

A Real-Time Hand Gesture System based on Evolutionary Search

Juan Wachs*, Helman Stern and Yael Edan

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er-Sheva, Israel, 84105, {helman, yael,juan}@bgu.ac.il

Michael Gillam, Craig Feied, Mark Smith and Jon Handler

Institute for Medical Informatics, Washington Hospital Center, 110 Irving Street, NW, Washington, DC, 20010, {gillam, feied, smith, handler}@medstar.net

Abstract - In this paper, we consider a vision-based system that can interpret a user's gestures in real time to manipulate objects within a medical data visualization environment. Dynamic navigation gestures are translated to commands based on their relative positions on the screen. Static gesture poses are identified to execute non-directional commands. This is accomplished by using Haar-like features to represent the shape of the hand. These features are then input to a Fuzzy C-Means Clustering algorithm for pose classification. A probabilistic neighborhood search algorithm is employed to automatically select a small number of Haar features, and to tune the fuzzy c-means classification algorithm. The gesture recognition system was implemented in a sterile medical data-browser environment. Test results on four interface tasks showed that the use of a few Haar features with the supervised FCM yielded successful performance rates of 95 to 100%. In addition a small exploratory test of the AdaBoost Haar system was made to detect a single hand gesture, and assess its suitability for hand gesture recognition.

Keywords: haar-like features, fuzzy c-means, hand gesture recognition, neighborhood search, computerized medical equipment.

1 Introduction

Computer information technology is increasingly penetrating into the hospital domain. It is important that such technology be used in a safe manner in order avoid serious mistakes leading to possible fatal incidents. Keyboards and mice are today principle method of human – computer interaction. Unfortunately, it has been found that a common method of spreading infection from one person to another include computer keyboards and mice in intensive care units (ICUs) used by doctors and nurses [1]. Introducing a more natural human computer interaction (HCI) will have a positive impact in today's hospital environment. The basis of human-human communications is speech and gesture (including facial expression, hand and body gestures and eye gaze). In FAcE MOUSE [2] a

surgeon can control the motion of the laparoscope by simply making the appropriate face gesture, without hand or foot switches or voice input. Here we explore only the use of hand gestures which can in the future be further enhanced by other modalities. A vision-based gesture capture system to manipulate windows and objects in within a graphical user interface (GUI) is proffered. Current research to incorporate hand gestures into doctor-computer interface have appeared in Graetzl et al. [3]. They developed a computer vision system that enables surgeons to perform standard mouse functions (pointer movement and button presses) with hand gestures. Zeng et al. [4] by tracking position of the fingers they gather quantitative data about the breast palpation process for further analysis. Much of the research on real-time gesture recognition has focused on exclusively dynamic or static gestures. In our work we consider hand motion and posture simultaneously. This allows for much richer and realistic gesture representations. Our system is user independent without the need of a large multi-user training set. We use a fuzzy c-mean discriminator along with Haar type features. In order to obtain a more optimal system design we employ a neighborhood search method for efficient feature selection and classifier parameter tuning. The real time operation of the gesture interface was tested in a hospital environment. In this domain non contact aspect of the gesture interface avoids the problem of possible transfer of contagious diseases through traditional keyboard/mice user interfaces.

A system overview is presented in Section 2. In Section 3 we describe the segmentation of the hand from the background. Section 4 deals with feature extraction and pose recognition. The results of performance tests for the FCM hand gesture recognition system appear in Section 5. In section 6 we report on a small exploratory test of the AdaBoost-Haar detector on one of our hand gestures. Section 7 concludes the paper.

2 System Overview

A web-camera placed above the screen (Figure. 1(a)) captures a sequence of images like those shown in Figure

1(b). The hand is segmented using color, B/W threshold and various morphological image processing operations. The location of the hand in each image is represented by the 2D coordinates of its centroid; and mapped into one of eight possible navigation directions of the screen (see Figure 2) to position the cursor of a virtual mouse. The motion of the hand is interpreted by a tracking module. At certain points in the interaction it becomes necessary to classify the pose of the hand. Then the image is cropped tightly around the blob of the hand and a more accurate segmentation is performed. The postures are recognized by extracting symbolic features, from the sequence of images. The sequence of features is interpreted by a supervised FCM that has been trained to discriminate various hand poses. The classification is used to bring up X-rays images, select a patient record from the database or move objects and windows in the screen. A two layer architecture is used. The lower level provides tracking and recognition functions, while the higher level manages the user interface.



Figure 1. Gesture capture

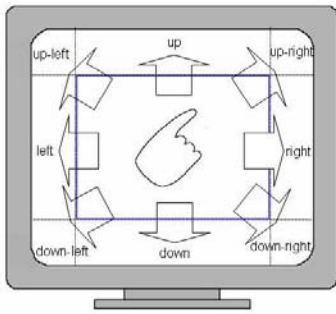


Figure 2. Screen navigation map

3 Segmentation

In order to track and recognize gestures, the CAMSHIFT [5] algorithm is used together with an FCM algorithm [6]. For CAMSHIFT, a probability distribution image of the hand color is created using a 2D hue-saturation color histogram [7]. This histogram is used as a look-up-table to convert the acquired camera images into a corresponding skin color through a process known as back propagation. Thresholding to black and white followed by morphological operations is used to obtain a single component for further processing to classify the gestures.

The initial 2D histogram is generated in real-time by the user in the ‘calibration’ stage of the system. The interface preview window shows an outline of the palm of the hand gesture drawn on the screen. The user places his hand within the template while the color model histogram is built (Figure 3), after which the tracking module (Camshift) is triggered to follow the hand. The calibration process is initiated by the detection of motion of the hand within the region of the template. In order to avoid false motion clues originated by non hand motion a background maintenance operation is maintained. A first image of the background is stored right after the application is launched, and then background differencing is used to isolate the moving object (hand) from the background. Since background pixels have small variations due changes in illumination over an extended period of time, the background image is dynamically maintained. Background variations are identified by threshold the absolute difference between two consecutive frames. If the difference is under some threshold t_1 , then the current images contain only a background, otherwise, an upper threshold level t_2 is checked to test whether the present object is a hand. In case that the current image is a background, the background stored image is updated using a running smoothed average.

$$B_{cc}(i, j) = (1 - \alpha) * B_{cc-k-1}(i, j) + \alpha * f(i, j) \quad (1)$$

Where B_{cc} is the updated stored background image at frame k , B_{cc-1} is the stored background image at frame $k-1$, α is the smoothing coefficient (regulates update speed), $f(i,j)$ is the current background image at frame k . Small changes in illumination will only update the background while huge changes in intensity will trigger the tracking module. It is assumed that the hand is the only skin colored object moving on the area of the template.

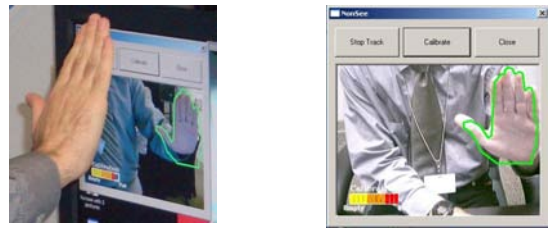


Figure 3. User hand skin color calibration

A low threshold and an open and a close morphology operations followed by the selection of the biggest component are applied to obtain a single connected blob, see Figure 4.



Figure 4. Image Preprocessing of the pose

4 Feature Extraction and Pose Recognition

4.1 Gesture Vocabulary

We currently provide three methods for generating mouse button clicks. The first two methods, “click, and double-click”, consists in moving the cursor to the desired position and holding the hand stationary for a short time, performing the gesture similar to Figure 5(a)(b) will activate the command ‘click’/‘double-click’ of the virtual sterile mouse in the current position of the cursor. The third method, “drag” (Figure 5(c)), after being activated as the previous ones, will perform the drag command on the current view, while the hand moves to one of the 8 directions. When the hand returns to the ‘neutral area’ the command is terminated.

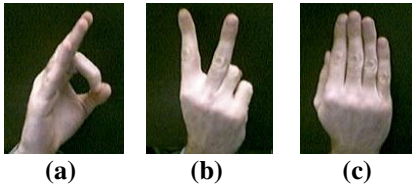


Figure 5. The gesture vocabulary

4.2 Hand Tracking and Pose Recognition

We classify hand gestures using a simple finite state machine (Figure 6). When the doctor wishes to move the cursor over the screen, he moves his hand out of the ‘neutral area’ to any of the 8 directions.

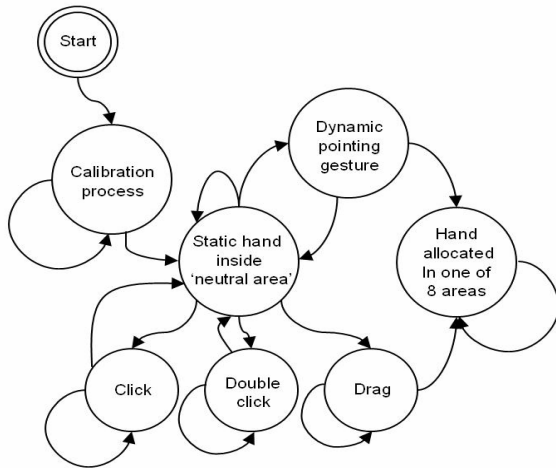


Figure 6. State machine for the gesture-based medical browser

The interaction is designed in this way because the doctor will often have his hands in the ‘neutral area’ without intending to control the cursor. While the hand is in

one of the 8 quadrants, the cursor moves in requested direction (Figure 7).

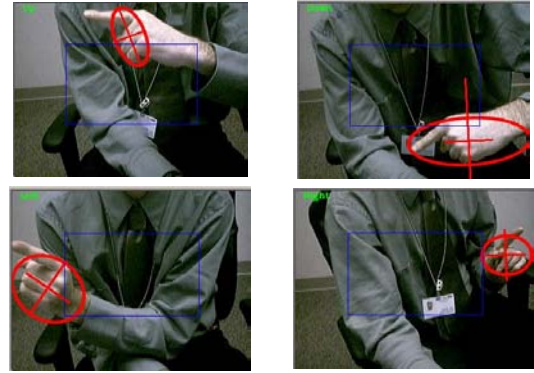


Figure 7. Four quadrants mapped to cursor movement

To facilitate positioning, we map hand motion to cursor movement. Small, slow hand (large fast) motion cause small (large) pointer position changes. In this manner the user can precisely control pointer alignment. When a doctor decides to perform a click, double-click, or drag with the virtual mouse, he/she places the hand in the ‘neutral area’ momentarily. This method differentiates between navigation and precise commands.

4.3 Haar Features

Basically, the features of this detector are weighted differences of integrals over rectangular sub regions. Figure 8(a)-(d) visualizes the set of available feature types, where black and white rectangles correspond to positive and negative weights, respectively. The feature types consist of four different edge-line features. The learning algorithm automatically selects the most discriminate features considering all possible feature types, sizes and locations. The feature types are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses. Their main advantage is that they can be computed in constant time at any scale.

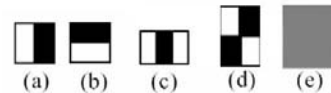


Figure 8. Extended integral rectangle feature set

Each rectangle feature is computed by summing up pixels within smaller rectangles:

$$f_i = \sum_{i \in I = \{1, \dots, N\}} \omega_i * RecSum(r_i) \quad (2)$$

With weights $\omega_i \in \mathfrak{R}$, rectangles r_i and their number N . Only weighted combinations of pixel sums of two rectangles are considered. The weights have opposite signs

(indicated as black and white in Figure. 8), and are used to compensate between differences in area. Efficient computation is achieved by using summed area tables. Average block features have been added to the original feature set of Viola-Jones. Features f_1 , f_2 , f_3 , and f_4 (Figure 8(a)-(d) respectively). The augmented rectangle feature f_5 (Figure 8.(e)) has been shown to extend the expressiveness and versatility of the original features leading to more accurate classification. Given that the basic resolution of the classifier is 100×100 , the exhaustive set of rectangle features is quite large ($> 750,000$). Even though computing each feature is efficient, the complete set is prohibitively expensive [8]. A rectangle r in the image, can be defined by the (x,y) position of its upper left corner, and by its width w and its height h . We constrain the total set of rectangles in an image, by using the relation: $x=w*n$, and $y=h*m$ where n, m are integer numbers. Hence, the total number of possible rectangles is $< 13,334$. In general, the total number of possible rectangles that fix in a square image with size L is $\Sigma(L/n)^2$, for $n=1$ to L .

4.4 Pose Recognition

In our system we reduce the Haar rectangular positions severely to a set of ‘selected’ rectangles v . These rectangles are limited to lie within a bounding box of the hand tracking window, and are obtained by dividing the window in m rows and n columns, and for each cell decide whether it is selected ‘1’ or not ‘0’. A more elaborated strategy enables one to define the type of feature for selected rectangles. Therefore, a set of rectangles in a window is defined by a tuple $\{n,m,t\}$, where n,m are columns and rows; and $t=\{t_1,\dots,t_i,\dots,t_v\}$ represent the type of feature of rectangle i indexed row wise from left to right. The feature type t can take integer values from 0 to 5, where 0 means that the rectangle is not selected, and 1,2,3,4,5 represent features of type f_1, f_2, f_3, f_4 and f_5 respectively. The hypothesis expressed in [8] is that a very small number of these features can be combined to form an effective classifier. As opposed to [8] our learning algorithm is not designed to select a single rectangle feature which best separates the positive and negative for each stage of a cascade of classifiers, instead, we evaluate a set of rectangle features simultaneously, which accelerates the process of feature selection. The Haar features selected are input into our hand gesture FCM recognition system architecture. Note that the feature sizes are automatically adjusted to fit into the dynamically changing bounding box created by out tracking system.

4.5 Optimal Feature Selection based on Evolutionary Search

The process of feature selection and finding the parameters of the FCM algorithm for classifying hand gesture sets uses a probabilistic neighborhood search (or Evolutionary Search) (PNS) method [9]. The PNS selects

samples in a small neighborhood around the current solution based on a special mixture type point distribution model:

$$PS(x|h) = \begin{cases} h, & x = 0 \\ h((1-h)^{|x|})/2, & x = \pm 1, \pm 2, \dots, \pm(S-1) \\ ((1-h)^{|x|})/2, & x = \pm S \end{cases} \quad (3)$$

Where,

S = maximum number of step increments.
 h = probability of no change
 x_j = a random variable representing the signed (positive or negative coordinate direction) number of step size changes for parameter P_j .
 $P_S(x|h) = P_r(x = s)$ the probability of step size s , given h .

Figure 9 shows an example of the convergence behavior of the PNS algorithm for 5 randomly generated starting solutions.

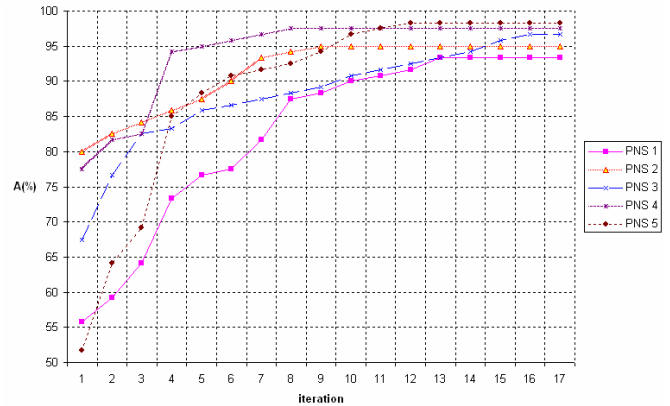


Figure. 9 Convergence curve for five sol. of the PNS alg.

Figure 10 shows the optimal set of features selected by this run. The feature f_4 and f_5 capture whether the posture is a kind of palm based gesture using diagonal line features and average grayscale. Inner-hand regions (such inside the palms) and normal size fingers are detected through f_1 , while f_3 captures the ring finger based on edge properties. Hence, this is quite different from traditional gesture classifiers which rely on parametric models or statistical properties of the gesture.



Figure. 10. Automatically selected features ($f_4, f_1, f_3, f_1, f_1, f_5$) for the 2×3 partition found by PNS.

Note, that the result is a set of common features for all three of our pose gestures. The optimal partition of the bounding box was 2x3 giving 6 feature rectangles. The parameter search routine found both the number of sub blocks and the type of Haar feature to assign to each.

5 Test of the Hand Gesture FCM Classifier

To evaluate the overall performance of the hand gesture tracking and FCM recognition system, we used the Azyxxi Real-time Repository™, which was designed to accommodate multi-data types. Figure 11 shows an example of the user screen of this database.



Figure. 11. Screen shot of Azyxxi controlled by gestures

The data-set consists of 20 trials of each of 4 tasks - Select Record of Patient, Browse X-ray collection, Select specific X-ray and Zoom in Damaged Area. The user was asked to perform the tasks sequentially. The total results for one experienced user are shown in Table 1. The success task rate shows how many times an action (part of the task) was performed correctly without catastrophic errors. Minor errors are related to inaccurate position of the cursor due to fast movements or changes in direction, while catastrophic errors occurred as a result of misclassification of the supervised FCM algorithm.

Table 1. Results of medical tasks using hand gestures

Task	Steps	Trials	Success Task
Select Record of Patient	1	19	94.74%
Browse X-ray collection	2	20	100%
Select specific X-ray	1	20	100%
Zoom in Damaged Area	2	19	94.74%

In general, the results of Table 1, indicate both the ability of the system to successfully track dynamic postures, and classify them with a high level of accuracy.

6 Test of the AdaBoost-Haar Classifier

Most of the work using Haar features is only for detection of a single object in an image and not as features embedded in a multi-object classification system as was done here. Never the less, because of the recent popularity

of using Haar features for object detection it was decided to run a small exploratory test, using only one of our pose gestures, to assess its suitability for multi-hand gesture classification.

For this test we used cascade of boosted classifiers based on an extended set of Haar-like features [10]. Only one gesture was adopted (Figure 5.(a)) to assess the performance of this method. Positive samples were created using an automated process which overlaid one instance of the gesture over several random backgrounds. The original instance was embedded in the background, twisted from -30 to 30 degrees, in the x,y and z axis a total of 1350 positives samples. A set of 5318 negative samples images were obtained from the Sebastian Marcel's repository web-site [11]. The testing including 50 images set and it was created automatically using the same methodology, see Figure 12.



Figure 12. The positive testing set for different angles

Using a cascade of 13 stages of classifiers, the primary results show an average hit rate of 52%, 48% missed and 34 false alarms. These results are for testing not training. The explanation for the low accuracy is due to the fact that the training and testing positive samples were artificially generated. The classifiers could not learn light and geometry changes properly because the single instance used did not create enough variations. Lighting conditions in a rotated image are different than a hand in real conditions. Gestures that include shadows, occlusion and change in geometry must be obtained by true life images. However, for the task of gesture detection, the rectangle features selected by AdaBoost are meaningful and easily interpreted. The features selected seem to focus on the property that the region of the palms is often darker than the far side of the hand (see Figure 13).



Figure 13. The four features selected by AdaBoost in the last stage. The features are shown overlaid on a typical training gesture. The features measures the difference in intensity between the region of the palm and the far side face of the hand

In our hand gesture FCM classifier, the tracking system uses color segmentation which adaptively compensates for variable illumination so there is no need to include varying illumination in our training samples. In the

FCM classifier we obtained good results using only 40 training samples per gesture. Because of the large training set required for the Haar Detector approach it does not look promising. However, further testing is required.

7 Conclusions

In this paper, we consider a vision-based system that can interpret a user's gestures in real time to manipulate windows and objects within a medical data visualization environment. A hand segmentation procedure first extracts binary hand blobs from each frame of an acquired image sequence. Dynamic navigation gestures are translated to commands based on their relative positions on the screen. Static gesture poses are identified to execute non-directional commands. This is accomplished by using Haar-like features to represent the shape of the hand. These features are then input to a Fuzzy C-Means Clustering algorithm for pose classification. A probabilistic neighborhood search algorithm is employed to automatically select a small number of visual features, and to tune a fuzzy c-means classification algorithm. Intelligent handling of features allows non-discriminating regions of the image to be quickly discarded while spending more computation on promising discriminating regions. The gesture recognition system was implemented in a sterile medical data-browser environment. Test results on four interface tasks showed that the use of these simple features with the supervised FCM yielded successful performance rates of 95 to 100%. In addition, a small exploratory test of the AdaBoost Haar detector was made to detect a single hand gesture from our pose set. The results indicated that the required large training set did not warrant the use of this method for our purposes. It should be noted that the Hand Gesture FCM Recognition system used here must find a bounding box around the gesture using color tracking and image processing, and then use the Haar features within the sub boxes of a partition of the bounding box. The AdaBoost Haar system doesn't use a bounding box but searches over the entire image and then classifies the image as containing a hand or not. Although our initial test points out a number of difficulties in using the AdaBoost Haar detector methodology for hand gesture recognition further tests are warranted.

8 Acknowledgement

This project was partially supported by the Paul Ivanier Center for Robotics Research & Production Management, Ben Gurion University.

References

- [1] M. Schultz, J. Gill, S. Zubairi, R. Huber, F. Gordin, "Bacterial contamination of computer keyboards in a teaching hospital," *Infect Control Hosp. Epidemiol.*, vol. 4, no. 24, pp. 302-303, 2003.
- [2] A. Nishikawa, T. Hosoi, K. Koara, D. Negoro, A. Hikita, S. Asano, H. Kakutani, F. Miyazaki, M. Sekimoto, M. Yasui, Y. Miyake, S. Takiguchi, and M. Monden. "FAce MOUSE: A Novel Human-Machine Interface for Controlling the Position of a Laparoscope," *IEEE Trans. on Robotics and Automation*, Vol. 19, No. 5, pp. 825-841, 2003.
- [3] C. Graetzel, T.W. Fong, S. Grange, and C. Baur, "A non-contact mouse for surgeon-computer interaction," *Technology and Health Care*, Vol. 12, No. 3, 2004, pp. 245-257.
- [4] T J. Zeng, Y. Wang, M.T. Freedman and S.K. Mun, "Finger tracking for breast palpation quantification using color image features", *SPIE Optical Engineering*, Vol. 36, No. 12, pp. 3455-3461, Dec. 1997..
- [5] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technical Journal*, pp. 1-15, 1998.
- [6] J. P. Wachs, H. Stern, and Y. Edan, "Cluster Labeling and Parameter Estimation for Automated Set Up of a Hand Gesture Recognition System," *IEEE Transactions in Man, Systems and Cybernetics, Part A* (in press).
- [7] J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes, *Computer graphics: principles and practice*, 2 Ed., Addison Wesley, 1987.
- [8] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In *IEEE Conf. on Computer Vision and Pattern Recogn.*, Kauai, Hawaii, December, 2001.
- [9] H. Stern, J. P. Wachs, Y. Edan, "Parameter Calibration for Reconfiguration of a Hand Gesture Tele-Robotic Control System", in *Proc. of JUSFA 2004 Japan – USA Symp. on Flexible Automat.*, Denver, Colorado, July 19-21, 2004.
- [10] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. *IEEE ICIP 2002*, Vol. 1, pp. 900-903, Sep. 2002.
- [11] Sebastien Marcel's Gesture Database Web Page. Available: www.idiap.ch/~marcel/Databases/main.html