Predictive Tracking in Vision-based Hand Pose Estimation using Unscented Kalman Filter and Multi-viewpoint Cameras

Albert Causo[1], Kentaro Takemura[1], Jun Takamatsu[1], Tsukasa Ogasawara[1], Etsuko Ueda[2] and Yoshio Matsumoto[3]
[1]*Nara Institute of Science and Technology*
[2]*Nara Sangyo University*
[3]*National Institute of Advanced Industrial Science and Technology*
Japan

1. Introduction

One of the major challenges in human-robot interaction is how to enable the use of unrestricted hand motion as a tool for communication. The direct use of hand as an input tool enables the user to connect to systems more naturally, allowing them to become an integral part of our daily lives. A vision-based approach, using cameras to capture data, supports non-contact and unrestricted movement of the hand. Nonetheless, the high degrees of freedom (DOF) of the hand is an essential issue to tackle in articulated hand motion tracking and pose estimation.

In this paper, we present our vision-based model-based approach, which uses multiple cameras and predictive filtering, to estimate the pose of the hand. We build on the research of Ueda *et al.*, whose work can separately estimate the global pose (wrist position and palm orientation) and the local pose (finger joint angles), but not simultaneously (Ueda et al., 2003). We address the problem through the use of a non-linear filter, Unscented Kalman Filter (UKF), to track the motion and simultaneously estimate the global and local poses of the hand.

The rest of the paper is organized as follows. Section 2 presents the related works and Section 3 discusses the UKF. Section 4 explains the hand pose estimation system and Section 5 details how we use the UKF for tracking and pose estimation. Experimental results and discussions are found in Section 6.

2. Related works

There are two main techniques to a vision-based hand pose estimation: appearance-based and model-based. Appearance-based approach uses two dimensional features such as silhouettes or colors, in order to compare the input image to a database of pose images and determine the hand pose (Athitsos & Sclaroff, 2003; Shimada et al., 2001). However, appearance-based approach is perspective-limited and usually gives solution only to a specific task problem (Pavlovic et al., 1997).

Source: Human-Robot Interaction, Book edited by: Daisuke Chugo,
ISBN 978-953-307-051-3, pp. 288, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

In model-based approach, the hand motion is modeled parametrically, giving a more precise and generic result. It tries to minimize the error between a predefined model of the hand and the observation data.
A full DOF hand has at least 27 parameters composed of 6 global (rotation and translation of the hand) and 21 local (finger joint angles). In a model based approach, the hand is represented as a model that describes its characteristics. The models can be classified in three groups: geometric model, statistical model, and physical based model. Geometric model uses various geometric primitives to represent the physical structure of the hand. Examples of geometric model include truncated quadrics (Stenger et al., 2001) and cardboard (Wu et al., 2001). On the other hand, statistical models define a hand shape as a variation of a mean model shape (Huang & Jeng, 2001). A physical based model considers the effect of various forces on the hand pose. Skeletal model covered with B-spline surface (Kuch & Huang, 1994), quadric surface (Ueda et al., 2003), or voxels (Causo et al., 2009) are examples of a physical based model.
Erol *et al.* classified hand pose estimation and motion tracking methods as either single-hypothesis tracking or multiple hypotheses tracking (Erol et al., 2007). In the former, the matching error between the model and the observation data is minimized by a best fit search. This technique includes optimization based methods like Gauss Newton (Rehg & Kanade, 1994), Genetic Algorithm (Lien & Huang, 1998), or Stochastic Gradient Descent (Bray et al., 2004) and physical-force models that uses force (Ueda et al., 2003), Unscented Kalman Filter (UKF) (Stenger et al., 2001; Causo et al., 2008) or Iterative Closest Point (ICP) algorithm (Delamarre & Faugeras, 1999).
Multiple hypotheses tracking, wherein multiple pose estimates are considered at each time frame, tries to address the issues of single hypothesis tracking such as the presence of singularity or spurious local minima. This includes tree-based search (Thayananthan et al., 2003), template matching (Shimada et al., 2001), and particle filtering (Lin et al., 2002).
Hand motion tracking is not a linear problem, but predictive tracking solutions for non-linear systems are available including Extended Kalman Filter (EKF), UKF, Gaussian sum filter, particle filter, and grid-based methods. Extended Kalman Filter is a straight-forward adaptation of the Kalman Filter to non-linear systems. Shimada *et al.* used EKF to estimate the pose of the hand and refine the 3D shape model even when using only a monocular camera and without any depth information (Shimada et al., 1998). A modified EKF through constraint fusion was used by Azoz *et al.* to localize and track an articulated arm (Azoz et al., 1998). Another extension of the Kalman Filter is the UKF (Julier & Uhlmann, 1997) which Stenger *et al.* used to track the motion of the hand modelled as truncated quadrics (Stenger et al., 2001).
Gumpp *et al.* used particle filtering (PF) to track the hand motion of the user in order to control a 20 DOF robot hand (Gumpp et al., 2006). Lin *et al.* parametrized the hand configuration space to be able to use a lower number of particles and consequently speeded up the computation (Lin et al., 2002). Thayananthan *et al.* and Stenger *et al.* both used grid-based filtering to search for the representative pose by traversing the tree nodes with high probabilities (Thayananthan et al., 2003; Stenger et al., 2004). They were able to do a fast search because the tree nodes' probabilities are updated during tracking and they skip the children of the nodes with small probabilities.

3. The Unscented Kalman Filter (UKF)

Unscented Kalman Filter belongs to the Kalman Filter (KF) family. It is a recursive estimator that uses information from the previous time frame in addition to the current observation measurement to make an estimate of the current state. Unlike the KF though, EKF and UKF are designed for non-linear systems. Extended Kalman Filter (EKF) is the more commonly used technique between the two. However, it requires the computation of Jacobian matrices, which is non-trivial in most cases.

In contrast, UKF uses unscented transformation method, which calculates the statistics of a random variable that undergoes non-linear transformation (Julier & Uhlmann, 1997). It is accurate up to the second order and requires fewer samples compared to a similar particle filter. Xiong *et al.* studied the performance of UKF under certain conditions and showed that it performs robustly in general tracking applications of non-linear systems (Xiong et al., 2006).

Figure 1 shows the overview of the UKF process, which is composed of two main parts, similar to the KF. First is the time-update, wherein the initial state estimate is computed by selecting sigma points and solving for its mean and covariance. The observation is also propagated in this step and its mean and covariance are also calculated. The second part is the measurement update. The Kalman gain and cross-covariance of the propagated state and the propagated observation are calculated and used to update the state and its covariance. The computational details are discussed next.

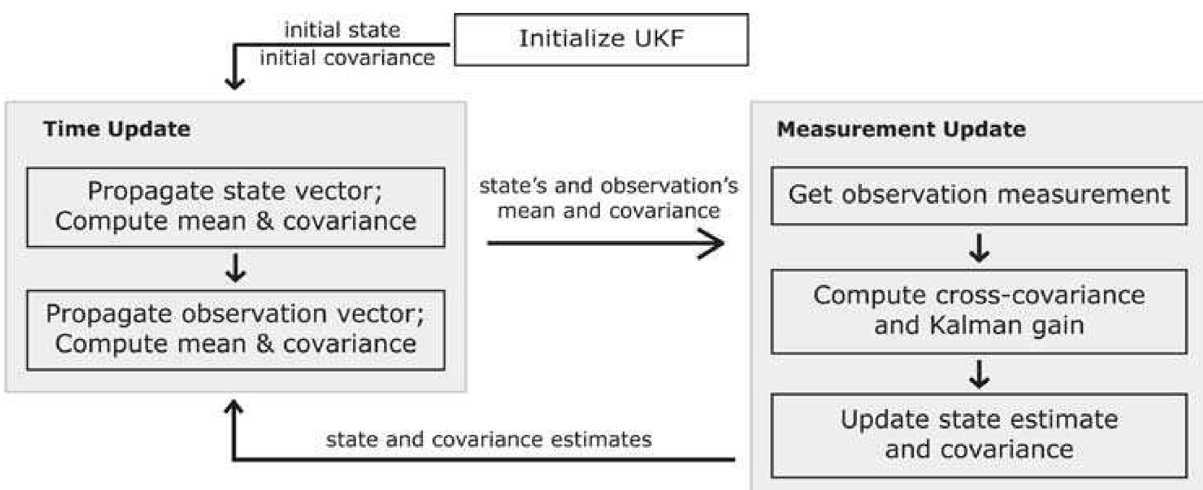


Fig. 1. The Unscented Kalman Filter (UKF) process. This is very similar to the Kalman Filter, except that the initial state estimate (under the Time Update box) is obtained from the sigma (particle) propagation.

For a given tracking problem, consider the state dynamics,

$$\mathbf{X_k} = f(\mathbf{X_{k-1}}, \mathbf{R_k}) \quad (1)$$

where:
f is the system dynamics,
$\mathbf{X_k}$ is the state vector of size n at time k, and
$\mathbf{R_k}$ is the state noise covariance.

UKF makes an initial estimate of the state vector, by selecting sigma points through Equation 2:

$$\mathbf{X_k^i} = \begin{cases} \mathbf{X_k^0} = \bar{\mathbf{X}}_{\mathbf{k-1}} & \\ \mathbf{X_k^i} = \bar{\mathbf{X}}_{\mathbf{k-1}} - (\phi)_i & i = 1,\ldots,n \\ \mathbf{X_k^i} = \bar{\mathbf{X}}_{\mathbf{k-1}} + (\phi)_{i-n} & i = n+1,\ldots,2n \end{cases} \quad (2)$$

where:
ϕ is the i_{th} column of $\sqrt{(n+\lambda)\mathbf{P_{k-1}}}$,
$\mathbf{P_{k-1}}$ is the covariance estimate from the previous iteration,
$\bar{\mathbf{X}}_{\mathbf{k-1}}$ is the state estimate from the previous iteration, and
λ is the scaling parameter.
$2n$+1 sigma points are selected to approximate the posterior mean and covariance of the state vector. The selection of the sigma points is deterministic and is set by adjusting the scaling parameter λ:

$$\lambda = \alpha^2(n+\kappa) - n \tag{3}$$

where:
α determines the distribution of the points around the mean and is set to a small positive value, and
κ is a secondary parameter set to 0 or $3 - n$.
Equation 1 is applied to $\mathbf{X_k^i}$ to obtain the propagated state vector $\hat{\mathbf{X}}_\mathbf{k}^\mathbf{i}$:

$$\hat{\mathbf{X}}_\mathbf{k}^\mathbf{i} = f(\mathbf{X_k^i}, \mathbf{R_k^i}) \tag{4}$$

The mean $\bar{\hat{\mathbf{X}}}_\mathbf{k}$ and the covariance $\hat{\mathbf{P}}_\mathbf{k}$ of the propagated sigma points are computed:

$$\bar{\hat{\mathbf{X}}}_\mathbf{k} = \sum_{i=0}^{2n} W_i \hat{\mathbf{X}}_\mathbf{k}^\mathbf{i} \tag{5}$$

$$\hat{\mathbf{P}}_\mathbf{k} = \sum_{i=0}^{2n} W_i \left[\hat{\mathbf{X}}_\mathbf{k}^\mathbf{i} - \bar{\hat{\mathbf{X}}}_\mathbf{k}\right]\left[\hat{\mathbf{X}}_\mathbf{k}^\mathbf{i} - \bar{\hat{\mathbf{X}}}_\mathbf{k}\right]^T \tag{6}$$

The weight W_i is computed according to the following:

$$\begin{aligned} &W_0 = \{\lambda/(n+\lambda)\} + (1-\alpha^2+\beta) \\ &W_i = 1/\{2(n+\lambda)\} \qquad\qquad i = 1,2,\ldots,2n. \end{aligned} \tag{7}$$

β is used to include information about the distribution of the state variable. It is found to be optimal at $\beta = 2$ for a Gaussian distribution.
Likewise, the observation vector is propagated using the propagated sigma points:

$$\hat{\mathbf{Y}}_\mathbf{k} = h(\hat{\mathbf{X}}_\mathbf{k}, \mathbf{S_k}) \longrightarrow \hat{\mathbf{Y}}_\mathbf{k} \approx \left\{\hat{\mathbf{Y}}_\mathbf{k}^\mathbf{0}, \hat{\mathbf{Y}}_\mathbf{k}^\mathbf{1}, \hat{\mathbf{Y}}_\mathbf{k}^\mathbf{2}, \ldots, \hat{\mathbf{Y}}_\mathbf{k}^\mathbf{2N}\right\} \tag{8}$$

where:
h describes the nonlinear observation function,
$\hat{\mathbf{Y}}_\mathbf{k}$ is the propagated observation vector,
$\hat{\mathbf{X}}_\mathbf{k}$ is the propagated state vector, and
$\mathbf{S_k}$ is the measurement noise covariance.
Then $\bar{\hat{\mathbf{Y}}}_\mathbf{k}$, the mean of the propagated observation vector, and its covariance $\hat{\mathbf{P}}_{\mathbf{yy_k}}$ are calculated using the same weights defined in Equation 7:

$$\bar{\hat{\mathbf{Y}}}_\mathbf{k} = \sum_{i=0}^{2n} W_i \hat{\mathbf{Y}}_\mathbf{k}^\mathbf{i} \tag{9}$$

$$\hat{\mathbf{P}}_{\mathbf{yy_k}} = \sum_{i=0}^{2n} W_i \left[\hat{\mathbf{Y}}_{\mathbf{k}}^{\mathbf{i}} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}}\right] \left[\hat{\mathbf{Y}}_{\mathbf{k}}^{\mathbf{i}} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}}\right]^T \tag{10}$$

Up until this point is the time update block of Fig. 1.

The succeeding steps are part of the measurement update. The observation vector $\mathbf{Y_k}$ is obtained from sensor measurements. Then the cross-covariance of the state and the observation vectors, $\hat{\mathbf{P}}_{\mathbf{xy_k}}$, is calculated in order to derive the Kalman gain $\mathbf{K_k}$.

$$\mathbf{P}_{\mathbf{xy_k}} = \sum_{i=0}^{2n} W_i \left[\hat{\mathbf{X}}_{\mathbf{k}}^{\mathbf{i}} - \bar{\hat{\mathbf{X}}}_{\mathbf{k}}\right] \left[\hat{\mathbf{Y}}_{\mathbf{k}}^{\mathbf{i}} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}}\right]^T \tag{11}$$

$$\mathbf{K_k} = \mathbf{P}_{\mathbf{xy_k}} \hat{\mathbf{P}}_{\mathbf{yy_k}}^{-1} \tag{12}$$

Finally, the state and covariance estimates are updated:

$$\bar{\mathbf{X}}_{\mathbf{k}} = \bar{\hat{\mathbf{X}}}_{\mathbf{k}} + \mathbf{K_k}(\mathbf{Y_k} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}}) \tag{13}$$

$$\mathbf{P_k} = \hat{\mathbf{P}}_{\mathbf{k}} - \mathbf{K_k} \hat{\mathbf{P}}_{\mathbf{yy_k}} \mathbf{K_k}^T \tag{14}$$

where $\bar{\mathbf{X}}_{\mathbf{k}}$ is the state estimate, and $\mathbf{P_k}$ is its covariance at time k. These values become the input to the next iteration, i.e., $\bar{\mathbf{X}}_{\mathbf{k}}$ becomes $\bar{\mathbf{X}}_{\mathbf{k-1}}$ and $\mathbf{P_k}$ becomes $\mathbf{P_{k-1}}$. Then the whole process repeats again.

Upon initialization of the filter, $\bar{\mathbf{X}}_{\mathbf{k-1}}$ and $\mathbf{P_{k-1}}$ in Equations 1 and 2 are set to some initial values and become $\bar{\mathbf{X}}_{\mathbf{0}}$ and $\mathbf{P_0}$, respectively. The scaling parameter values are adjusted heuristically.

For further discussion and details on the implementation of UKF, consult Julier & Uhlmann (Julier & Uhlmann, 1997) and Wan & Van der Merwe (Wan & van der Merwe, 2000).

4. Hand pose estimation using multi-viewpoint cameras

The vision-based hand pose estimation system takes its input from multiple cameras, which are positioned so that they see with the least amount of occlusion (Fig. 2).

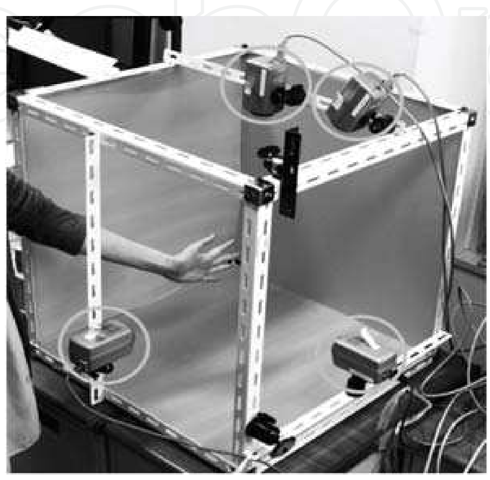


Fig. 2. Multiple viewpoint camera system.

The system is model-based as it uses a skeletal model of the hand (Fig.3). It represents the hand as a set of five manipulators with a common base point at the wrist. Each finger is a manipulator with several links and joints. The metacarpophalangeal (MCP) and carpometacarpal (CMC) joints have two DOFs each to account for flexion-expansion and abduction-adduction motions. The rest of the joints has one DOF each. The thumb has a special configuration. It has only 4 joints and 4 links, for a total of 5 DOFs. The wrist, which accounts for the global pose, has six DOFs for translation and rotation. The model has a skin composed of quadric surfaces, which will be referred to as surface model throughout this paper. The surface model represents the underlying skeletal configuration of the links and the joints. In summary, the skeletal model has 19 joints, 31 DOFs, 24 links and a total of 744 surface quadrics.

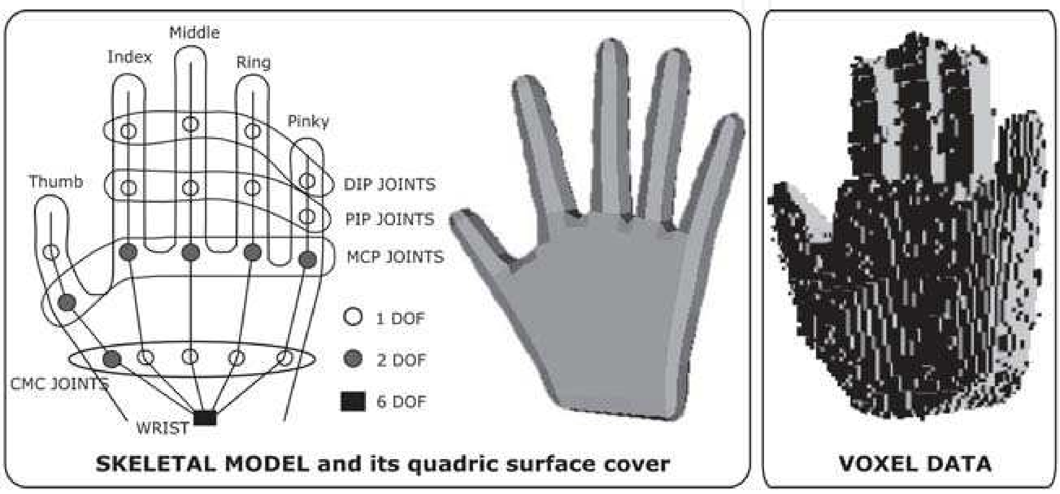


Fig. 3. The hand models. The skeletal model is covered with quadric surfaces. The voxel data is derived from the silhouettes of the input images.

The observation data of the system is the images from the cameras converted to voxel data by shape-from-silhouette technique (Szeliski, 1993). In other words, the voxel data represents the current hand pose as seen by the cameras.
Ueda *et al.,* minimized the error between the voxel data and the skeletal model using virtual force (Ueda et al., 2003). Although their technique has the advantages of being simple and fast, it cannot estimate the global and local poses simultaneously. It also has difficulty in recovering from erroneous estimation.
In our proposed approach, we estimated the global and the local parameters simultaneously using UKF. Instead of being limited to finger movements, it will also allow the palm's rotation and the wrist's translation to be estimated. This enables the hand pose estimation system to accept a more dynamic hand motion as input.

5. UKF in hand pose estimation

We present in this section how we used UKF to estimate the hand pose. We chose UKF over EKF or particle filter because of its simple implementation, fewer number of particles needed, and accuracy of up to the second order (Julier & Uhlmann, 1997). Moreover, for our system, the relationship between the observation data and the hand pose is non-linear.
Figure 4 illustrates the process of using UKF to estimate the skeletal pose of the hand using the voxel data as input. The step by step explanation is as follows:

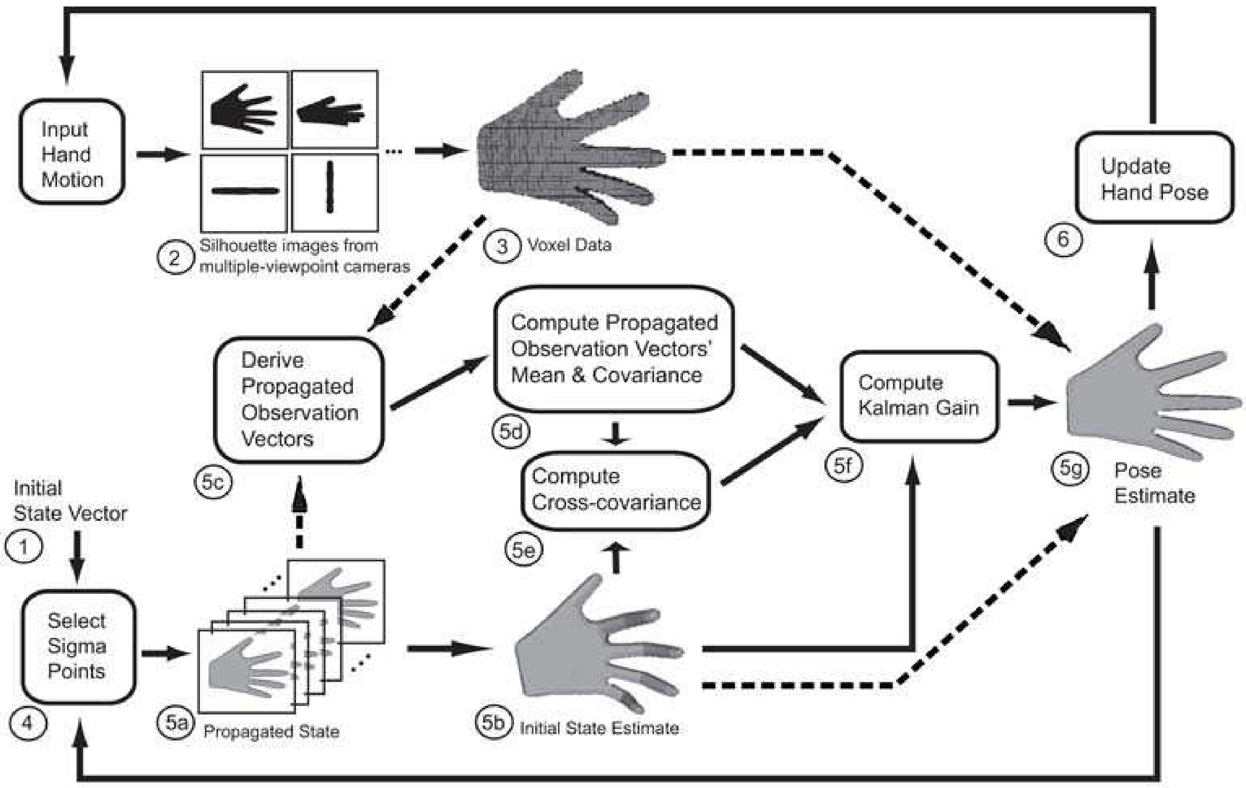


Fig. 4. Details of the proposed method. Dashed lines indicate comparison of models in order to obtain error measurements.

1. The state vector is set to $\bar{\mathbf{X}}_{k-1}$ while the state covariance is set to $\mathbf{P}_{k-1}$. During initialization, the state is set to zero ($\mathbf{X}_0$) while the state covariance is set to some value ($\mathbf{P}_0$).
2. The color image inputs from the multiple cameras are converted to silhouettes.
3. Using shape-from-silhouette approach, the silhouette images are converted to voxel data.
4. Sigma points are selected using $\bar{\mathbf{X}}_{k-1}$ and $\mathbf{P}_{k-1}$.
5. Hand pose is estimated using UKF:
   a. Apply Equation 1, the state dynamics equation, to the sigma points $\mathbf{X}_k^i$. This gives the propagated state vectors $\hat{\mathbf{X}}_k^i$, illustrated as variations of hand poses.
   b. Calculate the mean value of the propagated state vectors $\bar{\hat{\mathbf{X}}}_k$ and its covariance $\hat{\mathbf{P}}_k$. The $\bar{\hat{\mathbf{X}}}_k$ is the filter's initial state estimate.
   c. Propagate the observation vector $\hat{\mathbf{Y}}_k^i$ by computing the error between the propagated state $\hat{\mathbf{X}}_k^i$ and the voxel data.
   d. Calculate $\bar{\hat{\mathbf{Y}}}_k$, the mean value of the propagated observation vectors, and its covariance $\hat{\mathbf{P}}_{yy_k}$.
   e. Calculate the cross covariance $\hat{\mathbf{P}}_{xy_k}$.
   f. Compute Kalman gain $\mathbf{K}_k$ using Equation 12.
   g. Compute state estimate $\bar{\mathbf{X}}_k$ and its covariance $\mathbf{P}_k$. The hand pose estimate is defined by $\bar{\mathbf{X}}_k$.
6. Update the hand pose. $\bar{\mathbf{X}}_k$ and $\mathbf{P}_k$ become the next iteration's $\bar{\mathbf{X}}_{k-1}$ and $\mathbf{P}_{k-1}$, respectively.
7. The process repeats from Step 2 for the next iteration.

5.1 The state dynamics and composition of the state vector

A key factor in using a predictive filter is using the correct state dynamics. For the hand pose estimation, we used a second order dynamics or constant acceleration model, which describes the change in the state vector over time. It also captures the nature of the hand motion better than a constant velocity model does.

In Equation (15), $\mathbf{X_k}$ is the state vector at time $\mathbf{k}$, Δt is the time interval between frames, $\mathbf{V_k}$ is the noise covariance of the state vector, and $\mathbf{I}$ is the identity matrix. The state noise covariance accounts for all the disturbances not accounted for by the dynamics; it was determined heuristically in the experiments. The uncertainties of the dynamics are modeled to be independent for the position, velocity, and acceleration components.

$$\mathbf{X_k} = \begin{bmatrix} \mathbf{I} & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & \mathbf{I} & \Delta t \\ 0 & 0 & \mathbf{I} \end{bmatrix} [\mathbf{X_{k-1}}] + [\mathbf{V_k}] \tag{15}$$

The state vector $\mathbf{X}$ is composed of both global (rotation and translation) and local (finger joint angles) pose parameters and their respective first and second order derivatives (velocity and acceleration). In Equation 16, θ_n is either a global or local parameter, $\dot{\theta}_n$ is its velocity, and $\ddot{\theta}_n$ its acceleration.

$$\mathbf{X} = \left[\theta_1, \theta_2, \ldots, \theta_n, \dot{\theta}_1, \dot{\theta}_2, \ldots, \dot{\theta}_n, \ddot{\theta}_1, \ddot{\theta}_2, \ldots, \ddot{\theta}_n\right]^T \tag{16}$$

5.2 The observation function and composition of the observation vector

The observation function of our system is the non-linear process of obtaining the observation vector given a hand pose configuration. A given state vector, $\mathbf{X}$, specifies a combination of finger joint angles and wrist data that can be represented by the hand model as a particular pose. After converting the state vector to a hand model's pose, we obtain the error between the voxel data (Step 3 of Fig. 4) and the hand model. We designed the observation vector to contain the error measurement between the voxel data (the observed hand pose) and the hand model (the surface model).

The observation vector is composed of the geometric distance measured between the voxel data and the hand surface model. The distance is computed by checking whether each quadric Q_i of the surface model is located inside or outside of the voxel data. If it is outside, the Manhattan distance d_i between the center of the quadric and the nearest voxel is measured. If it is inside, d_i is set to zero. It is repeated for all the quadrics of the surface model. Figure 5 illustrates the process.

The distance values are stacked to form the observation vector $\mathbf{Y}$:

$$\mathbf{Y} = [\ldots, d_{i-1}, d_i, d_{i+1}, \ldots]^T \tag{17}$$

In order to lessen the computation time, the size of the observation vector can be decreased. In our experiments, instead of obtaining distance measurements from all quadrics (744 in total), we only sampled 140 quadrics. It made the computation time more manageable.

Additionally, at every time step, we always take distance measurements from the same set of quadrics. We are able to follow the motion of the finger links from one time frame to

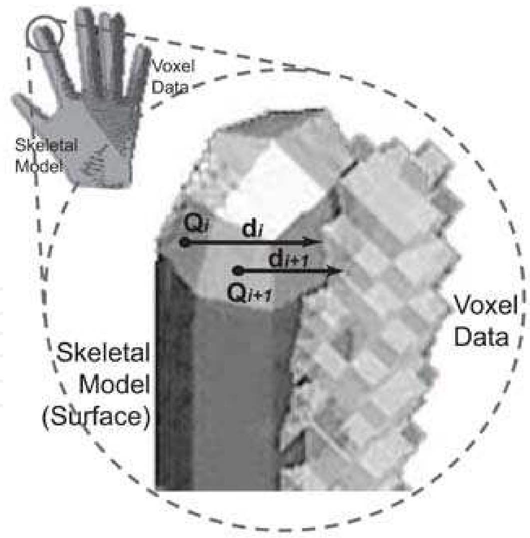


Fig. 5. The calculation of geometric error between surface model and voxel data.

another by keeping track of the changes in the distance values of the selected quadrics. That way, we can say that a sense of direction is encoded in the observation vector. Thus, the observation vector contains the magnitude of change of the finger link's motion, as well as its general sense of direction.

To interpret the observation vector, a zero error between the model and the observation (i.e., $\mathbf{Y} = 0$) implies that the hand model is completely inside the voxel data. Some value in the observation vector indicates that the fingers have moved in certain direction.

Finally, in Equation 13, $\mathbf{Y_k}$ is always set to zero, based on two things. First, comparing the voxel data to itself and computing distance measurements will just yield zero. Second, from the perspective of the filter, $\mathbf{Y_k = 0}$ can be interpreted to mean that the observation (sensor) measurement is not completely reliable and that we have to correct it through the $\mathbf{K_k}(\mathbf{Y_k} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}})$ term of Equation 13.

5.3 Initialization and filter tuning

Filter fine tuning and proper parameter initialization are important tasks when incorporating a predictive filter to a motion tracking solution. As mentioned above, the state vector is set to zero value ($\mathbf{X_0}$) at the initial step. The zero values assigned to the state parameters mean that the hand model is at its initial pose. The hand is said to be at initial pose when the palm is flat open and the fingers are extending away from the palm. Likewise, the state covariance matrix's diagonal is set to some value ($\mathbf{P_0}$).

The fine-tuning parameters of λ (see Equation 3) were also determined heuristically. For example, α was set to a small value between 1 and $1x10^{-4}$, κ was set to $(3 - n)$, and β was set to 2. Likewise, selection of the noise covariances $\mathbf{R_k}$ and $\mathbf{S_k}$ is also critical.

6. Experimental results and discussion

For all the experiments we have done, we used eight cameras in order to get finer voxel data. Additionally, the voxel resolution used was 2×2×2[mm] per octant or voxel unit. We also estimated a total of 15 hand pose parameters: 3 global and 12 local. The global parameters are roll, pitch and yaw. The local parameters are the 2 DOFs of the MCP and 1 DOF of the PIP. The following constraint gives the value of the DIP joint angle relative to the PIP:

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP}. \tag{18}$$

The proposed method was tested on several hand motions. Various hand motion data were obtained using a dataglove. These data, which we considered as the ground truth for all our experiments, were then used to create virtual versions of the different hand motions. These virtual motions were used as input to the pose estimation system and tracked. Proper initialization of the hand model and the voxel data (i.e., they must overlap initially) is necessary for filter convergence. Fortunately, the use of simulated motion eliminated this issue.

Figures 6, 7, and 8 show a hand motion that has been tracked successfully. The motion (Motion A) is that of a hand whose wrist is rotating and twisting, while the fingers (with the exception of the thumb) are simultaneously closing slowly. This motion involves three global and 12 local parameters. The wrist's roll, pitch, and yaw (see Fig. 6) and the four fingers' PIP (1 DOF) and MCP (2 DOFs) were estimated with good accuracy. Figure 7 shows only the MCP's expansion-flexion data (left column) and the PIP (right column).

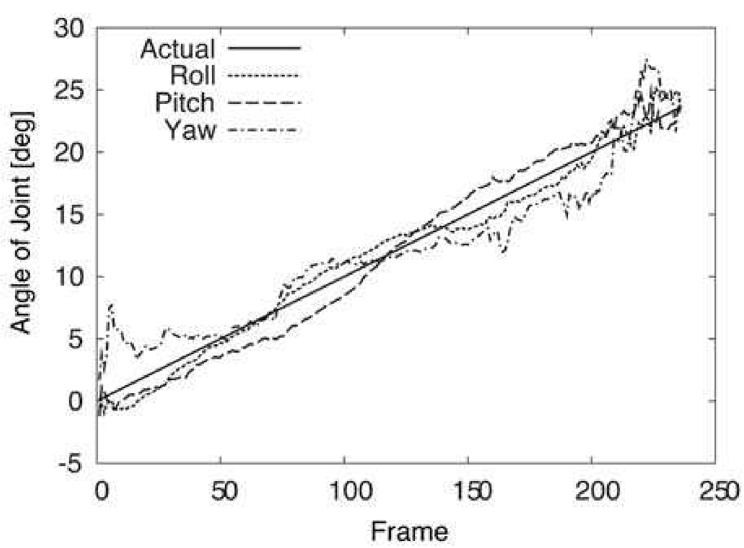


Fig. 6. Global estimation result when the fingers are closing simultaneously. Solid line is the ground truth values (actual); broken lines are the estimate (roll, pitch, yaw).

For Fig. 6 and Fig. 7, the black solid line is the ground truth value while the dotted and dashed lines are the estimate values. For all the fingers, the filter initially shows estimation errors by as much as 10 degrees, although it eventually converges to the desired value. The filter also gets lost but tries to get back on track. This can be seen as a noisy estimation in the pinky's MCP joint estimation (Fig.7 left side, top graph). We had to implement range constraints on the finger motion to ensure that awkward poses, for example fingers bending backward too much, do not happen. This can be seen as a plateau on the pinky's PIP estimation graph (Fig.7 right side, top graph).

Snapshots of the motion described above are shown in Fig.8. The top row is the virtually-generated motion and the bottom row is the result of the pose estimation. The numbers above each column of image correspond to the points in Fig. 7 when the images were taken. The local motion manifests in the images as the closing and opening of the fingers, while the global motion shows as the twisting of the wrist and palm.

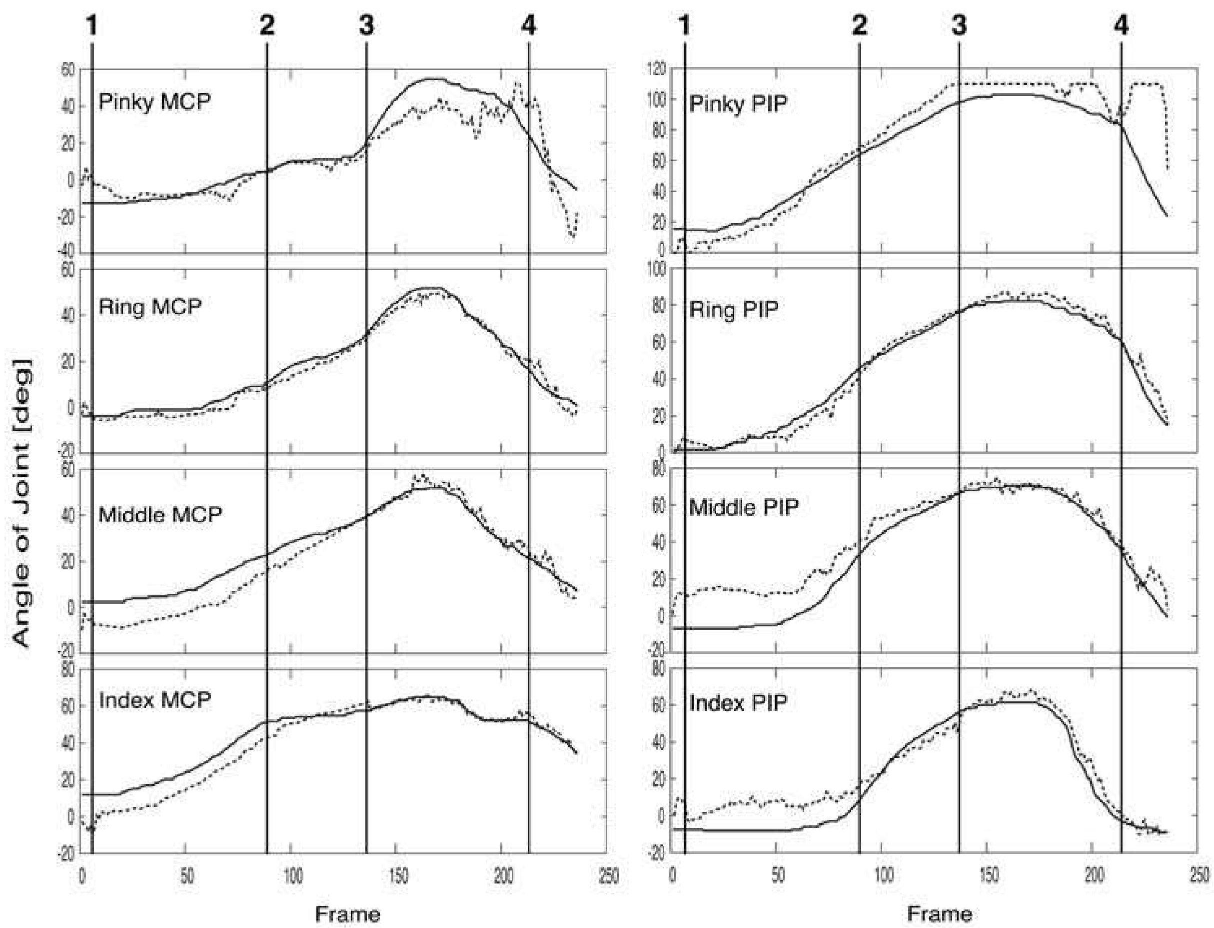


Fig. 7. MCP and PIP estimation results for fingers closing simultaneously while the wrist is rotating. Solid lines are the ground truth values; the dotted lines are the pose estimation result. The numbered vertical lines show when the snapshots in Fig. 8 were taken.

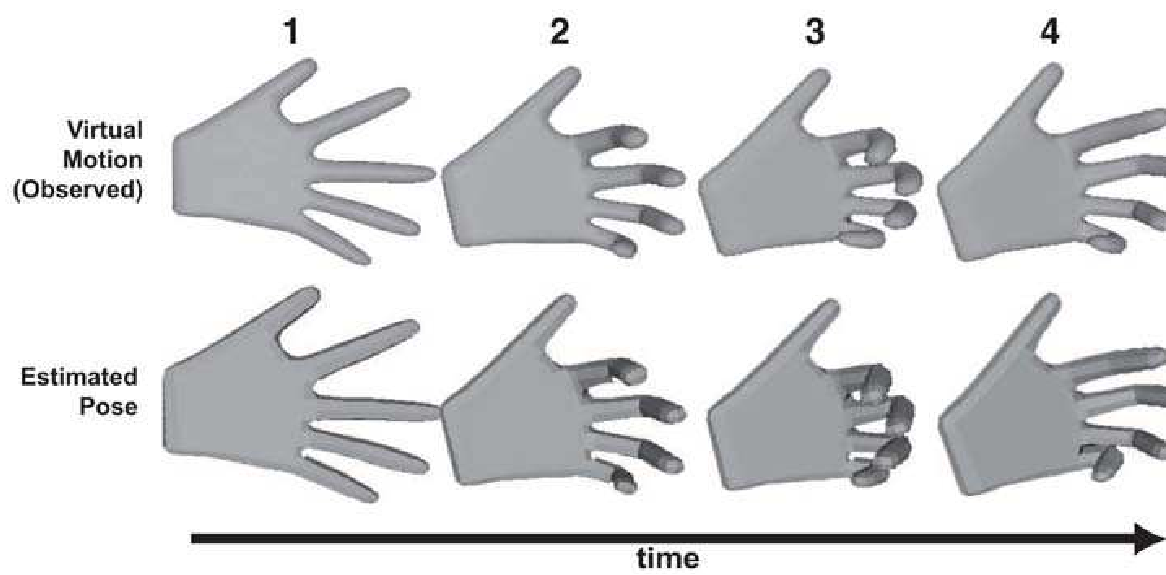


Fig. 8. Snapshots of estimation result. The numbers above each image column correspond to the points in Fig. 7 when the snapshots were taken. The motion is for a rotating wrist while the fingers are closing simultaneously.

Two more motions were tested to demonstrate the flexibility of the system. Snapshots of the estimation results are shown in Fig.9 (Motion B) and Fig.10 (Motion C). For both motions, the wrist is rotating and twisting due to roll, pitch, and yaw motions. In Fig.9, the hand is moving two fingers at a time. In Fig.10, the fingers successively bending towards the palm one by one, starting from the pinky toward the index finger and then opening in the reverse order.

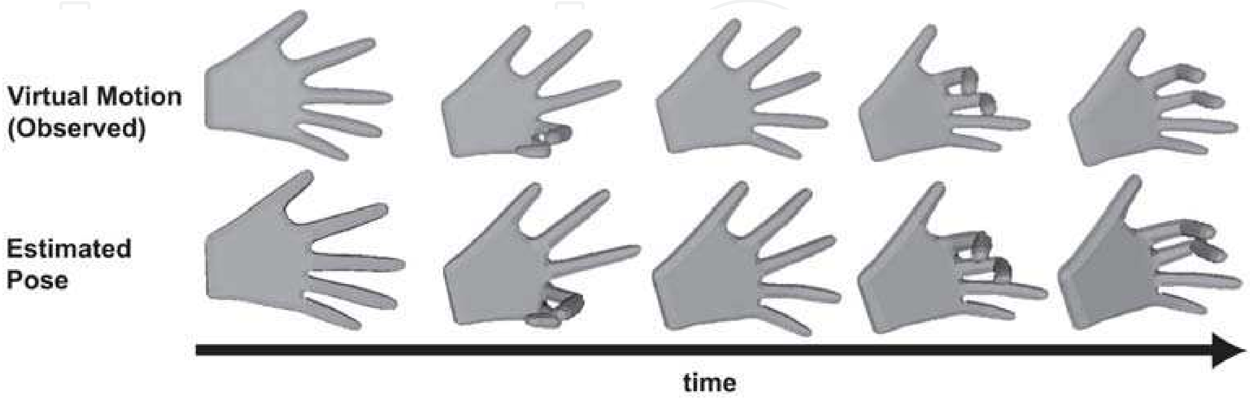


Fig. 9. Snapshots of the observed hand motion and their corresponding estimated hand poses. The motion is that of a hand rotating while the fingers are closing two at a time.

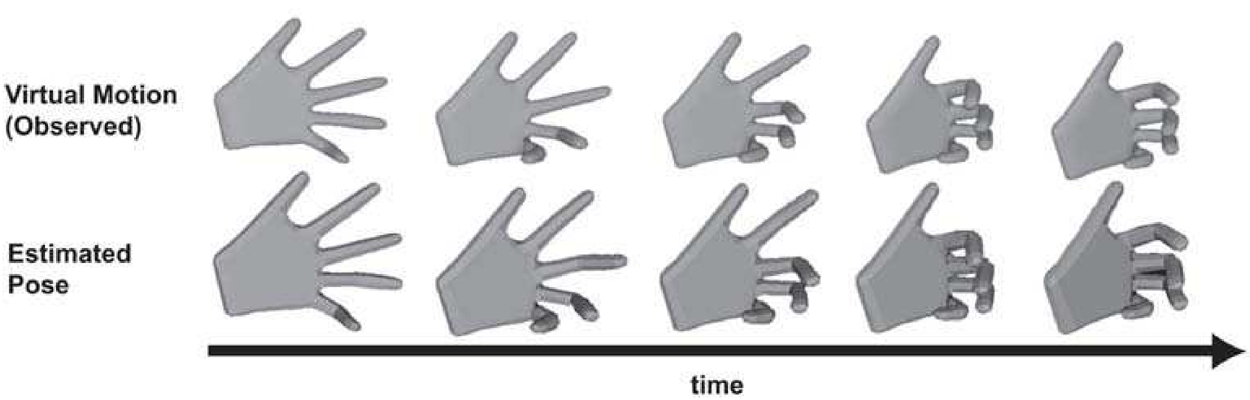


Fig. 10. Snapshots of the observed hand motion and their corresponding estimated hand poses. The motion is that of a hand rotating while the fingers are closing one at a time starting from the pinky going to the index.

To compare the accuracy of our estimation results, Fig.11 shows the average of absolute errors for all the joints estimated. The absolute errors range from 0.20 to 3.40 degrees per joint for every iteration. However, the actual change of angle per iteration of any joint, based on the ground truth data, is less than 1 degree only. We can interpret this range of absolute error as an indication of the filter's effort to converge to the ground truth value. Physically speaking, even a three degree motion of a joint is not easy to perceive due to the presence of the muscle and the skin covering the finger bones. Thus, the converging behavior is noticeable in the graphs of Fig.6 and 7 but imperceptible in the snapshots of Fig.8.

Furthermore, we compared our results with the original model-fitting approach's in (Ueda et al., 2003); a predictive filtering versus model-fitting comparison. Fig.12 establishes the robustness of using the UKF against using the virtual force based model-fitting. The figure shows the estimation result of both methods for the Index PIP joint. Both methods try to converge to the true value, but a closer look shows that the model-fitting has more difficulty in doing so. Between frames 100 to 200, the Index PIP is expanding and flexing (i.e., bending and stretching), and the UKF is able to track this movement quite well. The filter's estimation results fluctuate as it tries to converge to the true value yet manages to recover from the fluctuations. On the other hand, it takes some time for the model-fitting approach to recover from its over-estimation and overshoots its estimates. In short, the proposed method showed better error recovery than the model-fitting method.

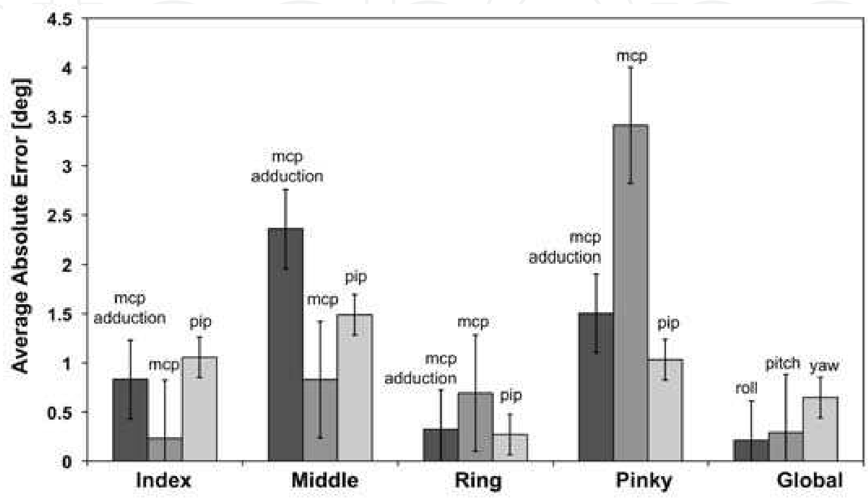


Fig. 11. Average absolute error of each DOF. The motion is that of a hand rotating while the fingers are closing simultaneously.

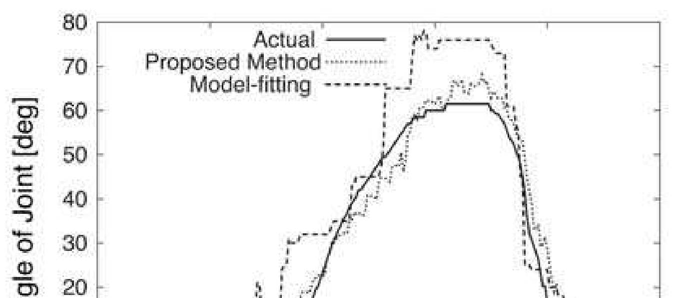


Fig. 12. Comparison of Index PIP estimation results of the original model fitting approach and the proposed method.

There are several issues to address when implementing a predictive filter in hand pose estimation. First is the composition of the state and observation vectors, more importantly, its size. In our experiments, the dimension of the state vector was 45: the 15 hand parameters (3 global + 12 local) and their respective first and second order derivatives while the observation vector's was 140. The size of the observation vector was adjusted until the optimum size is attained. A trade off between size and computation speed is needed here. If the observation vector is too small, there would not be enough information for the filter to process, but too big a size and the computation time increases considerably.

For the state vector, the size is largely determined by the dynamics model of the system. Since we chose a constant acceleration dynamics, we had to incorporate the first and second derivatives of the state variables in the state vector. Fortunately, inclusion of known hand constraints can help lessen the dimensions of the state vector. For example, we used the coupling constraint between the PIP and the DIP (Equation 18), thus shrinking the state vector by nine parameters.
The second important consideration in UKF is the noise covariance of the state (Equation 1) and the observation (Equation 8) vectors. The stability and convergence of the filter depend on the accurate choice of covariances (Xiong et al., 2006). In our case, all the covariances, listed in Table 1 were determined heuristically. The same noise covariances were applied to all the motions discussed here. Likewise, noise covariances for the observation measurement were also determined heuristically. We used different noise covariances for the different motions (see Table 2).

State Parameter	Covariance Value
θ	0.1
$\dot{\theta}$	0.01
$\ddot{\theta}$	0.001

Table 1. Covariance values used for the state vector.

Hand Motion	Covariance Value
Motion A (Fig.8)	0.001
Motion B (Fig.9)	0.1
Motion C (Fig.10)	0.1

Table 2. Covariance values used for the observation vector.

Lastly, the filter's computation speed is another important consideration. As mentioned before, for the UKF, computation speed depends largely on the size of the state vector and the observation vector. Minimizing either or both can result in faster computations, which in turn leads to a more stable and accurate filtering. Modifications to UKF, or its equivalent methods, to further lessen the number of sigma particles from $2n + 1$ have already been reported in the literature. For example, Julier *et al.* used only $n + 1$ number of particles (Julier & Uhlmann, 2002). La Viola compared the performance of EKF and UKF in head tracking and found that using quaternions to encode the joint angles resulted to better estimation, even by just using EKF (La Viola, 1996).
In our experiments, the computation speed of the filter is around 0.87 seconds for every iteration or roughly 1Hz. However, the usual frame capture speed of cameras is around 30Hz. Thus, there is a need to speed up the proposed method.

7. Conclusion and future work

We introduced a predictive filter, Unscented Kalman Filter, to a vision-based model-based system in order to estimate the global and local poses of the hand simultaneously. The UKF minimizes error between the hand model and the voxel data and computes the initial pose estimate by propagating $2n + 1$ sigma particles. We were able to show estimation results for up to 3 global and 12 local pose parameters in different motions and demonstrate better error recovery than a previous pose estimation technique. The results presented in this paper used virtually generated motion obtained from actual hand motion to verify our method.

Our future work includes the implementation of the proposed method in a real camera system and the use of a calibrated hand model. Moreover, an adaptation of the original UKF technique to the hand dynamics is necessary in order to speed up the computation and improve the accuracy and over-all stability of the filtering process.

8. References

Athitsos, V. & Sclaroff, S.J. (2003). Estimating 3D Hand Pose from a Cluttered Image, *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 432-442, Madison,WI, USA, Jun 2003

Azoz, Y.; Devi, L. & Sharma, R. (1998). Tracking Hand Dynamics in Unconstrained Environments, *Proc. of Third Int. Conf. on Automatic Face & Gesture Recognition*, pp. 274-279, Nara, Japan, Apr 1998

Bray, M.; Koller-Meir, E., Müller, P., Gool, L.V. & Schraudolph, N.N. (2004). 3D Hand Tracking by Rapid Stochastic Gradient Descent using a Skinning Model, *Proc. of the First European Conf. on Visual Media Production*, pp. 59-68, London, 2004

Causo, A.; Ueda, E., Kurita, Y., Matsumoto, Y. & Ogasawara, T. (2008). Model-based Hand Pose Estimation using Multiple View-point Images and Unscented Kalman Filter, *Proc. of the Seventeenth International Symposium Robot and Human Interactive Communication (RO-MAN 2008)*, pp. 291-296, Munich, Germany, Aug 2008

Causo, A.; Matsuo,M., Ueda, E., Takemura, K., Matsumoto, Y., Takamatsu, J. & Ogasawara, T. (2009). Hand Pose Estimation using Voxel-based Individualized Hand Model. *Proc. of the 2009 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*. pp. 451-456. Singapore, Jul 2009

Delamarre, Q. & Faugeras, O. (1999). 3D Articulated Models and Multi-view Tracking with Silhouettes, *Proc. of the Seventh IEEE Int. Conf. on Computer Vision*, Vol. 99, pp. 716-721, Kerkyra, Greece, Sep 1999

Erol, A.; Bebis, G., Nicolescu M., Boyle, R.D.& Twombly, X. (2007). Vision-based Hand Motion Estimation: A Review, *Comput. Vis. Image Underst.*, Vol. 108, No. 1-2 (Oct 2007) pages (52-73)

Gumpp, T.; Azad, P., Welke, K., Oztop, E., Dillmann, R. & Cheng, G. (2006). Unconstrained Real-time Markerless Hand Tracking for Humanoid Interaction, *Proc. of Sixth IEEE/RAS Int. Conf. on Humanoid Robots,* pp. 88-93, Genova, Italy, Dec 2006

Huang, C.L. & Jeng, S.H. (2001). A Model-based Hand Gesture Recognition System, *Machine Vision and Applications*, Vol. 12, No. 5 (Mar 2001) pages 243-258

Julier, S.J. & Uhlmann, J.K. (1997). A New Extension of the Kalman Filter to Nonlinear Systems, *Proc. of Conf. on Signal Processing, Sensor Fusion, and Target Recognition*, pp. 182-193, Orlando, FL, 21-24 Apr 1997

Julier, S.J. & Uhlmann, J.K. (2002). Reduced Sigma Point Filters for the Propagation of Means and Covariances through Non-linear Transformations, *Proc. of 2003 American Control Conf.*, pp. 887-892, Anchorage, AK, USA, 8-10 May 2002

Kuch, J.J. & Huang, T.S. (1994). Vision-based Hand Modeling and Tracking: A Hand Model, *Proc. of Twenty-Eighth Asilomar Conf. on Signal, Systems and Computers*, pp. 1251-1256, 31 Oct - 2 Nov 1994

La Viola Jr., J.J. (1996). A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion, *Proc. of American Control Conf.*, Vol. 3, pp. 2435-2440, Denver, CO, USA, Jun 2003

Lien, C.C. & Huang, C.L. (1998). Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing*, Vol. 16, No. 2, (Feb 1998) page numbers (121-134)

Lin, J.; Wu, Y. & Huang, T.S. (2002). Capturing Hand Motion in Image Sequences, *Proc. of IEEE Workshop on Motion and Video Computing*. Orlando, FL, pp. 99-104, Dec 2002

Pavlovic, V.; Sharma, R. & Huang, T. (1997). Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, (July 1997) page numbers (677-695)

Rehg, J.M. & Kanade, T. (1994). DigitEyes: Vision-based Hand Tracking for Human-Computer Interaction, *Proc. of IEEE Workshop on Motion of Non-Rigid And Articulate Objects*, pp. 16-22, Austin, TX, USA, Nov 1994

Shimada, N.; Shirai, Y., Kuno, J., & Miura, J. (1998). Hand Gesture Estimation and Model Refinement using Monocular Camera - Ambiguity Limitation by Inequality Constraint, *Proc. of the Third IEEE Int. Conf. on Face and Gesture Recognition*, pp. 268-273, Nara, Japan, Apr 1998

Shimada, N.; Kimura, K. & Shirai, Y. (2001). Real-time 3D Hand Posture Estimation based on 2-D Appearance Retrieval using Monocular Camera, *Proc. of IEEE ICCV Workshop on Recognition, Analysis, Tracking of Faces and Gestures in Real-Time Systems*, pp. 23-30, Vancouver, Canada, Jul 2001

Stenger, B.; Mendonca, P.R.S. & Cipolla, R. (2001). Model based 3D Tracking of an Articulated Hand, *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 310-315, Hawaii, USA, Dec 2001

Stenger, B.; Thayananthan, A., Torr, P. & Cipolla, R. (2004). Hand Pose Estimation using Hierarchical Detection. *Lecture Notes in Computer Science*, No. 3058, (2004) page numbers (105-116)

Szeliski, R. (1993). Rapid Octree Construction from Image Sequences. *CVGIP: Image Understanding*, Vol. 58, No. 1, (Jul 1993) page numbers (23-32)

Thayananthan, A.; Stenger, B., Torr, P.H.S. & Cipolla, R. (2003). Learning a Kinematic Prior for Tree-based Filtering, *Proc. of British Machine Vision Conf.*, Vol. 2, pp. 589-598, Norwich, UK, Sep 2003

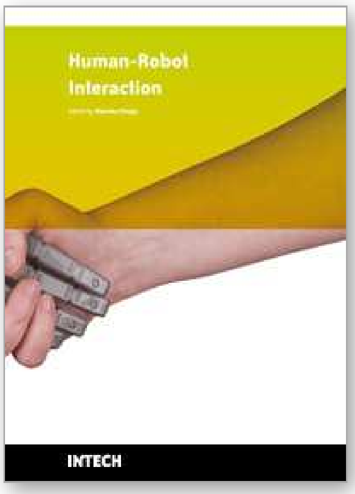
Ueda, E.; Matsumoto, Y., Imai, M. & Ogasawara, T. (2003). A Hand-Pose Estimation for Vision based Human Interfaces. *IEEE Transactions Industrial Electronics*, Vol. 50, No. 4, (Aug 2003) page numbers (676-684)

Utsumi, A. & Ohya, J. (1999). Multiple-hand Gesture Tracking using Multiple Cameras, *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 1, pp. 473-478, Ft. Collins, CO, USA, Jun 1999

Wan, E. & van der Merwe, R. (2000). The Unscented Kalman Filter for Nonlinear Estimation, *Proc. of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symp.*, pp. 153-158, Oct 2000

Wu, Y.; Lin, J.Y. & Huang, T.S. (2001). Capturing Natural Hand Articulation, *Proc. of the Eighth IEEE Int. Conf. on Computer Vision*, Vol. 2, pp. 426-432, Kerkyra, Greece, Sep 2001

Xiong, K.; Zhang, H.Y.& Chan, C.W. (2006). Performance Evaluation of UKF-based Nonlinear Filtering. *Automatica*, Vol. 42, No. 2, (Feb 2006) page numbers (261-270)



Human-Robot Interaction

Edited by Daisuke Chugo

ISBN 978-953-307-051-3

Hard cover, 288 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Human-robot interaction (HRI) is the study of interactions between people (users) and robots. HRI is multidisciplinary with contributions from the fields of human-computer interaction, artificial intelligence, robotics, speech recognition, and social sciences (psychology, cognitive science, anthropology, and human factors). There has been a great deal of work done in the area of human-robot interaction to understand how a human interacts with a computer. However, there has been very little work done in understanding how people interact with robots. For robots becoming our friends, these studies will be required more and more.

InTech Europe

University Campus STeP Ri

Slavka Krautzeka 83/A

51000 Rijeka, Croatia

Phone: +385 (51) 770 447

Fax: +385 (51) 686 166

www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai

No.65, Yan An Road (West), Shanghai, 200040, China

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元

Phone: +86-21-62489820

Fax: +86-21-62489821