Robust Finger Tracking for Wearable Computer Interfacing

Sylvia M. Dominguez
Electrical Engineering Dept.
UCLA & HRL Laboratories, LLC
Los Angeles, CA 90095
(310) 317-5380
sylvia@ee.ucla.edu

Trish Keaton
Information Sciences Laboratory
HRL Laboratories, LLC
Malibu, CA 90265
(310) 317-5759
pakeaton@hrl.com

Ali H. Sayed[1]
Electrical Engineering Dept.
University of California
Los Angeles, CA 90095
(310) 267-2142
sayed@ee.ucla.edu

ABSTRACT

Key to the design of human-machine gesture interface applications is the ability of the machine to quickly and efficiently identify and track the hand movements of its user. In a wearable computer system equipped with head-mounted cameras, this task is extremely difficult due to the uncertain camera motion caused by the user's head movement, the user standing still then randomly walking, and the user's hand or pointing finger abruptly changing directions at variable speeds. This paper presents a tracking methodology based on a robust state-space estimation algorithm, which attempts to control the influence of uncertain environment conditions on the system's performance by adapting the tracking model to compensate for the uncertainties inherent in the data. Our system tracks a user's pointing gesture from a single head mounted camera, to allow the user to encircle an object of interest, thereby coarsely segmenting the object. The snapshot of the object is then passed to a recognition engine for identification, and retrieval of any pre-stored information regarding the object. A comparison of our robust tracker against a plain Kalman tracker showed a 15% improvement in the estimated position error, and exhibited a faster response time.

Categories and Subject Descriptors

Gesture Interfaces, Kalman Filter Tracking, Robust Estimation.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Robust Estimation, Gesture Interfaces, Kalman Filter Tracking.

1. INTRODUCTION

A recent computing trend is wearable computing for the design of intelligent assistants to provide location-aware information access that can help users more efficiently accomplish their tasks. Thus imagine a user driving by a hotel or a restaurant while on a foreign trip. By pointing at either establishment, the machine would be able to convey to the driver recommendations about the hotel or the restaurant menu and opening hours [10]. In another scenario, while working on a digital desktop system [1],[9], the user's fingertip could be made to act as a mouse and used to 'point to', 'click[1] and 'drag' virtual objects. Computing and sensing in such environments must be reliable, persistent (always remains on), easy to interact with, transparent (user does not know it is there) and configured to support different needs and complexities. The success of such systems will rely heavily upon the ability to visually track and recognize the user's hand and pointing gestures in real-time.

A number of vision-based pointing gesture tracking algorithms has been proposed in the literature. These algorithms extract color segmentations, 3D stereo segmentations, and shape information from the machine's camera view in order to identify the user's hand and fingertip position. The algorithms, however, are complex and computationally intensive, and tend to slow down the response of the machine to a great extent. Some recent computer vision approaches for tracking applications speed up their computation time by reducing the image search area into a smaller window centered around the last known position of the moving object [1], [10]. The main drawback of these methods is that when the object moves faster than the frame capture rate, the object will move out of the window range forcing the algorithm to reset the image search area to the full view in order to recover the position of the object. The repeated reduction and expansion of the image search area slows down the system performance considerably. Some tracking solutions have attempted an improvement by gradually varying the search window's size according to the moving object speed [1]. The faster the object moves, the larger the search window becomes. Therefore, if the object is moving fast, the search window is large and the computation time for the vision algorithm increases. More advanced systems, such as in [5],[9], use state-space estimation techniques to center the smaller search window around a future predicted position of the user's fingertip, rather than around its current position. In this way, as the moving object speed increases, the predicted window position will accompany the speeding object keeping it inside the window's view sight. The window size thus remains small and centered around the object of interest regardless of its speed. This in turn keeps the memory allocations at a minimum, thus improving the system's response

[1] The work of A. H. Sayed was partially supported by NSF award ECS-9820765.

time. However, if the object abruptly changes its movement patterns (which introduces modeling uncertainties), then such systems tend to break down, and the user's hand is quickly lost from their search window view. Our robust estimation algorithm [3] is designed to model data uncertainties, such as those created by the user's random egomotion, and is thus more effective in keeping the user's hand inside the small search window, thereby speeding up the system's response time.

2. ROBUST FINGERTIP TRACKER

We have designed at HRL a wearable computer system called *"Snap&Tell"*, see [6], which enables a user to specify, segment, and recognize objects of interest, such as landmarks, by simply pointing at and encircling them with the user's fingertip. The *"Snap&Tell"* system accepts input from a color pencil camera, and segments the input video stream based on color. The color segmented image is then fed into a skin/non-skin discrimination algorithm to detect skin tones and extract the user's hand. Once the hand is extracted, shape and curvature analysis is used to determine the coordinate position of the fingertip.

To perform the tracking of the fingertip position in real-time, a robust state-space tracker is used to predict the future user's fingertip position. The predicted position coordinates are then used to center a small image search window around the expected fingertip position occurring in the next video frame. Accurate prediction of the fingertip region of interest speeds up the response time of the system; making it more memory and computationally efficient. At the conclusion of the pointing gesture, the algorithm determines if an object has been selected by the user, and extracts it from the scene by cropping the region of interest. The segmented object is then classified, and any pre-stored information associated with the object is made available to the user. A system block diagram for *"Snap&Tell"* is shown in Figure 1.

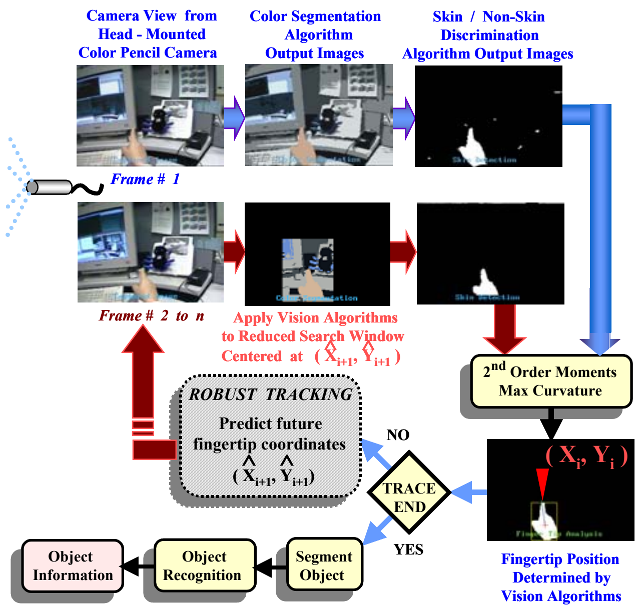
Camera View from Head - Mounted Color Pencil Camera
Color Segmentation Algorithm Output Images
Skin / Non-Skin Discrimination Algorithm Output Images
Frame # 1
Frame # 2 to n
Apply Vision Algorithms to Reduced Search Window Centered at ($\hat{X}_{i+1}$, $\hat{Y}_{i+1}$)
2nd Order Moments Max Curvature
ROBUST TRACKING
Predict future fingertip coordinates ($\hat{X}_{i+1}$, $\hat{Y}_{i+1}$)
NO
TRACE END
YES
(X_i, Y_i)
Fingertip Position Determined by Vision Algorithms
Object Information
Object Recognition
Segment Object

Figure 1. Block diagram for the "Snap&Tell" wearable computer interface.

2.1 Skin/Non-Skin Color Segmentation

To determine the skin-like regions in the current frame, we first perform a color segmentation based on the fast and robust mean shift algorithm [2]. By using the mean shift algorithm the number of dominant colors can be determined automatically, unlike the k-means clustering method where the initial number of classes must be chosen. Here, the intensity distribution of each color component in the current frame is viewed as a probability density function. The mean shift vector is the difference between the mean of the probability function on a local area and the center of this region. Mathematically, the mean shift vector associated with a region $S_{\vec{x}}$ centered on $\vec{x}$ can be written as:

$$\vec{V}(\vec{x}) = \frac{\int_{\vec{y}\in S_{\vec{x}}} p(\vec{y})(\vec{y}-\vec{x})d\vec{y}}{\int_{\vec{y}\in S_{\vec{x}}} p(\vec{y})d\vec{y}} \tag{1}$$

where $p(\cdot)$ is the probability density function. As shown in (1), the mean shift vector is proportional to the gradient of the probability density $\nabla p(\vec{x})$ at the point it is computed, and reciprocal to the probability density $p(\vec{x})$, such that

$$\vec{V}(\vec{x}) = c\frac{\nabla p(\vec{x})}{p(\vec{x})} \tag{2}$$

where c is a constant. Since the mean shift vector is along the direction of the probability density function maximum, we can exploit this property to find the actual location of the density maximum by searching for the mode of the density. One dominant color can be located by moving search windows in the color space using the mean shift vector iteratively. After removing all color inside the converged search window, one can repeat the mean shift algorithm again to locate the second dominant color. This process is repeated several times to identify a few major dominant colors which segment the image into like-color regions. The dominant colors of the current frame are used as the initial guess of dominant colors in the next frame, thus speeding up the computational time (adjacent frames are usually similar). After segmenting the current frame into homogeneous regions, we determine whether each region is skin-like by considering the mean hue and saturation values and geometric properties of the region. This region-based skin detection procedure is more robust to varying illumination conditions than pixel-based approaches [4],[11].

2.2 Shape Analysis

Once the skin-like regions have been segmented out, we clean up this image by applying morphological operations to minimize the number of artifacts being picked up as having skin-like color properties. Then the user's hand orientation with respect to the x-axis (i.e. pointing direction) is derived using central 2nd order moments, and the fingertip position is determined as the point of maximum curvature along the contour of the hand.

2.3 Robust State-Space Fingertip Tracking

To achieve computational efficiency, memory savings and real-time tracking, a robust state-space estimation algorithm is used to reduce the search area to a smaller search window centered around predictions of the fingertip position. The need for robust methods arises from the desire to control the influence of uncertain environment conditions on system performance including, for example, the effect of random variations in object speed and motion characteristics. A first step towards this objective is to formulate a robust state-space model that describes the user's fingertip motion in the presence of uncertain environments. Thus let T denote the frame capture rate for the wearable computer system (for our system 1/15 seconds/frame). Let also $\{ \alpha_{x,i} , \alpha_{y,i} \}$ denote the fingertip accelerations along the x and y directions (measured in pixels per second^2), and let $\{ v_{x,i} , v_{y,i} \}$ denote the speeds along these same directions during the i^{th} frame (measured in pixels/second). Then one could approximate the present fingertip position in the i^{th} frame $\{ x_i , y_i \}$ in terms of the previous frame fingertip pixel coordinates $\{ x_{i-1} , y_{i-1} \}$ and the pixel-shift per frame estimated by $\{ v_{i-1}T, \alpha_{i-1} T^2/2 \}$ such as,

$$x_i \approx x_{i-1} + v_{x,i-1}T + \alpha_{x,i-1} T^2/2 \tag{3}$$

$$y_i \approx y_{i-1} + v_{y,i-1}T + \alpha_{y,i-1} T^2/2 \tag{4}$$

$$v_{x,i} = v_{x,i-1} + \alpha_{x,i-1}T \tag{5}$$

$$v_{y,i} = v_{y,i-1} + \alpha_{y,i-1}T \tag{6}$$

These equations motivate the following state-space model with state vector s_i and measurement vector z_i.

$$s_i \overset{\Delta}{=} \begin{bmatrix} x_i & y_i & v_{x,i} & v_{y,i} & \alpha_{x,i} & \alpha_{y,i} \end{bmatrix}^T \tag{7}$$

$$s_{i+1} = Fs_i + Gu_i \tag{8}$$

$$z_i = Hs_i + v_i \ , \qquad z_i \overset{\Delta}{=} \begin{bmatrix} x_i & y_i \end{bmatrix}^T \tag{9}$$

with model parameters

$$F = \begin{bmatrix} 1 & 0 & T & 0 & 0.5T^2 & 0 \\ 0 & 1 & 0 & T & 0 & 0.5T^2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad H^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{11}$$

where the process noise u_i and the measurement noise v_i are assumed to be uncorrelated, with zero-mean white gaussian distributions and corresponding covariance matrices Q and R. The entries of these covariance matrices are chosen for optimality by testing the whiteness of the resulting innovations process of the Kalman filter (following the method of [7]). Our chosen values for R and Q meet a 95% confidence whiteness test [3]. In addition, the measurement vector z_i consists of the centered pixel coordinates that are provided by the vision algorithm locating the fingertip position. These coordinates can therefore be regarded as noisy measurements of the actual pixel coordinates $\{ x_i, y_i \}$. By using the assumed state-space mode (7)-(9), one can then proceed to employ a variety of estimation techniques to 'clean' z_i from measurement noise and to predict future movements of the $\{x_i, y_i\}$ fingertip coordinates. One such technique is the Kalman filter, which provides the optimal linear least-mean-squares (l.l.m.s.) estimate of the state variable given prior measurements.

The Kalman filtering formulation, however, assumes that the underlying model parameters $\{F,G,H,R,Q\}$ are accurate. When this assumption is violated, the performance of the filter can deteriorate and one is therefore motivated to consider robust variants; robust in the sense that they attempt to limit, in certain ways, the effect of model uncertainties on the overall filter performance. One way to model uncertainties is to treat the given parameters $\{F,G\}$ as nominal values and to assume that the actual values lie within a certain set around them. Thus, consider replacing (8) with a robust state-space model, such as

$$s_{i+1} = (F + \delta F_i)s_i + (G + \delta G_i)u_i \tag{17}$$

where the perturbations in $\{F,G\}$ are modeled as

$$\begin{bmatrix} \delta F_i & \delta G_i \end{bmatrix} = M\Delta_i \begin{bmatrix} E_f & E_g \end{bmatrix} \tag{18}$$

for some matrices $\{M, E_f, E_g\}$ and for an arbitrary contraction Δ_i, $\| \Delta_i \| \leq 1$. For generality, one could allow the quantities $\{M, E_f, E_g\}$ to vary with time as well. This is useful in the case when our model changes dramatically in a particular time instance, such as when the user starts walking or moves his/her head abruptly. The model (18) allows the designer to restrict the sources of uncertainties to a certain range space (defined by the matrix M), and to assign different levels of distortion by selecting the entries of $\{E_f, E_g\}$ appropriately, see [3],[8]. For the wearable computer system, there are several sources of uncertainties that may interfere with the accuracy of the assumed state-space model. The uncertainties can be due to the camera moving along with the user's head motion, changes in lighting conditions, the background and object moving independently from each other, or to the user's pointing finger abruptly changing directions at variable speeds and accelerations. All these factors changing constantly in time create different conditions of uncertainties. The authors are currently investigating more complex adaptive models for modeling the uncertainties associated with tracking humans for surveillance, monitoring and interfacing applications.

Let Π_0 be a chosen positive definite weighting matrix. Usually Π_0 is chosen as the variance matrix of s_0 , $\Pi_0 = E\{s_0 s_0^T\}$.

Let $\hat{s}_i$ stand for an estimate of s_i. Then the following equations describe a robust algorithm for determining $\hat{s}_i$ (for details see [8]):

Set the initial conditions to:

$$\hat{s}_{0|0} = P_{0|0} H^T R^{-1} z_0 \quad \text{and} \quad P_{0|0} = \left(\Pi_0^{-1} + H^T R^{-1} H\right)^{-1}$$

Step 1. If HM=0, then set $\hat{\lambda}_i = 0$ (non robust filter). Otherwise, select α (typically $0 < \alpha < 1$) and set

$$\hat{\lambda}_i = (1+\alpha)\left\| M^T H^T R^{-1} HM \right\|.$$

Step 2. Replace $\{Q, R, P_{i|i}, G, F\}$ by:

$$\hat{Q}_i^{-1} = Q^{-1} + \hat{\lambda}_i E_g^T \left[I + \hat{\lambda}_i E_f P_{i|i} E_f^T \right]^{-1} E_g$$

$$\hat{R}_{i+1} = R - \hat{\lambda}_i^{-1} HMM^T H^T$$

$$\hat{P}_{i|i} = \left(P_{i|i}^{-1} + \hat{\lambda}_i E_f^T E_f \right)^{-1}$$

$$= P_{i|i} - P_{i|i} E_f^T \left[\hat{\lambda}_i^{-1} I + E_f P_{i|i} E_f^T \right]^{-1} E_f P_{i|i}$$

$$\hat{G}_i = G - \hat{\lambda}_i F \hat{P}_{i|i} E_f^T E_g$$

$$\hat{F}_i = (F - \hat{\lambda}_i \hat{G}_i \hat{Q}_i E_g^T E_f)(I - \hat{\lambda}_i \hat{P}_{i|i} E_f^T E_f)$$

If $\hat{\lambda}_i = 0$, then simply set $\hat{Q}_i = Q$, $\hat{R}_{i+1} = R$, $\hat{P}_{i|i} = P_{i|i}$, $\hat{G}_i = G$, and $\hat{F}_i = F$.

Step 3. Update $\{\hat{s}_{i|i}, P_{i|i}\}$ as follows:

$$\hat{s}_{i+1} = \hat{F}_i \hat{s}_{i|i}$$

$$e_{i+1} = z_{i+1} - H\hat{s}_{i+1}$$

$$P_{i+1} = F\hat{P}_{i|i} F^T + \hat{G}_i \hat{Q}_i \hat{G}_i^T$$

$$R_{e,i+1} = \hat{R}_{i+1} + HP_{i+1}H^T$$

$$P_{i+1|i+1} = P_{i+1} - P_{i+1}H^T R_{e,i+1}^{-1} HP_{i+1}$$

$$\hat{s}_{i+1|i+1} = \hat{s}_{i+1} + P_{i+1|i+1} H^T \hat{R}_{i+1}^{-1} e_{i+1}$$

The estimates $\{\hat{s}_i, \hat{s}_{i|i}\}$, and the predicted fingertip coordinates for each frame, can be obtained by recursively iterating between steps 2 and 3. Note that for the case when $\hat{\lambda}_i = 0$, steps 2 & 3 are reduced to the standard time and measurement update Kalman equations. Moreover, P_i and $P_{i|i}$ will have the interpretation of error covariance matrices,

$$P_i \overset{\Delta}{=} E\{(s_i - \hat{s}_{i|i-1})(s_i - \hat{s}_{i|i-1})^T\}$$

$$P_{i|i} \overset{\Delta}{=} E\{(s_i - \hat{s}_{i|i})(s_i - \hat{s}_{i|i})^T\}$$

where now $\hat{s}_i$ and $\hat{s}_{i|i}$ will have the following interpretation,

$$\hat{s}_i \overset{\Delta}{=} \text{l.l.m.s. estimate of } s_i \text{ given } \{z_0, z_1, \ldots, z_{i-1}\}$$

$$\hat{s}_{i|i} \overset{\Delta}{=} \text{l.l.m.s. estimate of } s_i \text{ given } \{z_0, z_1, \ldots, z_{i-1}, z_i\}$$

3. EXPERIMENTAL RESULTS

Our choice for the state matrix F gives higher weight to the previous fingertip coordinates, while downplaying the estimates for acceleration and velocity. This model is fairly reasonable in situations when the user is standing still and pointing at an object, where the velocity is constant and acceleration is nonexistent. However, when the user is actively moving, uncertainties arise mainly in the acceleration and velocity models. Empirical evidence suggests the following choices for $\{M, E_f, E_g\}$, which account for the velocity and acceleration instabilities in a wearable system:

$$M = \begin{bmatrix} 0.5 & 0.5 & 0.25 & 0.25 & .125 & 0.125 \end{bmatrix}^T$$
$$E_g = \begin{bmatrix} 0 & 0 & 0.3 & 0.3 & 0.9 & 0.9 \end{bmatrix}$$
$$E_f = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 2 \end{bmatrix}$$

We applied this perturbation model along with our robust estimation algorithm to the task of tracking a typical fingertip trajectory of a user encircling an object of interest. We display a single frame of the results in Figure 2.

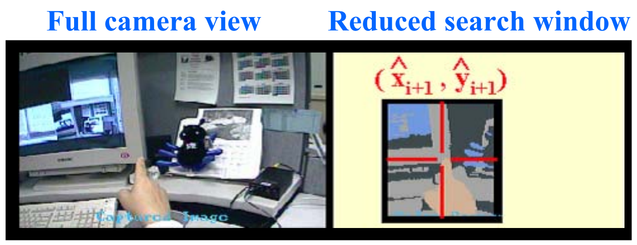

Figure 2. Successfully tracked fingertip using a robust state-space Kalman tracker.

Here we can see that our robust tracker accurately predicts the fingertip position, since the reduced search window is centered around the previously predicted fingertip coordinates, which nearly overlaps the actual present fingertip position. The *search window* size was chosen to be at least twice the size of the maximum estimation errors in the x and y directions, of our robust tracker previously applied to a training sequence representative of a typical pointing finger trajectory. Therefore, the more accurate the tracker is, the smaller the search window needed, and the faster the overall system response time will be. In this particular simulation, the response time of our overall system was 68% faster than the response obtained by a system that uses a full camera view to track the user's fingertip, and 23% faster when compared with systems such as [1] and [10] that use a small search window centered around the previous fingertip position (rather than the predicted future position). Furthermore, a comparison of the MSE results between the plain Kalman tracker, as in [5], and the robust Kalman tracker [3], showed over 15% improvement in the estimation error and response time using the robust tracker. These performance results are encouraging and merit future exploration. Figure 3 shows the fingertip tracking results over a sequence of frames, where the last frame shows the fingertip track and the extracted object of interest.

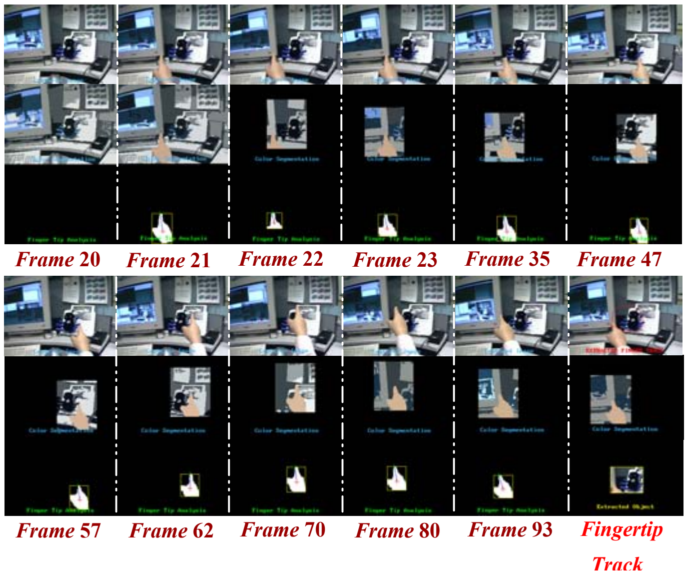

Figure 3. Real-time fingertip tracking using our robust tracker.

4. CONCLUDING REMARKS

The authors are currently investigating more complex adaptive models for modeling the uncertainties associated with user's head motion, walking, and changes in lighting conditions. One example is when the user starts walking while pointing at an object of interest. In this scenario, the uncertainties δF_i and δG_i would have larger values than when the user is standing still. Therefore, when our system would detect constant movement in the camera view (hinting walking motion), we would switch our robust tracker's perturbation model to the "walking" uncertainty model. Another situation is when the user moves his head abruptly, changing the camera view away from the user's hand and object of interest. At this point, our system would detect a complete change in background and it would switch our tracker to prediction without measurement updates (i.e., $\hat{s}_{i+1} = \hat{F}_i \hat{s}_i$) , until the camera view becomes stable again. Then we would return to our robust tracker with the "user-standing still" uncertainty model, and resume tracking the user's fingertip. In addition, we are exploring the use of this method towards robust tracking of people for purposes of surveillance and scene monitoring.

5. REFERENCES

[1] T. Brown and R.C. Thomas, "Finger tracking for the digital desk", *Proc. Australasian User Interface Conference (AUIC)*, vol. 1, pp. 11-16, Australian National University, Canberra, Australia, 2000.

[2] D. Comaniciu and P. Meer, "Robust analysis of feature space: color image segmentation", *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 750-755, San Juan, Puerto Rico, 1997.

[3] S. M. Dominguez, T. Keaton, A. H. Sayed, "Comparison of robust estimation and Kalman filtering applied to fingertip tracking in human-machine interfaces", *Proc. Asilomar Conference on Signal, Systems & Computers*, Pacific Grove, CA., Nov. 2001.

[4] K. Imagawa, Lu Shan, S. Igi, "Color-based hands tracking system for sign language recognition", *Proc. Conference on Automatic Face and Gesture Recognition*, pp. 462-467, Nara, Japan, 1998.

[5] C. Jennings, "Robust finger tracking with multiple cameras", *Proc. Conference on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 152-160, Corfu, Greece, 1999.

[6] T. Keaton, S. M. Dominguez, and A. H. Sayed, "Snap&Tell: a vision-based wearable system to support web-on-the-world applications", *Proc. Digital Image Computing – Techniques and Applications Conference*, Melbourne, Australia, Jan. 2002.

[7] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering", *IEEE Transactions on Automatic Control*, AC-15, pp. 175-183, 1970.

[8] A. H. Sayed, "A framework for state-space estimation with uncertain models", *IEEE Transactions on Automatic Control*, vol. 46, no. 7, pp. 998-1013, July 2001.

[9] A. Wu, M. Shah, and N. Da Vitoria Lobo, "A virtual 3D blackboard: 3D finger tracking using a single camera", *Proc. Conference on Automatic Face and Gesture Recognition*, pp. 536-543, Grenoble, France, 2000.

[10] J. Yang, W. Yang, M. Denecke, A. Waibel, "Smart sight: a tourist assistant system", *Proc. International Symposium on Wearable Computers*, vol. 1, pp. 73-78, San Francisco, CA., 1999.

[11] X. Zhu, J. Yang, and A. Waibel, "Segmenting hands of arbitrary color", *Proc. Conference on Automatic Face and Gesture Recognition*, pp. 446-453, Grenoble, France, 2000.