Real-time Multi-Objective Hand Posture/Gesture Recognition by Using Distance Classifiers and Finite State Machine for Virtual Mouse Operations

Alper Aksaç, Orkun Öztürk, and Tansel Özyer

Computer Engineering Department, TOBB University of Economics & Technology, Ankara, Turkey
aaksac@etu.edu.tr, oozturk@etu.edu.tr, ozyer@etu.edu.tr

Abstract

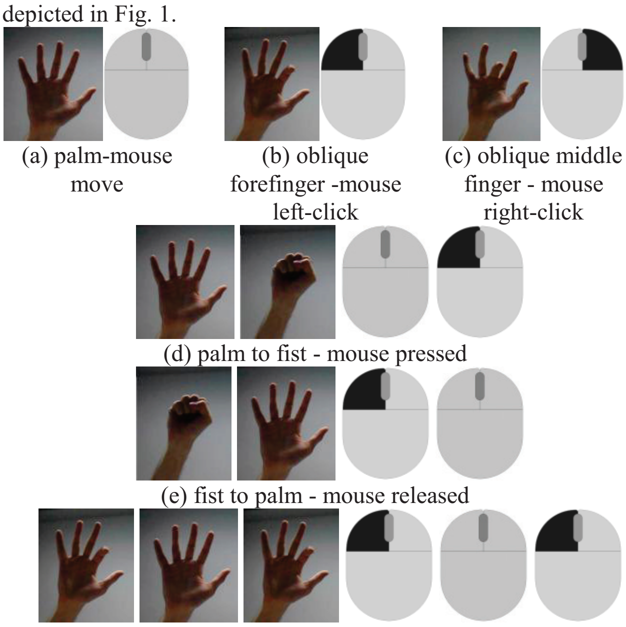
Cameras that are connected to computers record sequence of digital images of human hand in order to interpret human posture/gesture. Human hand posture/gesture recognition has been utilized for providing virtual reality mechanism and it is still an ongoing research in human-computer interaction (HCI) community. Virtual reality can be operated on a particular program but it will be more effective if the entire system can be controlled for the sake of generality. Another direction is the applicability of virtual reality in real time. In this paper, we have developed a virtual mouse system that can recognize the pre-defined mouse movements in real time regardless of the context. Our real time hand recognition system is three fold. 1) skin detection, 2) feature extraction and 3) recognition. For recognition, various features with their own objectives are constructed from hand postures andcompared according to the similarity measures and the best- matched posture is used as a mouse action to control the cursor of the computer.

1. Introduction

Recognition of hand postures is the fundamental problem that has to be solved while constructing a virtual mouse system.

Several approaches have been studied for solving a problem of recognition of hand postures/gestures. Pattern recognition has been utilized for this purpose. One particular way is decision-theoretic approach. There are several methods such as distance classifiers, template matching, conditional random field model (CRF), dynamic time warping model (DTW), Bayesian network, Fisher's linear discriminant model, time-delayed neural networks (TDNN), fuzzy neural networks, discriminant analysis. Some studies have followed hybrid models. Some examples are: k-nearest neighbor combined with Bayesian classifier; least-squares estimator with ANFIS network; incorporation of Markov chains and independent component analysis; hybrid statistical classifiers; use of self-organizing feature maps, simple recurrent network with hidden Markov model [1], [11 - 20]. Our method for recognition of hand postures is using distance classifiers and for the recognition of gestures, we have used finite state machines.

After the extraction of structures from hand postures, comparison of structures between real-time taken image (a frame of live video stream) and images that are belong to a posture which are taken earlier is made based on the similarity of structures. We took a posture as a selected one that has the biggest similarity value of comparison. In our system, similarity of fingers that inherently defines their own objectives are utilized. we defined four hand postures for the six mouse event. These postures and their corresponding mouse actions are depicted in Fig. 1.


(a) palm-mouse move

(b) oblique forefinger -mouse left-click

(c) oblique middle finger - mouse right-click

(d) palm to fist - mouse pressed

(e) fist to palm - mouse released

(f) open palm, oblique forefinger sequence - double left-click

Fig. 1. Hand postures/gestures and their correspondence mouse actions

The remainder of this paper is organized as follows: Section 2 gives description of technique used for hand skin detection and image processing that are made for enhancement. Extraction of key features is described in Section 3. Construction of polygonal structures from key points and the comparisons made are defined in Section 4. In Section 5, a summary of paper takes place.

2. Skin Detection

Using the color cue of skin regions for hand segmentation is a time-efficient process. Since the structural methods don't meet real-time necessity mostly, utilization of color attributes makes it more reliable. For this purpose, several methods have been proposed. Many of them use color-space transformation. Comparison between color-space transformations for skin detection discussed at [2, 3]. RGB is the most popular color space for most available image formats in many applications. Discrimination of skin color is one primary focus of interest. RGB can be converted with linear\nonlinear transformation in order to minimize the overlap between skin and non-skin pixels with robust parameters against varying illumination conditions [4]. The orthogonal RGB color space reduces the redundancy present in channels and represents the color with statistically independent components. Also, the luminance and chrominance components are explicitly separated; these spaces are prominent choice for skin detection. The YC_bC_r space represents color as luminance (Y) computed as a weighted sum of RGB values, and chrominance (C_b and C_r) computed by subtracting the luminance component from B and R values. The YC_bC_r space is one of the most popular choices for skin detection [2]. Working in the YC_bC_r space, it was found that the ranges of C_b and C_r most representative for the skin-color reference map were in (1) [5].

$$77 \le C_b \le 127 \text{ and } 133 \le C_r \le 173. \qquad (1)$$

We used this values (1) to threshold C_b and C_r components of transformed image under constant synthetic light and got good results to detect hand regions. In Fig. 2 original colored image (a) and thresholding binary image (b) containing skin regions is shown. Hence only using threshold values to determine flesh colored pixels is not completely efficacious, we used histograms of binary C_b and C_r images for employing histogram back projection and determining skin areas more reliably [6].

2.1. Histogram Back Projection

The back projection is the re-application of the modified histogram to the original image, functioning as a look-up table for pixel brightness values to record how well the pixels fit the distribution of pixels in a histogram model. We described above how to find skin-color pixel in a color image. Thus, we have a histogram of flesh color then we can use back projection to find flesh color areas in an image. By comparing images in Fig. 2(d) we showed the difference between generated binary images which one is the thresholding image and the other is the image generated by histogram back projection. Although back projected image has more noisy data, hand region is extracted better. These noises are not very important as we compute and use the biggest contour as an area of hand region. Put another way, noises are cleaned by selecting the biggest contour. Since good extraction of features is an important task for the high recognition rate of postures, after obtaining binary image we made a set of image processing operations to ensure better extraction of key features.

2.2. Image Processing

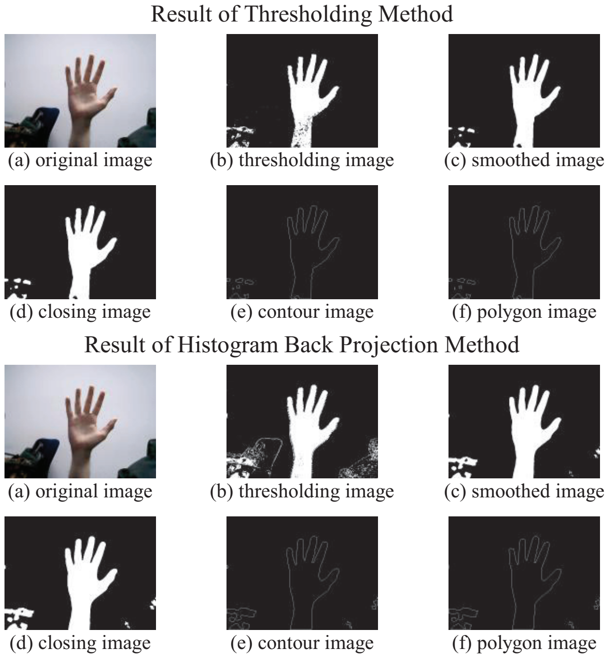
At the first step of an image processing phase median filter was applied to binary image to remove noisy data. Median filter runs through the image entry by entry, replacing each entry with the median of neighboring entries. In Fig. 2(c) median filter applied image is shown. After that step, hand region is still need to be improved by filling spaces to acquire a good contour of area. To do so, morphological image processing has been applied. We have used a close operation defined as follows in (2) where f is image, s is structuring element, $\oplus$ is the dilation operation, and $\ominus$ is the erosion operation. The result of applying closing operation on smoothed image is depicted in Fig. 2(d).

$$f \cdot s = (f \oplus s) \ominus s. \qquad (2)$$

2.2.1. Contour Finding

A contour is a list of points that represent, in one way or another, a curve in an image. An image contour is necessary in recognition. The centroid of interior region of hand and characteristic points of a contour of the region represent the structural features of hand [1]. If we are drawing a contour, it is common to approximate a contour representing a polygon with another contour having fewer vertices. The polygonal approximation of the shape consists on finding significant vertices along the contour such that these vertices constitute a good approximation of the original contour. A classic approach to this problem is to take the high curvature points (i.e., points with high absolute value of curvature) as significant vertices [7]. Contour representation of binary image in Fig. 2(d) is depicted in Fig. 2(e). We used Suzuki's algorithm for contour finding [8]. The centroid of the region and the vertices of a polygon that approximates a region contour constitute characteristic points that are used for defining a structural representation of an image [1]. For a contour depicted in Fig. 2(e), its corresponding polygon is shown in Fig. 2(f).

As can be seen in Fig. 2(f), there is more than one polygon already. But only one of the existent polygons is representing the hand region of image. For this reason, we presumed that the polygon which has the biggest area computed as the polygon corresponding to the hand region.


Result of Thresholding Method

(a) original image (b) thresholding image (c) smoothed image

(d) closing image (e) contour image (f) polygon image

Result of Histogram Back Projection Method

(a) original image (b) thresholding image (c) smoothed image

(d) closing image (e) contour image (f) polygon image

Fig. 2. Step by step contour extraction of hand region

3. Feature Extraction

After obtaining contour representation of hand region, we need to extract good features from contours to carry out a successful comparison between the patterns. Our experiments show that good features for the postures depicted in Section 1 would be fingertips. Hence we have defined our gestures for the mouse actions and these gestures either include all fingers or one finger missing or none of them, fingertip attributes would give fine clues. To extract fingertip points, we have made convexity analysis of contours. Convex hull with convexity defects aids in understanding the shape of the object or contour. Convexity defects are effective in resolving the shapes of complex objects.

3.1. Hull Creation

We used Sklansky's algorithm for convex hulls of polygons [9]. It is a sequential linear time algorithm used to compute the convex hull for a set of points. It has been proposed to overcome with failures on intersecting complex polygons. Although it fixes many failures significantly, the algorithm still fails with certain complex polygons. Since we are not concerned with complex polygons in postures, this is ideal for posture analysis and offers a complexity of O (N) [10]. Sklansky proposed using the 3-coins algorithm to find the convex hull of a simple n-polygon (in R^2).

The algorithm was run on the image showed in Fig. 2(f) and yielded the result showed in Fig. 3.

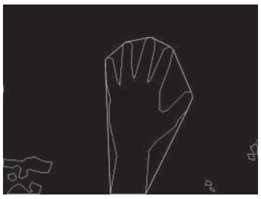
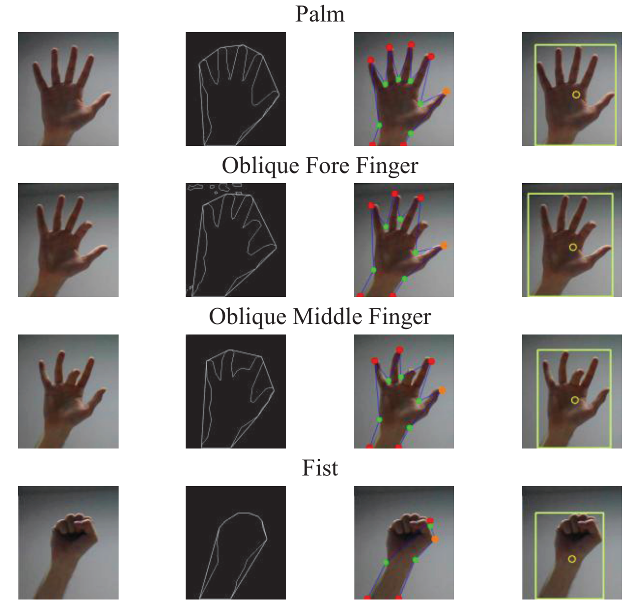


Fig. 3. Convex Hull

3.2. Defect Detection

From the hull and original contour, we have determined convexity defects (i.e., a finger concavity), which are basically holes in difference between contour and convexity hull. Each defect is represented with hull distance from contour [19]. Gained points of defects and lines concatenating nodes for defined postures are depicted Fig. 4. In the Fig. 4(c) images, red points represent start and end points of the convexity defects while green ones symbolizing depth points.

In this section, we have extracted key features from hand postures. Recognition of postures/gestures defined in the next section.



Palm

Oblique Fore Finger

Oblique Middle Finger

Fist

(a) original image (b) convex hull (c) convexity defects (d) bounding box and centroid

Fig. 4. Representation of defect detection for defined postures

4. Recognition of Postures/Gestures

4.1. Posture Recognition

For the recognition of postures, a distance classifier method is used. In the previous section, we have detected convex and concave points and then lines between these points. As a next step, we measured total line length (blue lines in Fig. 4(c)) and bounding box area of hand region (Fig. 4(d)), and then ratio of these measures. We took 10 observation images per posture and showed their corresponding ratio values in Table 1.

Table 1. Path Length

Fist	Palm
1009	1510
959	1392
1016	1524
963	1483
1042	1484
1040	1387
995	1473
1009	1491
1011	1495
985	1490

Table 2. Distance Measures (Fore Finger)

Error for Fore Finger	Error for Middle Finger
3172	73
3361	52
3256	41
3205	68
3474	42
3392	45
2690	173
3314	53
3281	58
2657	178

From Table 1, the biggest ratio value of the fist posture and the smallest ratio value of the palm posture are taken and the average value of these values is used as a threshold to distinguish these postures from each other. From samples, 1215 value is determined as the threshold value. We calculated this ratio value for every captured video frame and checked it against value 1215. If it was smaller than 1215, then we labeled it as a fist posture, and if it is not we took image for further processing.

In the images containing postures except fist, there are 4 or 5 concave (red) points that take place above the centroid of the hand region in the coordinate system. We labeled the one whose y value is the biggest as thumb. If there were 5 points, the closest point to the thumb along the x axis was labeled as forefinger and the second one as middle finger. These points were stored for a use in the next frame. For a frame having 4 finger points (either forefinger posture or middle finger posture), we get the location of the point closest to the thumb. Distance between this point (ɵ) and the previously stored points (θ) are calculated by the minimum mean square error equation below (3) and yielded results are shown in Table 2 and Table 3. First column shows the error value for forefinger and second column shows the error value for middle finger. If first value is smaller than the second one, we label it oblique forefinger posture; else labeling it oblique middle finger posture.

$$MSE(\theta) = E[(\theta - \text{ɵ})^2] \qquad (3)$$

For every posture, we captured 20 test images. In Table 4 ratio and in Table 5 and 6 error values along with postures are figured. Diagrams below the tables (Fig. 5 and Fig. 6) show the ratio of correctness.

We also obtained ratio values for fore finger and middle finger postures bigger than threshold value as we did for palm posture, so the ratio values for these postures wasn't shown in Table 5 and 6 needlessly.

Table 3. Distance Measures (Middle Finger)

Error for Fore Finger	Error for Middle Finger
433	4082
593	3218
1010	2521
674	2977
820	2669
866	2525
610	3293
500	3637
810	2965
481	3560

Table 4. Path Length

Fist	Palm
1016	1487
943	1459
915	1609
1037	1460
1076	1429
985	1427
1078	1463
1191	1441
1167	1454
1044	1448
1037	1436
1084	1443
1008	1463
1075	1459
1166	1479
1011	1477
1111	1483
973	1475
1075	1436
983	1503

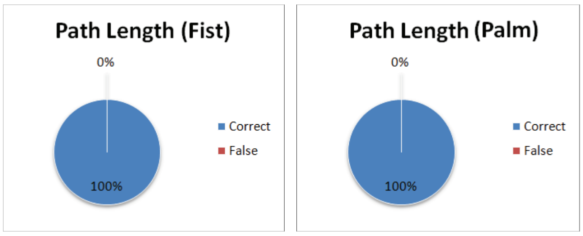


Fig. 5. Indicators of success for path length

Table 5. Distance Measures (Fore Finger)

Error for Fore Finger	Error for Middle Finger
0	0
4034	16
4250	4
4034	16
4141	9
4361	11
4250	14
4105	17
4034	16
3725	49
3626	64
3826	36
3725	49
4072	10
4292	12
4005	13
3793	29
3898	20
3557	65
3690	40

Table 6. Distance Measures (Middle Finger)

Error for Fore Finger	Error for Middle Finger
0	0
169	2725
226	2500
445	1933
0	0
81	13394
145	12740
145	12740
122	12913
36	14093
82	13421
445	10834
148	12769
409	11072
130	13025
493	10660
250	12205
554	10525
477	10970
292	12025

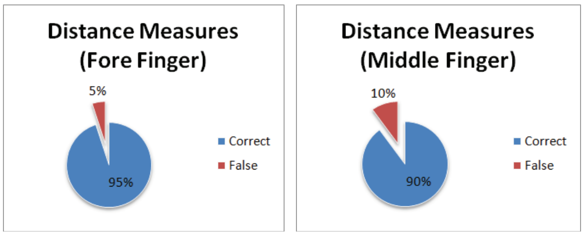


Fig. 6. Indicators of success for distance measures

4.2. Gesture Recognition

By the recognition of 4 postures we have handled 3 mouse actions already which are they “mouse move”, “left click” and “right click”. To recognize other gestures, we needed a mixture of these postures in a sequence. For this, we defined a gesture as a sequence of states and modeled by a finite state machine (FSM). FSM has been well received in various areas that are in common since it basically uses string matching between a data sequence and the state sequence of an FSM. Gesture recognition is also relevant with FSM in this aspect [22, 23].

4.2.1. Mouse Pressed and Released

These mouse actions are sequences of palm and fist postures. As shown in the diagram, mouse is pressed if posture turns from palm to fist or remain as fist. On the other hand, if posture returns to palm from fist it is described as mouse released action.

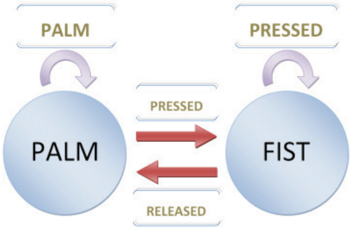


Fig. 7. The FSM of mouse released and pressed gestures with two states

4.2.2. Double Left Click

When oblique forefinger posture (mouse left click action) is detected in a frame, a timer is triggered and when another left click action is detected in another frame we look for if previous frame was palm and the timer is less than 500 milliseconds, if such is the case then we label it as double left click action.

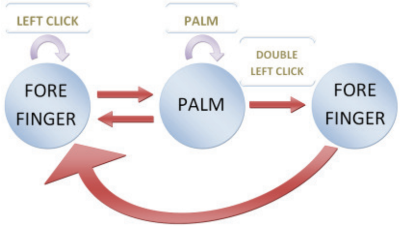


Fig. 8. The FSM of double left click gesture with three states

5. Summary

A distance classifier method for the recognition of hand postures and the finite state machines method for the hand gestures are presented in the paper. The main objective of research was to construct simple polygons from hand postures and recognize them by using distance classifiers and also was to classify hand gestures by defining FSMs. Besides this, detection of hand region by working on YC_bC_r color space, morphological operations that are made for enhancement, contour finding, and polygonal approximation is presented. On the basis of algorithms and techniques presented in the paper, a Virtual Mouse System was designed and implemented. We have used OpenCV library in Visual Studio C++ on Win7 OS for the implementation.

For a further work, we are on the removing of head region in the case of it has bigger area than a hand region by making a blob analysis.

Application video related to the subject can be downloaded from http://youtu.be/kQxiFaZbOfA.

Acknowledgement: This study was supported by Scientific and Technical Research Council of Turkey(Grant Number Tübitak EEEAG 109E241)

6. References

[1] Mariusz Flasinski, Szymon Myslinski, “On the use of graph parsing for recognition of isolated hand postures of Polish Sign Language”, Pattern Recognition, vol. 43, pp. 2249-2264, Issue 6, June, 2010.

[2] P. Kakumanu, S. Makrogiannis, N. Bourbakis, “A survey of skin-color modeling and detection methods", Pattern Recognition, vol. 40, pp. 1106-1122, ssue 3, March, 2007.

[3] Vladimir Vezhnevets, Vassili Sazonov and Alla Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques", *Cybernetics*, Citeseer, vol.85, pp. 85-92, 2003.

[4] Tarek M. Mahmoud, "A New Fast Skin Color Detection Technique ", *World Academy of Science, Engineering and Technology 43,* 2008.

[5] Francesca Gasparini and Raimondo Schettini, "Skin segmentation using multiple thresholding", *Proc. SPIE 6061, 60610F,* 2006.

[6] M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, "Skin detection in video under changing illumination conditions," *Pattern Recognition, 2000. Proceedings. 15th International Conference on* , vol.1, no., pp.839-842 vol.1, 2000.

[7] C. R. P. Dionisio, H. Y. Kim, “A Supervised Shape Classification Technique Invariant Under Rotation and Scaling,” in *Proc. Int. Telecommunications Symposium*, (Natal, Brasil), pp. 533-537, 2002.

[8] Satoshi Suzuki, KeiichiA be, “Topological structural analysis of digitized binary images by border following”, Computer Vision, Graphics, and Image Processing, Volume 30, Issue 1, April 1985, Pages 32-46

[9] Jack Sklansky, “Finding the convex hull of a simple polygon”, Pattern Recognition Letters, vol. 1, Issue 2, pp. 79-83, December, 1982.

[10] M.M. Youssef, K.V. Asari, R.C. Tompkins, J. Foytik, "Hull convexity defects features for human activity recognition," *Applied Imagery Pattern Recognition Workshop (AIPR), 2010 IEEE 39th* , pp.1-7, 13-15 Oct. 2010.

[11] Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi Ur Rehman, Israr Ahmed, "Pakistan Sign Language Recognition Using Statistical Template Matching", Proceedings of World Academy Of Science, Engineering and Technology, vol. 3, January, 2005.

[12] Liu Te-Cheng, Wang Ko-Chih, A. Tsai, Wang Chieh-Chih, "Hand posture recognition using Hidden Conditional Random Fields," *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pp.1828-1833, 14-17 July, 2009.

[13] Suk Heung-Il, Sin Bong-Ke, Lee Seong-Whan, "Recognizing hand gestures using dynamic Bayesian network," *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, vol., no., pp.1-6, 17-19 Sept. 2008.

[14] H.-I. Suk, Sin Bong-Kee, Lee Seong-Whan, "Robust modeling and recognition of hand gestures with dynamic Bayesian network," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* , pp.1-4, 8-11 Dec., 2008.

[15] H.H. Aviles-Arriaga, L.E. Sucar, C.E. Mendoza, B. Vargas, "Visual recognition of gestures using dynamic naive Bayesian classifiers," *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop on* , pp. 133- 138, 31 Oct.-2 Nov., 2003.

[16] Heung-Il Suk, Bong-Kee Sin, Seong-Whan Lee, “Hand gesture recognition based on dynamic Bayesian network framework”, Pattern Recognition, vol. 43, Issue 9, pp. 3059-3072, September, 2010.

[17] P. Modler, T. Myatt, "Recognition of separate hand gestures by Time-Delay Neural Networks based on multi-state spectral image patterns from cyclic hand movements," *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pp.1539-1544, 12-15 Oct., 2008.

[18] B. Tusor, A.R. Varkonyi-Koczy, "Circular fuzzy neural network based hand gesture and posture modeling," *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE* , pp.815-820, 3-6 May, 2010.

[19] Daniel B. Dias, Renata C. B. Madeo, Thiago Rocha, Helton H. Biscaro, Sarajane M. Peres, "Hand movement recognition for Brazilian Sign Language: A study using distance-based neural networks," *Neural Networks, IEEE - INNS - ENNS International Joint Conference on,* pp. 697-704, 2009 International Joint Conference on Neural Networks, 2009.

[20] Wen Gao, Gaolin Fang, Debin Zhao, Yiqiang Chen, “A Chinese sign language recognition system based on SOFM/SRN/HMM”, Pattern Recognition, vol. 37, Issue 12, pp. 2389-2402, December, 2004.

[21] Cristina Manresa, Javier Varona, Ramon Mas and Francisco J. Perales, "Hand Tracking and Gesture Recognition for Human-Computer Interaction", Electronic Letters on Computer Vision and Image Analysis 5(3), pp. 96-104, 2005.

[22] Pengyu Hong, Matthew Turk, Thomas S. Huang, "Constructing Finite State Machines for Fast Gesture Recognition", In *Proceedings of the International Conference on Pattern Recognition - Volume 3* (ICPR '00), vol. 3., pp. 691-694, IEEE Computer Society, Washington, DC, USA, 2000.

[23] Pengyu Hong; Turk, M.; Huang, T.S.; , "Gesture modeling and recognition using finite state machines," *Proceedings. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000*, pp.410-415, 2000.