Fingertip Detection and Finger Identification Approach for Hand Gesture Recognition using Microsoft Kinect

Ahmed Al Marouf*, Md. Kamrul Hasan and Hasan Mahmud

***Abstract*—Hand gesture recognition is one of the recent research attractions in human computer interaction (HCI) and augmented reality (AR). Recognizing different hand gestures depend mostly on the posture of the hand which varies based on the orientation of fingers. To classify them, the orientation of fingertips and the names of individual finger are vital features to be considered. The most challenging task in hand gesture recognition is to determine the palm point and use the point to find all the fingertips. In this paper, a novel approach is proposed to determine the center of the palm and to detect the fingertips. The procedure of fingertip detection includes adaptive Hill Climbing algorithm applied on distance graphs. This paper also propose a novel approach for finger identification based on the relative distances among the fingertips and valley points. The experimental results shows up to 94% accuracy based on its inputs.**

Keywords*—human computer interaction (HCI), fingertip detection, finger identification, graph-based detection, palm point determination.

I. INTRODUCTION

Human computer interaction is a vast research area and interaction techniques are one of the main concerns of it. The nowadays technology are improving the interaction techniques to minimize the barrier of human and computer. Different types of interactive devices are developed and the technology market is growing rapidly holding the hand of these devices. However, better priced and user-friendly interaction techniques are the main concern of these devices. Interaction by touch or multi-touch, voice recognition, facial recognition, movements of eyes etc. are newly developed interaction techniques, which are gaining researchers' attention day by day.

However, better priced and user-friendly interaction techniques are the main concern of these devices. Interaction by touch or multi-touch, voice recognition, facial recognition, movements of eyes etc. are newly developed interaction techniques, which are gaining researchers' attention day by day.

A new interaction technique is gesture while gesture based interfaces and recognition systems are very popular nowadays. "Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of conveying meaningful information or interacting with the environment" [1]. Gestures can be static such as the user assumes a certain pose or configuration or dynamic such as with pre-strike, stroke and post-stroke phases. S. Mitra et al. [1] introduced that gestures could broadly be of three types: hand and arm gestures; head and face gestures and body gestures. Recognizing hand gestures is important for sign languages, human-robot interaction and other entertainment applications for allowing users to play and interact in virtual environments [1]. Using different gestures human can interact thus operate computer systems. Gesture is a common interaction technique which could be used in any kind of computer systems such as desktop, laptop, tablets or even in cell phones.

Gestures could be constructed by different organs such as head movements, body motions or hand movements. Each of these gestures are different from one another in case of computer vision. Hand gesture is used widely in many countries and cultures. It is easy to use and social conventions already exist regarding hand gestures. Different hand gestures are used to welcome someone, to express goodbye, showing accomplishment etc. While interacting with computers human mostly prefer hand or both hands as instructions provided to system are done by handheld devices like mouse, keyboard, keypads etc. Most of these interactions are done by using the fingers or the whole hand including the palm area. In case of gesture recognition, before using hand gestures in real-time systems the positions of the fingers should be identified properly which is possible by detecting fingertips and finger names. Therefore, the fingertip detection and finger identification is the main challenging work in this context.

Virtual reality or augmented reality, vision based games, sign language applications, human robot interactions are some of the applications of hand gesture recognition, specifically fingertip detection and finger identification [2]. 2D or 3D images could be used to detect different gestures. Many depth based cameras like Bumble Bee, Microsoft Kinect etc. could be used for capturing images in real time. In 2D images the overlapped fingers or multidirectional fingers could not be detected. On the other hand, 3D sensors use depth camera with vision or RGB camera which overcomes the problem of segmentation methods for 2D images. In this paper, we have used system having Microsoft Kinect

*Corresponding author.

Ahmed Al Marouf* is with DIU Human Computer Interaction (HCI) Research Lab, Department of Computer Science and Engineering, Daffodil International University, Email: marouf.cse@diu.edu.bd

Md. Kamrul Hasan and Hasan Mahmud are with the Systems and Software Lab (SSL) Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. E-mail: hasankt@iut-dhaka.edu and hasan@iut-dhaka.edu.

Manuscript received September 30, 2018; revised November 16, 2018; Accepted December 03, 2018.

which consist two different cameras: RGB and Depth camera. IR sensor and PrimeSense sensor are used to calculate the depth information. Fig 1 shows the different components of Microsoft KINECT.

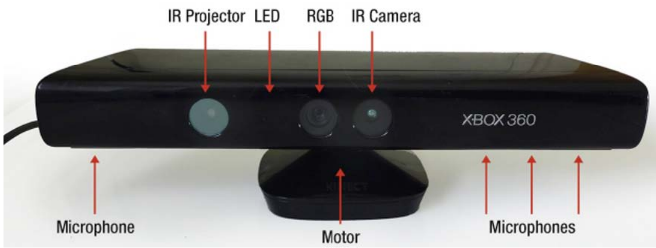

Fig. 1: Different Components of Microsoft Kinect [3], [4]

The idea of Natural User Interaction (NUI) [5] or Natural Computing (NC) [6] is followed in this paper, which does not include any use of wearable sensors, markers or color gloves but the users hand movement. Different methods to detect the fingertips [5], [6], [7] and finger names are introduced by researchers [8], [9], [10].

In this paper, we have proposed novel approaches to find the fingertips and finger names using the information of center of the palm. For finger identification, we have proposed a new model, 4Y - model which is based on adaptive Hill climbing Algorithm for finger identification.

In the rest of the paper, section II discusses about the related works done by different researchers on this problem. Section III illustrates the proposed method to determine the palm point of the hand and in section IV the fingertip detection method is presented. The second proposed algorithm of finger identification is demonstrated in section V. The detail experimental analysis and results are shown in section VI and finally the section VII concludes with some future scopes mentioned.

II. Literature Review

Fingertip detection and finger identification is the major challenge in the context of hand gesture recognition.As the hand gesture solely depends on the fingertips positions and finger orientation, the main target is to find out a better way to determine the fingertips. Different types of images (2D or 3D) could be used for gesture recognition. Several ways have been adopted to determine the fingertips from 2D or 3D images.

Template matching using fingertip templates [9],template matching [10], [11], [12], contour [13], based on finger shape [7], [14] are the state-of-the-art methods used for 2D images for hand gesture and fingertip detection. But each method has its advantages along with some lacking like input images are captured from inferred camera [9-11], “clear contrast between hand and background” [12], “illumination of the workspace is controlled and generally uniform” [8] or using a colorful gloves [15], [16]. A compare and contrast on the fingertip detection methods for 2D images have been introduced in [16] and a robust fingertip detection method has been proposed. On the other hand for 3D images the state-of-the-art methods are based on 3D cameras like Microsoft KINECT, Bumble bee etc. and implemented some previously used methods.

Depth-Skin-Background Mixture Model (DSBMM) segmentation algorithm has be proposed in [4] which is based on Artificial Neural Network (ANN). Xin Zhang et al. [4] proposed a dual mode (side and frontal) switching algorithm to identify the hand pose in writing mode and ellipse fitting technique for palm point detection. M. Elmezain et al. [17] proposed a hand gesture spotting method using Hidden Markov Model (HMM). M.K. Bhuyan et al. [7] proposed a hand calibration algorithm consists Extraction of Geometric features and Hand modeling. In [7] five circles intersecting with the points in segmented image are considered as fingertip points. They have proposed hand segmentation using skin segmentation, detection of centroid, normalization constant, determination of orientation and hand & forearm segmentation. They have proposed a hand calibration algorithm consists extraction of geometric features and hand modeling.

J. L. Raheja et al. [5] used distance transform on the inverted images to determine the center of palm. Nevertheless, the calculation of determination of coordinates is not mentioned. This paper [5] is helpful for palm point identification policy which was performed by applying the distance transform on the inverted binary images of hand. The paper introduced color difference for different hand. D. Uebersax et al. [31] proposed a heuristic based approach to determine the center of the palm and palm region.

Meenakshi et al. [5] proposed a hand gesture recognition system based on shape parameters. For different scenario of the system changes its interest and the shape of the parameters. For different gestures unique code in generated which helps to classify the gesture easily. Z. Ren et al. [18] [18] proposed a system of robust part based hand gesture recognition using different camera. The gesture is recognized based on finger earth mover’s distance with a commodity depth camera [18].

D. D. Yang et al. [19] an effective robust fingertip detection method for finger writing character recognition system on plane surfaces. Template matching [20], [21], [22], [23], axis-boundary intersection [24], [25], conic fitting, contour tracking was performed on the method. To detect the fingertip, position is identified from hand contour and then circle feature matching is done. S. Iyer et al. [20] considered detection of fingers with a depth based hand detector in static frames. True positive, false positive approach is taken in case of template matching and area percentage, whereas, skin model is used for segmentation. For edge detection two important papers are followed in this paper are [21], [22]. Image segmentation is done based on edge detection using boundary code [20], [26], [27]. But it is for the 2D cameras [28], [29], [4], [30], not for 3D camera such as Microsoft Kinect. Lejeune et al. [12] a new jump edge detection method implemented on the basis of canny approximation of edge detection. For ellipse fitting algorithm, Walter Gander et al. [23] least-square fitting of circles and ellipses. Andrew Fitzgibbon et al. [24] a direct approach of non-linear least square fitting of ellipse. The non-generalization and calibration problem of Microsoft Kinect has been elaborately discussed and mentioned by many researchers in [31], [32], [33], [34], [35]

In this paper, we proposed a new approach to determine the center of the palm thus palm point using ellipse fitting algorithm and a graph based approach to determine the fingertips. The center of the palm or palm point could be determined various ways. The comparison between an existing model and proposed model to determine the center of the palm is also a vital element in this paper. Moreover, in this paper, we have proposed a novel approach to determine the palm point of hand which could be applicable for both hands. The palm point information is then used to determine the five different fingertips as well as finger names.

III. PROPOSED APPROACH

The process of fingertip detection includes five steps. From Fig 2, we can illustrate those steps as stated image acquisition, preprocessing, image processing, palm point determination and graph based algorithm for fingertip detection. Pre-processing includes the thresholding, RGB to gray-scale conversion and color & depth calibration. Determining the minimum depth value from the Kinect camera, determining the segmentation threshold, cropping the region of interest and edge detection are the steps of processing. Depth and color segmentation & calibration is done for hand segmentation. For fingertip detection we have merged two existing idea from state of the art in a manner that it minimizes the limitations of both approach. Ellipse fitting technique is applied for palm point determination and point-to-point scanning is done for fingertip locating. In this section, each of the steps are discussed elaborately.

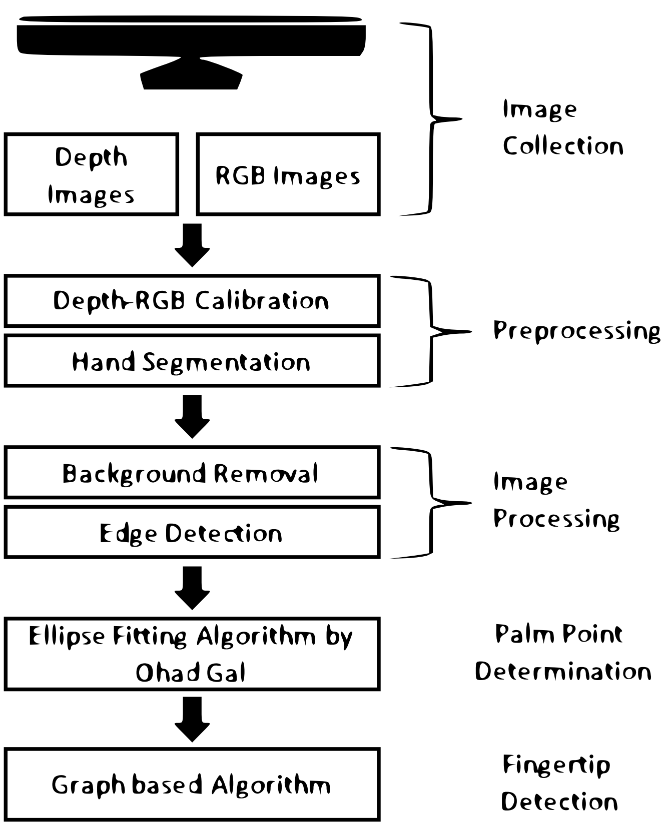

Fig. 2: Flowchart of Fingertip Detection process.

A. Image Acquisition

The images are captured using Microsoft KINECT. Fig 1 shows the hardware architecture of KINECT which integrates two cameras: RGB and Depth. Color image is captured by the RGB camera while the IR camera and the PrimeSense sensor calculate the depth information of the image.

Fig. 3: Image Acquisition. (RGB image from NTU MSR Hand Gesture Dataset [33])

Therefore, color and depth images are collected from the Microsoft KINECT, but which a slight time gap. Because of the unwanted time gap the color and depth images are not calibrated. The depth information of Microsoft Kinect is 11 bit with 2048 levels [4]. The depth value d_{raw} of a point in image could be defined as calibration procedure described in [30]

$$d = KXtang(Hd_{raw} + L) - O \quad (1)$$

Where d is the depth of that specific point in cm, K = 12.36 cm, H = 3.5×10-4 rad, L = 1.18 rad and O = 3.7 cm.

(a) (b) (c)

Fig. 4: Forearm segmentation for the hand. (a) gray image, (b)forearm of hand and (c) Region of Interest (ROI)

B. Preprocessing

Because of some minor hardware problem like low resolution and strong noise, only depth or RGB image is not satis-

factory to use. The color-depth non-synchronization problem introduced in [4] is not a negligible issue as the color and depth images are not captured and updated at the exact time by the Microsoft Kinect sensor [4]. A complex DSB-MM (Depth-Skin Background Mixture Model) is proposed in [4], which integrates the information coming from three individual steps named depth model, skin model and background model. Depth - color synchronization or calibration could be done to solve this problem. The calibration process is a stochastic process like trial and error. The one-to-one mapping between the depth and color image is the output of calibrated image.
As the user will be interacting with the system by hand, the nearest point will be hand from the Microsoft Kinect camera [3][21]. The nearest point in millimeters will be different for each of the images as it totally depends on the users hand position and orientation. As the depth information contains the distance values of each pixel in the image from Microsoft Kinect, the minimum depth value in the calibrated image will indicate the hand pixel [21]. In the calibrated images the minimum depth value and its surroundings up to a certain threshold (introduced as segmentation threshold) is being considered as Region of Interest (ROI). The segmentation threshold is a dynamic variable having values around 240 $\pm$ 5 mm.

C. Image Processing

The region of interest is containing the hand part only. The segmented images are converted to gray-scale images and binary images (Fig 5). All the binary images automatically remove the background of images, so additional background removal techniques are not necessarily applied here. The binary images are considered for the edge detection algorithms. Several edge detection procedures (Sobel, Prewitt, Roberts etc.) are tested (Fig 6) while Canny's approximation gives the better output to the images. From the detected edges, the edge coordinate points are recorded directly. Huge sized arrays are used to store the coordinate points in Matlab. The Canny's approximation method outperforms the other existing edge detection method. Therefore the edges could be detected clearly to be useful for further processing. The edge values are one of the important feature in our proposed approach.

Fig. 5: Gray-scale image (a) and binary image (b) after segmentation.

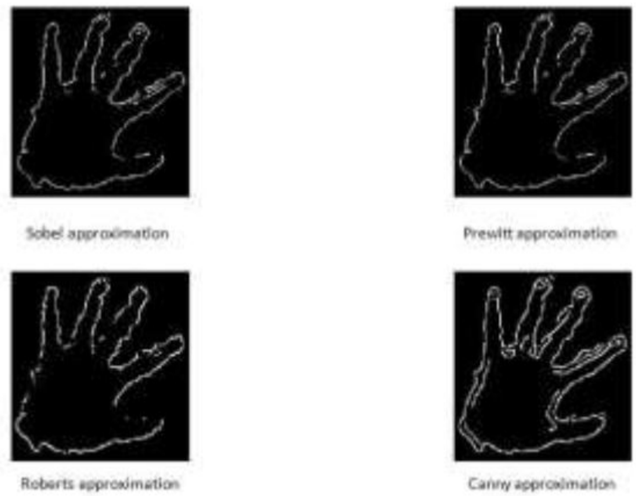

Fig. 6: Sobel, Roberts, Prewitt and Canny Approximation of edge detection on the RoI

D. Palm Point Determination

In this section, we have proposed a new approach to determine the center of the palm or palm point by using Ohad Gal's ellipse fitting algorithm. As the state-of-the-art shows [6], [20] the palm point can be used for further processing. But the limitation in determination of palm point is not considered as a key contribution. In this paper, we have used Ohad Gal's ellipse fitting algorithm to determine the palm point, whereas in the state-of-the-art there is not enough evidence to apply this algorithm on this context.
The palm point is approximately the center of the palm and its coordinates are used for further processing. The fingertips along with the finger shape of human hand are generally in shape of ellipse with different radius, but with no exception. Therefore, ellipse fitting algorithms will be applicable in calculating the coordinates of edge of the region of interest. The Ohad Gal's algorithm [32] uses the Least-Squares criterion for estimation of the best fit to an ellipse from a given set of points *(x,y)*. The LS estimation is done for the conic representation of an ellipse (with a possible tilt). Conic Ellipse representation will be according to (2). Tilt/orientation for the ellipse occurs when the term xy exists (i.e. $b \ = 0$).

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2}$$

In this paper, we have used the Ohad Gal's ellipse fitting algorithm [34] to detect the palm point. The ellipse fitting algorithm takes huge number of coordinates and generates the center of palm. We converted the gray-scale images into binary images to find out the edge points. The morphological operation dilation is done using dilation threshold and structural element for each image. The edge indices of the palm are detected by Canny's approximation and the edge points are considered as inputs to the ellipse fitting algorithm. The ellipse fitting algorithm is based on least square method and center of the palm are calculated thoroughly by mathematical equations. After the dilation process the edge points of palm area used to be like ellipses. All the ellipses have centers of their own and the ellipse fitting algorithm finds the center of the ellipses.

Ohad Gal's ellipse fitting algorithm [34] is directly used to get the parameters for the image. The input for the algorithm are the boundary points and the output is a structure containing sub axis radius of x and y axis; orientation in radius of the ellipse; center coordinates of the non-tilt ellipse; center coordinates of the tilted ellipse; size of the long and short axis and status of detection of an ellipse. The center coordinates of tilted ellipse is the center of palm which is palm points. We have considered this ellipse fitting algorithm because the palm area of the hand is not exactly a circle and it tends to be an ellipse.
Therefore, ellipse fitting algorithm would be applicable for this problem. The Fig 7 shows different hand postures and the palm point detected.

IV. Fingertip Detection

In this paper, we proposed a novel approach to detect the fingertips with the help of the palm point determined in the previous step. We consider the palm point as the base point and calculate the distances of all the edge points from the palm point which generates a distance graph.

A. Distance Graph Generation

The Euclidean distance equation is used to calculate the distances, where, (x_p, y_p) is the coordinate of palm point and (x_i, y_i) is the edge points.

$$Distance = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2} \tag{3}$$

The Euclidean distance graph is plotted and the zigzag curve showing zigzag patterns indicates the distances of all the edge points from the palm point. The Euclidean distance graph shown in Figure 9 illustrates that there are five peak values and four valley values in the curve. There could be more than five peaks recorded in the curve as noises. The multiple peaks problem could be easily eliminate by adaptive Hill Climbing algorithm.
As the distance of fingertips are maximum from the palm point and the distance of valleys are minimum from the palm point, the five peak points and four valley points defines the five fingertips and four valley points of the fingers, respectively. Therefore, the coordinates of the maximum and minimum distance points outputs the required fingertips and valley points, respectively.

B. Graph-based Algorithm to find the fingertips and valley points

Input for the adaptive hill-climbing algorithm will be all the (x, y) coordinates and the output would be the coordinates of five fingertips and four valley points, respectively. All the findings are stored in vectors named as *VectorFingertips* and *VectorValleys* for five fingertips and four valley points, respectively. The algorithm keeps track of the local maxima and local minima of graph. The local maxima are considered to be the fingertips and local minima are considered to be the valley points.

V. Finger Identification

Finger identification from images refers to the recognition of each five fingers differently. With some exceptions, generally human has five fingers named as Thumb, Index, Middle, Ring and Little. From the fingertips determined in the previous step, the name of the fingers could be identified. In this paper, we propose the 4Y model which is a novel finger identification algorithm based on geometric calculation and general biometric features of human hand. The geometric calculations, general biometric features, algorithm's input-output and the 4Y model is explained in the next sections.

A. Geometric Calculations

A simple Euclidean distance calculation formula given in equation (1) is required to calculate the distances between the fingertips (F_1, F_2, F_3, F_4, F_5) and valley points (V_1, V_2, V_3, V_4). Total eight (8) distances should be measured as D_1 (F_1–V_1), D_2 (V_1-F_2), D_3 (F_2–V_2), D_4 (V_2- F_3), D_5 (F_3–V_3), D_6 (V_3-F_4), D_7 (F_4–V_4), D_8 (V_4-F_5). These distance values does not have any unit to measure as they are measured from coordinates.

B. General Biometric Features

In [33] some of the basic assumptions were made from the orientation of the human hand an algorithm was proposed. For our proposed algorithm we are making some general biometric assumptions about the human hand and these are.

- Considering human hand with five fingers but with no exceptions.
- Show the five fingers of the hand in front of the Microsoft Kinect for better calibration. The hand should be perpendicular to the device and fully stretched.
- Distance between Thumb and Index fingertips is largest of all, in a stretched hand. So the addition of distances from these two fingertips to the common valley point will also be largest of all on the same orientation.
- With the change in orientation of fingers, the orientation or position of valley points also change.
- Index and Ring fingers are nearest from the Thumb and Little fingers, respectively.

C. Algorithm's Input-Output

Input for this model algorithm will be the five fingertips (F_1, F_2, F_3, F_4, and F_5), four valley points (V_1, V_2, V_3, and V_4) and the palm point P (x_p, y_p). The output of this finger identification algorithm is an image where the finger names are associated with the fingertips. Thus all five fingers are identified accurately. As the output of the algorithm is visualized in Fig 10, there could be four Y patterns found if we connect the valley points with the palm point, and so the model is named 4Y model for finger identification. This

Fig. 7: Palm point determination using Ohad Gal's Algorithm. The RGB images and the palm points on the right of it.

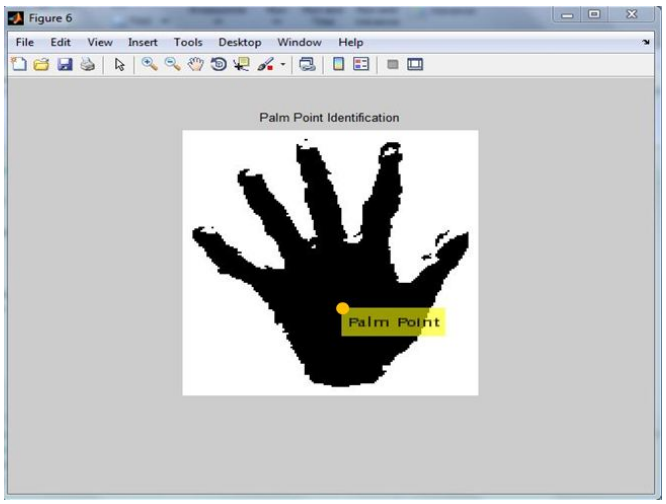

Fig. 8: Palm point (yellow dot) outputted by the Ohad Gal's ellipse fitting algorithm.

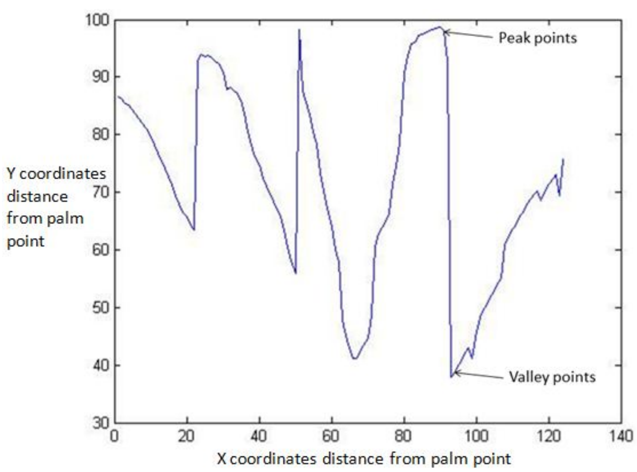

Fig. 9: Euclidean Distance graph for fingertip detection

Algorithm: Adaptive Hill Climbing Algorithm for graph-based fingertip detection.

1. Start the iteration from the first (x, y) point and store it as one of the fingertips in *VectorFingertips*.
2. If (*nextPoint*<*currentPoint*), store it in *VectorValleys* and continue to overwrite on it, until a greater value of *nextPoint* come.
3. Find the next peak point starting from the valley point and store in *VectorFingertips*.
4. Continue step 2 and 4 until the two vectors are full.

Algorithm: 4Y Model for Finger Identification

1. Calculate all the eight distances (D_1, D_2 ..... D_8) between the fingertips (*VectorFingertips*) and valley points (*VectorValleys*) using Euclidean Distance equation.
2. As for F_1 and F_5 there is only one distance calculated, start the identification from these points.
3. If $((D_1 + D_2) > (D_7 + D_8))$, then F_1 is in the *THUMB* finger and F_5 is in the *LITTLE* finger.
4. As F_2 is nearest from F_1, F_2 is in the *INDEX* finger and for the same reason F_4 is in the *RING* finger.
5. The only fingertip left F_3 is in the *MIDDLE* finger.

algorithm will identify finger names even if the fingers are touching each other firmly. The valley points will be closer to the fingertips but with the same assumptions the algorithm could identify the fingers. Fig 11 shows the visual output of the algorithm.

VI. Experimental Analysis and Results

In this section, the experimental analysis and results of the proposed approaches for finding fingertips and identification of finger names are illustrated. The experiments includes the preprocessing, processing, palm point identification, fingertip detection and finger identification as mentioned in the proposed approach section. The experiments are performed by implementing all the steps in order to validate the proposed algorithms.

A. Dataset

The dataset used for the experiment is the NTU Microsoft-Kinect-Hand Gesture dataset [18-19], which contains the color image and the corresponding depth map obtained from the Kinect sensor. It contains 10 different gestures taken from 10 subjects (persons), and for each gesture, there are 10 variations. Thus, there are 101010 = 1000 cases. As the dataset has 10 different gestures, but our algorithm is only validate for the stretched hand. Therefore, only 100 cases could be used

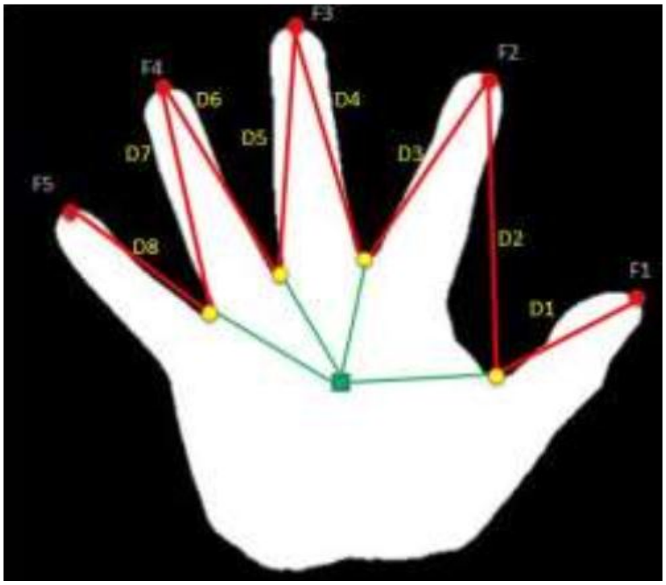

Fig. 10: Demonstration of 4Y Model for Finger Identification.

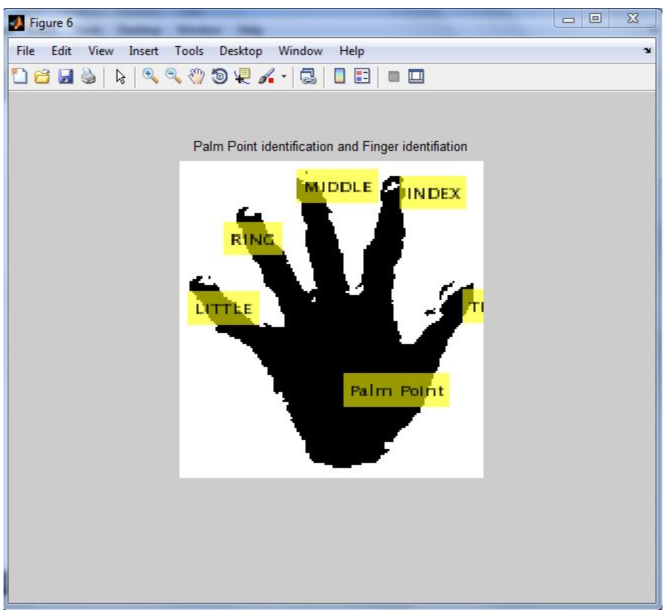

Fig. 11: Fingers are correctly identified by 4Y Model.

for testing this experiment. The input images are two types: RGB image and depth image. The RGB image represents the RGB pixel values and the depth image represents the depth values.

The experiment starts with loading the input images. The experiment process follows the steps mentioned in proposed approach sequentially. We have used Matlab 14a version for all the simulation and implementation process. Firstly, the pre-processing includes the thresholding, RGB to gray-scale conversion and color & depth calibration. Secondly, determining the minimum depth value from the Kinect camera, determining the segmentation threshold, cropping the region of interest and edge detection are performed as a part of the processing. Thirdly, depth and color segmentation and calibration is done for hand segmentation. Then the Canny's edge detection and background removal is performed as part of image processing step. Therefore, the graph-based adaptive hill-climbing algorithm is applied to find the five fingertips and four valley points. Lastly, the4Y model is implemented on the fingertips and valley points to get the name of the fingers.

B. Results

We have tested the 4Y model for finger identification using 100 test images and depth data. The contingency matrix or the confusion matrix [39] in the Table I gives the actual reading of the data that are used to determine the specificity and sensitivity [38] of the proposed algorithm.

TABLE I
CONFUSION MATRIX FOR 4Y MODEL

	Predicted Class					
Actual Class	Finger Name	Thum b	Index	Middl e	Rin g	Littl e
	Thumb	90	7			3
	Index	6	92	2		
	Middle		1	96	3	
	Ring			3	88	9
	Little				6	90

C. Sensitivity and Specificity of the Proposed Algorithm

Sensitivity and specificity are the statistical measures to measure the performance of any binary classification test [38]. As in our algorithm classification is done, so sensitivity and specificity could be a good measurement to judge the algorithm effectively. Sensitivity measures the proportion of actual positives which are correctly identified as such [38]. Specificity measures the proportion of negatives which are correctly identified as such [38]. To calculation of these measurements follows the equations as (4) and (5). True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), False Negative (*FN*) are determined from the confusion matrix. These four quantities could be used to measure other five major terms to justify the proposed algorithm. Precision or Positive predictive value (*PPV*) [38] is the positivity measurement from the positive findings of the data. The calculation of *PPV* can be done using (6).

$$Sensitivity = TP/(TP + FN) \tag{4}$$

$$Specificity = TN/(TN + FP) \tag{5}$$

$$PPV = TP/(TP + FN) \tag{6}$$

Negative predictive value (NPV) [36] is the negativity measurement from the negative findings of the data and Fall-out or false positive rate (FPR) [38] is the measurement of positivity which are rejected from the positive findings of the data. The measurement of the NPV and FPR could be performed by the following equations.

$$NPV = TN/(TN + FN) \tag{7}$$

$$FPR = FP/(FP + TN) \tag{8}$$

False Discovery Rate (FDR) [38] is the measurement of false alert which is found from one minus the positive predictive value of the findings.

$$FDR = 1 - PPV = FP/(TP + FP) \tag{9}$$

Accuracy [38] from the confusion matrix could be measured by the division of true-ness and total positivity and negativity of the findings. The calculation of accuracy would be done by the (10).

$$ACC = (TP + TN)/(TP + TN + FP + FN) \tag{10}$$

The Matthews correlation coefficient [38] is used in machine learning as a measure of the quality of classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1 [38]. The calculation of MCC for all five fingers are visualized in Fig. 12.

$$MCC = \frac{TP + TN + FP + FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

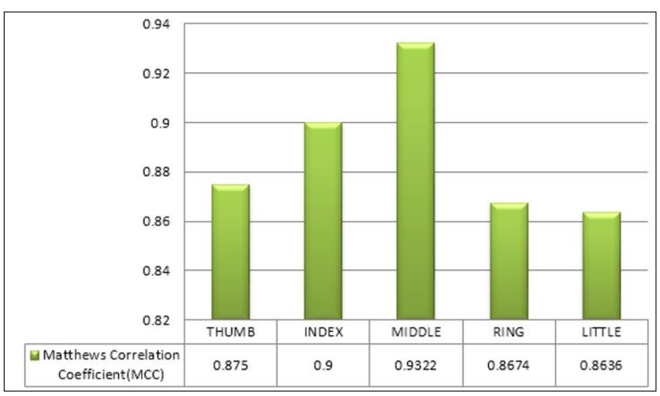

Fig. 12: Matthews Correlation Coefficient for five fingers

For testing the accuracy F1 score or F-measure is been used in the state-of-the-art researches. F1 score is the harmonic mean of precision and sensitivity which follow (12).

$$F1 - score = 2TP/(2TP + FP + FN) \tag{11}$$

All the measurements mentioned above are useful to understand the efficiency and effectiveness of the proposed algorithms. Table II illustrates all the measurement values and Table III demonstrates the accuracy and percentage of error in determination of palm point along with the actual points. Eight random simulated values are presented in the Table 3

TABLE II
SENSITIVITY, SPECIFICITY, NEGATIVE PREDICTIVE VALUE, FALL-OUT OR FALSE POSITIVE RATE, FALSE DISCOVERY RATE AND ACCURACY MEASUREMENT FROM THE CONFUSION MATRIX

	THUMB	INDEX	MIDDLE	RING	LITTLE
True Positive(TP)	90	92	96	88	90
True Negative(TN)	390	392	393	391	388
False Positive(FP)	10	8	7	9	12
False Negative(FN)	10	8	4	12	10
Sensitivity	0.90	0.92	0.96	0.88	0.90
Specificity(SPC)	0.9750	0.9800	0.9825	0.9775	0.9700
Precision(PPV)	0.900	0.920	0.932	0.907	0.882
Negative Predictive Value(NPV)	0.9750	0.9800	0.9899	0.9710	0.9750
Fall-out/False Positive Rate(FPR)	0.0250	0.0200	0.0175	0.0225	0.0300
False Discovery Rate(FDR)	0.100	0.080	0.068	0.093	0.118
F1 Score	0.90	0.92	0.945	0.8934	0.8910
Accuracy	0.960	0.968	0.978	0.958	0.956

which provides 98.37% accuracy. The overall accuracy of the proposed system is around 94%. The comparison of simulation and confusion matrix could be illustrated in the Fig 13. The implementation of experiments could be illustrated in Figure 15.The comparison between the accuracy and F1 score can be visualized from the figure 14.

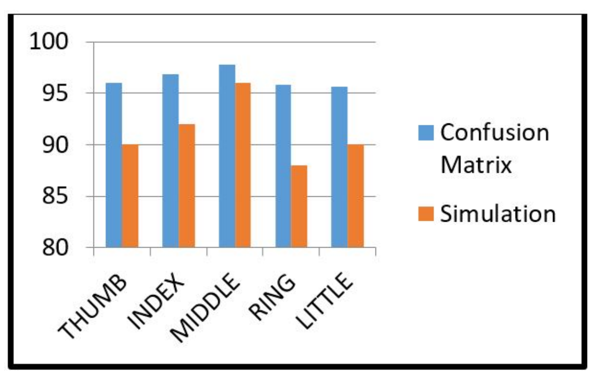

Fig. 13: Accuracy in Simulation and Confusion Matrix.

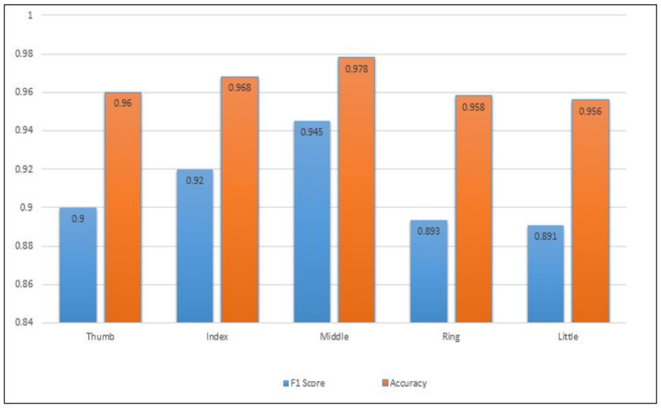

Fig. 14: Comparison between Accuracy and F1 score

TABLE III
THE ACCURACY AND PERCENTAGE OF ERROR IN DETERMINATION OF PALM POINT ALONG WIOTH THE ACTUAL POINTS

Serial No	Actual Point	Resulting Output Point	Accuracy (%)	Percentage of Error (%)
1	(76,100)	(79,102)	97.23%	2.77%
2	(55,100)	(50,100)	98.333%	1.667%
3	(42,45)	(45,43)	98.864%	1.136%
4	(48,102)	(48,100)	98.666%	1.334%
5	(56,98)	(56,98)	100%	0%
6	(65,96)	(64,97)	99.135%	0.865%
7	(78,105)	(75,105)	98.336%	1.664%
8	(98,92)	(97, 100)	96.446%	3.554%

VII. CONCLUSION

For different Human Computer Interaction (HCI) applications the fingertips location and the finger names could be used. Sign language, Calculator, Gesture recognition and spotting, Trajectory analysis, Real time Character recognition using hand gesture are such applications where the output ingredients could be used.

In this paper, we have proposed novel approaches to solve three different problem: palm point determination, fingertips detection and finger identification. For palm point determination we have proposed Ohad Gal's ellipse fitting algorithm, which provides 98.37% accurate result. For fingertip detection we have proposed graph-based adaptive hill-climbing algorithm and for finger identification the proposed 4Y model shows satisfactory results. In future, the filtering techniques could be applied to get a noise free RGB image and consider them as input. Applying state-of-the-art edge detection algorithm could also provide more appropriate edge points which may consider for the fingertip algorithm. The algorithm only works for the static hand gestures but this can be modified to work for different posture and orientations of hand gestures.

REFERENCES

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Trans. Sys. Man Cyber Part C*, vol. 37, no. 3, pp. 311–324, May 2007.

[2] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction," vol. 19, pp. 677–695, 1997.

[3] "Project 4 kinect tracker," *http://web.uvic.ca/rayhan/csc560 /projects/project4/components.html*.

[4] "Kinect camera," *http://www.xbox.com/en-US/ kinect/def*.

[5] X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, "A new writing experience: Finger writing in the air using a kinect sensor," *IEEE Computer Society Journal of Multimedia at work*, vol. 20, pp. 85–93, 2013.

[6] J. L. Raheja, A. Chaudhary, and K. Singal, "Tracking of fingertips and centres of palm using kinect," *3rd International Conference on Computational Intelligence, Modeling Simulation*, pp. 248–252, 2011.

[7] P. Meenakshi, "Hand gesture recognition based on shape parameters," *Proceedings of IEEE International Conference on Image Information Processing (ICIIP 2011)*, 2011.

[8] M. Bhuyan, D. R. Neog, and M. K. Kar, "Fingertip detection for hand pose recognition," *International Journal of Computer Science and Engineering(IJCSE)*, vol. 4, 2012.

[9] J. L. Crowley, F. Berard, and J. Coutaz, "Finger tracking as an input device for augmented reality," *Proceedings of. International Workshop on Automatic Face and Gesture Recognition*, pp. 195–200, 1995.

[10] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface system," *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, p. 411 – 416, May 2002.

[11] ——, "Real-time fingertip tracking and gesture recognition," *IEEE proceedings Computer Graphics and Applications*, vol. 22, pp. 64–71, 2002.

[12] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," *Proceedings. of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, p. 462–467, 2000.

[13] T. Keaton, S. M. Dominguez, and A. H. Sayed, "Snaptell: a multi-modal wearable computer interface for browsing the environment," *Proceedings of Sixth International Symposium on Wearable Computers (ISWC)*, p. 75 – 82, 2002.

[14] V. H. Christian and B. M. François, "Bare-hand human computer interaction," *Proceedings of the 2001 workshop on Perceptive user interfaces*, pp. 1–8, Nov. 2001.

[15] F. Parvini, D. McLoed, C. Shababi, B. Navai, and S. Zali, B.and Ghandeharizadeh, "An approach to glove-based gesture recognition," *13th International Conference on HCI*, pp. 236–245, 2009.

[16] L. Lamberti and F. Camastra, "Real-time hand gesture recognition using a color glove," *16th International Conference on Inage Analysis and Processing*, vol. 6978, pp. 365–373, 2011.

[17] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Hand trajectory-based gesture spotting and recognition using hmm," *6th International Conference on Image Processing (ICIP 2009)*.

[18] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, 2013.

[19] M. Baldauf, S. Zambanini, P. Frohlich, and P. Reichl, "Markerless visual fingertip detection for natural mobile device interaction," *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, ACM MobileHCI '11*, p. 539–544, 2011.

[20] S. V. Iyer, "Detection of fingers with a depth based hand-detector in static frames," *Master of Science Thesis, The University of Texas at Arlington*, 2013.

[21] T. Uemura, G. Koutaki, and K. Uchimura, "Image segmentation based on edge detection using boundary code," *International Journal of Innovative Computing, Information and Control*, vol. 7, 2011.

[22] A. Lejeune, S. Piérard, M. Van-Droogenbroeck, and J. Verly, "A new jump edge detection method for 3d cameras," *Master of Science Thesis from INTELSIG Laboratory, Montefiore Institute, University of Liège, Belgium*, 2011.

[23] W. Gander, G. H. Golub, and R. Strebel, "Least-squares fitting of circles and ellipses," *Institut fur Wissenschaftliches Rechnen Eidgen ossische Technische Hochschule*.

[24] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *Published on Pattern Analysis and Machine Intelligence*, vol. 21, May 1999.

[25] D. D. Nguyen, T. C. Pham, and J. W. Jeon, "Finger extraction from scene with grayscale morphology and blob analysis," *Proceedings of IEEE Int'l Conf. Robotics and Biomimetic*, p. 324–329, 2008.

[26] 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.

[27] J. Lin, Y. Wu, and T. S. Huang, "Capturing human hand motion in image sequences," *Proceedings of IEEE Int'l Workshop on Motion and Video Computing*, p. 1–6, 2002.

[28] C. P. Chen, C. Yu-Ting, L. Ping-Han, T. Yu-Pao, and T. Shawmin, "Realtime hand tracking on depth images," *Visual Communications and Image Processing (VCIP)*, pp. 1–4, 2011.

[29] D. Uebersax, J. Gall, M. V. Bergh, and L. V. Gool, "Real-time sign language letter and word recognition from depth data," *2011 IEEE International Conference on Computer Vision Workshops*.

[30] V. Frati and D. Prattichizzo, "Using kinect for hand tracking and rendering in wearable haptics," *IEEE World Haptics Conference (WHC 2011)*, pp. 317–321, 2011.

[31] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in human computer-interaction," *Proceedings of ICICS*, pp. 1–5, 2011.

[32] O. Gal, "fit$_{e}llipse.mfile, matlabcentral[online].''$

[33] L. Yi, "Hand gesture recognition using kinect," *Master's Thesis, Dept. of Computer Engineering and Computer Science (CECS), University of Louisville*, May 2012.

[34] "Detector performance analysis using roc curves – matlab simulink example," *www.mathworks.com.*, Retrieved on November 21, 2017.

[35] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, pp. 77–89, 1997.

[36] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica (BBA)- Protein*.

Ahmed Al Marouf is currently pursuing M.Sc. Engg in Computer Science and Engineering (CSE) from Islamic University of Technology (IUT), Gazipur, Bangladesh. He received his Bachelor degree from the Department of Computer Science and Engineering (CSE), IUT in 2014. He is a graduate researcher of Systems and Software Lab (SSL) in the CSE department of IUT. His research interest lies within HumanComputer Interaction, Biometric technologies. He is currently working as a lecturer in Department of Computer Science and Engineering (CSE) of Daffodil International University (DIU), Dhaka, Bangladesh. He also acts as the Technical Lead of DIU HCI (Human Computer Interaction) Research Lab.

Md. Kamrul Hasan Md. Kamrul Hasan has received his PhD from Kyung Hee University, South Korea. Currently he is working as an Associate Professor of CSE Department in Islamic University of Technology (IUT), Gazipur, Bangladesh where he has been serving for ten years. Previously, He obtained a B.Sc. in CIT degree from IUT. He has long experience in software as a developer and consultant. His current research interest is in intelligent systems and AI, software engineering, cloud computing, data mining applications and social networking. Dr. Kamrul is the founding director of the Systems and Software Lab (SSL) in the CSE department of IUT.

Hasan Mahmud has received his Bachelor degree in Computer Science and Information Technology (CIT) from Islamic University of Technology (IUT), Bangladesh in 2004. Then he joined as a faculty member in Computer Science and Engineering (CSE) department at Stamford University Bangladesh. He did his Master of Science degree in Computer Science from University of Trento (UniTN), Italy in 2009. He had received University Guild Grant Scholarship for the two years (2007-2009) Master's study and also awarded with early degree scholarship. He has different research articles published in several international journals and conferences. From 2009 he is working as an Assistant Professor in the department of Computer Science and Engineering (CSE) of Islamic University of Technology (IUT), Bangladesh. He is now pursuing his PhD degree in CSE under the guidance of Prof. Dr. M. A. Mottalib and Dr. Md. Kamrul Hasan at IUT. His research interest focuses on Human-Computer Interaction, Gesture based Interaction, Machine learning. He is the co-founder of Systems and Software Lab (SSL) in the CSE department of IUT.