A Vision based Geometrical Method to find Fingers Positions in Real Time Hand Gesture Recognition

Ankit Chaudhary
Computer Vision Research Group, BITS Pilani
Rajasthan, INDIA
ankitc.bitspilani@gmail.com

Jagdish L. Raheja
Machine Vision Lab, CEERI/CSIR, Pilani
Rajasthan, INDIA
jagdish@ceeri.ernet.in

Shekhar Raheja
Digital Systems Group, CEERI/CSIR, Pilani
Rajasthan, INDIA
shekhar.raheja@rediffmail.com

Abstract— **A novel method to calculate the bended finger's angle has presented here which could be used to control the electro-mechanical robotic hand. It is assumed that the robotic hand has the human hand like joints and same number of degree of freedom as human hand. In many applications, equipment like human hand is needed to do the same kind of operation like human do. These days it is easy to make the electro-mechanical robotic hand which has five fingers and same joints but it is not easily controllable as the human hand for accurate job. In our method the hand gesture will be interpreted for controlling the robotic hand. The angles for all the fingers will be calculated and that could be further passed to the robotic hand for controlling its finger. User would perform gesture according to the action as he wants to be done by robotic hand. Here finger positions are detected using geometric analysis of hand in the extracted image, while region of interest cropping from the image make the algorithm faster.**

Index Terms— **Human Computer Interface, Action Recognition, Real Time Systems, Digital Image Processing, Skin Filter, Natural Computing, Gesture Recognition**

I. Introduction

There are several scenarios where people need a robot which can have a human hand like capability and execution of suspected ammunitions, removal of land mines in war field or a narrow place where human cannot reach. The robotic hand should be able to bend finger like human and it should be easily controllable. This paper describes a novel method which could be used to control such an electro-mechanical robot hand using gesture recognition generated by the human hand. In the past researchers have done significant work in this area but they used a wired glove in which sensors were planted or using colors on fingers to recognize the gesture clearly in the image. This work requires no sensor, color or paper clip to detect fingers in the image. The detection of moving fingers in real time needs a fast and robust implementation. On the other hand, processing time is also very critical factor in real time applications as Ozer [26] states "Designing a real-time video analysis is truly a complex task".

Our vision based real time system that can be used to control a remotely located robotic hand, would be able to perform same operation as the human hand. User has to show his natural hand (without wearing any mechanical-electronic equipment) to the camera and the palm should face the camera. User can show any hand to system (right of left) and also there is no restriction on the direction of the hand. If the palm is facing the camera, the hand can be in any direction to control the electro-mechanical hand. Now user would bend his fingers to hold an object (virtual object), and the robotic hand will do the same operation to hold the actual object on its location. The movement of user hand will change the movement of robot hand in real time. This work is extension of our earlier work discussed in [29][43].

This vision based system detects fingertips in the real time from live video and calculate fingers bending angle. The process from the human gesture to robotic hand action is explained in figure 1. First captured 2D image would be preprocessed and skin filter would be applied. Segmentation method is able to extract the hand gesture from the image frame even if there are skin colors like objects in the gesture background. The processed image would be cropped to get only area of interest to make further processing faster. In the cropped image, fingertips

doi:10.4304/jsw.7.4.861-869

and center of palm would be detected and then system would measures distance between centre of palm and fingertips. The calculated angle for each finger could be passed as input to the robotic hand, so that robotic hand can bend its finger accordingly. System is able to detect fingertips, center of palm and angles continuously without any system error. In this paper fingertip detection based gesture recognition was done without using any training data or any learning based approach.

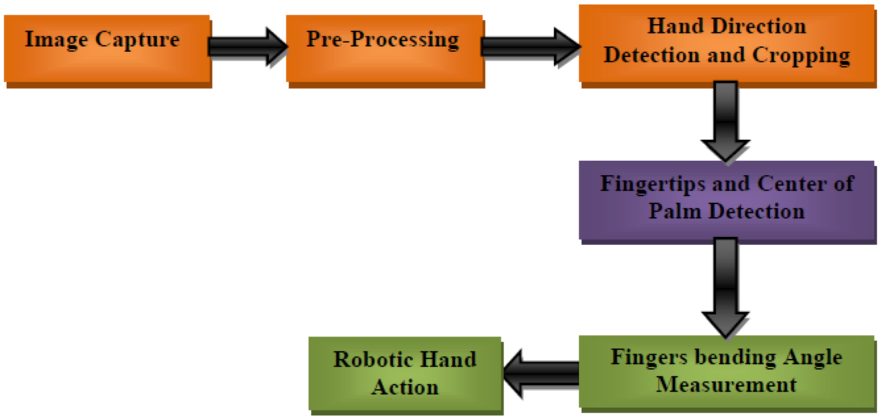

Figure 1. Block diagram flow of the system in different phases

II. RELATED WORK

Many applications could be found in literature of real time robotic control in human computer interaction, computer games control [8], human robot interaction [36], and sign language recognition [34]. Bhuyan [1] developed a gesture recognition system using edge detection and hand tracking and FSM, TGR classification techniques for developing a platform for communication with robots. Dastur [7] controlled a robotic arm by gestures recognition using HAAR classifier. Chaudhary [3], Hardenberg [11], Hoff [12], Li [18], Man [19] and Mohammad [20] also have used gesture recognition to control robots/electro-mechanical gadgets in their applications. Raheja [28] controlled robotic hand using human hand movement where he used a PCA based pattern matching. Many researchers [10][15-17][21][23][32-33][37][42] have used fingertip detection in their research work to find out information about the human hand according to their applications. A survey on fingertip detection can be found in [4].

In the literature we found several issues in existing approaches in the past work. Garg [9] used 3D images in his method to recognize the hand gesture, but this process is complex and also not time efficient. Yang [40] analyses the hand contour to select fingertip candidates, then finds peaks in their spatial distribution and checks local variance to locate fingertips. This method is not invariant to the orientation of the hand. There are other methods, which are using directionally variant templates to detect fingertips [14][30]. Few other methods are dependent on specialized instruments and setups like using infrared camera [24], stereo camera [41], fixed background [6][27] or color markers on hand to simplify the constraints. Many fingertip detection methods are based on hand segmentation technique because it decreases pixel area which is going to process, by selecting only areas of interest. However most segmentation methods cannot do a clearly hand segmentation under some conditions like fast hand motion, cluttered background, poor light condition [40].

Poor hand segmentation performance usually invalidates fingertip detection methods. Few researchers [24-25][31] used infrared camera to get a reliable segmentation. Few other researchers [6][13][27][35][39] limited the degree of the background clutter, finger motion speed or light conditions to get a reliable segmentation in their work. Some of fingertip detection methods cannot localize accurately multidirectional fingertips. Researchers [2][6][27][35] assumed that the hand is always pointing upward to get precise localization. Nolker [22] also calculated the angles in hand posture using neural network, but she did it with 3D modeling.

III. IMAGE PRE-PROCESSING

Real time video was captured using a simple web camera in 2D, which was connected to a windows XP® running PC. There can be varying lightning conditions, in which user is giving input to system.

A. Skin Filter

A HSV color space based skin filter was applied on the captured RGB format image to reduce lighting effect. The resultant image was segmented to get a binary image from the original one. Binary images are bi-level images where each pixel is stored as a single bit (0 or 1). Smoothening to the image was needed, as the output image had some jagged edges as clearly visible in figure 2(c). There can be some noise in the filtered images due to false detected skin pixels or some skin color objects (like wood) in background, it can generate few unwanted spots in the output image as shown in the right corner of figure 2(e). To remove these errors, the biggest BLOB was applied to the noisy image; the error free image is shown in figure 3. The only limitation of this filter is that the BLOB should be the biggest one. In this masking, background would be eliminated, so false detected skin pixels would not exist in the background.

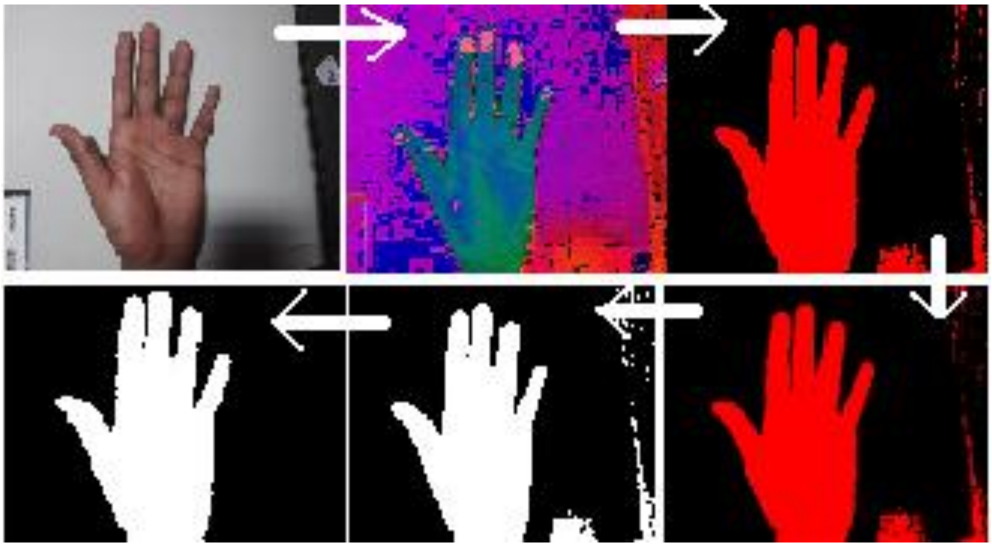

Figure 2. Binary Image formation (a) Target image (b) HSV conversion (c) Filtered image (d) Smoothen image (e) Binary image (f) Biggest BLOB

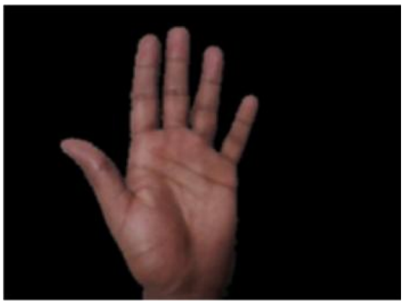

Figure 3. The masked image

B. Hand Direction Detection

In this system, the user can give directionally free input by showing hand gesture to the camera. Hand can be on any angle in any direction but the palm should be toward the camera for the accurate control of robotic hand. System first has to find out the direction of hand to extract the area of interest. For this a 4-way scan of pre-processed image was performed as shown in figure 4 and histograms were generated based on skin color intensity in each direction. In all four scans the maximum value of skin pixels was chosen from histograms and it was noted that maximum value of sin pixels represents the hand wrist and obviously opposite end of this scan would represent the fingers in the hand (as this is also clear in figure 4). Histograms generation equations were:

$$H_x = \sum_{y=1}^{n} imb(x,y)$$

$$H_y = \sum_{x=1}^{m} imb(x,y)$$

Here *imb* represents the binary silhouette and m, n represents the row and columns of the matrix *imb*. The yellow bar showed in figure 4 corresponds to the first skin pixel in the binary silhouette scanned from the left to right direction. Similarly the other bars correspond to respective directions as shown in figure 4. For this input image frame the red bar had greater magnitude than other bars, the hand wrist was in downward direction of the frame and consequently the direction of fingers were in the upward direction.

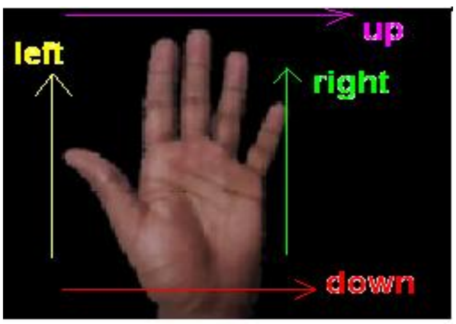

Figure 4. Image scanning

C. Image Cropping

Image cropping eliminates unwanted region to be processed in further steps, hence avoids unnecessary computational time. In the histograms which we generated in hand direction detection, it was observed that at the point where hand wrist ends, a steeping inclination of the magnitude in the histogram starts. As starting point of image where inclination was found and the other points of the first skin pixel in other three scans, would give the coordinates where the image is to be cropped. The equations for cropping the image were*:*

$$imcrop = \begin{cases} origin_{image}, for\ Xmin < X < Xmax \\ \qquad\qquad\qquad Ymin < Y < Ymax \\ 0, \qquad\qquad elsewhere \end{cases}$$

Where imcrop represents the cropped image, Xmin, Ymin, Xmax, Ymax represent the boundary of the hand in the image. Some results of image cropping for different inputs are shown in figure 5. In all the histograms in figure 5 it is clear that at the wrist point, a steeping inclination starts in the scanning direction.

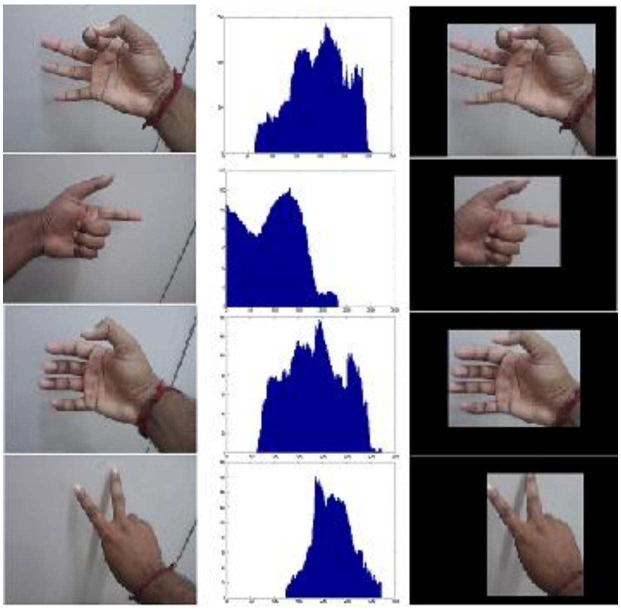

Figure 5. Results of hand cropping process

IV. Fingertip and Centre of Palm Detection

Now we have the cropped image which has area of interest, in which we will try to find out fingertips and center of palm for further use in system.

A .Detection of fingertips

Fingertips have been used in different systems for different purposes. In our approach the hand direction is already known from previous steps. A scan was done in the cropped binary image from the wrist to the finger and number of pixels were calculated for each row or column based on the hand direction, as hand can be either horizontal or vertical. The intensity values for each pixel were assigned from 1 to 255 in increased manner from wrist to fingertip proportionality. So, each skin pixel on the edges of the fingers is assigned a high intensity value of 255. Fingertips were detected taking a threshold of value 255 as shown in figure 6. Mathematically the fingertip detection can be explained as:-

$$pixel_{count}(y) = \sum_{X=Xmin}^{Xmax} imb(x,y)$$

$$modified_{image}(x,y) = round(\ x * 255/\ pixel_{count}(y)\ )$$

$$Finger_{edge}(x,y) = \begin{cases} 1 & if\ \ modified_{image}(x,y) = 255 \\ 0 & otherwise \end{cases}$$

Here $Finger_{edge}$ would give the fingertips points.

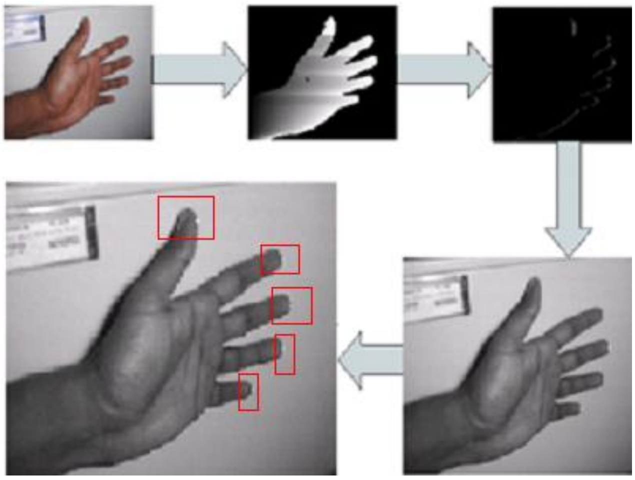

Figure 6. Fingertip detection process

The line having high intensity pixel, is first indexed and checked whether differentiated value lie inside an experimentally set threshold for a frame of resolution 240x230, if it is true it represents a fingertip. A result of fingertip detection process is shown in figure 7 where fingertips are detected as white dots.

Figure 7. Fingertips and centre of palm detected in a real time system

B. Centre of Palm detection

Automatic centre of palm (COP) detection in a real time input system is a challenging task, but it opens a new set of applications where hand gesture recognition can be used. The exact location of the COP in the hand can be identified by applying a mask of dimension 30x30 to the cropped image and counting the number of on pixels lying within the mask. This process was made faster using summed area table of the cropped binary image for calculating the masked values [5]. In the summed area table the value at any point (x, y) is the sum of all the pixels above and to the left of (x, y), inclusive. As shown in equation:-

$$sum(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

The summed area table can be computed efficiently in a single pass over the image as:-

$$sum(x,y) = i(x,y) + sum(x-1,y) + sum(x,y-1) \\ -sum(x-1,y-1)$$

Once the summed area table has been computed, the task of evaluating any rectangle can be accomplished in constant time with just four array references (figure 8) as:-

$$\sum_{\substack{A(x)<x' \le C(x) \\ A(y)<y' \le C(y)}} i(x',y') \;\; = sum(A) + sum(C) - sum(B) - sum(D)$$

The value of the rectangular mask over a region can be calculated by simply four lookups. This improves the speed of the computation by a factor of 250. The COP was calculated as the mean of the centers of all the regions that have a sum of more than a threshold as shown in figure 8.

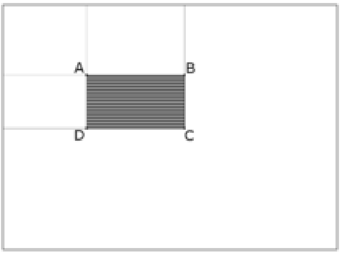

Figure 8. Finding the sum of a rectangular area [38]

From the experiments this threshold came as 832. Some results of fingertips and COP detection are shown in Figure 13.

V. ANGLE CALCULATION

The fingertips and COP information is now known to us, which would be used to detect the position of fingers in the gesture, made by user in one frame of input.

A. Distance between COP & Fingertips

The distance between each fingertip and COP can be calculated by subtracting their coordinates as shown in figure 9.

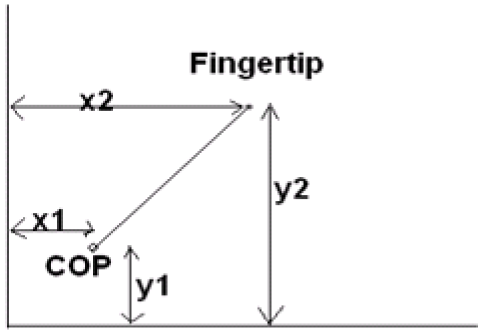

Figure 9. Distance calculation between COP and Fingertips

B. Finger bending angle

Initially user has to show all fingers open gesture to the system which will be recorded as the reference frame for the system and the in this frame bending angle of all fingers would be marked as 180^0 as shown in figure 10.

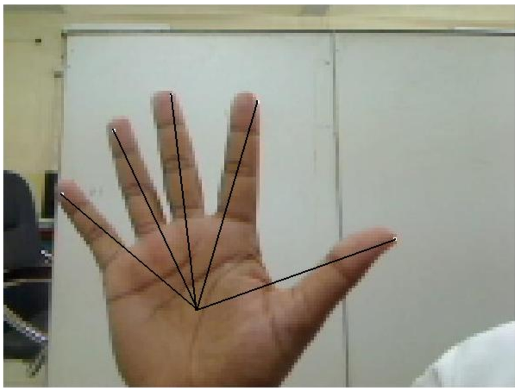

Figure 10. The reference frame

The distance between any fingertip and the COP would be the maximum in this position. As user starts bending the fingers in either direction (forward or backward), distances among fingertips and COP would decreases.

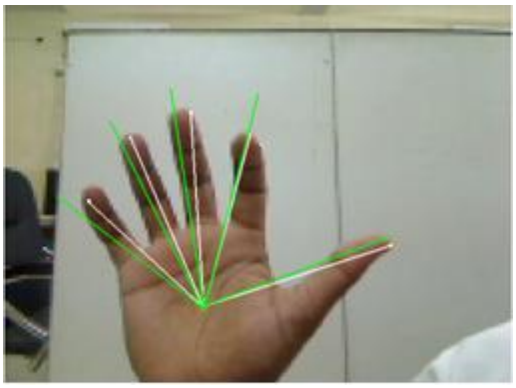

Figure 11. Green lines shows the reference distances and white lines shows the current distances

The values calculated from the reference frame would be stored for each finger. If the user changes the position of his fingers, the distance between COP and fingertips would be compared with the reference distances as shown in figure 11. The angle on which fingers were bended, would be calculated on comparing these distances. In our method it can be from 180^0 to 90^0 as after bending more than 90^0, base of fingers in the hand would be detected as fingertips. Through the experiments, the distance between COP and fingertip is assumed to be the 1/3rd of the reference distance on 90^0 and when the angle is 180^0, the distance between COP and fingertip is assumed to be equal to the reference distance as shown in figure 12.

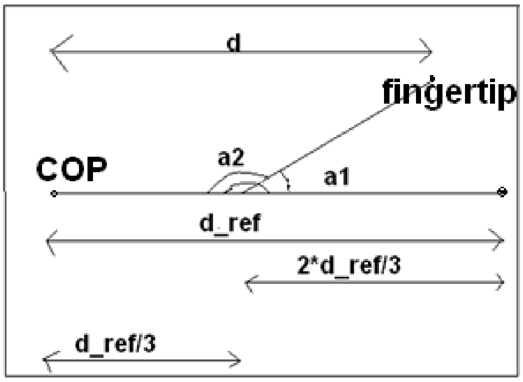

Figure 12. Angle approximation.

From the figure 12 it is clear that when d = d_{ref}, angle $a1=0^0$ and when d= d_{ref} /3, angle $a1=90^0$.

So, we can express angle a1 as:

$$a1 = 90° - \frac{d - d_{ref}/3}{2d_{ref}/3} * 90°$$

Angle of fingers bending, a2 from the figure 12, would be:

$$a2 = 180° - a1$$

$$a2 = 180° - 90° + \frac{d - d_{ref}/3}{2d_{ref}/3} * 90°$$

$$a2 = 90° + \frac{d - \frac{d_{ref}}{3}}{\frac{2d_{ref}}{3}} * 90$$

Table 2 shows results of the calculating different distances among fingertips and COP, and their corresponding angles for the respective user inputs. This method works on any hand gesture input, shown in any direction. Even if the user is moving the hand position, this information is also passed to remote robotic hand and robotic hand would also move in the same direction.

VI. PERFORMACNE

Table 1 shows the comparative analysis of time taken by the system in different steps. The simulation of system was done in MATLAB® running on Windows XP® and Pentium®4, 2.80 GHz processor. Maximum time was taken by preprocessing part and after that, image cropping and hand direction detection took longest time. It is clear that if we take few assumptions, the system will run much faster, but for the robustness we are not putting any condition to the user.

The users was giving input with their hand (one hand at a time, either right or left), in random directions. System was tested in different conditions for long time to check its sustainability in the commercial deployment and it performed excellently with different users. It is free from the user's hand geometry and would work same for everyone. In the past Bhuyan [1] have done experiment with dynamic gestures and was getting accuracy of 80-95% in his work. Raheja [28] tested accuracy of 90% in his robotic hand control work. Currently we are getting accuracy of 90-95% and trying to improve our system to make it more robust.

TABLE I.
TABULATION OF COMPUTATIONAL TIME

S.No.	Action	Time taken (in ms)
1.	BS Formation	45(30.6%)
2.	Hand Direction & Cropping	39(26.5%)
3.	COP Detection	22(15%)
4.	Fingertips Detection	25(17%)
5.	Angle of bending	16(10.9%)
6.	Total	147(100%)

VII. CONCLUSION

This paper gives a novel approach of the hand gesture recognition where no learning or training method was used. The usage of the system is very easy and robust. User has to show the hand in front of the camera and the fingers bending angle would be calculated. These angles could be passed to the electro-mechanical robotic hand. User can show any hand to camera in any direction, and fingertips and centre of palm would be detected in the captured image. The gesture was extracted even from the complex background and cropping of ROI made algorithm faster.

The bending angles of fingers were calculated using a time efficient geometric modeling method. The user can control the robotic hand using his gesture without wearing any gloves or markers. The results are satisfactory and this technology can be used in a many real life applications to save human life. In the future work we will implement this work into a small chip where a small camera would be in built and it would be able transfer gesture interpretations over the wireless network.

ACKNOWLEDGMENT

This research is being carried out at Machine Vision Lab, Central Electronics Engineering Research Institute (CEERI/ CSIR), A Govt. Of India Laboratory, Pilani, INDIA. It was a part of "Controlling the electro-mechanical hand using hand gesture" project. Authors would like to thank The Director, CEERI Pilani for providing research facilities and for his active encouragement and support.

.

TABLE II.
DISTANCES (NO. OF PIXELS) BETWEEN COP AND FINGERTIPS AND CORRESPONDING ANGLES (IN DEGREES)

S. No.	Image	Finger1		Finger2		Finger3		Finger4		Finger5	
		Distance	Angle	Distance	Angle	Distance	Angle	Distance	Angle	Distance	Angle
1. Ref		207.05	180	229.506	180	254.283	180	255.765	180	246.14	180
2.		176.78	160.3	211.32	169.3	237.103	170.9	201.479	151.3	236.24	174.6
3.		188.138	167.7	214.308	171.1	235.936	170.3	243.298	173.4	237.191	175.1
4.		125.032	126.5	192.276	158.1	199.063	150.7	206.461	154	142.302	123
5.		144.9	139.5	149.933	133.2	146.512	122.8	106.075	101	138.679	121.1

Figure 13. fingertips and COP detections in several hand postures

REFERENCES

[1] M.K. Bhuyan, D. Ghosh, and P.K. Bora, "Designing of Human Computer Interactive Platform for Robotic Applications," TENCON 2005 IEEE Region 10, pp.1-5, Nov. 2005.

[2] T. Brown, R.C. Thomas, “Finger tracking for the Digital Desk”, Proceedings of First Australasian User Interface Conference (AUIC 2000), Canberra, Australia, pp.11-16, Feb, 2000.

[3] A. Chaudhary, J.L. Raheja, “ABHIVYAKTI: A Vision Based intelligent System for elder and Sick persons”, Proceedings of 3rd IEEE International Conference on Machine Vision, Hong Kong, pp. 361-364, Dec 2010.

[4] A. Chaudhary, J.L. Raheja, K. Das, S. Raheja, "A Survey on Hand Gesture Recognition in context of Soft Computing", Published as Book Chapter in “Advanced Computing” CCIS, Springer Berlin Heidelberg, Vol. 133, pp. 46-55, 2010.

[5] F. Crow, "Summed-area tables for texture mapping". SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, New York, USA, pp. 207–212.

[6] J.L. Crowley, F. Berardand, J. Coutaz, “ Finger Tacking As an Input Device for Augmented Reality” Proceedings of International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland pp.195-200, Jun 1995.

[7] J. Dastur, A. Khawaja, "Robotic Arm Actuation with 7 DOF Using Haar Classifier Gesture Recognition," Second International Conference on Computer Engineering and Applications (ICCEA), vol.1, pp. 27-29, March 2010.

[8] W. Freeman, “Computer vision for television and games”, Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, page 118, 1999.

[9] P. Garg, N. Aggarwal, S. Sofat, “Vision Based Hand Gesture Recognition”, World Academy of Science, Engineering and Technology, vol. 49. pp. 972-977.

[10] G. Gastaldi and et al., “a man-machine communication system based on the visual analysis of dynamic gestures”, International conference on image processing, Genoa, Italy, pp. 397-400, Sep, 2005.

[11] C. V. Hardenberg, F. Berard, “Bare hand human computer interaction”, Proceedings of the ACM workshop on Perceptive user interfaces, Orlando, Florida, USA, pp.1-8, Nov 2001.

[12] W.A. Hoff, J.C. Lisle, "Mobile robot control using a small display," Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.4, pp. 3473- 3478, Oct. 2003.

[13] T. Keaton, S. M. Dominguez, A. H. Sayed, “SNAP & TELL: a multimodal wearable computer interface for browsing the environment” Proceedings of Sixth International Symposium on Wearable Computers, Seattle, WA, USA pp.75–82, Oct 2002.

[14] H. Kim, D.W. Fellner, “Interaction with hand gesture for a back-projection wall”, Proceedings of Computer Graphics International, Greece, pp. 395-402, June, 2004.

[15] J. M. Kim, W. K. Lee, “Hand shape Recognition using fingertips” In the proceedings of fifth International Conference on Fuzzy Systems and Knowledge Discovery, Jinan, Shandong, China, pp.44-48, Oct 2008.

[16] B. Lee, J. Chun, “Manipulation of virtual objects in marker-less AR system by fingertip tracking and hand gesture recognition”, Proceedings of 2nd international conference on interaction science: Information Technology, Culture and Human, Seoul, Korea, pp. 1110-1115, 2009.

[17] D. Lee, Y. Park, “Vision-Based Remote Control System by Motion Detection and Open Finger Counting”, IEEE Transactions on Consumer Electronics, Vol. 55, issue 4, pp. 2308-2313, Nov 2009.

[18] Z. Li, S. Wachsmuth, J. Fritsch, G. Sagerer, "View-adaptive manipulative action recognition for robot companions," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1028-1033, Oct. 29 2007.

[19] W. T. Man, S. M. Qiu, W. K. Hong, "ThumbStick: a novel virtual hand gesture interface," IEEE International Workshop on Robot and Human Interactive Communication, pp. 300- 305, Aug. 2005.

[20] Y. Mohammad, T. Nishida, S. Okada, "Unsupervised simultaneous learning of gestures, actions and their associations for Human-Robot Interaction," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2537-2544, Oct. 2009.

[21] D.D. Nguyen, T.C. Pham, J.W. Jeon, “Fingertip Detection with Morphology and Geometric Calculation”, IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, pp. 1460-1465, Oct 2009.

[22] C. Nolker, H. Ritter, "Parametrized SOMs for hand posture reconstruction," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, vol.4, pp.139-144, 2000.

[23] C. Nolker, H. Ritter, “Visual Recognition of Continuous Hand Postures”, IEEE Transactions on neural networks, Vol. 13, No. 4, pp. 983-994, July 2002.

[24] K. Oka, Y. Sato, H. Koike, “Real time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems”, Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR.02), Washington D.C., USA, pp.411–416, 2002.

[25] K. Oka, Y. Sato, H. Koike, “Real time fingertip tracking and gesture recognition” IEEE Computer Graphics and Applications, vol. 22, Issue 6. pp. 64–71.

[26] I. B. Ozer, T. Lu, W. Wolf, “Design of a Real Time Gesture Recognition System : High Performance through algorithms and software” IEEE Signal Processing Magazine, pp. 57-64.

[27] F. K. H. Quek, T. Mysliwiec, M. Zhao, “Finger Mouse: A Free hand Pointing Computer Interface”, Proceedings of International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp.372-377, Jun 1995.

[28] J.L. Raheja, R. Shyam, U. Kumar, P.B. Prasad, "Real-Time Robotic Hand Control Using Hand Gestures," Second International Conference on Machine Learning and Computing (ICMLC), 2010, pp.12-16, Feb. 2010.

[29] J.L. Raheja, K. Das, A. Chaudhary, "An Efficient Real time method of fingertip detection", Proceeding of 7th International Conference on Trends in Industrial Measurements and Automation (TIMA 2011), Chennai, INDIA, pp. 447-450, Jan 2011.

[30] A. Sanghi, H. Arora, K. Gupta, V. B. Vats, "A Fingertip Detection and Tracking System as a Virtual Mouse, a Signature Input Device and an Application Selector", Seventh International Conference on Devices, Circuits and Systems, Cancun, Mexico, pp.1-4, April 2008.

[31] Y. Sato, Y. Kobayashi, 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, Grenoble, France, pp.462–467, 2000.

[32] A.E. Sawah and et al., "a framework for 3D hand tracking and gesture recognition using elements of genetic programming", 4th Canadian conference on Computer and robot vision, Montreal, Canada, pp. 495-502, May, 2007.

[33] M. C. Shin, L. V. Tsap, and D. B. Goldgof, "Gesture recognition using bezier curves for visualization navigation from registered 3-d data", Pattern Recognition, Vol. 37, Issue 5, pp.1011–1024, May 2004.

[34] T. Starner, A. Pentland, "Real-time american sign language recognition from video using hidden markov models", In SCV95, pp. 265–270, 1995.

[35] A. Tomita, R. J. Ishii, "Hand shape extraction from a sequence of digitized grayscale images", Proceedings of twentieth International Conference on Industrial Electronics, Control and Instrumentation, IECON '94, Bologna, Italy, vol.3, pp.1925–1930, 1994.

[36] J. Triesch, C.V.D. Malsburg, "Robotic gesture recognition", In Gesture Workshop, pp. 233–244, 1997.

[37] R. Verma, A. Dev, "Vision based Hand Gesture Recognition Using finite State Machines and Fuzzy Logic", International Conference on Ultra-Modern Telecommunications & Workshops, pp. 1-6, Oct, 2009.

[38] http://en.wikipedia.org/wiki/Summed_area_table

[39] Y. Wu, Y. Shan, Z. Zhangy, S. Shafer, "VISUAL PANEL: From an ordinary paper to a wireless and mobile input device" Technical Report, MSRTR 2000, Microsoft Research Corporation, 2000.

[40] D. Yang, L.W. Jin, J. Yin and Others, "An effective robust fingertip detection method for finger writing character recognition system", Proceedings of the Fourth International Conference on Machine Learning And Cybernetics, Guangzhou, China, pp.4191–4196, Aug 2005.

[41] H. Ying, J. Song, X. Renand, W. Wang, "Fingertip Detection and Tracking Using 2D and 3D Information", Proceedings of the seventh World Congress on Intelligent Control and Automation, Chongqing, China, pp.1149-1152, 2008.

[42] H. Zhou, Q. Ruan, "A Real-time Gesture Recognition Algorithm on Video Surveillance", 8th international conference on Signal Processing, Beijing , China, vol.3, Nov 2006.

[43] A. Chaudhary, J.L. Raheja, K. Das, "A Vision based Real Time System to Control Remote Robotic Hand Fingers", Proceedings of IEEE International Conference on Computer Control and Automation (ICCCA 2011), Jeju Island, South Korea, 1-3 May, 2011, pp. 118-122.

Ankit Chaudhary received his Master of Engineering from Dept. of Computer Science, Birla Institute of Technology & Science, Pilani, India and Ph.D. in Computer Vision, from CEERI/Dept. of Computer Science, BITS Pilani, India. His areas of research interest are Computer Vision, Artificial Intelligence and Machine Learning.

J. L. Raheja received his M.Tech from IIT Kharagpur, INDIA and Ph.D. from Technical University of Munich, GERMANY. Currently he is senior scientist at CEERI, Pilani, INDIA. His areas of research interest are Digital Image Processing, Embedded Systems and Human Computer Interface. He has been a DAAD fellow. He has published more than 80 papers in International Journals and Conferences of repute.

Shekhar Raheja received his B.Tech in electronics engineering from Rajasthan Technical University and currently pursuing his M.S from TU Kaiserslautern, Department of Computer. Science, Germany. His areas of research interest are Digital Image Processing, Real Time & Embedded Systems. He has worked as research assistant at CEERI Pilani.