Fingertip Detection for Hand Pose Recognition

M.K. Bhuyan, Debanga Raj Neog and Mithun Kumar Kar
Department of Electronics and Electrical Engineering,
Indian Institute of Technology Guwahati, India.
E Mail:{mkb, debanga, mithun}@iitg.ernet.in

***Abstract*— In this paper, a novel algorithm is proposed for fingertip detection and finger type recognition. The algorithm is applied for locating fingertips in hand region extracted by Bayesian rule based skin color segmentation. Morphological operations are performed in the segmented hand region by observing key geometric features. A probabilistic modeling of the geometric features of finger movement has made the finger type recognition process significantly robust. Proposed method can be employed in a variety of applications like sign language recognition and human robot interactions.**

Keywords- Hand Posture, Human Computer Interaction, Hand Modelling.

I. INTRODUCTION

In Human Computer Interaction (HCI), gesture based interface gives a new direction towards the creation of a natural and user friendly environment. Recently in HCI, the detection of finger and finger types has received growing attention in applications like sign language, vision based finger guessing games and in applications related to real-time systems and virtual reality and recognizing pointing gestures in the context of human–robot interaction [1].

In applications that use hand gesture input, the user's hands are be used to direct the focus in the scene, requiring that the position and pose of the hands be determined; for example, the user can point to, something, which can be described in words using three dimensional computer audio. Changing the hand posture could indicate that some activity should take place. A gesture, in fact, can be defined as a sequence of hand postures, or a change from one posture to another. This frequently translates to a very few simple mouse actions or menu selections, in the context of a user interface. The most complicated part of dynamic hand gesture recognition is the sign language recognition with both local and global motion [11].Many methods for hand pose detection use skin color-based hand detection and take geometrical features for hand modeling.

The block diagram as shown in Fig. 1 summarizes the proposed algorithms, in which a novel method is proposed for fingertip detection and finger type recognition. These two parameters can be effectively used for static hand pose recognition. In the proposed method, hand is first separated from the forearm and some key geometric features of pre-defined gestures are obtained by hand calibration. Gaussian distribution is considered to include the variations in the finger positions in a gesture. The input gesture is recognized by using the minimum distance classifier. The proposed scheme is described in more details in the sections to follow.

II. RELATED WORK

As in HCI, the detection of finger and finger types are very important in applications like sign language, vision based finger guessing games and in applications related to real-time systems and virtual reality and recognizing pointing gestures in the context of human–robot interaction. The detection of finger and finger types is still challenging.

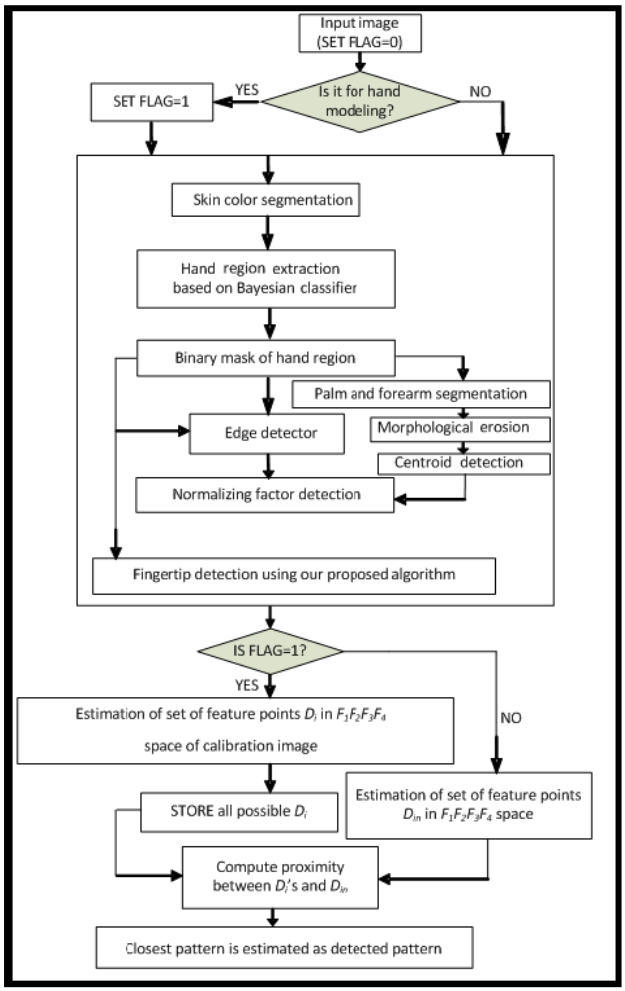

Figure 1: Block diagram of our proposed algorithms

Many methods for fingertip detection uses skin color based hand detection and morphological operations for hand segmentation and appearance based skeleton models [15], [6]. Sung Kwan Kang *et al.* proposed a method for fingertip detection by using contour extraction and morphological operations to detect finger blobs [13]. A similar idea about fingertip recognition based on fingertip detection and tracking appeared in [11]. Sigal*et al.* uses a classifier incorporating Bayesian decision theory for skin detection [4]. They processed real-time skin segmentation in video sequences. They used skin color histogram over time, which are dynamically updated based on a feedback from the current segmentation. In our work we apply Bayesian classifier model and used training images database provided by Jones and Rehg [2].

Kerdvibulvech*et al.* proposed a method for detecting the positions of fingertips of a hand playing guitar [16]. They used a set of Gabor filter kernels to compute a lower dimensional representation of image and applied artificial neural network to the resulting feature vector. They estimated the fingertip positions by using an LLM network. Dung Duc Nguyen *et al.* proposed a method for fingertip detection from a stereo camera by using grayscale morphology and blob analysis [6]. They extracted the skin region using disparity image information. But their accuracy of detection was affected by using cluttered background. Some authors used curvature measurement and shape analysis for finding the fingertip locations [5], [10].

Finger detection techniques that use model based analysis of hand posture were introduced in [3], [6]. Here authors modeled the hand image to analyze the hand postures. They modeled the hand with skeleton structure with considering physiological constraints. They take many constraints related to finger joint movements, joint angle limits and movement types, flexion of joints and adduction and abduction of metacarpophalangeal (MP) joints [3]. Matsumoto *et al.* used the similar idea based on skeletal hand model [10]. They used the multi viewpoint camera system for the observation of hand poses. They used a voxel model with estimation algorithm for estimating different hand poses. However accuracy and speed give constraints to their method.

Finger detection techniques that do not use the skeletal model were introduced in [9], [5], [11]. Yu Sun *et al.* estimated the fingertip force directions by imaging the color patterns in the fingernails and surrounding skin [9]. Barrho*et al.* proposed a method for the fingertip detection problem in different manner [15]. They used generalized Hough transform (GHT) to match a fingertip template with a hand image and a probabilistic model for localization of fingertip. The result has some false detection since there are many regions which have the same shape as the fingertip. Parker *et a* proposed a method for hand pose and gesture recognition by using geometrical properties of hand by detecting fingertips and shape analysis with histogram matching for hand pose recognition [5]. They have not considered the abduction and adduction movement of fingers, which may be very crucial in practical applications.

In an attempt to overcome some of the above mentioned difficulties in recognizing hand posture, we propose a model-based method for modeling a gesturing hand. A novel method is proposed for fingertip detection and finger type recognition. These two parameters can be effectively used for static hand pose recognition.

III. PROPOSED HAND POSE DETECTION PROCESS

A. Skin Segmentation

Here we find the skin color distribution by analyzing the color histogram of the images in RGB color space. This is done by two class classifiers. The prior histograms used for classification are pre-computed using the offline database provided by Jones and Rehg [2]. The conditional probabilities of foreground (fg) and background (bg) are respectively $P(fg/rgb)$ and $P(bg/rgb)$. We give some threshold T to distinguish the classification boundary based on the criteria that the ratio of $P(fg/rgb)$ and $P(bg/rgb)$ exceeds T, which is based on a relative risk factor associated with misconception.

$$T < \frac{P(fg/rgb)}{P(bg/rgb)} = \frac{P(rgb/fg)P(fg)}{P(rgb/bg)P(bg)} \tag{1}$$

If $P(fg)$ is the probability of an arbitrary pixel in an image being skin.
We can write:

$$T \times \frac{1-P(fg)}{P(fg)} < \frac{P(rgb/fg)}{P(rgb/bg)} \tag{2}$$

Where $P(fg)$ =0.09 in the training database.

With given $P(fg)$ the threshold T can be empirically found by computing "Receiver Operating Characteristics" (ROC curve) over a collection of 100 images [2]. Here a threshold was chosen such that 85% correct classification is achieved while having 25% chance of false alarm. The choice of the threshold is made by the fact that the optimum value of the threshold lies near the bend of the ROC curve. In our experiment the value of T is computed as 0.06.The result of pixel classification gives a binary mask, in which '0's correspond to background pixels and '1's correspond to foreground pixels. We used maximum area criteria to eliminate noise effects, while extracting the hand.

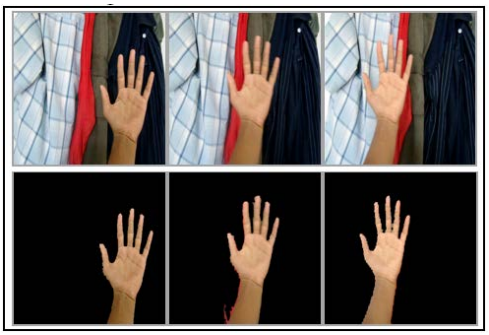

Figure 2: Skin color segmentation

B. Detection of Centroid

We have obtained the mask of the segmented hand from the skin color segmentation approach. One of the most relevant information that we can extract from the mask corresponding to the segmented hand is the centroid of the mask. To reduce the uncertainty in the centroid position due to the change in finger position, we computed the centroid from an eroded image of the mask for extracting only the palm region of the hand. The centroid $(\overline{c_x}, \overline{c_y})$ is computed by the following equation:

$$\overline{c_x} = \frac{\sum_{\forall x_I \in S} x_I}{N}, \overline{c_y} = \frac{\sum_{\forall y_I \in S} y_I}{N} \tag{3}$$

where, *S* is the skin region in the mask, *N* is the total number of pixels in the skin region

C. Normalization Constant

To remove the effect of the relative distance of the hand from the camera there is a necessity for normalizing the features viz. distance of the segmented hand. We first computed the edge of the mask of segmented hand by using canny edge detector. The normalization constant related to overall hand size is then computed by the following equation:

$$N_0 = \frac{\sum_{(x_e, y_e) \in E} \sqrt{(x_e - \overline{c_x})^2 + (y_e - \overline{c_y})^2}}{M} \tag{4}$$

Where, *E* is the set of all pixels belonging to the edge, *M* is the total number of pixels in the set *E*.

D. Determination of Orientation

The orientation of the hand can be estimated by using the information of position of fingertips and centroid with respect to the image frame. Orientation is defined as the angle of axis of the least moment of inertia. It is obtained by minimizing $I(\theta)$ with respect to θ ,

$$I(\theta) = \sum_{(m,n)} \sum_{\in R} [(P_{i,y} - \overline{c_y}) \cos\theta - (P_{i,x} - \overline{c_x}) \sin\theta]^2 \tag{5}$$

That gives,

$$\theta = \frac{1}{2} \tan^{-1} \theta \left[\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right] \tag{6}$$

Where,

$$\mu_{p,q} = \sum_{(m,n)} \sum_{\in R} (P_{i,x} - \overline{c_x})^p (P_{i,y} - \overline{c_y})^q \tag{7}$$

$(P_{i,x}, P_{i,y})$ *and* $(\overline{c_x}, \overline{c_y})$ are the positions of fingertips and the centroid of the hand palm region respectively. For each segmented hand region we can calculate the orientation and construct a reference line R_{ref} .

E. Hand and Forearm Segmentation

The thickness of contour of the segmented skin region is used for separating the hand from the forearm region. We determine the two intersection points of the orientation vector with the already computed rectangle boundary. The middle points of the line connecting the centroid and each intersection point is computed. Assuming that the wrist position lies within these two points, we could find out the thickness of the hand contour in the perpendicular direction of the orientation vector along the line joining the middle points. The region where the thickness is minimum is considered as the wrist region, which is shown in Fig. 3 [14].

Figure3: Forearm segmentation

IV. PROPOSED HAND CALIBRATION

A. Extraction of GeometricFeatures

One possible approach for detection of corners in the segmented hands, the location in the curvature where the curvature can be unbounded can be used. It is also expected that the fingertips of the hand are also included among those corners. But if due of some error in segmentation, the boundary of segmented hand does not become smooth, there can be very large number of corners detected and extraction of fingertips will become very complicated. That is why we have proposed a new way to extract the fingertips.

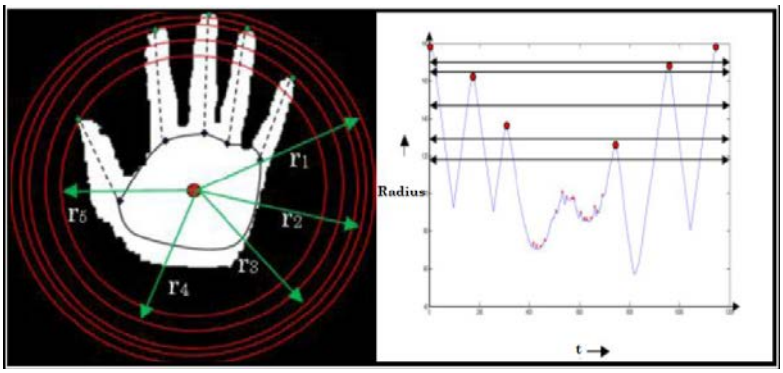

Figure 4: Proposed algorithm for fingertip detection

We first plot the distance of hand contour, $r(t)$ from the center of mass $(\overline{c_x}, \overline{c_y})$ by traversing the contour in a cyclic manner as shown in the Fig. 4. Since the axis of each finger is actually not passing through the center of mass, the peaks detected are actually not exactly at the fingertip positions, but this can give us a good approximate estimation for initializing our algorithm. Our fingertip detection algorithm can be summarized as given below:

Step 1:

Show the five fingers of the hand in front of the camera for calibration. The hand should be perpendicular to the camera and fully stretched for this.

Step 2:

(a)Find the most distant point in the hand contour from the centroid and construct a circle about the centroid. Decrease the radius of the circle in optimal step size till five peaks of the graph are eliminated. Consider the radii corresponding to the five peaks to be r_1, r_2, r_3, r_4 and r_5. The minimum radius r_5 is also denoted as $r_{\min}$. Name the peaks as p_1, p_2, p_3, p_4 and p_5 and corresponding contour traversal parameter t of these peaks as t_1, t_2, t_3, t_4 and t_5 according to the sequence of peak elimination. The whole process is illustrated in Fig. 4.

(b)Select r_A, r_B, r_C, r_D and r_E such that $r_1 < r_A < r_2, r_2 < r_B < r_3$, $r_3 < r_C < r_4$, $r_4 < r_D < r_5$, $r_5 < r_E < r_5 - \delta$; where δ is a safety margin. Here, δ is set as $\delta = \frac{r_{\min}}{3}$.

Step 3:

Compute six sets of points in the hand contour by finding six sets of solutions of the equations $r(t) = r_A, r(t) = r_B, r(t) = r_C, r(t) = r_D$, $r(t) = r_E$ and $r(t) = r_5 - \delta$. Let us consider the solution sets to be S_1, S_2, S_3, S_4 , S_5 and S_6. For every set $S_{i, i\in\{1,\ldots,6\}}$, the point $s \in S_i$ belongs to the finger corresponding to the peak p_j , if

$$j = \arg(\min_{k=1:5}(|s - t_k|)) \tag{8}$$

Step4:

Construct five axes for all finger by joining *s* values belonging to each $p_{j,\, j\in\{1,\ldots,5\}}$ .The intersection points of these axes with the eroded hand mask are R_1, R_2, R_3, R_4 *and* R_5. Consider these points as the point of rotation the fingers.

Step5:
Now for recognizing the type of fingers in the hand, if the left hand is shown for calibration, classify the peaks from left to right with respect to the reference axis R_{ref} as T (Thumb or Finger I), I (Index or Finger II), M (Middle or Finger III), R (Ring or Finger IV) and L (Little or Finger V).

Step6.
The actual intersection of the axes and the hand contour will give the actual positions of fingertips P_T, P_I, P_M, P_R *and* P_L .

B. Hand Modeling

Here a model-based approach is proposed for mathematical modeling of hand image. This skeletal model can model both the physical structure and physiological constraints of hand. The human hand consists of 5 fingers, each of which contains three joints. Except for the thumb, there are two degrees of freedom for MP joints and one degree of freedom for proximal interphalangeal (PIP) joints and distal interphalangeal (DIP) joints [13]. Taking into account all degrees of freedom for each joint and also considering global hand pose, the human hand has roughly 27degrees of freedom. Human hand joints can be classified as flexion, twist, directive, or spherical according to the type of movement or possible rotation axes [3]. The hand model is shown in Fig. 5.

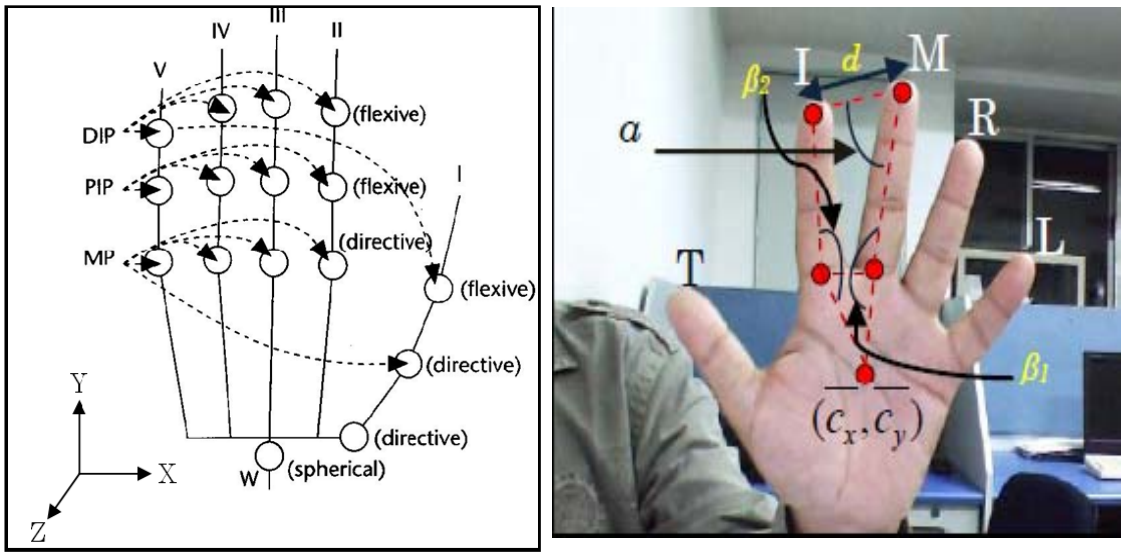

Figure 5: Hand model showing angle of rotation of fingers and a normal stretched hand with all the fingers named as T, I, M, R and L

The movement of a particular segment of the hand is produced by rotations of its proximal joint and can be specified by the joint's rotation angles. Here we define local coordinate systems on every joint position and represent any joint rotation by a sequence of rotations occurring around the three axes of the local coordinate system.

Physical structure
Several assumptions were made for mathematical modeling of hand with respect to hand structure.

- The hand remains perpendicular to camera so that so that the line joining centroid and fingertip of middle finger lies parallel to Y axis as shown in Fig. 5.
- Each finger from II to V was constructed from 3 straight line segments. The first segment starts from the intersection point of middle lines of each finger with the morphological cross section of palm area. Here we take these cross section points as MP joints of the finger. Next segments are joints by MP-PIP joints, PIP-DIP joints and DIP and fingertip. Here we are assuming only the adduction and abduction of the metacarpophalangeal (MP) joints. The axis of abduction and adduction motion is referenced along Y axis. Here we are not considering the flexion of the interphalangeal (PIP) joints and metacarpophalangeal (MP) joints.
- The thumb is modeled separately, because of the MP joints of the thumb has two degree of freedoms with large rotation angle and the bending direction of the thumb is different from the other fingers. We assume that the line joining the MP joint with the thumb fingertip is straight and it has only abduction and adduction movement about Z axis.

Physiological Constrain

- The abduction or adduction angle between the adjacent fingers from II to V for static constraints are taken as $-15^o \le Q^z_{MP,s} \le 15^o$, where Z gives the axis of rotation to the adduction or abduction movement in local joint centered coordinate system.MP denotes for the metacarpals (palm bones) Here *s* denotes the number of finger from II to V.
- The four fingers are planar manipulators with the exception of the MP joint. The MP joint of finger III displays limited adduction and abduction motion [3]. The total abduction and adduction angle movement for thumb is also taken as 30^0.

V. PROPOSED RECOGNITION ALGPRITHM

The fingers of a normal stretched hand is named as T, I, M, R and L for thumb, index, middle, ring and little fingers respectively. The positions of the MP joints are computed by the previously proposed skeletal hand model. Figure 5 shows the hand model constructed by I and M fingers. Subsequently, ten such models are constructed from a hand by considering all pairs of fingertips and the corresponding MP joints. For each model, we can compute 4 key features $\alpha_{i,j} d_{i,j}, \beta_{1,i,j}$ and $\beta_{2,i,j}$. Considering a 4D feature space $F_1F_2F_3F_4$, all possible 10 models can be mapped to the $F_1F_2F_3F_4$ space.

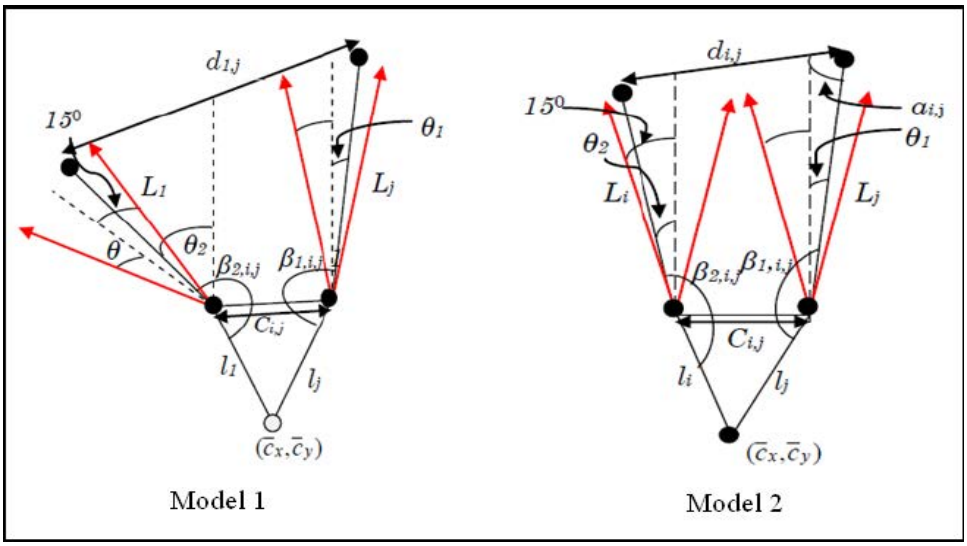

Figure 6: Modeling of abduction and adduction angles about MP joints for Model-1 and Model-2 of the proposed algorithm

Furthermore, the proposed algorithm can be used for distinguishing different hand gestures shown by two, three and four fingers. For gestures with single finger, template matching is done for finger type detection. Five finger gesture is a trivial case, where all the five fingers will be present. Table I shows the finger combinations we have considered as valid gestures for our experiments.

TABLE 1
ACCEPTABLE FINGER COMBINATIONS

Number of fingers	Number of all possible combinations	Finger combinations considered
2	10	(T,I),(I,M),(I,L)
3	10	(T,I,M),(I,M,R),(T,I,L)
4	5	(T,I,M,L),(I,M,R,L)

These 8 patterns will map to $F_1F_2F_3F_4$ space into 8 sets $D_{k(k=1:8)}$. For the selection of the gesture patterns, the proximity between the patterns in the $F_1F_2F_3F_4$ space is considered. We selected those patterns which are well separated in $F_1F_2F_3F_4$ space as well as easy to gesticulate. We are also defining $d_{avg}, \alpha_{avg}, \beta_{1,avg}$ and $\beta_{2,avg}$ for a particular pair of fingers which represent average values of $d, \alpha, \beta_{1,}$ and β_2 respectively. Figure 4 shows all these parameters for gesture modeling. Based on the analysis of hand structure, we can analyze the gesture

classes based on the variation of abduction and adduction angles of each finger. For unique orientation of the finger axis of thumb, the gestures incorporating thumb are analyzed separately. The reference axis for thumb is defined as the line joining the centroid and the estimated MP joint of the finger. For other fingers, the reference axis is the line parallel to the middle finger axis and passing through the MP joint of respective fingers.
Table II shows an approximate variation of abduction and adduction angle of each finger in different gesture classes.We have proposed two models for analyzing possible variations in the features d , α , β_1 an β_2 in a pair of fingers as shown in Fig. 5.

- Model 1: All pair of fingers including thumb and
- Model 2: All pair of fingers not including thumb.

We considered L_1 , L_2 , L_3 , L_4 and L_5 are the lengths of the thumb, index, middle, ring and little finger respectively. The red arrows signify the possible range of abduction and adduction angle for a particular finger. Here, θ is the offset angle from the reference axis of the thumb. Again, θ_1 and θ_2 are abduction and adduction angle for a particular gesture. Now, the features are computed from the two models by using Euclidean geometry as: Now, the features are computed from the two models by using Euclidean geometry as

$$d_{i,j} = \sqrt{L_i^{\,2} + C_{i,j}^{\,2} + 2L_iC_{i,j}\sin(\theta_2) + Lj^2 - 2L_j\sqrt{L_i^{\,2} + C_{i,j}^{\,2} + 2L_iC_{i,j}\sin(\theta_2)} \times \cos(90 + \theta_2 - \sin^{-1}(\frac{L_i\sin(\theta_2)}{\sqrt{L_i^{\,2} + C_{i,j} + 2L_iC_{i,j}\sin(\theta_2)}}))} \quad (9)$$

$$\alpha_{i,j} = \sin^{-1}(\frac{\sqrt{L_i^{\,2} + C_{i,j} + 2L_iC_{i,j}\sin(\theta_2)}}{d_{i,j}}\cos(\theta_1 - \sin^{-1}(\frac{L_i\cos(\theta_2)}{\sqrt{L_i^{\,2} + C_{i,j} + 2L_iC_{i,j}\sin(\theta_2)}}))) \quad (10)$$

$$\beta_{1,i,j} = 90^0 + \theta_1 + \cos^{-1}(\frac{l_j^2 + C_{i,j}^2 - l_i^2}{2l_jC_{i,j}}) \quad (11)$$

$$\beta_{2,i,j} = 90^0 + \theta_1 + \cos^{-1}(\frac{l_i^2 + C_{i,j}^2 - l_j^2}{2l_iC_{i,j}}) \quad (12)$$

Finger indexed with 1 to 5 represents thumb, index, middle, ring and little finger respectively.

- $d_{i,j}$: the d-feature of the $i_{th,}$ and j_{th} fingers.
- $\alpha_{i,j}$: the α -feature of the $i_{th,}$ and j_{th} fingers.
- $\beta_{1,i,j}$: β_1 -feature of the $i_{th,}$ and j_{th} fingers.
- $\beta_{2,i,j}$: β_2 -feature of the $i_{th,}$ and j_{th} fingers.
- L_i : the length of $i_{th,}$ finger (i=1:5).
- $C_{i,j}$: the distance between MP joints $i_{th,}$ and j_{th} fingers

TABLE 2
RANGE OF ABDUCTION AND ADDUCTION ANGLE FOR DIFFERENT CLASSES OF GESTURES

Possible Gestures	Range of abduction and adduction angles in degree (Mean angle is shown in brackets)				
	T	I	M	R	L
TI	θ to θ +30^0 (θ +15^0)	-15^0 to 15^0 (0)			
IM		-15^0 to 0^0 (-7.5^0)	0^0 to 15^0 (7.5^0)		
IL		-15^0 to 15^0 (0)			-15^0 to 15^0 (0)
TIM	θ to θ +30^0 (θ +15^0)	-15^0 to 0^0 (-7.5^0)	0^0 to 15^0 (7.5^0)		
IMR		-15^0 to 0^0 (-7.5^0)	0^0 (0)	0^0 to 15^0 (7.5^0)	
TIL	θ to θ +30^0 (θ +15^0)	-15^0 to 15^0 (0)			-15^0 to 15^0 (0)
TIML	θ to θ +30^0 (θ +15^0)	-15^0 to 0^0 (-7.5)	0^0 to 15^0 (7.5^0)		-15^0 to 15^0 (0)
IMRL		-15^0 to 0^0 (-7.5^0)	0^0 (0)	0^0 to 15^0 (7.5^0)	-15^0 to 15^0 (0)

As mentioned earlier, $D_{k(k=1:8)}$ is the set of points in $F_1F_2F_3F_4$ plane of gesture classes $G_{j(j=1:8)}$. For every gesture class with N fingertips, there will be $M = \binom{N}{2}$ points in $F_1F_2F_3F_4$ plane. Due to uncertainty of abduction and adduction angles, we have modeled 4D Gaussian distribution for every point in $F_1F_2F_3F_4$ plane.

Let us consider the pair of index and middle finger for representation. The average of four features d, α, β_1 and β_2 are computed by putting the mean angles in θ_1 and θ_2 as $\theta_1 = 7.5^0$ and $\theta_2 = -7.5^0$ in equations (9), (10), (11) and (12). Let us consider the average values, which will be considered as the mean of the distributions, are $d_{avg}, \alpha_{avg}, \beta_{1,avg}$ and $\beta_{2,avg}$. The maximum possible ranges of the four features are computed by putting the ranges of θ_1 and θ_2 as, $\theta_1 \in [0^0, 15^0]$ and

The ranges obtained for each of the features are calculated as, $\Delta d, \Delta\alpha, \Delta\beta_1$ and $\Delta\beta_2$, which will be used for computing the variances of the features.

Now, we can define a 4D-normal distribution of j_{th} cluster of k_{th} gesture pattern with features $(d_{k,j}, \alpha_{k,j}, \beta_{1,k,j}, \beta_{2,k,j})$ as:

$$f(d_{k,j}, \alpha_{k,j}, \beta_{1,k,j}, \beta_{2,k,j}) = e^{-\left(\frac{(d_{k,j}-d_{avg})^2}{2(\Delta d/2)^2} + \frac{(\alpha_{k,j}-\alpha_{avg})^2}{2(\Delta\alpha/2)^2} + \frac{(\beta_{1,k,j}-\beta_{1,avg})^2}{2(\Delta\beta_1/2)^2} + \frac{(\beta_{2,k,j}-\beta_{2,avg})^2}{2(\Delta\beta_2/2)^2}\right)} \quad (13)$$

We determine all distributions $D_{k(k=1:8)}$ for 8 valid gesture classes during the calibration. The set can be expressed as follows:

$$D_k = \{f(d_{k,j}, \alpha_{k,j}, \beta_{1,k,j}, \beta_{2,k,j})\}_{j=1:M} \quad (14)$$

where,

$$M = \begin{cases} 1; & for\ 2\ fingertips \\ 3; & for\ 3\ fingertips \\ 6; & for\ 4\ fingertips \end{cases} \quad (15)$$

Let us consider $\overline{f}_{in}$ is the vector consisting of the fingertip positions and MP joints positions of the input gesture. As depicted in Fig. 4 a circle centered at centroid is initially constructed such that the circle will enclose all the peaks of the contour distance profile, which is measured by computing the distance of the hand contour from the centroid in a cyclic manner. The circle is uniformly shrunk to final position with radius equal to $r_{min} - \delta$, in which the contour distance profile will just exclude all the valid peaks $i.e.$ fingertips of the profile. This process will exclude all unwanted small peaks (artifacts) in the palm boundary, which may be generated due to the inherent constraints of skin color based segmentation. The input gesture pattern $\overline{f}_{in}$ is then transformed to D_{in} in the $F_1F_2F_3F$ space. Next, a proximity measure is used for computing the proximity of D_{in} from the pre modeled gesture pattern distribution $D_{k(k=1:8)}$. Mahalanobis distance is used for computing the proximity between D_{in} and $D_{k(k=1:8)}$ and the corresponding gesture pattern is recognized on the basis of minimum distance criteria.

VI. EXPERIMENTAL RESULTS

In order to validate the performance of the proposed algorithm, we have taken 8 gestures which are shown in Fig. 7. For recognizing all the static gestures, all the experiments are executed in the normal illumination condition. The experiments have been tested on a set of 20 users and each user has performed a predefined set of 8 gestures. Therefore, we have 160 gestures to evaluate the performance of the designed system. To avoid the difficulty of matching image features between different views, the user's hand was fixed perpendicular to camera and only the rotation of hand about the axis perpendicular to palm is considered.

TABLE 3
PERCENTAGE ACCURACY OF GESTURE PATTERN RECOGNITION

Gesture Patterns	Accuracy in%
TI	92
IM	93
IL	94
TIM	93
IMR	95
TIL	93
TIML	92
IMRL	95
Total	93.37

The results of fingertips detection are shown in Fig. 7. The estimated centroid is also shown along with the detected fingertips. Table 3 summarizes the percentage accuracy of the proposed gesture recognition system. As it can be observed in Table 3, the system has a high recognition rate of different finger types, which could reach up to 95% for some special cases. This high recognition shows the efficacy of the proposed system for possible deployment for the HCI-based application.

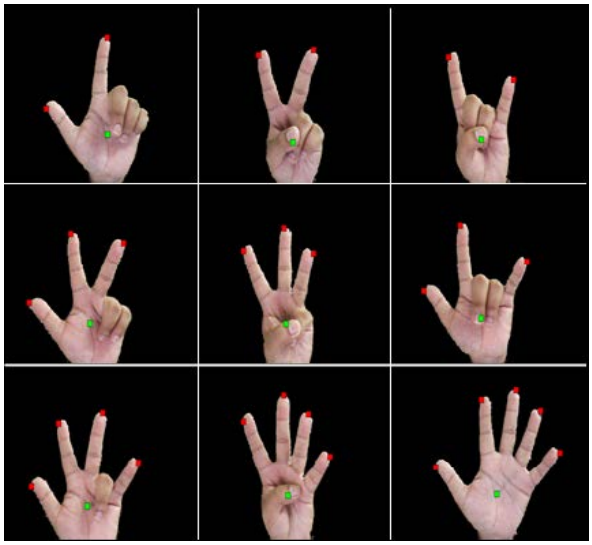

Figure 7: Detected fingertips of 8 gesture patterns

VII. CONCLUSION

Most of the traditional appearance based methods have not probabilistically modeled the fingers for hand pose detection. In this paper, a probabilistic model of abduction and adduction movements of fingers has been proposed. Our proposed method is less sensitiveness towards the inaccurate skin color segmentation in fingertip detection process. Hand calibration process has made it more robust against size inconsistency between the model and the hand of the user. The high recognition rate has shown the effectiveness of our algorithm over existing methods like silhouette matching- based gesture recognition. The geometrical features of hand extracted during calibration can provide valuable information for finger spelling in sign language recognition and gesture animation. A more number of finger combinations can be included for the proposed method by inclusion of increased features and consideration of flexion angles. In our future works we will like to increase the number of the features along with the removal of uncertainties in the surrounding environment such as changing backgrounds and shadows to get improved results.

VIII. REFERENCES

[1] V. I. Pavlovic, R. Sharma and T. S. Huang, “Visual Interpretation of Hand Gestures for Human-Computer Interaction,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677–695, 1997.

[2] M. J. Jones and J. M. Rehg, “Statistical Color Models with Application to Skin Detection,”Proc. IEEE Int’l. Conf. Computer Vision and Pattern Recognition, vol.1, pp. 274–280, 1999.

[3] J. Lee and T. L. Kunii, “Model-Based Analysis of Hand Posture,” IEEE Computer Graphics and Applications, pp. 77–86, 1995.

[4] L. Sigal, S. Sclaroff and V. Athitsos, “Skin Color-Based Video Segmentation under Time-Varying Illumination,”IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 7, pp. 862–877, 2004.

[5] J. R. Parker and M. Baumback, “Finger Recognition for Hand Pose Determination,”Proc. IEEE Int’l Conf. Systems, Man, and Cybernetics, pp. 2492–2497, 2009.

[6] D. D. Nguyen, T. C. Pham and J. WookJeon, “Finger Extraction from Scene with Grayscale Morphology and BLOB Analysis,” Proc. IEEE Int’l Conf. Robotics and Biomimetics, pp. 324–329, 2008.

[7] K. Nickel and R. Stiefelhagen, “Visual recognition of pointing gestures for humanrobot interaction,”Image and Vision Computing., vol.25, pp. 1875-1884, 2007.

[8] X. Yin and M. Xie, “Finger identification and hand posture recognition for human robot interaction,”Image and Vision Computing, vol. 25, pp. 1291-1300, 2007.

[9] Y. Sun, J. M. Hollerbach and S. A. Mascaro, “Estimation of Fingertip Force Direction With Computer Vision,”IEEE Trans. Robotics, vol. 25, no. 6, pp. 1356–1369, 2009.

[10] E. Ueda, Y. Matsumoto, M. Imai and T. Ogasawara, “A Hand-Pose Estimation for Vision-Based Human Interfaces,” IEEE Trans. Industrial Electronics, vol.50, no.4, pp.676–684, 2003.

[11] X. Yuan and J. Lu, “Virtual Programming with Bare-Hand-Based Interaction,”Proc. IEEE Int’l Conf. Mechatronics Automation, pp. 896–900, 2005.

[12] J. Lin, Y. Wu and T. S. Huang, “Capturing Human Hand Motion in Image Sequences,”Proc. IEEE Int’l Workshop on Motionand Video Computing, IEEE Computer Society, pp. 1–6, 2002.

[13] S. K. Kang, M. Y. Nam and P. Rhee, “Color Based Hand and Finger Detection Technology for User Interaction,”Proc.IEEE Int’l Conf. Convergence and Hybrid Information Technology, pp. 229–236, 2008.

[14] Y. Xu, J. Gu, Z. Tao and D. Wu, “Bare Hand Gesture Recognition with a Single Color Camera,”Proc. IEEE Int’l Conf. Imageand Signal Processing, pp. 1–4, 2009.

[15] J. Barrho, M. Adam and U. Kiencke, “Finger Localization and Classification in Images based on Generalized Hough Transform and Probabilistic Models,”Proc. IEEE Int’l Conf. Control, Automation,Robotics and Vision, pp. 1–6, 2006.

[16] C. Kerdvibulvech and H. Saito, “Vision-Based Detection of Guitar Players Fingertips Without Markers,”Proc. IEEE Int’l Conf. Computer Graphics, Imaging and Visualisation, pp. 419-428, IEEE Computer Society, 2007.