

Multi touch Gesture Generation and Recognition Techniques: A Study

Mr. Ramdas Pandurang Bagawade¹, Ms. Pournima Akash Chavan², Ms. Kajal Kantilal Jadhav³

Asst. Prof, Computer Engineering, PES's College of Engineering, Phaltan, India¹

B.E. Scholar, Computer Engineering, PES's College of Engineering, Phaltan, India^{2,3}

Abstract: A huge number of users are using smart phones to communicate with each other. A smart phone user is exposed to various threats when they use their phone for communication. These threats can disorganization the operation of the smart phone, and transmit or modify user data rather than original[1]. So applications must guarantee privacy and integrity of the information. Single touch mobile security is unable to give efficient performance for confidential data. Hence we are moving towards multitouch mobile security for high security. In computing, multi-touch is authentication technology that enables a surface to recognize the presence of more than one touch points of contact with the touch screen [2]. By using multiple touch points to authenticate user for access confidential data in mobile phones. we are presenting our study about biometric gestures to authenticate user through multitouch finger points for more security[1].

Keywords: Multitouch, biometric gesture, authentication, security, smart phone Finger-tracking, Android Operating system.

I. INTRODUCTION

Today's IT admins face the troublesome task of managing the unnumerable amounts of mobile devices that connect with enterprise networks a day for communication through network. Securing mobile devices has become increasingly important now days as the numbers of the devices in operation and the uses to which they are put have expanded in world wide. The problem is compounded within the enterprise as the ongoing trend toward IT users or organizations is resulting in more and many more employee-owned devices connecting to the corporate internet. Authentication is a nothing but process in which the credentials provided are compared to those on file in a database of valid users' information on a operating system. If the credentials match, the process is completed and the user is granted authorization for access to the system. The permissions and folders came backoutlineeach the surroundings the user sees and also themethod he willmove with it, as well asthe amount of access and different rights comparable tothe number of allottedcupboard space and different services [1].

The generally a computer authentication process is to use alphanumerical usernames or textbased and passwords. This method has been shown to have some disadvantages. For example, users tend to pick passwords that can be easily guessed and recognized by other hard to remember. To device this problem, some researchers have developed authentication techniques that use multitouch biometric gesture as passwords for authentication.

Multi-touch, in a computing environment, is an interface technology that enables input gestures on multiple points

on the surface of a device. Although most generally used with touch screen devices on handheld devices, such as smart phones and tablets, and othermulti-touch has been used for other surfaces as well, including touch pads and whiteboards, tables and walls[2].

In other words, multi-touch refers to the capability of a touch screen (or a touchpad) to recognize two or more points of contact on the surface simultaneously. The constant following of the multiple points permits the portable interface to acknowledge gestures, that modify advanced practicality similar to pinch-to-zoom, pinch. wherever gesture recognition is much of deciphering human gestures via mathematical algorithms. Gestures will originate from any bodily motion howevernormallyoriginate from the face or hand and alternative human biometric gestures but the identification and recognition of posture, and human behaviours is additionallythe topic of gesture recognition techniques.

We used Equal Error Rate (EER) [2] to measure accuracy. This is the rate at which False Acceptance Rate (FAR) [2] and False Rejection Rate (FRR). To find out whether using multiple gestures would improvethesystem's performance, we combined scores of 2 differentgestures from the same user in the same order and evaluatedthe EER of the combined gestures as:

$$FAR = \frac{\text{Wrongly accepted individuals}}{\text{Total number of wrong matching}}$$

$$FRR = \frac{\text{Wrongly rejected individuals}}{\text{Total number of correct matching}}$$



II. DEVELOPING A GESTURE AUTHENTICATION TECHNIQUE [1]

Biometric systems are an effective way to authenticate valid users generally based on the “something they are” property in mobile authentication. The goal of biometric identification is that the automatic verification of identity of a living person by proving over some distinctive gestures that solely he possesses in authentication method.

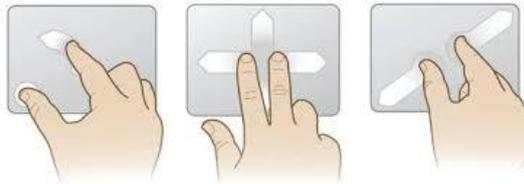


Figure 1: Multi touch behaviour[2]

The biometric authentication system has two phases: **enrollment** phase and **authentication** phase. If new user must first record his secret hand signs at the first enrollment phase to the system. The process is performing the hand signs at the user’s discreet choice with sufficient space for hand movement during registration phase.

III. GESTURE TAXONOMY [1]

1. Parallel: All fingertips are moving in the same direction during the gesture. For example, a bush swipe, during which all 5 fingers move from left to right the screen.
2. Closed: If all fingertips are moving inward toward the center of the hand. For example, a pinch gesture.
3. Opened: All fingertips are moving outward from the center of the hand. For example, a reverse pinch gesture.
4. Circular: All fingertips are rotating around the center of the hand. For example, a clockwise or counterclockwise rotation[1].



Figure 1: Single touch[1]

IV. MATCHING TOUCH SEQUENCES TO SPECIFIC FINGERS

a. Hidden Markov Models [3]

Hidden Markov Models (HMMs) are statistical models and simplest versions of dynamic Bayesian Networks, where the system being modelled is a Markov process with an unobserved state. It is a collection of finite states connected by transitions, much like Bayesian Networks.

Each state has two probabilities: a transition probability, and an output probability distribution. Parameters of the model are determined by training data[4][5].

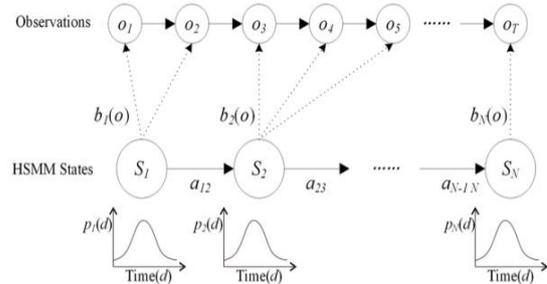


Figure 2: Hidden Markov Models [3]

hidden states, as well as N-dimensional observable symbols.



Figure 3: Multi touch Movement [1]

The conventional HMM is expressed as the following [4]. HMM is the mathematical tool to model signals, objects ... that have the temporal structure and follow the Markov process. HMM can be described compactly as $\lambda = (A, B, \pi)$ (Figure 4b) where,

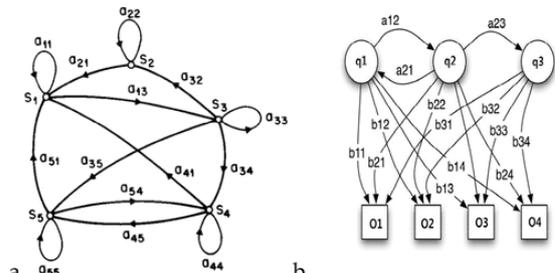


Figure 4: Conventional Hidden Markov Model [4]

$A = \{a_{ij}\}$: the state transition matrix

$$a_{ij} = P[q_{t+1}=s_j | q_t=s_i],$$

$$1 \leq i \leq N, 1 \leq j \leq N,$$

$$1 \leq i \leq N$$

$B = \{b_i(k)\}$: the observation symbol probability distribution

$$b_j(k) = P[O_t=v_k | q_t=s_j],$$

$$1 \leq j \leq N,$$

$$1 \leq k \leq M, b_j(k) = P[O_t=v_k | q_t=s_j],$$

$$1 \leq j \leq N, 1 \leq k \leq M$$



$\pi = \{\pi_i\}$: the initial state distribution

$\pi_i = P[q_1 = s_i] \pi_i = P[q_1 = s_i]$

1. Set of states: $S = \{s_1, s_2, \dots, s_N\}$
2. State at time t : q_t
3. Set of symbols: $V = \{v_1, v_2, \dots, v_M\}$

Given the observation sequence $O = O_1 O_2 \dots O_T$ and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O | \lambda)$, i.e., the probability of the observation sequence given the model.

Now let us consider following two states:

1. Training: based on the input data sequences $\{O\}$, we calculate and adjust $\lambda = \bar{\lambda}$ to maximize likelihood $P(O | \lambda)$
2. Recognizing: based on $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ for each class, we can then assign the class in which the likelihood $P(O | \lambda)$ is maximized.

The observation symbol probability distribution $P[O_t = v_k | q_t = s_i]$ can be discrete symbols or continuous variables. If the observations are different symbols.

$$B(i, k) = P(O_t = k | q_t = s_i)$$

$$B(i, k) = P(O_t = k | q_t = s_i)$$

If the observations are vectors in R^L , it is common to represent $P[O_t | q_t]$ as a Gaussian:

$$P[O_t = y | q_t = s_i] = N(y; \mu_i, \Sigma_i)$$

$$P[O_t = y | q_t = s_i] = N(y; \mu_i, \Sigma_i)$$

$$N(y; \mu, \Sigma) = \frac{1}{(2\pi)^{L/2} |\Sigma|^{1/2}} \exp[-\frac{1}{2}(y-\mu)^T \Sigma^{-1} (y-\mu)]$$

$$N(y; \mu, \Sigma) = \frac{1}{(2\pi)^{L/2} |\Sigma|^{1/2}} \exp[-\frac{1}{2}(y-\mu)^T \Sigma^{-1} (y-\mu)]$$

A more flexible representation is a mixture of M Gaussians:

$$P[O_t = y | q_t = s_i] = \sum_{m=1}^M P(M_t = m | q_t = s_i) \times N(y; \mu_{m,i}, \Sigma_{m,i})$$

$$P[O_t = y | q_t = s_i] = \sum_{m=1}^M P(M_t = m | q_t = s_i) \times N(y; \mu_{m,i}, \Sigma_{m,i})$$

where M_t is a hidden variable that specifies which mixture component to use and $P(M_t = m | q_t = s_i) = C(i, m)$ is the conditional prior weight of each mixture component. In our approach, we both implement continuous and discrete output variable distribution for 1st and 2nd HMM stages respectively [3][6].

a. Dynamic Time Warping

Dynamic TimeWarping (DTW) is an algorithm that compares two different sequences that may possibly vary in time. For example, if two video clips of different people walking a particular path were compared, the DTW algorithm would detect the similarities in the walking pattern, despite walking speed differences, accelerations or decelerations. [3][7]

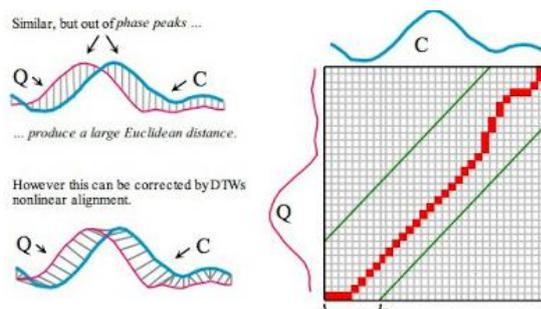


Figure 4: Dynamic time warping[3]

The algorithm begins with a set of template streams, describing each gesture available in the system database. This results in high computation time, and hence, limitations in recognition speed. Additionally, the storing of many templates for each gesture results in costly space usage on a resource-constrained device.

Consider a training set of N sequences $fS_1; S_2; \dots; S_N$, where each S_g represents sample of the same gesture class. Then, each sequence S_g composed by a set of feature vectors at each time t , $S_g = fsg_1; \dots; sgl_{Lg}$ for a certain gesture category, where L_g is the length in frames of sequence S_g . Let us assume that sequences are ordered according to their length, so that $L_{g1} \leq L_g \leq L_{g+1}$; $8g \in [2; \dots; N]$, the median length sequence is $S = S_{dN/2}$. This sequence S is used as a reference and the rest of sequences are aligned with it using the classical Dynamic Time Warping with Euclidean distance [4], in order to avoid the temporal deformations of various samples from an equivalent gesture class. Therefore, once the alignment method, all sequences have length $L_{dN/2}$.

We define the set of warped sequences as $\sim S = f \sim S_1; \sim S_2; \dots; \sim S_N$. Consider a training set of N sequences $fS_1; S_2; \dots; S_N$, where each S_g represents a sample of the same gesture class. Then, each sequence S_g composed by a set of feature vectors at each time t , $S_g = fsg_1; \dots; sgl_{Lg}$ for a certain gesture category, where L_g is the length in frames of sequence S_g . Let us consider that sequences are ordered according to their length, so that $L_{g1} \leq L_g \leq L_{g+1}$; $8g \in [2; \dots; N]$, the median length sequence is $S = S_{dN/2}$ [4].

$$P(q, \lambda) = \sum_{k=1}^M \alpha_k \cdot P(q)_k,$$

$$P(q)_k = e^{-\frac{1}{2}(q-\mu_k)^T \cdot \Sigma_k^{-1} \cdot (q-\mu_k)},$$

This sequence S is used as a reference, and the remaining of sequences are assigned with it using the classical Dynamic TimeWarping with Euclidean distance [3], in order to remove the temporal deformations of different samples from the same gesture category. Hence, after the alignment process, all sequences have length $L_{dN/2}$. We define the set of warped sequences as $\sim S = f \sim S_1; \sim S_2; \dots; \sim S_N$ [3].

Input: A gesture $C = \{c_1, \dots, c_n\}$ with corresponding GMM model $\lambda = \{\lambda_1, \dots, \lambda_m\}$, its similarity threshold value μ , and the testing separate $Q = \{q_1, \dots, q_n\}$, Cost Matrix M is defined, where $N(x)$, $x = (i, t)$ is the set of three upper-left location of x in M .

Output: Working path of the detected gesture, if any.

```
//Initialization
for i=1:m do
    for j=1:n do
        M(i,j)=v
    end
end
```

```

for j=1:v do
M(0,j)=0
end
for t=0:v do
for i=1:m do
x=(i,j)
M(x)=D(qi,λi)+minχ'∈N(χ)M(χ')
End
end
if m(m,t)< then
W={argminχ'∈N(χ)M(χ')}
Return
End
end [4]

```

b. Artificial Neural Networks

Artificial Neural Networks (ANNs) are networks of weighted, directed graphs where the nodes are artificial neurons, and the directed edges are connections between them. The most common ANN structure is the feed forward Multi-Layer Perceptron. Feed forward means that the signals only travels one way through the net[4][8]. For input pattern p , the i -th input layer node holds $x_{p,i}$.

Net input to j -th node in hidden layer:

$$net_j^{(1)} = \sum_{i=0}^n w_{j,i}^{(1,0)} x_{p,i}$$

Now Output of j -th node in hidden layer:

$$x_{p,j}^{(1)} = S\left(\sum_{i=0}^n w_{j,i}^{(1,0)} x_{p,i}\right)$$

Then Net input to k -th node in output layer:

$$net_k^{(2)} = \sum_j w_{k,j}^{(2,1)} x_{p,j}^{(1)}$$

Finally Output of k -th node in output layer:

$$o_{p,k} = S\left(\sum_j w_{k,j}^{(2,1)} x_{p,j}^{(1)}\right)$$

Network error for p :

$$E_p = \sum_{k=1}^K (l_{p,k})^2 = \sum_{k=1}^K (d_{p,k} - o_{p,k})^2$$

Neurons are arranged in layer wise, with the outputs of each neuron in the same layer being connected to the inputs of the neurons in that layer. Finally, the output layer neurons are assigned a value. Each output layer neuron show the particular class of gesture, and the record is assigned to however class's neuron has the highest value During training, the gesture class for each neuron in the output layer is known, and the nodes can be assigned the "correct" value.

Critical Analysis

A critical analysis based on the results achieved by is shown in this section. ANNs, HMMs, and DTW algorithms were implemented on a mobile phone, and measured in performance according to recognition speed, accuracy and time needed to train[3]. Since Bayesian Networks are a super class of HMMs which have been tweaked towards gesture classification, they are not considered. Thus according to recognition, accuracy and training time we can say that DTW gives better performance as compare to HMM and ANN. These results are summarized below:

Table 1: Comparison between different algorithms [3]

No.	Algorith m	Recogn- ition Speed	Accura cy	Trainin g Time
1	HMMs	10.5ms	95.25%	Long
2	ANNs	23ms	90%	Medium
3	DTW	8ms	95.25%	No Training

V. FINGER TRACKING

Firstly we need adjust finger tracking parameters, that's why we need to activate the calibration in the tab in on-screen display [5][9].

a. Projection Signatures:

Projection signatures are performed directly on the resulting threshold binary image of the hand [5]. The core process of this algorithm is consists of adding the binary pixels row by row along a diagonal (the vertical in this case). Previous knowledge of the hand angle is therefore required. A low-pass filter is applied on the signature (row sums) in order to reduce low frequency variations that create many local maxima and cause the problem of multiple positives (more than one detection per fingertip). The five maxima thereby obtained correspond to the position of the five fingers.

b. Geometric Properties:

The second algorithm is based on the geometric properties and, as shown on line 3 of figure 5, uses a contour image of the hand on which a reference point is set. This point can be determined either by finding the centre of mass of the contour (barycenter or centroid) or by fixing a point on the wrist [6].



Figure 5: Hand Movement [5]

Euclidean distances from that point to every contour point are then computed, with the five resulting maxima assumed to correspond to the finger ends [5]. The minima



can be used to determine the intersections between fingers (finger valleys). The geometric algorithm also required filtering in order to reduce the problem of multiple positives.

c. Circular Hough Transform:

The circular Hough transform is applied on the contour image of the hand but could as well be performed on an edge image with complex background if no elements of the image exhibit the circular shape of the fingertip radius. This can be done efficiently for finger ends by eliminating points that are found outside the contour image. The inconvenient is that the set of discard points contains a mix of finger valleys and false positive that cannot be sorted easily [5].

d. Color Markers:

While the three previous algorithms rely only on the hand characteristics to find and track the fingers, the marker algorithm tracks color markers attached to the main joints of the fingers. Each color is tracked individually using colour segmentation and filtering [5].

This permits the identification of the different hand segments. The marker colors should therefore be easy to track and should not affect the threshold, edge or contour image of the hand. Respecting these constraints makes it possible to apply all algorithms to the same video images and therefore to compare each algorithm degree of accuracy and precision with respect to the markers [5].

Comparisons:

Table 2: Comparison between different techniques [5]

Properties	Projecti on Signatur e	Geometr ic Properti es	Circular Hough Trans-form	Colo r Mak ers
Locates fingers	Good	Good	Good	Good
Locates fingertips	Poor	Normal	Normal	Good
Locates finger ends and valleys	Poor	Good	Good	Good
Work with complex background	Poor	Good	Normal	Good
Precision	Good	Good	Good	Good
Accuracy	Poor	Good	Good	Good

All the presented algorithms have succeeded, in various degrees, in detecting each finger. The projection signatures algorithm can only roughly identify a finger, but the circular Hough transform and geometric properties algorithms can find both finger intersections and finger end points, it is important to note that in the case where finger are folded, the end points don't correspond to the fingertips[5].

CONCLUSION

In this paper we have studied gesture authentication techniques for multi touch finger points such as pinch, zoom, expand, swap up, rotate. We have also studied parallel, circular, open and close gesture taxonomies.

For performing matching touch sequences to specific figures we have studied three techniques such as HMM, DTW and ANN, out of which DTW is best as per the results shown in table1.

For figure tracing we have studied four techniques such as projection signatures, geometric properties, circular hough transform and color markers out of which color markers is best as per the results shown in table2.

REFERENCES

- [1] KalyaniDevidas: DushmanAndroid Software based Multi-touch Gestures Recognition for Secure Biometric Modality
- [2] Memon, K. Isbister, N. Sae-Bae, N. and K. Ahmed, "Multitouch gesture based authentication," IEEE Trans. Inf. Forensics Security, vol. 9, no. 4, pp. 568–582, Apr. 2014
- [3] Methods for Multi-touch Gesture Recognition:Daniel Wood
- [4] <http://journals.sagepub.com/doi/full/10.5772/50204>
- [5] Finger Tracking Methods Using EyesWebAnne-Marie Burns1 and Barbara Mazzarino2
- [6] https://www.cse.buffalo.edu/~jcorso/t/CSE555/files/lecture_hmm.pdf
- [7] DWT:Probability-based Dynamic Time Warping and Bag-of-Visual -and-Depth-Words for Human Gesture Recognition
- [8] https://en.wikipedia.org/wiki/Artificial_neural_network
- [9] <http://whatis.techtarget.com/definition/gesture-recognition>

BIOGRAPHIES



Prof. Ramdas Pandurang Bagawade, Received the B.E. degree in Computer Engineering from Vidya Pratishans College of Engineering, Baramati in 2008. Also completed M.E. Degree in 2013 from Pune University. Presently he is working in the Department of Computer Engineering at PES's COE Phaltan as an Assistant Professor



Miss Pournima Akash Chavan, Presently she is doing her Bachelor of Engineering from PES's College of engineering, Phaltan



Miss Kajal Kantilal Jadhav, Presently she is doing her Bachelor of Engineering from PES's College of engineering, Phaltan