IMPROVED VIRTUAL MOUSE POINTER USING KALMAN FILTER BASED GESTURE TRACKING TECHNIQUE

D.R.A.M. Dissanayake[1], U.K.R.M.H. Rajapaksha[2] and M.B Dissanayake[3]

Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka.
Email: [1]madushan.amd@gmail.com, [2]ehazitha@gmail.com, [3]maheshid@ee.pdn.ac.lk

ABSTRACT

Human computer interaction (HCI) has become an emerging area of research in recent past. Image based HCI techniques can lead the modern technology to the level of self-controlling smart devices. In this research the possibility of implementing a virtual mouse controlled by video inputs is considered. Proposed algorithm is capable of tracking pointer movement along with right, left and double click actions, using colored thumb, index and middle fingers. The method presented use color filtering algorithms to initially detect human hand gestures as control inputs from the input video, to be used as mouse pointer movements and other required clicking functionalities. Kalman filter tracking was employed to overcome errors due to sudden occlusions and to handle measurement fluctuations of the system implemented. A cost minimization algorithm has been introduced to eliminate distractions due to presence of unwanted noisy objects, as well as to distinguish detections corresponding to clicking action and pointer movement which enable continuous smooth tracking of the pointer movement. Finally, the improvements obtained in accuracy and reliability by using the Kalman filter and cost minimization technique over traditional method is investigated and presented.

***Key words*:** HCI, Kalman filter, Image based tracking, Hungarian algorithm

1. INTRODUCTION

With the rapid growth in computer technology, HCI has played a major role in bringing the power of computers and other smart devices to people in a way that are both cost effective and user friendly. In HCI, mouse controlling has drawn more attention recently and various kinds of both software and hardware have been designed. Since spacing is currently a major limiting factor, hardware for HCI is not encourage much. Instead of hardware, virtual machines are the best solution and virtual mouse is one of the superlative approaches for mouse controlling. Inputs to a virtual mouse can be given in various ways but the most popular method is web camera based hand gestures.

Though hand gesture based web camera system for mouse controlling is not a new topic to the research field, pointer fluctuations due to measurement errors and sudden measurement disappearance have great impact on the smooth and continuous pointer movement. In most virtual mouse projects [1, 2] they have considered only the raw measurements to control the mouse pointer and there aren't ways to predict the pointer movement under sudden occlusions or measurement disappearance due to lighting or environmental changes. In [2] they have added a code to reduce the pointer fluctuations based on the previous and the current finger tips position pixels, but still the results are inadequate. In our approach we have introduced an implementation of Kalman filter to overcome above mentioned drawbacks to improve the pointer movement. Other than that in order to give a proper functionalities of a mouse, basic clicking actions are also inserted while ensuring the smooth and continuous pointer movement. Among various approaches for developing HCI for gesture tracking in computer vision, HCI Using Interface has been chosen for our application by pasting colored stickers on fingers.

The rest of the paper is organized as follows. Section 2 presents methodology, while Section 3 and 4 discuss Kalman filter and Cost minimization based motion tracking. Section 5 analyzes the results and Section 6 concludes the paper.

2. METHODOLOGY

Initially an image frame is captured from the video which is acquired via in- build web camera of the laptop, to be used as the input image. Input control signals to maneuver mouse pointer are obtained from the captured frame by color segmentation. Since skin detection is neither reliable nor efficient way to determine fingertips, three blue color strikers are pasted on thumb, index and middle fingers. Color segmentation is done by the color filtering algorithm based on

HSV color space [3] to extract only blue color objects in the input frame. Maximum and minimum thresholds of hue, saturation and value channel levels corresponding to blue color are tuned experimentally using histograms settings. Morphological operations are then performed on the obtained mask to remove noise and fill in holes.

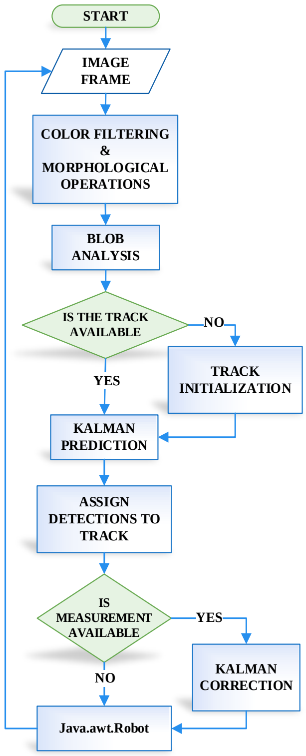

Figure 1: Over view of the methodology

Corrected or predicted centroid values for pointer detection and movements are determined by Kalman algorithm with cost minimization algorithm. Then these values are sent to java.awt.Robot class [4] to bridge the MATLAB and mouse driver in laptop/pc, since the ability to control mouse pointer movement and clicking functionalities are not available in MATLAB. When no detections are associated with the track for several consecutive frames, the algorithm assumes that the users hand has left the field of view and pause the pointer. Image processing toolbox, computer vision toolbox and basic functions, classes and objects in MATLAB 2014b have been used to develop the algorithm.

3. KALMAN FILTER BASED TRACKING

Since the pointer movement is controlled by using the centroid details of the filtered blob corresponding to index finger, tracking of the index finger location is very crucial in order to guarantee the smooth pointer movement. Due to the filtering errors and background noises (other control inputs are also background noises when considering the pointer movement) sometimes the detection corresponding to the index finger may change or disappear. These kind of situations effect the smoothness of the pointer movement.

To overcome this problem we have introduced the implementation of Kalman filter algorithm for index finger movement tracking. The Kalman filter is a recursive predictive algorithm based on the state space representation of the systems. It estimates the state of the dynamic system by using the state space parameters and the noise values. And by using the measurements and the measurement error values it improves the estimated state. Thus by using the Kalman filter algorithm the effects on noises, nondeterministic centroids movements can be avoided.

Consider the discrete time state space model,

$$\underline{x}(k+1) = A * \underline{x}(k) + B * \underline{u}(k) + \underline{w}(k) \qquad (01)$$

$$\underline{z}(k) = H * \underline{x}(k) + \underline{v}(k) \qquad (02)$$

where, $\underline{x}(k)$, $\underline{u}(k)$, $\underline{w}(k)$, $\underline{z}(k)$, $\underline{v}(k)$, A, B, H are the state vector of the system, the control input, the process noise, the measured state, the measurement noise, the state transition model, the control model and the measurement model respectively.

Kalman filter algorithm consists of two steps, Time update and the measurement update. In the first step the state is predicted with the dynamic model.

In the second step estimated state is corrected with the observation model, so that the error covariance of the estimator is minimized [5]. It is an optimal estimator which tries to minimize the state estimation error. This procedure is repeated for the each time step with the state of the previous time step as initial value.

In the case of tracking mouse pointer movement, initially the state space model for the process is derived by describing the movement on two orthogonal axis x and y. Mouse pointer movement can be derived as linear, constant acceleration model in discrete time state space [6], where the x(k), y(k) is the x, y coordinate position and the k^{th} time step and the Δt is the sampling time.

For x coordinate, model can be derived as follows.

$$x(k+1) = x(k) + \dot{x}(k)\Delta t + \frac{\ddot{x}(k)}{2} * \Delta t^2 \qquad (03)$$

$$\dot{x}(k+1) = \dot{x}(k) + \ddot{x}(k+1) * \Delta t \qquad (04)$$

$$\ddot{x}(k+1) = \ddot{x}(k) \qquad (05)$$

Equations for y coordinates can be obtained by replacing x in eq.(03), eq.(04) and eq.(05) with y. From the above equations we can derive the state space model for the pointer movement as follows.

$$\underline{x}(\mathrm{k}) = \begin{bmatrix} x(k) \\ y(k) \\ \dot{x}(k) \\ \dot{y}(k) \\ \ddot{x}(k) \\ \ddot{y}(k) \end{bmatrix} \quad \underline{z}(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} \quad \underline{u}(k) = [\underline{0}] \qquad (06)$$

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \Delta t^2/2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Delta t^2/2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (07)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (08)$$

4. COST MINIMIZATION BASED DISTORTION ELIMINATION

When the tracking of index finger which controls the pointer movement is continuing, noisy detections can occur. When flickering or sudden environmental changes take place, artifacts may get detected on top of fingertip detections. This is due to the fact that color filtering algorithm is not adaptive.

Detections corresponds to the clicking functions, objects detected as thumb and middle finger tips also act as noisy detections for the pointer movement. When there are several detections, appearing at the same time in a single frame, there should be a method to distinguish between nose and desired signal. Otherwise pointer tends to follow noisy detection. From detections in the current frame, assigning object corresponds to the pointer movement, to the pre-defined track is done by cost minimization algorithm. The cost takes into account the Euclidean distance between the predicted centroid value of the pointer movement given by the prediction stage of the Kalman algorithm and the centroids of the detections in the current frame which are the measurements.

Results are stored in a matrix to form a cost matrix and perfect assignment for the track among the detections are found by solving the assignment problem. James Munkres's variant of the Hungarian assignment algorithm [7] is used to solve the assignment problem represented by the cost matrix while minimizes the total cost.

5. RESULST AND DISCUSSION

In this research we mainly considered about linear motion and curvy linear motion to verify the effectiveness of applying Kalman algorithm under certain occlusion conditions. The noise parameters were tuned according to practical knowledge such that the best matching is obtained. 1.3cm wide and 1cm high blue color sticker was used for labeling in the experiment. Distance from the camera to the sticker was 52 cm. 4cm wide and 32cm high obstacle was placed at 27 cm away from the camera.

5.1. Linear Motion

The linear motion of the mouse pointer under occlusion condition is considered. Figure 2 shows the results of applying Kalman algorithm for the linear motion with occlusion.

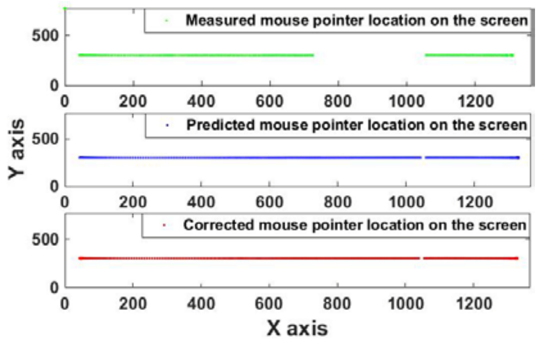

Figure 2: Variation of measured mouse pointer, Predicted mouse pointer, Corrected mouse pointer

As seen by the results, the Kalman algorithm is very good at tracking the linear motion at occlusion conditions. Normally in this kind of applications fully occluded environments don't last for a longer time. Thus the Kalman algorithm definitely helps to improve the tracking.

Figures 3 and 4 show the pointer variation in detail. Figure 3 shows the full scale variation along x axis and zoomed y axis. Figure 4 shows the zoomed portion of both x and y axes. Although we see a straight line in figure 2, it is not exactly straight when zoomed. That difference can be seen in the figure 3. But when the Kalman algorithm is applied to the pointer movement we can clearly observe that higher fluctuations are reduced and motion is converged to a certain fixed path. Kalman predictions not only depend on the past few measurements but also on the measurement noise and process noise. That's why the tracked path does not change suddenly with the measurements. When the measurements are not available the corrected pointer location is same as the predicted pointer location. Figure 4 shows the further zoomed part of the Figure 3. That also shows the impact of Kalman algorithm to reduce measurement fluctuations and thus this method has helped to improve the smoothness of mouse pointer movements.

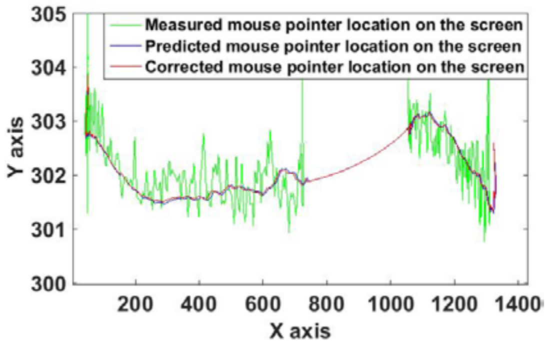

Figure 3: Variation of measured mouse pointer, Predicted mouse pointer, Corrected mouse pointer

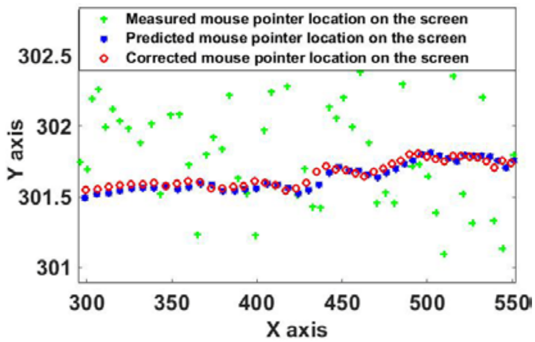

Figure 4: variation of measured mouse pointer, Predicted mouse pointer, Corrected mouse pointer

5.2. Partial Circular Motion (curvilinear motion)

The curvilinear motion of the index finger under occlusion condition is also considered in this research. The effectiveness of the usage of Kalman algorithm to predict curvilinear motion is experimentally investigated. Figure 5 shows the Kalman filter predicted and corrected outputs tend to move away from the curved line. The filter output has slightly followed along the tangential line to the curve at the point where the measurements are lost. That is because the Kalman filter was designed to predict linear motion, not the curvilinear motions. But still the predictions are pretty good to understand the motion.

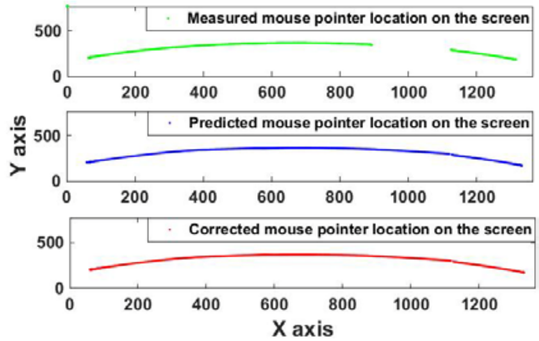

Figure 5: Variation of measured mouse pointer, Predicted mouse pointer, Corrected mouse pointer

6. CONCLUTION

Obtained results emphasis that the Kalman algorithm helps to improve the pointer movement under occlusion conditions and minimizes fluctuations in measurements to get a better estimates of the coordinates for linear motions. For the linear motion the algorithm shows very much effectiveness for the application. When the motion is not linear it is able to track the tangential line of the motion. That is, it tries to track the linear part of the measured motion because the original Kalman algorithm has been developed for the linear motion tracking. Results implies that, implemented methodology has improved the accuracy, smoothness and reliability of mouse pointer movement. The future improvements of this research would be, analyzing the use of adaptive color filtering and realizing of advance mouse control actions.

7. REFERENCES

[1] A. Banerjee, A.Ghosh, K. Bharadwaj, and H. Saikia, "*Mouse control using a web camera based on colour detection*", International Journal of Computer Trends and Technology (IJCTT), volume 9,number 1, Mar 2014

[2] H. Park, "*A Method for controlling mouse movement using a real-time camera*", Department of Computer Science, Brown University, Providence, RI, USA.

[3] S. Sural, G. Qian, and S. Pramanik,"*Segmentation and histogram generation using the HSV color space for image retrieval,*" International Conference on Image Processing (ICIP), pp. 589-592, 2002.

[4] Robot (Java Platform SE 7) - Oracle

Documentation. At: http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html

[5] G. Welch, and G. Bishop, ”*An introduction to the Kalman Filter*”, UNC-Chapel Hill, TR 95-041, pp.2-7, July 24, 2006.

[6] N. Obolensky, “*Kalman Filtering methods for moving vehicle tracking*”, Thesis for Master of Science, University of Florida, pp. 4-12, 2002.

[7] J. Munkres, “Algorithms for assignment and transportation problems”, Journal of the Society for Industrial and Applied Mathematics, Volume 5, Number 1, March, 1957.