A Real-Time Hand Gesture Interface for Medical Visualization Applications

Juan Wachs[1], Helman Stern[1], Yael Edan[1], Michael Gillam[2], Craig Feied[2], Mark Smith[2], Jon Handler[2]

[1]Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er-Sheva, Israel, 84105,
{helman, yael,juan}@bgu.ac.il.
[2]Institute for Medical Informatics, Washington Hospital Center, 110 Irving Street, NW, Washington, DC, 20010,
{feied,smith,handler,gillam}@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 features, fuzzy c-means, hand gesture recognition, neighborhood search, computerized medical databases.*

Introduction

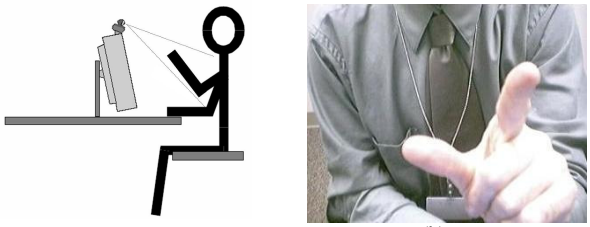
Computer information technology is increasingly penetrating into the hospital domain. It is important that such technology be used in a safe manner in order to avoid serious mistakes leading to possible fatal incidents. Keyboards and mice are today's principle method of human – computer interaction. Unfortunately, it has been found that a common method of spreading infection from one person to another includes computer keyboards and mice in intensive care units (ICUs) used by doctors and nurses (Schultz et al. 2003). Many of these deficiencies may be overcome by introducing a more natural human computer interaction (HCI) , especially an adaptation of speech and gesture (including facial expression, hand and body gestures and eye gaze). In FAce MOUSe (Nishikawa et al. 2003) a surgeon can control the motion of the laparoscope by simply making the appropriate face gesture, without hand or foot switches or voice input. Gaze, is used as one of the diagnostic imaging techniques for selecting CT images by eye movements (Yanagihara and Hama, 2000).
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 within a graphical user interface (GUI) is proffered. Current research to incorporate hand gestures into the doctor-computer interface have appeared in Graetzel et al. (Graetzel et al. 2004). 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. (Zeng et al. 1997) use the tracking position of the fingers to collect 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. Section 6 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 cues, a 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 (of the Haar type) 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.



(a) (b)

Fig. 1. Gesture Capture

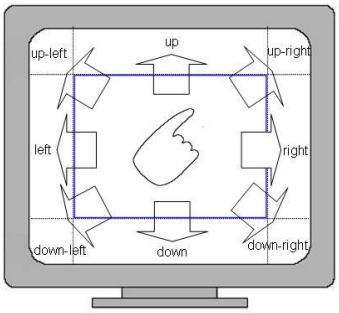


Fig. 2. Screen navigation map

3 Segmentation

In order to track and recognize gestures, the CAMSHIFT (Bradski 1998) algorithm is used together with an FCM algorithm (Wachs et al. 2006). For CAMSHIFT, a hand color probability distribution image is created using a 2D hue-saturation color histogram (Foley et al. 1987). This histogram is used as a look-up-table to convert the acquired camera images into a corresponding skin color probability image through a process known as back propagation. A backprojected image is a probability image learned at the end of the calibration process, and assigns to each pixel a likelihood (0 to 1) of it being classified as a hand pixel. 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/her hand within the template while the color model histogram is built (Fig. 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 immediately 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 must be dynamically changed. Background variations are identified by a threshold applied to the absolute difference between every two consecutive frames. If the differ-

ence 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 backgroundstored image is updated using a running smoothed average.

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

In (1) Bcc_k is the updated stored background image at frame k, Bcc_{k-1} is the stored background image at frame k-1, α is the smoothing coefficient (regulating 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. The process of calibration takes a couple of seconds, and is necessary for every new user since every user has a slightly different skin color distribution and changes in artificial/daylight illumination affects the color model.

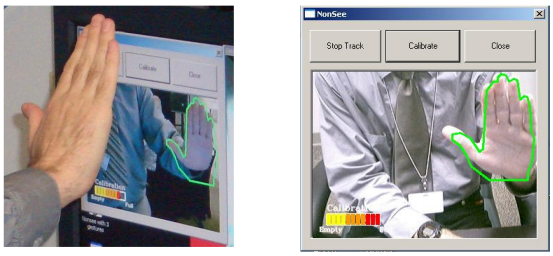


Fig. 3. User hand skin color calibration

A low threshold and open and close morphology operations followed by largest component selection are applied to obtain a single connected blob (see Fig. 4).

Fig. 4. Image processing 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 of 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.

(a) (b) (c)

Fig. 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 direction regions.

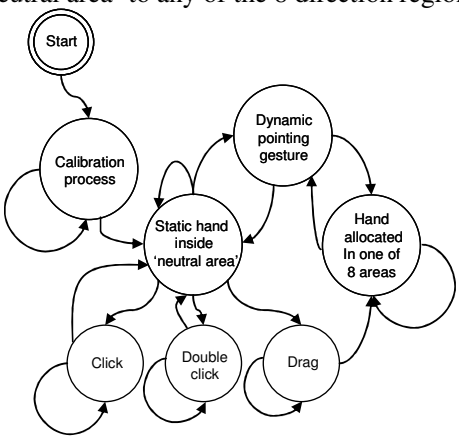


Fig. 6. State machine for the gesture-based medical browser

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 regions, the cursor moves in requested direction (Figure 7).

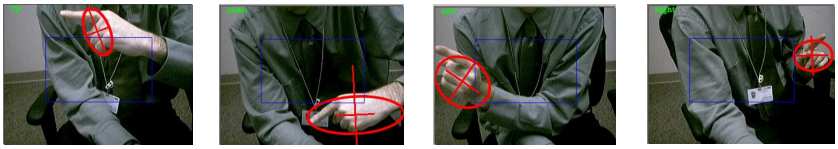


Fig. 7. Four quadrants mapped to cursor movement

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, and use the original image without preprocessing.

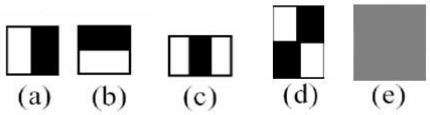


Fig. 8. Extended integral rectangle feature set

Each rectangular feature is computed by summing up pixel values within smaller rectangles:

$$f_i = \sum_{i \in I=\{1,\ldots,N\}} \omega_i * \mathrm{Re}\, cSum(r_i) \qquad (2)$$

In (2) $\omega_i \in \Re$ are weights, r_i is the i*th* rectangle, and N is the number of rectangles. 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. We have added a block average feature (see Fig. 8(c)) to f_1 , f_2 , f_3 , and f_4 (see Fig. 8(a)-(d)) selected from the original feature set of Viola-Jones. The augmented rectangle feature f_5 (Fig. 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 100x100, the exhaustive set of rectangle features is quite large (> 750,000). Even though computing each feature is efficient, the complete set is prohibitively expensive (Viola and Jones 2001). 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 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 and m are integer numbers. Hence, the total number of possible rectangles is less than 13,334.

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. For each cell a binary variable is used to decide whether it is selected or not. A more elaborate 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 $t=\{t_1,\ldots,t_i,\ldots,t_v\}$ 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 indicates 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 Viola and Jones is that a very small number of these features can be combined to form an effective classifier. As opposed to Viola and Jones method, 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 ges-

ture FCM recognition system architecture. Note, that the feature sizes are automatically adjusted to fit into a dynamically changing bounding box created by out tracking system.

4.5 Optimal Feature and Recognition Parameter Selection

The process of feature selection and finding the parameters of the FCM algorithm for classifying hand gesture sets uses a probabilistic neighborhood search (PNS) method (Stern, et al. 2004). The PNS selects samples in a small neighborhood around the current solution based on a special mixture point distribution model:

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

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.

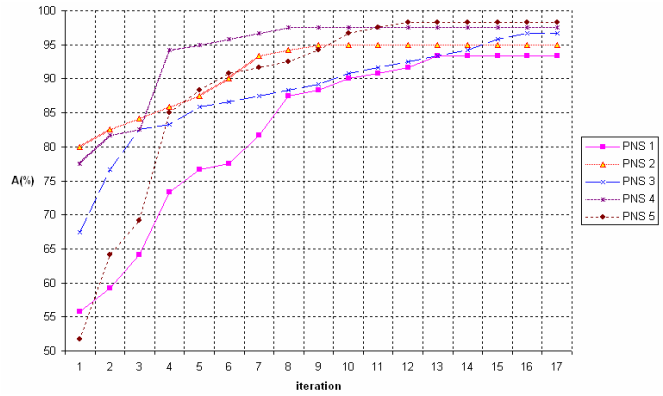


Fig. 9. Convergence curve for five sol. of the PNS alg.

Figure 10 shows the optimal set of features selected by this run. The features f_4 and f_5 capture characteristics of a 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.

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.

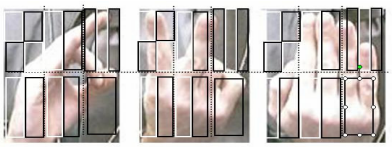


Fig. 10. Automatically selected features (f_4, f_1, f_3, f_1 ,f_1 , f_5) for the 2x3 partition

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 ™ (Azyxxi 2003), which was designed to accommodate multi-data types. 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. 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.

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%

6 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 (Wachs et al. 2005). 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 percent ,which is considered accurate enough for medical browsing and navigation tasks in hospital environments. The explanation for the 5% drop in accuracy is due to confusion between gestures 'b' and 'c' ('double click' and 'drag'), which points to the fact that the training and testing positive samples were not large enough. The classifiers could not learn light and geometry changes properly because the small cadre used did not create enough variations. Gestures that include shadows, occlusion and change in geometry must be obtained by true life images to enrich the training dataset. All of this will be done in a future study. Another issue that must be addressed is the false triggers obtained as a result of a fast moving objects moving between the hand and the camera. An approach to tackle this weakness is to keep a generic 2D histogram of the skin color distribution, and to compare the distance of the candidate object color histogram to the one stored. The generic histogram can be created from a training set offline. Catastrophic errors due to confusion between gestures can be reduced significantly by using the probabilities of gesture occurrences in a transition matrix based on the state machine presented in Fig. 6.

An appealing alternative method for fast recognition of a large vocabulary of human gestures suggests using Haar features to reduce dimensionality in hand attention images, instead of using the MEF space (Cui and Weng 1999).

Additional future work will include recognition of dynamic two handed manipulation gestures for zooming an image, rotating an image, etc. We are interested, as well, to experiment with larger gesture vocabularies to enhance the interaction flexibility to the system.

Acknowledgments

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

References

Azyxxi Online Source (2003) http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=14967

Bradski GR (1998) Computer vision face tracking for use in a perceptual user interface. In Intel Technical Journal, pp 1-15

Cui Y and Weng J (1999) A Learning-Based Prediction-and-Verification Segmentation Scheme for Hand Sign Image Sequence. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21 , num.8, pp 798 – 804.

Foley JD, van Dam A, Feiner SK and Hughes JF (1987) Computer graphics: principles and practice, 2 Ed, Addison Wesley

Graetzel C, Fong TW, Grange S, and Baur C (2004) A non-contact mouse for surgeon-computer interaction. J Tech and Health Care 12:3:245-257

Lienhart R and Maydt J (2002) An Extended Set of Haar-like Features for Rapid Object Detection. In IEEE ICIP 2002 vol:1, pp 900-903

Nishikawa A, Hosoi T, Koara K, Negoro D, Hikita A, Asano S, Kakutani H, Miyazaki F, Sekimoto M, Yasui M, Miyake Y, Takiguchi S, and Monden M (2003) FAce MOUSe: A Novel Human-Machine Interface for Controlling the Position of a Laparoscope. IEEE Trans on Robotics and Automation 19:5:825-841

Schultz M, Gill J, Zubairi S, Huber R, Gordin F (2003) Bacterial contamination of computer keyboards in a teaching hospital. Infect Control Hosp Epidemiol 24:302-303

Stern H, Wachs JP, Edan Y (2004) Parameter Calibration for Reconfiguration of a Hand Gesture Tele-Robotic Control System. In Proc of USA Symp on Flexible Automat, Denver, Colorado, July 19-21

Viola P and Jones M (2001) Rapid object detection using a boosted cascade of simple features. In IEEE Conf on Computer Vision and Pattern Recogn Kauai, Hawaii

Wachs JP, Stern H, and Edan Y (2006) Cluster Labeling and Parameter Estimation for Automated Set Up of a Hand Gesture Recognition System. In IEEE Trans in SMC Part A (in press)

Wachs JP, Stern H (2005) Hand Gesture Interf for Med Visual App Web Site. Available: http://www.imedi.org/docs/references/gesture.htm

Yanagihara Y, Hiromitsu H (2000) System for Selecting and Generating Images Controlled by Eye Movements Applicable to CT Image Display, Medical Imaging Technology, September, vol.18, no.5, pp 725-733

Zeng TJ, Wang Y, Freedman MT and Mun SK (1997) Finger tracking for breast palpation quantification using color image features. SPIE Optical Eng 36:12, pp 3455-3461