Model-Based Hand Tracking Using an Unscented Kalman Filter

B. Stenger, P. R. S. Mendonça and R. Cipolla
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ, UK
[bdrs2|prdsm2|cipolla]@eng.cam.ac.uk
svr-www.eng.cam.ac.uk/research/vision/

Abstract

This paper presents a novel method for hand tracking. It uses a 3D model built from quadrics which approximates the anatomy of a human hand. This approach allows for the use of results from projective geometry that yield an elegant technique to generate the projection of the model as a set of conics, as well as providing an efficient ray tracing algorithm to handle self-occlusion. Once the model is projected, an Unscented Kalman Filter is used to update its pose in order to minimise the geometric error between the model projection and a video sequence on the background. Results from experiments with real data show the accuracy of the technique.

1 Introduction

Hand tracking has great potential as a tool for better human-computer interaction. This paper introduces a new method for hand tracking that estimates the pose of a 3D hand model constructed from truncated quadrics by using an *Unscented Kalman Filter* [11, 14]. The use of quadrics as building blocks for the model permits the application of powerful techniques concerning the projective geometry of such surfaces [13, 5, 3], as well as the handling of self-occlusion. An image of the model is then compared to a video sequence, and the filter is then used to estimate the pose of the model (and its covariance matrix), minimising the geometric error between the projection of the model and edges detected in the image. An overview of the tracking system is shown in the flowchart on figure 1.

The next section presents a brief literature survey of hand tracking. Section 3 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 4. Section 5 shows experimental results from real data, and the conclusions are presented in section 6.

2 Literature Review

Different methods have been proposed to capture human hand motion. Rehg and Kanade [12] introduced the use of a highly articulated 3D hand model for the tracking of a human

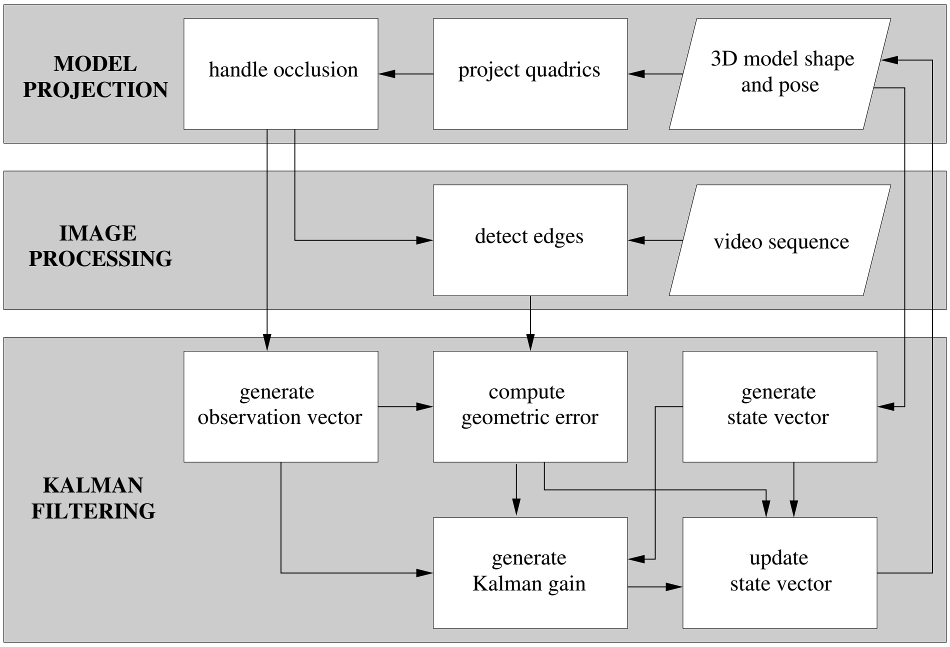

Figure 1: Flowchart of the tracking system. A detailed description of each stage of the process is given in section 4

hand. 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. A nonlinear least squares method is used to minimise the error between the measured joint and tip locations and the locations predicted by the model. The system runs in real-time, however, dealing with occlusions and handling background clutter remains a problem. Heap and Hogg [7] used a deformable 3D hand shape model. The hand is modelled as 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 [4], Cipolla and Hollinghurst presented a stereo handtracking system using a 2D model deformable by affine transformations. Wu and Huang [15] proposed a two-step algorithm to estimate the hand pose, first estimating the global pose and subsequently finding the configuration of the joints. However, their algorithm relies on the assumption that all fingertips are visible. Recently, Isard and MacCormick [9] have presented a vision based drawing system. The 2D hand shape is modelled with B-splines and partitioned sampling is used to track contours in real-time.

3 Theoretical Background

3.1 Projective Geometry of Quadrics and Conics

A quadric 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 $\mathbf{X}^{\mathrm{T}}\mathbf{Q}\mathbf{X} = 0$ [13]. The

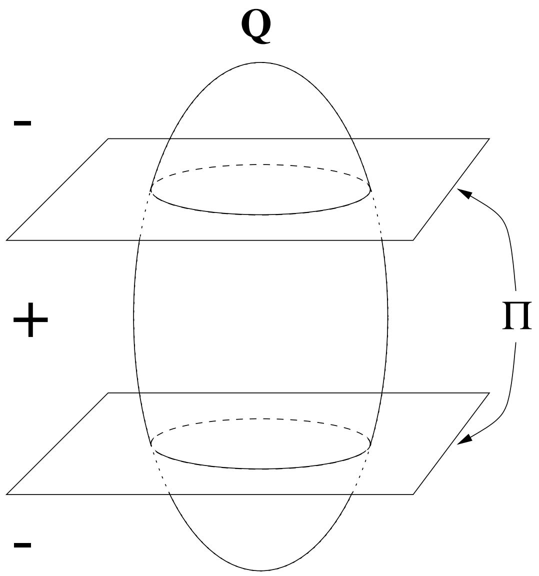

Figure 2: A truncated Quadric $\mathbf{Q}_\Pi$, e.g. a truncated ellipsoid, can be obtained by finding points on quadric $\mathbf{Q}$ which satisfy $\mathbf{X}^\mathrm{T}\Pi\mathbf{X} \geq 0$.

image of a quadric $\mathbf{Q} = \left[\begin{smallmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^\mathrm{T} & c \end{smallmatrix}\right]$ seen from a normalised projective camera $P = [\mathbb{I} \mid \mathbf{0}]$ is a conic $\mathbf{C}$ given by $\mathbf{C} = c\mathbf{A} - \mathbf{b}\mathbf{b}^\mathrm{T}$ [3]. The tangent $\mathbf{l}$ to the conic at the point $\mathbf{x} \in \mathbf{C}$ is given by $\mathbf{l} = [l_1,\ l_2,\ l_3]^\mathrm{T} = \mathbf{C}\mathbf{x}$. 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 $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, 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 modelling 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:

$$\mathbf{X}^\mathrm{T}\mathbf{Q}\mathbf{X} = 0 \tag{1}$$

$$\text{and} \qquad \mathbf{X}^\mathrm{T}\Pi\mathbf{X} \geq 0, \tag{2}$$

where Π is a matrix representing a pair of clipping planes (see figure 2).

3.2 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) [10, 1], the CONDENSATION algorithm [8], or Monte Carlo methods [6]. The Unscented Kalman Filter (UKF), an alternative to the EKF, has been proposed by Julier and Uhlmann [11]. It is provably superior to the EKF in most practical situations. It is also computationally more efficient, thus permitting higher frame rates of the tracking algorithm. The computation of Jacobian matrices is avoided, which is 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 parameterisation of the underlying distribution. This is also in contrast to random sampling methods such as CONDENSATION or Monte Carlo based techniques which demand a larger number of sample points, and are therefore computationally expensive.

Consider the nonlinear state transition equation

$$\mathcal{X}(k+1) = \mathbf{f}\Big(\mathcal{X}(k), \mathbf{u}(k+1), k+1\Big) + \mathbf{v}(k+1), \tag{3}$$

where $\mathbf{f}$ describes the system dynamics, $\mathcal{X}(k)$ is the n-dimensional state of the system at timestep k, $\mathbf{u}(k+1)$ is a control input vector, 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

$$\mathcal{Z}(k+1) = \mathbf{h}\Big(\mathcal{X}(k+1), \mathbf{u}(k+1), k+1\Big) + \mathbf{w}(k+1), \tag{4}$$

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 a set of $2n$ sample points σ_l, $l = 1, 2, ..., 2n$, as the columns of $\pm\sqrt{n\Sigma_{\mathcal{X}}(k|k)}$.
2. Compute $\mathcal{X}_0(k|k) = \hat{\mathcal{X}}(k|k)$ and $\mathcal{X}_l(k|k) = \sigma_l + \hat{\mathcal{X}}(k|k)$.
3. Compute $\mathcal{X}_l(k+1|k)$ by applying the system equation (3) to $\mathcal{X}_l(k|k)$.
4. Compute the predicted state $\hat{\mathcal{X}}(k+1|k)$ as

$$\hat{\mathcal{X}}(k+1|k) = \frac{1}{2n+1} \sum_{l=0}^{2n} \mathcal{X}_l(k+1|k). \tag{5}$$

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

$$\hat{\mathcal{Z}}(k+1|k) = \frac{1}{2n+1} \sum_{l=0}^{2n} \mathcal{Z}_l(k+1|k). \tag{6}$$

7. 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 $\hat{\mathcal{Z}}(k+1|k)$.
8. Update the Kalman gain matrix $\mathbf{G}(k+1)$.
9. Update the estimate of the state vector

$$\hat{\mathcal{X}}(k+1|k+1) = \hat{\mathcal{X}}(k+1|k) + \mathbf{G}(k+1)\nu(k+1). \tag{7}$$

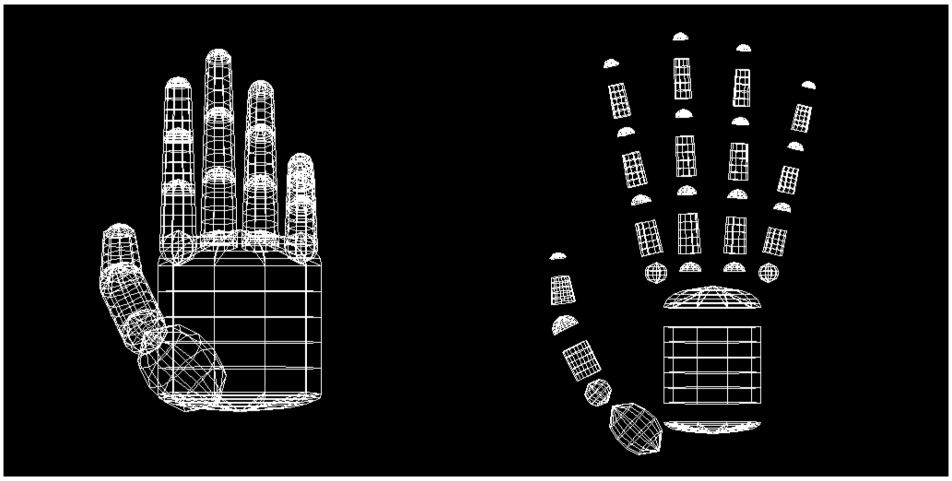

Figure 3: The 27 DOF hand model is constructed from 37 truncated quadrics. Front view (left) and exploded view (right) are shown.

4 3D Model Based Tracking

4.1 Description of the Hand Model

The hand model is built using a set of quadrics $\mathbf{Q}_i, i \in 1, ..., 37$, approximately representing the anatomy of a real human hand as shown in figure 3. Similar to Rehg [12], 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 pose of the thumb. The DOF for each joint correspond to the DOF of a real hand. 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 modelled 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 hand.

4.2 Generation of the Contours

Each clipped quadric of the hand model is projected individually as described in section 3.1, generating a list of clipped conics. For each conic matrix $\mathbf{C}$ we use eigendecomposition to obtain a factorisation given by $\mathbf{C} = \mathbf{T}^{-\mathrm{T}}\mathbf{D}\mathbf{T}^{-\mathbf{1}}$, where

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{8}$$

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 centred 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

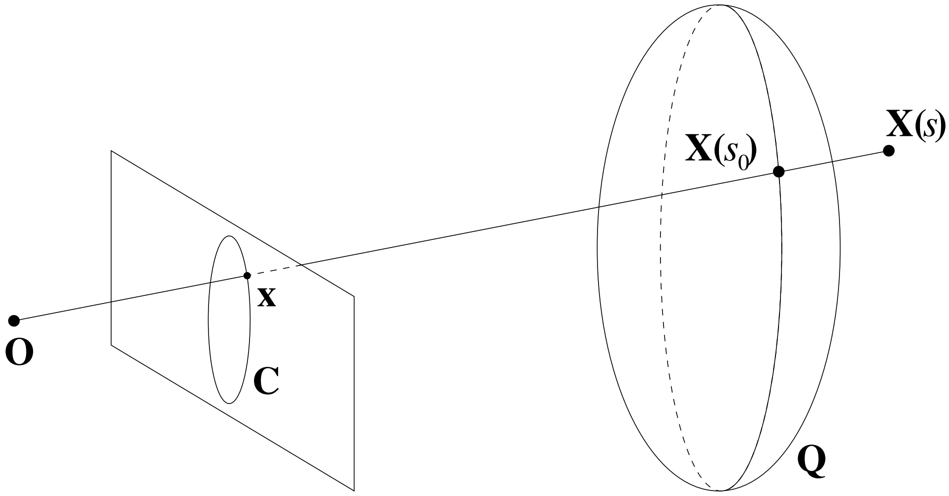

Figure 4: A quadric $\mathbf{Q}$ and its projection $\mathbf{C}$ on the image plane.

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 $\mathrm{rank}(\mathbf{D}) = 3$ we draw an ellipse, for $\mathrm{rank}(\mathbf{D}) = 2$ we draw a pair of lines.

The next step is the handling of occlusion, achieved by a simple ray tracing algorithm. Consider a point $\mathbf{x}$ on the conic $\mathbf{C}$, obtained by projecting the quadric $\mathbf{Q}$, as shown in figure 4. The camera centre and $\mathbf{x}$ define a 3D ray L. Each point $\mathbf{X} \in L$ is given by $\mathbf{X}(s) = \left[\begin{smallmatrix}\mathbf{x}\\ s\end{smallmatrix}\right]$, where s is a free parameter determining the depth of the point in space, such that the point $\mathbf{X}(0)$ is at infinity and $\mathbf{X}(\infty)$ is at the camera centre. The point of intersection of the ray with the quadric $\mathbf{Q}$ is found by solving the equation

$$\mathbf{X}(s)^{\mathrm{T}}\mathbf{Q}\mathbf{X}(s) = 0 \tag{9}$$

for s. Writing $\mathbf{Q} = \left[\begin{smallmatrix}\mathbf{A} & \mathbf{b}\\ \mathbf{b}^{\mathrm{T}} & c\end{smallmatrix}\right]$, the unique solution of (9) is given by $s_0 = -\mathbf{b}^{\mathrm{T}}\mathbf{x}/c$.

In order to check if $\mathbf{X}(s_0)$ is visible, (9) 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 5 shows an example of the projection of the hand model with occlusion handling.

4.3 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 modelling 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 neighbourhood of the projected hand model.

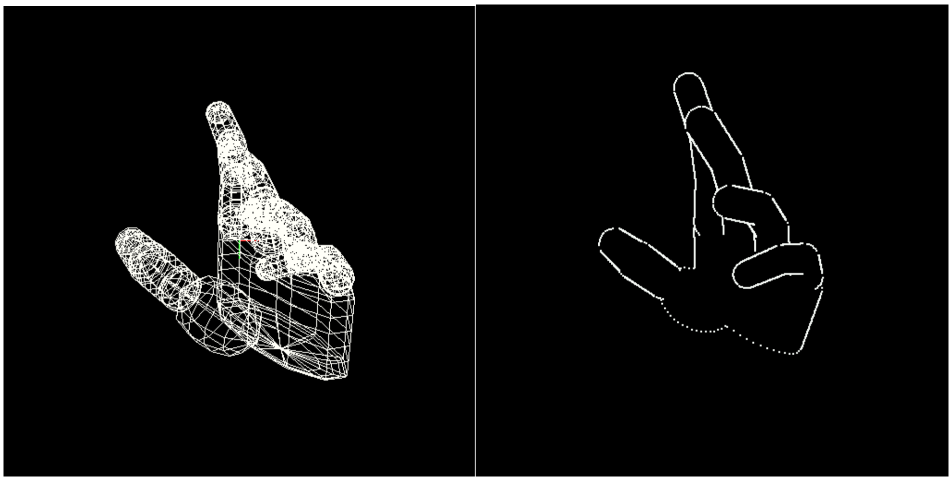

Figure 5: Occlusion handling is achieved by a simple ray tracing algorithm. The 3D model (left) and its generated contour (right) are shown.

Let $S_i = \{\mathbf{X}_i^1, \mathbf{X}_i^2, ..., \mathbf{X}_i^{N_i}\}$ be the set of visible (not occluded) points on the contour generator of $\mathbf{Q}_i$, and $\mathbf{C}_i$ be the projection of $\mathbf{Q}_i$. The image of each point $\mathbf{X}_i^j$ is denoted by $\mathbf{x}_i^j$. The vector $\mathbf{n}_i^j$ normal to $\mathbf{C}_i$ at $\mathbf{x}_i^j$ can be obtained as described in section 3.1.

For each point $\mathbf{x}_i^j$ we look for edges along the normal $\mathbf{n}_i^j$ (see for example [2]). For this the intensity values in the image are convolved with the derivative of a Gaussian kernel and an edge is assigned to the position $\mathbf{e}_i^j$ with the largest absolute value. The observation vector $\mathcal{Z}$ is constructed by stacking the inner products $\mathbf{n}_i^{j\mathrm{T}} \mathbf{e}_i^j$ into a single vector.

The predicted observation vector for the UKF is obtained as follows: By projecting the hand model corresponding to the state vector $\mathcal{X}_0(k+1|k)$ we obtain a reference contour, for which a list of image points $I_0 = \{\mathbf{x}_i^j\}$ 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 $\mathbf{n}_i^{j\mathrm{T}} \mathbf{x}_i^j$, where the points $\mathbf{x}_i^j$ are in the list I_l. The predicted observation can be found according to equation 6.

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 weighted average of the contour points, $\mathbf{e}$ is the corresponding edge in the image and $\mathbf{n}$ is the corresponding normal vector 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 and the edges in the image.

5 Experimental Results

Real data experiments were designed to test the proposed tracking algorithm. Two sequences of one hundred 360×288 greyscale images of a pointing hand and an open hand, respectively, were acquired. The parameters of the hand model were initially set manually

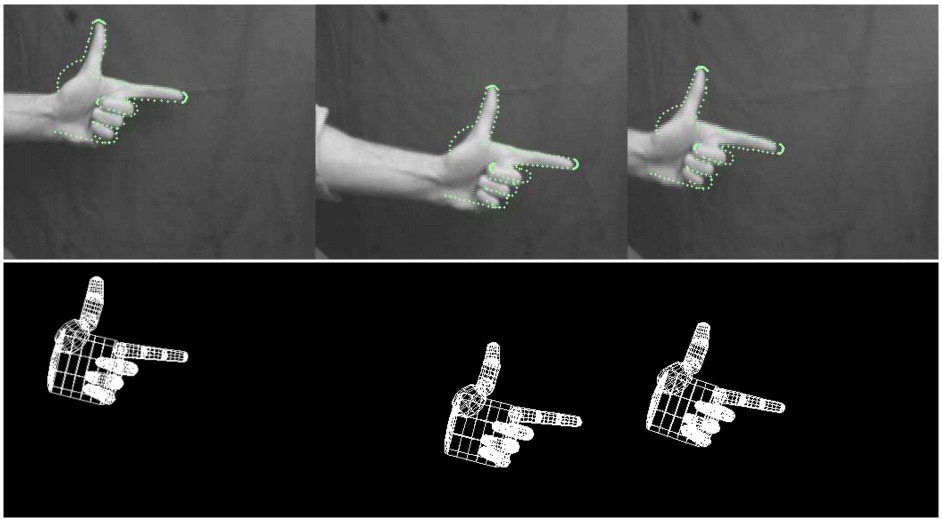

Figure 6: Results of the tracking algorithm: The images in the top row show the frames with the contours of the hand model superimposed. The images in the bottom row show the corresponding 3D pose. The hand is tracked continuously over the complete sequence of 100 frames.

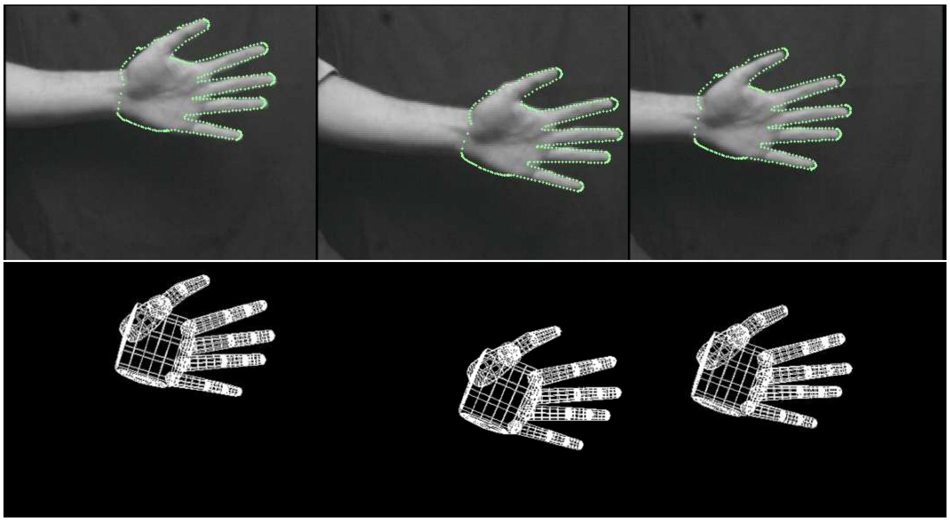

Figure 7: Tracking of an open hand. The top row shows the contours superimposed on the images, the bottom row shows the estimated 3D pose of the hand model.

to match the pose of the hand in the first frame of each sequence. Three DOF were given to the motion of the hand: translation in x- and y-directions and rotation about the z-axis. The dynamics of the hand was modelled using a second order process, i.e. using position, velocity and acceleration. The state vector was therefore

$$\mathcal{X} = \left[x,\; y,\; \theta_z,\; \dot{x},\; \dot{y},\; \dot{\theta}_z,\; \ddot{x},\; \ddot{y},\; \ddot{\theta}_z\right]^{\mathrm{T}}. \tag{10}$$

The results of the tracking algorithm are shown in figures 6 and 7. The images in the top row show the contours superimposed on selected frames of the sequence, the images in the bottom row show the corresponding 3D pose of the hand model. The tracking system operates at a rate of 3 frames per second on a Celeron 433MHz machine.

It can be seen that the system is accurate, succeeding in obtaining the correct pose of the hand. Although it is not operating in real-time yet, we expect to achieve this goal by code optimisation and using a faster machine. Also, a less complex model, tailored to the particular application, can be used for speeding up the processing.

6 Conclusion

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 parameters of the model. Results with real data demonstrate the efficiency of the proposed method.

6.1 Future Work

It is desirable to estimate as many pose and motion parameters as possible. Preliminary experiments suggest that, in order to increase the number of parameters, it is first necessary to refine the shape of the model to obtain a better agreement between its projection and the edges in image. This is currently done by hand, but the same framework presented here could be used to estimate the shape of the model from a set of still images, in an off-line stage to be carried out before the tracking.

The use of multiple cameras in order to reduce ambiguity is under development. A possible approach is to increase the length of the observation vector by stacking the measurements carried out on different images.

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

[2] 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.

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

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

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

[6] 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.

[7] 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, Vermont, USA, October 1996.

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

[9] 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, 2000.

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

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

[12] 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.

[13] 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.

[14] E. 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, Alberta, Canada, October 2000.

[15] Y. Wu and T. 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, September 1999.