Hand Pose Recognition Using Geometric Features

M.K. Bhuyan, Debanga Raj Neog and Mithun Kumar Kar
Department of Electronics and Communication Engineering,
Indian Institute of Technology Guwahati, India, 781039
E-mail: {*mkb, debanga, mithun*}*@iitg.ernet.in*

***Abstract*—In this paper, a novel approach for hand pose recognition by using key geometrical features of hand is proposed. A skeletal hand model is constructed to analyze the abduction and adduction movements of the fingers and these variations are modeled by multidimensional probabilistic distributions. For recognizing hand poses, proximity measures are computed between input gestures and pre-modeled gesture patterns. The proposed algorithm is more robust to the improper hand segmentation and side movements of fingers. Experimental results show that the proposed method is very much suitable for the applications related to Human Computer Interactions (HCI).**

***Index Terms*—Skeletal hand model, Morphological operation, Proximity measure.**

I. Introduction

In Human Computer Interaction, gesture based interface gives a new direction towards the creation of a natural and user friendly environment. With the emergence of novel technologies, different human to human communication schemes have been introduced into human computer interactions (HCI) [1]. Matsumoto *et al*. proposed a method to estimate hand pose by taking all joint angles with a 3D skeletal hand model [2]. They used the multi viewpoint camera system for the observation of hand poses and reconstructed the hand poses by using a voxel model. Subsequently, estimation algorithm is used for recognition of hand poses. However accuracy and speed give constraints to their method.

A model-based analysis of hand posture was introduced in [3], [4], [5]. Here, modeling of the hand image was performed to analyze the hand postures. They modeled the hand with a skeleton structure by considering physiological constraints. The constraints are finger joint movements, joint angle limits, movement types, flexion of joints, adduction and abduction of metacarpophalangeal (MP) joints [3]. Huang *et al*. proposed a method for capturing human hand motion in an image sequence by determining the hand pose from the palm by using an iterative closed point algorithm and factorization method [6]. Sun *et al*. estimated the fingertip force directions by imaging the color patterns in the fingernails and surrounding skin [7]. Parker *et al*. proposed a method of fingertip detection for hand pose recognition. They took geometrical properties of fingers and used template matching for finding the location of fingertips. They used shape histograms as templates for finger classification [8]. They have not considered the abduction and adduction movement of fingers. Barrho *et al*. proposed a method for the fingertip detection problem in different manner [9]. They used generalized Hough transform (GHT) to match a fingertip template with a hand image and a probabilistic model for localization of fingertips, but existence of false detection has limited its usability.

We have proposed a model-based method for modeling a gesturing hand to overcome some of the above mentioned difficulties in recognizing hand posture. 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. In the proposed method, hand is first separated from the forearm and some key geometric features of predefined gestures are obtained by hand calibration. The variation of finger positions are modeled by considering the probabilistic variation of key geometric features. The input gesture is recognized by using the minimum distance classifier. In the following sections, the proposed algorithm will be explained in detail.

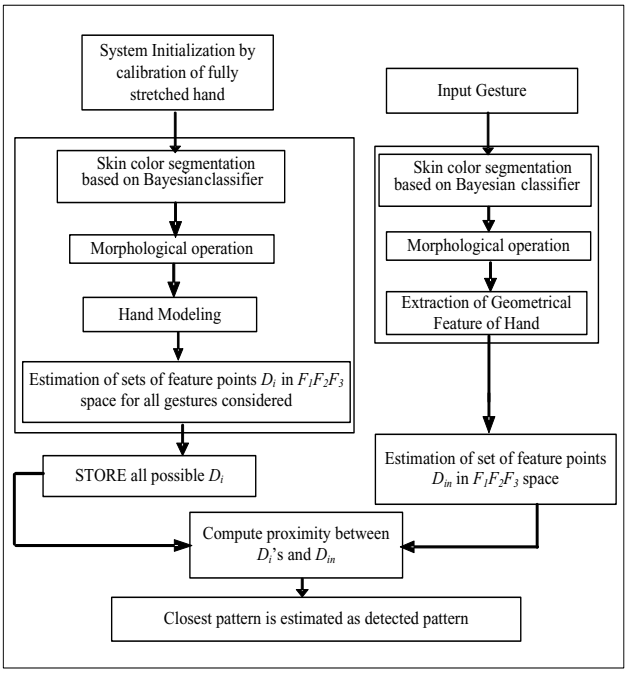


Fig. 1. Block diagram of the proposed scheme.

II. Proposed Hand pose Detection Process

A. Preprocessing Steps

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 [10]. The conditional probabilities

978-1-61284-091-8/11/$26.00 ©2011 IEEE

of foreground (fg) and background (bg) are $P(fg/rgb)$ and $P(bg/rgb)$ respectively. We give some threshold T_0 to distinguish the classification boundary between skin and non skin region based on the criteria that if the ratio of $P(fg/rgb)$ and $P(bg/rgb)$ exceeds T_0, which is based on a relative risk factor associated with misconception, the region is classified as skin color region.

$$T_0 < \frac{P(fg/rgb)}{P(bg/rgb)} = \frac{P(rgb/fg)P(fg)}{P(rgb/bg)P(bg)} \quad (1)$$

If $P(fg)$ is the probability of an arbitrary pixel in an image being skin, we can write

$$T_0 \times \frac{1-P(fg)}{P(fg)} < \frac{P(rgb/fg)}{P(rgb/bg)} \quad (2)$$

Where $P(fg)$ = 0.09 in the training database [10]. With given $P(fg)$ the threshold T_0 can be empirically found by computing "Receiver Operating Characteristics" (ROC curve) over a collection of 100 images excluded from the training data. 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_0 is computed as 0.06 [11].

The centroid of the eroded skin region is computed for extracting the approximate center of the palm region of hand. The centroid $(\overline{c_x}, \overline{c_y})$ is computed as:

$$\overline{c_x} = \sum_{\forall (x_I, y_I) \in S} x_I/N_p,\ \overline{c_y} = \sum_{\forall (x_I, y_I) \in S} y_I/N_p \quad (3)$$

In the above equation, S is the set of all pixels belonging to eroded skin region in its binary mask and N_p is the total number of pixels in the same region. The extracted skin region is normalized to take care of the variation of relative distance of the gesturing hand from the camera. The normalization simply nullifies the motion of the hand normal to the direction of the camera. The normalization constant related to overall hand size is computed by the following equation:

$$N_0 = \sum_{\forall (x_e, y_e) \in E} \sqrt{(x_e - \overline{c_x})^2 + (y_e - \overline{c_y})^2}/M_p \quad (4)$$

Where, E is the set of all pixels belonging to the edge of skin region and M_p is the total number of pixels in the set E.

The orientation of the hand can be estimated by using the information of boundary points and centroid of skin region with respect to the image frame. Orientation is defined as the angle of axis of the least moment of inertia. The orientation vector can be treated as the reference axis R_{ref} for hand. We determine the two intersection points of the orientation vector with the bounding rectangle of skin color region. The middle points of the lines connecting the centroid and each intersection point are computed and joined. The forearm could be detected and separated by locating the wrist part which can be recognized as two parallel lines originating at the object boundary that expands into the shape of a hand. The position where width of skin colored region changes most along the perpendicular direction of constructed line, determines the boundary between hand and forearm [12].

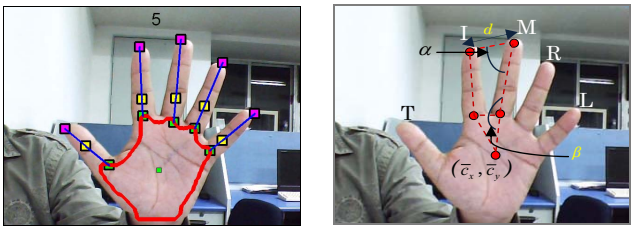
B. Extraction of Geometric Features

The algorithm for extracting key geometric features of hand is summarized as given below:

Step 1: Show five fingers of hand in front of camera such that plane of hand remains perpendicular to the optical axis of camera. Compute the distances of the boundary points of hand from the centroid $(\overline{c_x}, \overline{c_y})$ in a cyclic manner and store them in an array $r(t)$, where t is indices of pixels while traversing the boundary points. Smoothen the data in $r(t)$ using B-spline interpolation and compute the local maxima. Sort the values of the maxima and store top five elements of sorted array as $r_1 > r_2 > r_3 > r_4 > r_5$ and their indices in $r(t)$ as t_1, t_2, t_3, t_4 and t_5.

Step 2: 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, which is set as $\delta = r_5/3$. Compute six sets of points 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. The following algorithm is then implemented.

FOR i=1 to 6
  Consider $s_{i,j}$ as the j^{th} element of set $S_{i,i=1:6}$
  N_i is the total no of elements in S_i .
  FOR j=1 to N_i
    $index = \arg(\min_{p=1:5}(|s_{i,j} - t_p|))$, $index \in \{1,...,5\}$
    Include $s_{i,j}$ in set P_{index}
  END
  FOR k=1 to 5
    IF number of elements in $P_k = 1$ and $P_k = \{p_{k,1}\}$
      Include spatial coordinates corresponding to $p_{k,1}$ in set M_k,
    ELSEIF number of elements in $P_k = 2$ and $P_k = \{p_{k,1}, p_{k,2}\}$
      Include midpoint of spatial coordinates
      corresponding to $p_{k,1}, p_{k,2}$ in set M_k
    ELSE DO NOTHING
    END
  END
  CLEAR $P_{index,index=1:5}$
END



(a) (b)

Fig. 2. (a) Results of proposed algorithm for fingertip detection, (b) A normal stretched hand with fingers named as T, I, M, R and L.

Step 3: Construct five finger axes by joining pixels corresponding to the elements of each of the sets $M_{k,k=1:5}$. Now for recognizing the type of fingers in the hand,

if the left hand is shown for calibration, classify the peaks spatially 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). The just opposite procedure will be applicable for the right hand. For each finger the actual intersection of the finger axis with the hand contour and eroded hand mask will give the actual positions of fingertips P_T, P_I, P_M, P_R, P_L and positions of MP joints R_1, R_2, R_3, R_4, R_5 respectively.

C. Hand Modeling

For mathematical modeling of hand, a model-based approach is proposed considering both the physical structure and physiological constraints of hand.

- The rotation of the proximal joints of hand is responsible for the movement of a particular segment of the hand, which can be specified by the joint's rotation angles. As shown in Fig. 3, we can define a local coordinate system on joint positions and any joint rotation can be modeled as a sequence of rotations occurring around the three axes of the local coordinate system defined [3].
- Only the adduction and abduction of the metacarpophalangeal (MP) joints are considered for modeling. The MP joint of finger III displays limited adduction and abduction.
- The hand is kept perpendicular to the optical axis of the camera to ensure that the line joining centroid and fingertip of middle finger lies parallel to Y axis. The axes of abduction and adduction motion are referenced along Y axis.
- The thumb is modeled separately by assuming that the line joining the MP joint of thumb with the thumb fingertip is straight and it has only abduction and adduction movement about Z axis. The total abduction and adduction angle movement for thumb is taken as 30^o.
- The abduction or adduction angle between the adjacent fingers from II to V for static constraints are taken as

$$-15^o \leq Q^z_{MP,s} \leq 15^o \tag{5}$$

where Z gives the axis of rotation of the adduction or abduction movement in local joint centered coordinate system [1]. Here, s denotes the indices of fingers from II to V.

III. Proposed Algorithm for Fingertype Recognition

A. Feature Selection

For a normal hand gesture showing all five fingers for calibration, we can select 10 pairs of finger combinations. Now, based on the information of relative distance and relative angle generated by fingertip and MP joint positions, we can compute 3 key features for each pair of fingers. The 3 features we are considering are $d_{i,j}, \beta_{i,j}$ and $\alpha_{i,j}$, which are shown

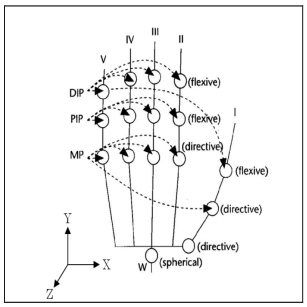


Fig. 3. Hand model showing angle of rotations of fingers.

for the case of I and M finger pair in Fig. 2(b). Here, (i, j) denotes the indices of pair of fingers, where the indices for thumb, index, middle, ring and little fingers are 1,2,3,4 and 5 respectively. Thus, spatial information of each pair of finger can be mapped to a relative 3D feature space $F_1F_2F_3$. For our proposed algorithm for fingertype recognition, we have considered 8 hand gestures consisting of two, three and four fingers and the valid gestures considered are given in Table I. If M_0 denotes the total number of finger pairs possible for a particular gesture, then the gesture can be mapped to M_0 distinct points in $F_1F_2F_3$ space. Thus, 8 selected patterns will map to $F_1F_2F_3$ space into 8 sets $D_{i,i=1:8}$. For the selection of the gesture patterns, the patterns well separated in $F_1F_2F_3$ space as well as easy to gesticulate are considered. 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. For gestures with single finger, template matching can be done for finger type detection. Five finger gesture is a trivial case, where all the five fingers will be present.

B. Modeling the Distribution of Features

Table I 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, β and α in a pair of fingers. Model 1 considers all pair of fingers each including thumb and Model 2 is considered for finger pairs without including thumb.

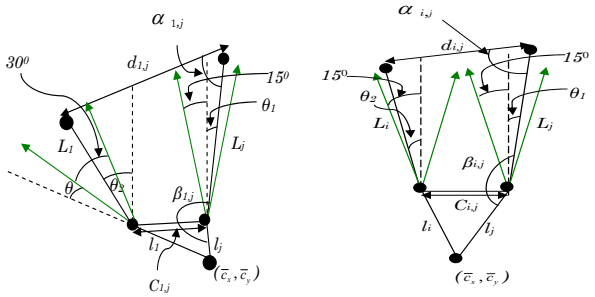


Fig. 4. Proposed Model-1 and Model-2 for modeling abduction and adduction movement of finger about MP joint.

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

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

The proposed Model 1 and Model 2 are shown in Fig. 4. The notations used in the figure are summarized below:

- $d_{i,j}$: d-feature of the i_{th} and j_{th} fingers.
- $\beta_{i,j}$: β-feature of the i_{th} and j_{th} fingers.
- $\alpha_{i,j}$: α-feature of the i_{th} and j_{th} fingers.
- L_i: length of i_{th} finger
- $C_{i,j}$: distance between MP joints of i_{th} and j_{th} fingers
- l_i: distance of MP joint of i_{th} finger from centroid

In this, fingers indexed with 1 to 5 represent thumb, index, middle, ring and little finger respectively. The green arrows in Fig. 4 are indicating the possible range of abduction and adduction angle for a particular finger. Here, θ is the offset angle of the thumb from the reference axis of the thumb computed during calibration . Again, θ_1 and θ_2 are abduction and adduction angle for a particular pair of fingers in a gesture. The features are computed from the two models by using Euclidean geometry as shown in Table II.

TABLE II
COMPUTATION OF FEATURES

Features	Equations for feature computation
$d-feature$	$d_{i,j} = \sqrt{A_c^2 + L_j^2 - 2L_jA_c \times Y}$ (6)
$\beta-feature$	$\beta_{i,j} = 90^0 + \theta_1 + \cos^{-1}(\frac{l_j^2 + C_{i,j}^2 - l_i^2}{2l_jC_{i,j}})$ (7)
$\alpha-feature$	$\alpha_{i,j} = \sin^{-1}(\frac{A_c}{d_{i,j}}\cos(\theta_1 - \sin^{-1}(\frac{L_i\cos(\theta_2)}{A_c})))$ (8)

where,

$$A_c = \sqrt{L_i^2 + C_{i,j}{}^2 + 2L_iC_{i,j}\sin(\theta_2)}$$

and

$$Y = \cos(90^0 + \theta_2 - \sin^{-1}(\frac{L_i\sin(\theta_2)}{A_c}))$$

Here we consider that $G_{j,j=1:8}$ represent the 8 gesture classes. Therefore $D_{k,k=1:8}$ are the sets of points in $F_1F_2F_3$ space for the gesture classes $G_{j,j=1:8}$. For every gesture class with N fingertips, there will be $M_0 = \begin{pmatrix} N \\ 2 \end{pmatrix}$ points in $F_1F_2F_3$ space. Due to uncertainty of abduction and adduction angles, we have modeled each of the three features as a Gaussian distribution for every point in $F_1F_2F_3$ space for a particular gesture.

Let us consider that the average values of the three features, which will be considered as the mean of the distributions, are d_{avg}, β_{avg} and α_{avg}, which are computed by putting the mean angles of θ_1 and θ_2, obtained using the information in Table I and equations (6), (7) and (8). The maximum possible ranges of three features are computed by putting the ranges of θ_1 and θ_2. The ranges obtained for each of the features are denoted as Δd, $\Delta\beta$ and $\Delta\alpha$, which will be used for computing the variances of the feature distributions.

Now, we can define a normal distribution for each of the 3 features based on the information obtained from the geometric features and modeling of feature variations. For each of the three features d, β and α which can be denoted as F_1, F_2 and F_3, we can define a normal distribution as,

$$f(F_i) = \frac{1}{\sqrt{2\pi(\Delta F_i)^2}} e^{-\frac{(F_i - F_{i,avg})^2}{2(\Delta F_i)^2}}\Big|_{i\in\{1,2,3\}} \quad (9)$$

We determine all the sets $D_{k,j}$ for 8 valid gesture classes during the calibration, where $D_{k,j}$ represents j_{th} cluster of k_{th} gesture in $F_1F_2F_3$ space. The set can be expressed as follows:

$$D_{k,j} = \{f(d) \times f(\beta) \times f(\alpha)\}\big|_{j\in\{1,\ldots,M_0\},k\in\{1,2,3,\ldots,8\}} \quad (10)$$

where, M_0 denotes the total number of distributions in set D_k as defined before.

C. Input Gesture Classification

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 Section II.B, a contour distance profile is generated, which is measured by computing the distance of the contour of hand boundary from the centroid in a cyclic manner. The distance profile is interpolated using B Spline method and we will just exclude all the valid peaks with value less than $r_{min} - \delta$. 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. As described earlier in fingertip detection algorithm, we perform morphological operations to find out more accurate positions of fingertips. The input gesture pattern $\overline{f}_{in}$ is then transformed to D_{in} in the $F_1F_2F_3$ space. Next, a proximity measure is used for computing the proximity of D_{in} from the pre-modeled gesture pattern distributions D_{pre} such that pre-modeled gestures considered contain same number of fingertips as the input gesture pattern. Mahalanobis distance is used for computing the proximity between D_{in} and all applicable D_{pre}. While measuring the proximity between D_{in} and D_{pre}, the elements of D_{in} and distributions of D_{pre} are considered in same order. Subsequently, the corresponding gesture pattern is recognized on the basis of minimum distance criteria.

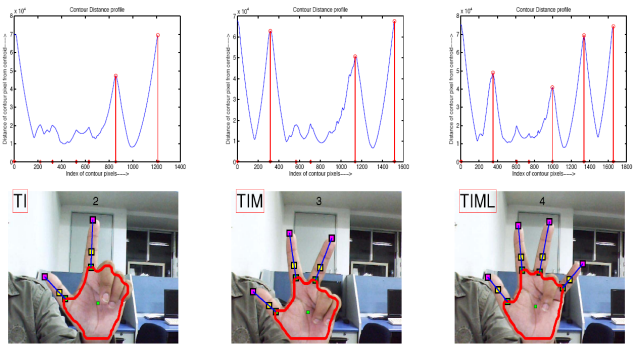


Fig. 5. Some results of fingertip detection and fingertype recognition.

IV. Experimental Results

In our experiment total 20 users were taken for testing purpose, where each user has performed a predefined set of 8 gestures. To evaluate the performance of the proposed recognition system, all gestures are processed under normal illumination conditions. In our experimental setup, the users hand was fixed perpendicular to the optical axis of the camera and only the rotation of hand about the axis perpendicular to palm is considered to tackle the difficulty of matching image features in different viewing conditions. Some of the results of our proposed fingertype recognition algorithm are shown in Fig. 5. The percentage recognition rate of different gesture patterns are summarized in Table III.

TABLE III
Percentage Recognition Rate of Gesture Patterns

Gesture Patterns	Recognition Rate in %
TI	93
IM	93
IL	94
TIM	92
IMR	95
TIL	93
TIML	92
IMRL	94
Total	93.25

The accuracy is calculated by considering the success and failure rate of each predefined gestures in the recognition process. As it can be observed in Table III, 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 applications.

V. Conclusions

In this paper, a simple and robust method is proposed for fingertip detection and hand pose recognition. Our proposed fingertip detection algorithm is more robust against inaccurate color segmentation. A skeletal hand model is constructed based on the information of positions of fingertips and MP joints. The distance and angle features of the hand are modeled as 3D Gaussian distributions, which has considerably increased the performance of our algorithm by eliminating the effects of the side movement of the fingers. The experimental results with high recognition rate show the effectiveness of our algorithm in recognizing the fingertypes of hand gestures. The proposed method can provide valuable information of finger spelling for the applications related to sign language recognition and gesture animation and can be useful in applications like remote robot control. In our future works we will like to extend our proposed algorithm by including the flexion angles of the hand joints in the skeletal hand model, which will allow more natural interaction with the computer.

Acknowledgements

This work was supported in part by a grant from MHRD, Government of India under National Mission for Education through ICT for the project entitled “Development of Indian Sign Language Recognition and Education System for Hearing Impaired Students of India”.

References

[1] Vladimir I. Pavlovic, Rajeev Sharma and Thomas 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] Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai and Tsukasa Ogasawara, “A Hand-Pose Estimation for Vision-Based Human Interfaces,” *IEEE Trans. Industrial Electronics*, vol. 50, no. 4, pp. 676–684 , 2003.

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

[4] Dung Duc Nguyen, Thien Cong Pham and Jae Wook Jeon, “Finger Extraction from Scene with Grayscale Morphology and BLOB Analysis,” *Proc. IEEE Int’l Conf. Robotics and Biomimetics*, pp. 324–329, 2008.

[5] Sung Kwan Kang, Mi Young Nam and Phill Kyu 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.

[6] John Lin, Ying Wu and Thomas S. Huang “Capturing Human Hand Motion in Image Sequences,” *Proc. IEEE Int’l Workshop on Motion and Video Computing, IEEE Computer Society*, pp. 99–104, 2002.

[7] Yu Sun, John M. Hollerbach and Stephen A. Mascaro, “Estimation of Fingertip Force Direction With Computer Vision,” *IEEE Trans. Robotics*, vol. 25, no. 6, pp. 1356–1369 , 2009.

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

[9] Jorg Barrho, Mathias Adam and Uwe 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.

[10] 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

[11] Leonid Sigal, Stan Sclaroff and Vassilis 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.

[12] Yishen Xu, Jihua Gu, Zhi Tao and Di Wu, “Bare Hand Gesture Recognition with a Single Color Camera,” *Proc. IEEE Int’l Conf. Image and Signal Processing*, pp. 1–4, 2009.

[13] Kai Nickel and Rainer Stiefelhagen, “Visual recognition of pointing gestures for humanrobot interaction,” *Image and Vision Computing.*, vol. 25, pp. 1875-1884, 2007.

[14] Xiaoming Yin and Ming Xie, “Finger identification and hand posture recognition for humanrobot interaction,” *Image and Vision Computing.*, vol. 25, pp. 1291-1300, 2007.

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