is the observation model and $\mathbf{w}(k+1)$ is the measurement noise. An overview of the filtering algorithm is given in algorithm 1.

Algorithm 1 Unscented Kalman Filtering (UKF) Algorithm.

1: Select $2n+1$ points according to

$$\mathcal{X}_l(k|k) = \begin{cases} \hat{\mathcal{X}}(k|k) & l = 0 \\ \hat{\mathcal{X}}(k|k) - \sigma_k^l & l = 1, ..., n \\ \hat{\mathcal{X}}(k|k) + \sigma_k^{l-n} & l = n+1, ..., 2n \end{cases}, \tag{7}$$

where σ_k^l is the lth column of the matrix $\sqrt{n\Sigma_{\mathcal{X}}(k|k)}$.

2: Compute $\mathcal{X}_l(k+1|k)$ by applying the system equation (5) to $\mathcal{X}_l(k|k)$.

3: Compute the predicted state $\hat{\mathcal{X}}(k+1|k)$ (and the error covariance matrix)

$$\hat{\mathcal{X}}(k+1|k) = \frac{1}{2n+1}\sum_{l=0}^{2n} \mathcal{X}_l(k+1|k). \tag{8}$$

4: Compute $\mathcal{Z}_l(k+1|k)$ by applying the observation equation (6) to $\mathcal{X}_l(k+1|k)$.

5: Compute the predicted observation $\hat{\mathcal{Z}}(k+1|k)$ as

$$\hat{\mathcal{Z}}(k+1|k) = \frac{1}{2n+1}\sum_{l=0}^{2n} \mathcal{Z}_l(k+1|k). \tag{9}$$

6: Compute the innovation $\nu(k+1) = \mathcal{Z}(k+1) - \hat{\mathcal{Z}}(k+1|k)$ from the current measurement $\mathcal{Z}(k+1)$ and the predicted observation $\hat{\mathcal{Z}}(k+1|k)$.

7: Update the Kalman gain matrix $\mathbf{K}(k+1)$.

8: Update the estimate of the state vector (and the error covariance matrix)

$$\hat{\mathcal{X}}(k+1|k+1) = \hat{\mathcal{X}}(k+1|k) + \mathbf{K}(k+1)\nu(k+1). \tag{10}$$

3. 3D-Model Based Tracking

3.1. Description of the Hand Model

The hand model is built using a set of quadrics $\mathbf{Q}_i, i \in \{1, ..., 39\}$, approximately representing the anatomy of a real human hand, as shown in figure 3. Similar to Rehg and Kanade [21], we use a hierarchical model with 27 degrees of freedom (DOF): 6 for the global hand position, 4 for the pose of each finger and 5 for the pose of the thumb. The DOF for each joint correspond to the DOF of a real hand. Only 7 DOF are currently tracked in our system, global hand motion and thumb extension. All other joints are kept fixed. Starting from the palm and ending at the tips, the coordinate system of each quadric is defined relative to the previous one in the hierarchy.

The palm is modeled using a truncated cylinder, its top and bottom closed by half-ellipsoids. Each finger consists of three segments of a cone, one for each phalanx. They are connected by hemispheres, representing the joints. The phalanges of the thumb are represented by an ellipsoid, a truncated cylinder and a truncated cone. Hemispheres are used for the tips of fingers and thumb. The shape parameters of each quadric are set by taking measurements from a real

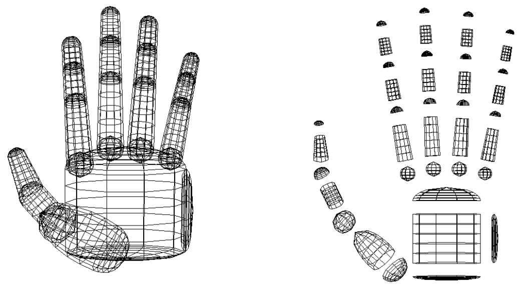

Figure 3. The 27 DOF hand model is constructed from 39 truncated quadrics. Front view (left) and exploded view (right) are shown.

hand.

3.2. Generation of the Contours

Each clipped quadric of the hand model is projected individually as described in section 2.1, generating a list of clipped conics. For each conic matrix $\mathbf{C}$ we use eigendecomposition to obtain a factorization given by

$$\mathbf{C} = \mathbf{T}^{-\mathrm{T}}\mathbf{D}\mathbf{T}^{-1} \quad , \tag{11}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{12}$$

with $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbb{I}$. The diagonal matrix $\mathbf{D}$ represents a conic aligned with the x- and y-axis and centered at the origin. The matrix $\mathbf{T}$ is the representation in homogeneous coordinates of a Euclidean transformation that maps this conic onto $\mathbf{C}$. We can therefore draw $\mathbf{C}$ by drawing $\mathbf{D}$ and transforming the rendered points according to $\mathbf{T}$. The drawing of $\mathbf{D}$ is carried out by different methods, depending on its rank. For $\mathrm{rank}(\mathbf{D}) = 3$ an ellipse, for $\mathrm{rank}(\mathbf{D}) = 2$ a pair of lines is drawn.

The next step is the handling of self-occlusion, achieved by comparing the depths of points in 3D space. In section 2.1 it was shown how to obtain the depth of a point $\mathbf{X}(s_0)$ on the contour generator of a quadric $\mathbf{Q}$. In order to check if $\mathbf{X}(s_0)$ is visible, equation (1) is solved for each of the other quadrics $\mathbf{Q}_i$ of the hand model. In the general case there are two solutions s_1^i and s_2^i, yielding the points where the ray intersects with quadric $\mathbf{Q}_i$. The point $\mathbf{X}(s_0)$ is visible if $s_0 \geq s_j^i \quad \forall i, j$, in which case the point $\mathbf{x}$ is drawn. Figure 4 shows an example of the projection of the hand model with occlusion handling.

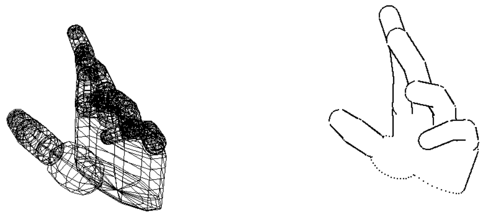

Figure 4. Handling self-occlusion: The 3D model (left) and its generated contour (right) are shown.

3.3. Construction of the State and Observation Vectors

The state vector $\mathcal{X}$ contains the global pose of the hand and the configuration of the joints. Additionally, components modeling the hand motion, such as velocity and acceleration, can be included. In the most general case the state vector will have dimension $27n$, where $n - 1$ is the order of the dynamic model.

The observation vector $\mathcal{Z}$ is obtained by detecting edges in the neighborhood of the projected hand model. Let $S_i^v = \{\mathbf{X}_i^{v,1}, \mathbf{X}_i^{v,2}, ..., \mathbf{X}_i^{v,N_i}\}$ be the set of visible (not occluded) points on the contour generator of $\mathbf{Q}_i$ as seen from camera $\mathbf{P}_v$, and let $\mathbf{C}_i^v$ be the projection of $\mathbf{Q}_i$ on $\mathbf{P}_v$. The image of each point $\mathbf{X}_i^{v,j}$ is denoted by $\mathbf{x}_i^{v,j}$. The vector $\mathbf{n}_i^{v,j}$ normal to $\mathbf{C}_i^v$ at $\mathbf{x}_i^{v,j}$ can be obtained, as described in section 2.1. For each point $\mathbf{x}_i^{v,j}$ one looks for edges along the normal $\mathbf{n}_i^{v,j}$ (see for example [3]). The intensity values in the images are convolved with a derivative of a Gaussian kernel and an edge is assigned to the position $\mathbf{e}_i^{v,j}$ with the largest absolute value. The observation vector $\mathcal{Z}$ is constructed by stacking the inner products $\mathbf{n}_i^{v,j\mathrm{T}}\mathbf{e}_i^{v,j}$ into a single vector.

The predicted observation vector for the UKF is obtained by projecting the hand model corresponding to the state vector $\mathcal{X}_0(k+1|k)$ on each image. One then obtains a set of reference contours, for which a list of image points $I_0 = \{\mathbf{x}_i^{v,j}\}$ is computed together with the corresponding normals. Each of the remaining state vectors $\mathcal{X}_l(k+1|k)$ is used to compute new contours and new lists I_l of image points. The vectors $\mathcal{Z}_l(k+1|k)$ are then constructed by stacking the inner products $\mathbf{n}_i^{v,j\mathrm{T}}\mathbf{x}_i^{v,j}$, where the points $\mathbf{x}_i^{v,j}$ are in the list I_l. The predicted observation can be found according to (9).

Each component of the innovation vector will then have the form $\mathbf{n}^{\mathrm{T}}(\mathbf{e} - \tilde{\mathbf{x}})$, where $\tilde{\mathbf{x}}$ is the average position of the contour points in a single image, $\mathbf{e}$ is the corresponding edge in that image and $\mathbf{n}$ is the corresponding normal vector

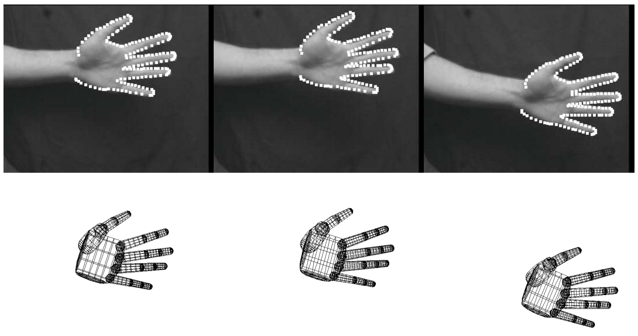

Figure 5. Tracking of an open hand with a single camera. Top row: contours superimposed on frames. Bottom row: estimated 3D pose of hand model.

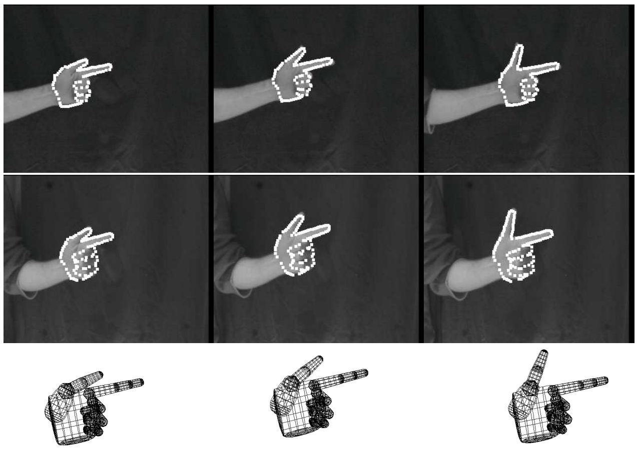

Figure 6. Tracking of a pointing hand with extension of the thumb using two cameras. Top and middle row: model projected on images from camera 1 and 2, respectively. Bottom row: corresponding 3D pose as seen in view 1.

of the reference contour. Therefore, a component of the innovation is the distance between the average contour point and the corresponding edge. Thus the innovation can be interpreted as the error in pixels between the projection of the hand model on each image and the edges in that image.

4. Experimental Results

Real data experiments were designed to test the proposed tracking algorithm. Three sequences of eighty 360×288 grey-scale images of a hand in front of a dark background were acquired. The parameters of the hand model were manually set to match the pose of the hand in the first frame. Six DOF were given to the motion of the hand and one DOF was given to the configuration of the thumb. This allows for

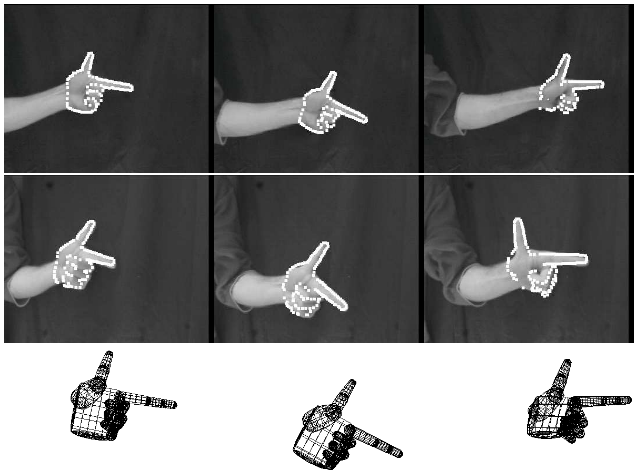

Figure 7. Tracking of a pointing hand from two views. Top and middle row: model projected on images from camera 1 and 2, respectively. Bottom row: corresponding 3D pose as seen in view 1.

the modeling of arbitrary rigid body motion plus the flexion of the thumb, simulating a “point and click” interface. The dynamics of the hand and thumb were modeled using a second order process, i.e. using position, velocity and acceleration.

The results of the tracking algorithm can be seen in figures 5, 6 and 7. Figure 5 illustrates the result of tracking an open hand using a single camera. The images in the top row show the contours superimposed on selected frames of the sequence, and the bottom row shows the estimated 3D pose of the hand model. The tracking of a pointing hand from two views is shown in figures 6 and 7. The first two rows show the projection of the model on images from each view. The corresponding 3D pose of the hand can be seen in the bottom row. From figure 6 it is clear that the configuration of the thumb is accurate. The quality of the estimation of the 3D position and orientation of the model is demonstrated in figure 6.

For a single camera the tracking algorithm operates at a rate of 3 frames per second on a Celeron 433MHz machine. Note that the computational complexity grows linearly with the number of cameras. Although the system is not operating in real-time yet, it is expected that this goal can be achieved by code optimization and using a faster machine. Moreover, the algorithm can be implemented on distributed systems in a straightforward manner.

5. Summary and Conclusions

This paper presented a novel model-based hand tracking system. The use of quadrics to build the 3D model yields a practical and elegant method for generating the contours of the model, which are then compared with the image data. This measurements are used by an Unscented Kalman filter to estimate the current motion and configuration parameters of the model. Results with real data demonstrate the efficiency of the proposed method. The system is easily scalable from single to multiple views, and from rigid to articulated models. Real-time performance is expected to be achieved for an implementation on a distributed system, using one processor per camera.

Acknowledgements

This work has been supported by EPSRC, award ref. 301321, and the Gottlieb Daimler- and Karl Benz-Foundation. Paulo R. S. Mendonça would like to acknowledge the financial support of CAPES, Brazilian Ministry of Education, grant BEX1165/96-8.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. In *IEEE Nonrigid and Articulated Motion Workshop*, pages 90–102, San Juan, Puerto Rico, June 1997.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Number 179 in Mathematics in science and engineering. Academic Press, Boston, 1988.

[3] A. Blake and M. Isard. *Active Contours : The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag, London, 1998.

[4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 8–15, Santa Barbara, USA, June 1998.

[5] T. J. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 239–245, Fort Collins, USA, June 1999.

[6] R. Cipolla and P. J. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, Cambridge, UK, 1999.

[7] R. Cipolla and N. J. Hollinghurst. Human-robot interface by pointing with uncalibrated stereo vision. *Image and Vision Computing*, 14(3):171–178, Apr. 1996.

[8] G. Cross and A. Zisserman. Quadric reconstruction from dual-space geometry. In *Proc. 6th Int. Conf. on Computer Vision*, pages 25–31, Bombay, India, Jan. 1998.

[9] Q. Delamarre and O. D. Faugeras. Articulated models and multi-view tracking with silhouettes. In *Proc. 7th Int. Conf. on Computer Vision*, volume II, pages 716–721, Corfu, Greece, Sept. 1999.

[10] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 126–133, Hilton Head Island, USA, June 2000.

[11] A. Doucet, N. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science Series. Springer-Verlag, New York, 2001.

[12] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, June 1996.

[13] A. J. Heap and D. C. Hogg. Towards 3-D hand tracking using a deformable model. In *2nd International Face and Gesture Recognition Conference*, pages 140–145, Killington, USA, Oct. 1996.

[14] D. C. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, Feb. 1983.

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

[16] M. Isard and J. MacCormick. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. 6th European Conf. on Computer Vision*, volume 2, pages 3–19, Dublin, Ireland, June 2000.

[17] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[18] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. American Control Conference*, pages 1628–1632, Seattle, USA, June 1995.

[19] I. A. Kakadiaris and D. N. Metaxas. Model-based estimation of 3-D human motion with occlusion based on active multi-viewpoint selection. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 81–87, San Francisco, June 1996.

[20] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intell.*, 2(6):522–536, Nov. 1980.

[21] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In J.-O. Eklundh, editor, *Proc. 3rd European Conf. on Computer Vision*, volume II of *Lecture Notes in Computer Science 801*, pages 35–46. Springer-Verlag, May 1994.

[22] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. 5th Int. Conf. on Computer Vision*, pages 612–617, Cambridge, USA, June 1995.

[23] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford Classic Texts in the Physical Sciences. Clarendon Press, Oxford, UK, 1998. Originally published in 1952.

[24] E. A. Wan and R. van der Merve. The Unscented Kalman Filter for nonlinear estimation. In *Proc. of IEEE Symposium 2000 on Adaptive Systems for Signal Processing, Communications and Control*, pages 153–158, Lake Louise, Canada, Oct. 2000.

[25] Y. Wu and T. S. Huang. Capturing articulated human hand motion: A divide-and-conquer approach. In *Proc. 7th Int. Conf. on Computer Vision*, volume I, pages 606–611, Corfu, Greece, Sept. 1999.

[26] Y. Wu, L. J. Y., and T. S. Huang. Capturing natural hand articulation. In *Proc. 8th Int. Conf. on Computer Vision*, volume II, pages 426–432, Vancouver, Canada, July 2001.