Skin and Motion Cues Incorporated Covariance Matrix for Fast Hand Tracking System

Mohd Shahrimie Mohd Asaari, Shahrel Azmin Suandi*, Bakhtiar Affendi Rosdi

Intelligent Biometric Group, School of Electrical and Electronic Engineering, Engineering Campus,
Universiti Sains Malaysia (USM), 14300, Nibong Tebal, Penang, Malaysia

*Corresponding Author: shahrel@eng.usm.my

Copyright ©2014 Horizon Research Publishing All rights reserved.

Abstract Hand tracking is one of the essential elements in vision based hand gesture recognition system. The tracked hand image can provide a meaningful gesture for more natural ways of Human Computer Interaction (HCI) systems. In this paper, we present a fast hand tracking method based on the fusion of skin and motion features incorporated covariance matrix. First, hand region is detected using a fusion of skin and motion cues, and a region of interest (ROI) is created around the detected region. During the tracking, skin and motion features are extracted around top, left and right corners of the ROI and hand displacement is measured using ROI based tracker. To increase the robustness, we incorporate a covariance matrix of the ROI window as a region descriptor to represent the target object. In the consecutive frames, we measure the distance descriptor covariance matrix (DDCM) between the target object and the covariance matrix extracted from new ROI position. When DDCM is not satisfying a certain acceptable threshold, ROI position is adjusted by shifting the ROI window around nearest neighbor to obtain a set of candidate regions. We assign a candidate region which has the smallest DDCM as the correct estimated ROI position. The experimental result shows that our approach can track the hand gesture under several real-live scenarios with a detection rate above 95% with the tracking speed at 42fps on average.

Keywords Hand Detection, Hand Tracking, Region of Interest, Hand Gesture, Covariance Matrix

1 Introduction

Currently, vision based object tracking has become an important task within the field of computer vision. The availability of high quality and low-priced video cameras, and the increasing need for automated video analysis has gained a great interest in object tracking algorithms [16]. One of the pertinent applications of object tracking is in vision based hand gesture recognition systems. In vision based hand gesture, the detection and tracking of the human hands are the foundation steps of the recognition system, where hands must be localized in every image sequence. The effectiveness of hand tracking greatly depends on its ability to function re-

liably in real time so that an instant inter- action would be achievable. On the other hand, to guarantee more accessibility, such tracking system should not require user to wear special clothes or cumbersome devices such as colored markers. Moreover, in vision based, hand tracking is a challenging task due to difficulty to deal with variations in its appearance mainly caused by shape and pose change, rapid and erratic motion, illumination changes and occlusion problem. Therefore, by solving these issues, it will directly enhance the performance of the whole gesture system [11].

In the last decade, various approaches to detect and track hand gesture have been proposed. Color-based technique is a straightforward and simple approach that is often utilized to detect skin color regions in the image [2, 8, 10]. Although it is very popular, it has some drawbacks. First, any skin colored objects (includes face, wall, clothes, etc.) will remain in the image. Second, there are cases where other skin color regions are occluded, which can hamper accurate detection. For example, Koh et al. [10] tracked hand gesture using a skin color information based on hand appearance model which considers both shape and color information using Active Appearance Model (AAM). During initialization, the system verifies the user's hand and the appearance model using Mahalanobis distance, and then parametrically constructs the skin color model using Gaussian distribution. Although the system is robust to various illumination conditions, it is not adapted to the illumination changing while the system runs. Moreover, the system restricted to a particular viewpoint and difficult to adapt with non-rigid appearance that varies greatly during natural hand motion. In [2, 8] the moving hand is tracked by computing blobs and hand location is predicted using Kalman Filter. In their work, they assumed that the process and measurement noise are Gaussian and hand is moving in constant velocity. However, this assumption restricts the hand gesture movement in natural ways. Cheng et al. [3] have developed a real-time hand gesture tracking using a combination of motion, skin and edge detection. When there are other skin color objects moving rapidly, they used movement justification and background subtraction to absorb the problem. They managed to detect and track hand under complex background assuming that the background is uniform and hand moves in constant speed. In their method, the edge is difficult to be identified when the hand moves fast due to image ghosting. Additionally, the background modeling is hard to achieve when the

scene is exposed to illumination effect and contain a large amount of changes. In [1], hand detection is achieved using a fusion of skin color and motion features. A rectangular ROI is then defined to represent the detected hand location. The hand tracking is achieved using ROI based tracker, which utilizes the distributions of skin and motion pixels which are obtained from the ROI edges. They managed to track the hand under cluttered environment with a promising detection rate at real-time speed, by assuming the difference between frame interval is relatively small and no unrelated skin colored objects move close to the ROI.

In 2006, Porikli et al. [12] proposed a method to track non-rigid object using covariance matrix representation [14] and model update mechanism using a mean manifold. In their approach, they represent the target object with a set of features as a covariance matrix and manage to capture the correlation of spatial and statistical properties between the same representations. At each consecutive frame, they search the whole image to find a candidate region that has a covariance matrix which is most similar to the target object. The model update mechanism effectively adapts the object deformations and appearance changes. Even though the covariance tracker efficiently detects the non-rigid objects, the tracking performance decreases when the background and target object has a very less color variation. Moreover, most of the computational power in the covariance tracker is spent to compare the target model with the candidate regions from the whole image which makes the system hard to achieve a real time performance.

In this paper, we propose a real-time hand tracking system using the ROI based tracker based on the fusion of skin and motion features. In this approach, we advocate the efficient integration of region descriptor covariance matrix [14] into the ROI based tracking framework in order to gain a more robust tracker. We also introduce the temporal frame difference into the covariance feature mapping to reduce the tendency of fusing the background pixels into the region descriptor. By incorporating the covariance matrix, the ROI based tracker has a specific reference model of the object being tracked. This is to efficiently reduce the drifting problem when the hand is moving fast, erratic, and when there are unwanted skin colored objects move close to the ROI. In our proposed method, we shift the previous hand ROI position to the nearest neighborhood location based on previous detection position to obtain a set of candidate regions rather than to search from the whole image. Therefore, we save a lot of the computational power used to compare the target model with the candidate regions. For complete flow of the hand gesture tracking system, we refer the readers to Fig.(1).

The remaining of this paper is organized as follow. In sections 2 and 3, we describe the detail of the tracking algorithm. In section 4, we describe the detail experimental results to evaluate the robustness and the efficiency of the proposed algorithm. Finally we conclude the contribution of this paper and identify areas for future development in section 5.

2 Initialization Stage

2.1 Skin Segmentation

In this work, we extract skin pixels using YC_bC_r pixels distribution. YC_bC_r color space is derived in such way that the illumination component is concentrated only in Y com-

Figure 1. Flow chart of the hand gesture tracking system.

ponent while color is contained in C_b and C_r chrominance components. The illumination can vary significantly over the skin region under different lighting effect thus difficult to select range of skin color values. Therefore, to reduce illumination effect, only chrominance components are used to select range of skin color value. Fig.(2) shows the histogram of C_b and C_r components. Since the histograms are likely similar to normal distributions, the mean and standard deviation are taken into account for skin segmentation as shown in Eq.(1) where, μ_{cb} and μ_{cr} are overall mean and σ_{cb} and σ_{cr} standard deviation of C_b and C_r components respectively. The result of skin regions segmentation, $S_t(x, y)$ is depicted in Fig.(3).

$$S_t(x, y) = \begin{cases} 1, & \text{if } \begin{cases} \mu_{cr} - \sigma_{cr} < range\_cr < \mu_{cr} + \sigma_{cr} \\ \mu_{cb} - \sigma_{cb} < range\_cb < \mu_{cb} + \sigma_{cb} \end{cases} \\ 0, & \text{Otherwise} \end{cases} \quad (1)$$

Figure 2. Histogram of C_b and C_r components.

Figure 3. Skin Color Regions Segmentation.

2.2 Motion Segmentation

When object moves in spatial space, motion information can be extracted by examining the local grey level change. To detect the moving object, the simplest technique is to use frame differencing method because it has great detection speed. Let $F_t(x, y)$ be the t^{th} RGB image sequence and $Fd_t(x, y)$ is the frame difference between t^{th} and $(t - 1)^{th}$ frame defined in Eq.(2). The frame difference image is then converted to grey level image using Eq.(3), where RGB are the elements of $Fd_t(x, y)$. The binary image, $Fdb_t(x, y)$ of the frame difference is obtained using Eq.(4), where T is a threshold value. In this procedure, the threshold value is obtained experimentally and determined as $T = 0.05\bar{X}$, where $\bar{X}$ is the average pixels value of grey level image. The result of motion segmentation is depicted in Fig.(4).

$$Fd_t(x, y) = |F_t(x, y) - F_{t-1}(x, y)| \quad (2)$$

$$Fdg_t(x, y) = 0.299R + 0.587G + 0.114B \quad (3)$$

$$Fdb_t(x, y) = \begin{cases} 1, & Fdg_t(x, y) \div 255 \geq T \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

Figure 4. Motion Segmentation.

Figure 5. Hand Region Extraction.

2.3 Defining Hand ROI

After segmenting the frames using skin and motion features, the hand region can be located by combining these two features using logical AND operator. By combining these two features, face regions and other skin colored objects that remains after skin segmentation are discarded. The combination of skin and motion image $SM_t(x, y)$, are defined in Eq.(5) and Eq.(6), respectively, where $f_m[M, N]$ is a median and $f[i, j]$ is a 3×3 masking window used to remove the unwanted impulse noise pixels remained in the $M \times N$ image. The hand ROI, which is defined as a rectangular bounding box around the binary image $SM_t(x, y)$, can be obtained using Eq. (7), where $[C]$ and $[R]$ are vectors that contains horizontal and vertical coordinates extracted from white pixels of $SM_t(x, y)$ binary image. The result of hand region extraction is depicted in Fig.(5).

$$SM_t(x, y) = (S_t(x, y) \& Fdb_t(x, y)) * f_m[M, N] \quad (5)$$

$$f_m[M, N] = \text{median}\{f[i, j]\} \quad (6)$$

$$ROI_t(x, y) = \text{rectangle}(SM_t(x, y), [m, n, w, h]) \quad (7)$$

where, $w = \max[C] - \min[C]$, $h = \max[R] - \min[R]$, $m = \min[C]$ and $n = \min[R]$

3 Tracking Stage

3.1 ROI Based Tracker

Once the hand has been localized, the skin and motion pixels available around the hand ROI are used as important information to track the new hand location in the successive frame. Base on the information from $ROI_t(x, y)$ in Eq.(7), the skin and motion pixels are extracted by scanning around top, left and right corners of the ROI as in the following equations

$$right_t(i, j) = \text{rectangle}(SM_t(x, y), [p, n, 0.2w, h]) \quad (8)$$

$$left_t(i, j) = \text{rectangle}(SM_t(x, y), [q, n, 0.2w, h]) \quad (9)$$

$$top_t(i, j) = \text{rectangle}(SM_t(x, y), [m, r, w, 0.4h]) \quad (10)$$

where $p = m - 0.2w$, $q = \max[C]$ and $r = n - 0.2h$. In this work, the dimensional size of the abovementioned ROIs are determined empirically. The amount of white pixels available from each ROI can provide useful information to update new hand position, which can be obtained from the Eq. (11), (12) and (13).

$$sumright = \sum_{j=1}^j \sum_{i=1}^i right_t(i, j) \quad (11)$$

$$sumleft = \sum_{j=1}^j \sum_{i=1}^i left_t(i, j) \quad (12)$$

$$sumtop = \sum_{j=1}^j \sum_{i=1}^i top_t(i, j) \quad (13)$$

The ROI based tracking algorithm is then constructed as indicated in Table 1, where the minimum and maximum values of row and column coordinate from the scanning regions are used to calculate the right, left, bottom and top ROI displacement. The illustration of the ROI tracking procedure is depicted in Fig.(6).

Table 1. ROI Based Tracking Algorithm.

```

If ( sumright > sumleft )
    find the right displacement and shift the ROI to the right
elseif ( sumleft > sumright )
    find the left displacement and shift the ROI to the left
else
    keep the previous ROI column coordinate
end
If ( sumtop > 0 ) % shift the ROI either to top or bottom
    If ( the min row coordinate < half of the topt(i, j) height )
        find the top displacement and shift ROI to the top
    elseif ( the min row coordinate > half of the topt(i, j) height )
        find the bottom displacement and shift ROI to the bottom
    end
else
    keep the previous ROI row coordinate
end

```

Figure 6. Illustration of ROI Based Tracking Algorithm from step interval t to $t + 20$. Right, left and top scanning regions are illustrated in yellow, blue and red bounding box, respectively.

3.2 Hand Region Covariance Matrix

Tracking the hand using the raw pixel value from skin and motion information is not sufficient especially when hand is moving very fast or other skin region is moving rapidly near the hand ROI. As illustrated in Fig.(6), ROI Based tracking algorithm tracks the moving hand by only utilizing the information extracted near the edges of the moving region without considering the region inside the ROI that contains a lot of useful appearance attributes such as the intensity, color and frame difference values. To improve the ROI Based tracking algorithm, we incorporate a covariance matrix [14] of a set of features extracted from ROI window as a region descriptor to represent the target object. In the consecutive frame, we construct another covariance matrix corresponding to the new ROI position and compute the distance descriptor covariance matrix (DDCM) from the target object to verify the new ROI position, so that it always similar with the target object. Let $R_t(x, y)$ be the $W \times H \times d$ dimensional features image extracted from $ROI_t(x, y)$ as a function of feature mapping, Φ as illustrated in Eq. 14,

$$R_t(x, y) = \Phi(ROI_t(x, y)) \quad (14)$$

where, the function Φ can be any mapping such as intensity image, RGB color image, frame difference values, etc. We arrange the features vector which includes the spatial and appearance attributes which directly relate the pixels coordinate as described in Eq.15

$$\mathbf{F}_{(x,y)} = [\mathbf{f}^1 \ \mathbf{f}^2 \ \mathbf{f}^3 \ \mathbf{f}^4 \ \dots \ \mathbf{f}^d]^T \quad (15)$$

where, $\mathbf{F}_{(x,y)}$ is the d -dimensional features vector inside $R_t(x, y)$. In this work, we select 10 features ($d = 10$) where, $\mathbf{f}^1$ and $\mathbf{f}^2$ are the pixel locations in x and y , $\mathbf{f}^3$ is the intensity image obtained from Eq.(3), $\mathbf{f}^4$, $\mathbf{f}^5$, $\mathbf{f}^6$ and $\mathbf{f}^7$ are the first

Table 2. Example of features vector arrangement if the ROI size is 28×44 .

$\mathbf{f}^1$	1	2	3	4	.	.	.	26	27	28
$\mathbf{f}^2$	1	1	1	1	.	.	.	44	44	44
$\mathbf{f}^3$	170	184	176	181	.	.	.	52	39	24
$\mathbf{f}^4$	170	170	16	7	.	.	.	7	10	28
$\mathbf{f}^5$	170	174	186	181	.	.	.	4	16	18
$\mathbf{f}^6$	170	166	8	17	.	.	.	1	16	2
$\mathbf{f}^7$	170	174	186	181	.	.	.	0	4	6
$\mathbf{f}^8$	208	212	241	240	.	.	.	40	33	33
$\mathbf{f}^9$	188	192	201	200	.	.	.	29	31	31
$\mathbf{f}^{10}$	174	178	161	160	.	.	.	14	25	25

Table Note :

$\mathbf{f}^1 = x$ location, $\mathbf{f}^2 = y$ location, $\mathbf{f}^3 = I(x, y)$

$\mathbf{f}^4 = \left| \frac{\partial I(x, y)}{\partial x} \right|$, $\mathbf{f}^5 = \left| \frac{\partial I(x, y)}{\partial y} \right|$, $\mathbf{f}^6 = \left| \frac{\partial^2 I(x, y)}{\partial x^2} \right|$, $\mathbf{f}^7 = \left| \frac{\partial^2 I(x, y)}{\partial y^2} \right|$

$\mathbf{f}^8 = R(x, y)$, $\mathbf{f}^9 = G(x, y)$, $\mathbf{f}^{10} = B(x, y)$

and second derivatives of the intensity image with respect to x and y , $\mathbf{f}^8$, $\mathbf{f}^9$ and $\mathbf{f}^{10}$ are the RGB color values extracted from $ROI_t(x, y)$, respectively. The first and second derivative images are calculated through the filters $[-1 \ 0 \ 1]^T$ and $[-1 \ 2 \ -1]^T$ respectively. Although the variance of the pixel locations (x, y) is the same for all the regions of the same size, they are still important since the correlation with the other features are used at the non-diagonal entries of the covariance matrix [14]. The detail arrangement for a sample of feature vectors extracted from $R_t(x, y)$ region with 28×44 window size and concatenated into rows vectors is depicted in Table 2.

From Table 2, consider that we have rows entries of $\mathbf{f}^1$ and $\mathbf{f}^2$, which are denoted as vectors $\mathbf{X}$ and $\mathbf{Y}$ respectively, where $\mathbf{X}$ and $\mathbf{Y}$ are random variables, each with finite variance, then the covariance of two random variables, $\mathbf{X}$ and $\mathbf{Y}$, can be mathematically represented in Eq. (17).

$$\mathbf{X} = [x_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_{n-1} \ x_n]^T \quad (16)$$

$$\mathbf{Y} = [y_1 \ y_2 \ y_3 \ y_4 \ \dots \ y_{n-1} \ y_n]^T$$

$$Cov(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^n [(x_i - \bar{x})(y_i - \bar{y})] \quad (17)$$

where, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$. This relationship can be generalized as a multivariate situation to relate a set of d -dimensional features inside $W \times H$ window size of $R_t(x, y)$ using a covariance matrix form as described in Eq. (18)

$$Cov_{R_t} = \frac{1}{WH} \sum_{i=1}^{WH} (\mathbf{f}_i - \mu_{R_t})(\mathbf{f}_i - \mu_{R_t})^T \quad (18)$$

where, Cov_{R_t} is the $d \times d$ covariance matrix of $R_t(x, y)$ and μ_{R_t} is the mean vector corresponding to the features point $\mathbf{f}$ inside $R_t(x, y)$.

3.3 Distance Descriptor Covariance Matrix

In every consecutive frame, we construct another covariance matrix corresponding to the new ROI position $R_{t+1}(x, y)$ and denote as $Cov_{R_{t+1}}$. We compare the new covariance matrix, $Cov_{R_{t+1}}$ with the target covariance matrix Cov_{R_t} , by measuring the distance descriptor covariance matrix (DDCM) between each other. The distance measure between the two matrixes is not suitable to be measured using an arithmetic subtraction because the space of covariance matrix is not a vector space [14]. Therefore, we use the distance measure proposed in [6] which uses the sum of squared

algorithms of generalized eigenvalues to measure the dissimilarity of two covariance matrixes as describe in Eq.(19)

$$DDCM(Cov_{R_t}, Cov_{R_{t+1}}) = \sqrt{\sum_{d=1}^n \ln^2 \lambda_d(Cov_{R_t}, Cov_{R_{t+1}})} \quad (19)$$

where, $\{\lambda_d(Cov_{R_t}, Cov_{R_{t+1}})\}_{d=1, \dots, n}$ are the generalize eigenvalues of $Cov_{R_{t+1}}$ and Cov_{R_t} which are obtained from Eq.(20)

$$(Cov_{R_t})x_d = \lambda_d(Cov_{R_{t+1}})x_d \quad (20)$$

where, $\{x_d\}_{d=1, \dots, n}$ are the generalize eigenvectors. The ROI position is adjusted when the DDCM is not satisfying for a certain threshold value, which can be obtained from a set of experiments. The examples of the DDCM measurement are illustrated in Fig.(7). To adjust the ROI position, we shift the ROI window around nearest neighbor to obtain a set of candidate regions. At each candidate regions, we compute its covariance matrix and its distance covariance matrix about the target object and select the region which has the smallest distance as the estimated new ROI position. Then, the covariance matrix is updated corresponding to the previous estimated selected candidate region. Fig.(8) shows the condition where the ROI position of the successive frame needs to be adjusted when the distance covariance matrix from the target object is exceeding the threshold value, T_{DDCM} . The red bounding box in Fig.(8b) represents the neighboring region to obtain a set of candidate regions to estimate the correct ROI position based on the smallest distance covariance matrix.

Figure 7. DDCM measure for a set of difference ROI region.

Figure 8. Situation where ROI position is adjusted when the covariance matrix distance is not satisfying T_{DDCM} value. (a) ROI position set as target object, (b) ROI position needs to be adjusted. The red bounding box represents nearest neighbor to obtain candidate region.

4 Result And Discussion

The experiments were executed using MATLAB 7 which performed on Intel Core Duo at 2GB RAM, under Windows 7 operating system. The frames sequences were captured at 30fps using low price USB camera with a resolution of 352×288 . The camera focused mainly upper part of the gesturer body and the actor was not restricted to wear either short or long sleeved clothing. During the initialization, the

Figure 9. From top to bottom; Gestures performed by six different individuals for trajectory “1”, “0”, “2”, “S”, “C” and “9”.

actor makes significant movement using his hand and during this process, other part of the body is allowed to move in small scale. Our experiments only consider on tracking one hand because our main focus is to extract the hand motion trajectory which will be used as trajectory based hand gesture recognition system.

We prepared the databases in two groups, where the first group is used to evaluate tracking performance for trajectory based gestures and the second group is to evaluate tracking performance under several real live scenarios. We consider that the estimated hand position is accurate if it is fall within 9×9 neighborhood of the ground truth target object centre position, where the tracking rate is defined as the ratio of successful estimated frames over total frames. Since each individual has different ways of performing the same gestures and there are also variations in skin color, we performed experiment that was carried out by six different individuals on six types of gesture using the databases from the first group. The results are listed in Table 3, where the average tracking rate is 95.43 % with the tracking speed at 42fps on average. The samples tracking results are illustrated in Fig.(9).

To further demonstrate the robustness and efficiency of the proposed algorithm, we also performed another series of experiments to track hand movement in several real live tracking scenarios. We used the databases from the second group and the experiment results are summarized in Table 4. In Fig.(10), we observed that the proposed algorithm can successfully track the moving hand under the cluttered environment (Scene A) that included other skin colored objects. This

Table 3. Hand gesture tracking rate of six difference person for six types of gesture.

		Gesture Type (Trajectory Based)					
Person		"1"	"0"	"2"	"S"	"C"	"9"
1	TF	184	180	199	172	185	169
	TR (%)	91.85	97.22	94.47	97.09	97.3	95.26
	TS (fps)	43	43	43	43	44	43
2	TF	196	241	233	229	240	229
	TR (%)	88.27	98.76	94.42	96.51	91.67	93.01
	TS (fps)	42	42	41	41	42	41
3	TF	238	243	243	243	240	241
	TR (%)	98.31	98.76	97.94	97.53	98.75	96.68
	TS (fps)	42	42	41	41	41	42
4	TF	243	227	241	243	235	241
	TR (%)	97.94	97.8	95.85	93.82	95.24	97.66
	TS (fps)	40	42	42	43	42	41
5	TF	166	202	229	241	243	197
	TR (%)	93.97	97.52	96.94	97.92	97.94	97.46
	TS (fps)	43	44	43	43	43	43
6	TF	286	287	284	277	290	281
	TR (%)	93.71	87.8	89.43	92.78	91.38	96.44
	TS (fps)	42	41	42	43	41	42

Table Note : TF = Total Frames, TR = Tracking Rate,
TS = Tracking Speed

is expected because in our tracking procedures, we use the fusions of skin and motion features in ROI Based tracking algorithm which is able to discard the unrelated skin colored object that remains in the image.

In addition to this experiment, we tested the proposed algorithm in handling the situation when left hand moved rapidly near the ROI and it occluded with right hand (Scene B). In this case, the right hand is the object of interest. When both of the moving hands exhibit both motion and color features, the ROI Based tracking intents to drift to the wrong position as it does not have a significant reference model of the object being tracked. By incorporating the covariance matrix to represent the target object, the occlusion between both hands was successfully absorbed to prevent the hand ROI to drift to the wrong position as illustrated in Fig.(11). Since the moving hand can undergo non-rigid transformation, the target covariance matrix is updated in every step interval to adapt to this situation. We observed that the covariance update mechanism has improved the ROI Based algorithm to track the non-rigid hand shapes and appearances (Scene C) as shown in Fig.(12). The proposed method also does not make any assumption on the motion of the tracked object. Therefore, it helps to improve the tracking rate by reducing the drifting problems during unpredictable hand motion. As shown in Fig.(13), the proposed tracking algorithm has the ability to solve the tracking complexity when the hand is moving fast and rapidly changed its direction (Scene D). The covariance matrix also has an advantage property, in which it is invariant to the mean changes such as identical shifting of color values as described in [12]. As such, it enhances the tracking ability when the hand moves under different illumination effect (Scene E) as illustrated in Fig.(14).

As a qualitative benchmark, we analyzed the performance of our proposed algorithm by comparing with Covariance tracker [12], Kalman Filter based tracker [2] and ROI based tracker [1]. We tested these three algorithms on image sequence of Scene F as shown in Fig.(15). As illustrated from trajectory plot, in the case when the hand moves fast and erratically changed direction, the ROI based tracker easily fails its tracking state. On the other hand, the Kalman Filter based tracker provides a very smooth tracking trajectory, but there

Table 4. Hand tracking under real live scenarios.

Scene	TF	TR (%)	TS (fps)
Scene A: Moving under cluttered environment	219	97.72	42
Scene B: Occlusion with other moving skin regions	369	97.28	42
Scene C: Rapidly change shape and appearance	240	97.23	43
Scene D: Rapidly change speed and direction	292	94.88	41
Scene E: Moving under different illumination effect	292	94.63	42
Scene F: Moving fast, rapidly change direction and occlusion with face region	292	88.01	42
Scene G: Long video sequence under complex environment	737	95.93	42

Table Note : TF = Total Frames, TR = Tracking Rate,
TS = Tracking Speed

are drifting time occur when the hand rapidly changed its direction, therefore the tracking failed at this point. This is expected because it was learnt that the Kalman Filter is difficult to deal with non-linear stochastic process estimation. In the case when the hand occluded with skin color regions, the target model in the Covariance tracker tends to resemble with background pixels. When this situation occurs, the model update strategy proposed in [12] has a tendency to contaminate with the unrelated background pixels which hampers a correct detection. In our procedures, the occlusion problem is addressed by utilizing the motion features in ROI based tracker to guide the covariance matrix to search a candidate regions based on previous ROI location, thus reducing the false detection. Although we observed that the Covariance tracker provides a close comparable tracking performance with our proposed method, it is far from real time implementation because it is high computational cost as compared to other algorithms as depicted in Fig.(16) because most computational power is used to search the candidate regions from the whole image. Finally, we tested our procedure on a long video sequence captured under complex environment (Scene G) with the trial duration more than 25 seconds to evaluate the long term stability. The trajectory plot and sample of the tracking images are illustrated in Fig.(17). From the trajectory plot, we observed that the proposed procedure gives the best result as compared to others methods.

Figure 10. Hand motion tracking performed on Scene A **Figure 11.** Hand motion tracking performed on Scene B

Figure 12. Hand motion tracking performed on Scene C

Figure 13. Hand motion tracking performed on Scene D

Figure 14. Hand motion tracking performed on Scene E

5 Conclusion and Future Development

In this work, a method of hand gesture tracking system using ROI based method incorporated with covariance matrix is proposed. In ROI based method, the extraction of skin and motion information only focuses around the ROI to speedup the computation time and others unrelated moving objects are easily discarded. Covariance matrix framework introduced in ROI based algorithm offers the ability to deal with noisy problem by adjusting ROI position by selecting the best candidates region as a correct estimated position. The experimental results demonstrate the robustness and validity of the proposed algorithm with average tracking rate is above 95% at speed of 42fps on average. In the future, we will work to improve the trajectory smoothness by incorporating the proposed algorithm with Adaptive Kalman Filter to extract the trajectory and then integrate with gesture recognition engine to produce a meaningful HCI system.

Figure 15. Trajectory plot and sample tracking results of proposed algorithm (blue rectangle), Covariance tracker (red rectangle), ROI based tracker (green rectangle) and Kalman Filter based tracker (purple rectangle) performed on Scene F.

Figure 16. Comparison of computational time of proposed algorithm with Covariance tracker, ROI based tracker and Kalman Filter based tracker performed on Scene F.

Figure 17. Trajectory plot and sample tracking results of proposed algorithm (blue rectangle), Covariance tracker (red rectangle), ROI based tracker (green rectangle) and Kalman Filter based tracker (purple rectangle) for long term stability evaluation on Scene G.

Acknowledgements

This work was supported by the USM Research University - Postgraduate Research Grant Scheme (RU-PRGS), (1001/PELECT/8034030) and the USM Postgraduate Incentive Grant, (1001/PELECT/8021023).

REFERENCES

- [1] Asaari MSM, Suandi SA (2010) Fusion of motion and color cue for hand detection and tracking. World Engineering Congress pp 240–245
- [2] Binh N, Shuichi E, Ejima T (2005) Real time hand tracking and gesture recognition system. International Conference on Graphic, Vision and Image Processing pp 111–115
- [3] Chen F, Fu C, Huang C (2003) Hand gesture recognition using a real-time tracking method and hidden markov models. Image Vision Computing 21(8):745–758
- [4] Comaniciu D, Ramesh V, Meer P (2004) Real-time tracking of non-rigid objects using mean shift. IEEE Conference on Computer Vision and Pattern Recognition pp 142–149
- [5] Elmezain M, Al-Hamadi A, Niese R, Michaelis B (2010) A robust method for hand tracking using mean-shift algorithm and kalman filter in stereo color image sequences. International Journal of Information Technology 6(1):24–28
- [6] Forstner W, Moonen B (1999) A metric for covariance matrices. Technical Report, Dept. of Geodesy and Geoinformatics, Stuttgart University
- [7] Hee S, Gregorij K, Ruzena B (2008) Hand tracking and motion detection from the sequence of stereo color image frames. International Conference on Industrial Technology pp 1–6
- [8] Imagawa K, Lu S, Igi S (1998) Color based hands tracking system for sign language recognition. International Conference on Face and Gesture Recognition pp 462–467
- [9] Isard M, Blake A (1998) Conditional density propagation for visual tracking. International Journal of Computer Vision 29(1):5–28
- [10] Koh E, Won J, Bae C (2009) On-premise skin color modeling method for vision-based hand tracking. International Symposium on Consumer Electronics pp 908–909
- [11] McAllister G, McKennan S, Ricketts I (2002) Hand tracking for behavior understanding. Image and Vision Computing 20(12):827–840
- [12] Porikli F, Tuzel O, Meer P (2006) Covariance tracking using model update based on means on riemannian manifolds. IEEE Conference on Computer Vision and Pattern Recognition pp 728–735
- [13] Shan C, Tan T, Wei Y (2007) Real-time hand tracking using a mean shift embedded particle filter. Pattern Recognition 40(7):1958–1970
- [14] Tuzel O, Porikli F, Meer P (2006) Region covariance: A fast descriptor for detection and classification. European Conference on Computer Vision pp 589–600
- [15] Utsumi A, Tetsutani N, Igi S (2002) Hand detection and tracking using pixel value distribution model for multiple-camera-based gesture interactions. Conference on Knowledge Media Networking pp 31–36
- [16] Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. Acm Computing Surveys (CSUR) 38(4):13
- [17] Zhai H, Wu X, Han H (2005) Research of a real time hand tracking algorithm. International Conference on Neural Networks and Brain pp 1233–1235
- [18] Zhang Q, Zhang M, Hu J (2009) Hand gesture contour tracking based on skin color probability and state estimation model. Journal of Multimedia 4(6):349–355