9

Face and Gesture Recognition for Human-Robot Interaction

Dr. Md. Hasanuzzaman[1] and Dr. Haruki Ueno[2]
[1]Department of Computer Science & Engineering, University of Dhaka
[2]National Institute of Informatics, The Graduate University for Advanced Studies, Tokyo
[1]Bangladesh, [2]Japan

1. Introduction

This chapter presents a vision-based face and gesture recognition system for human-robot interaction. By using subspace method, face and predefined hand poses are classified from the three largest skin-like regions that are segmented using YIQ color representation system. In the subspace method we consider separate eigenspaces for each class or pose. Face is recognized using pose specific subspace method and gesture is recognized using the rule-based approach whenever the combinations of three skin-like regions at a particular image frame satisfy a predefined condition. These gesture commands are sent to robot through TCP/IP wireless network for human-robot interaction. The effectiveness of this method has been demonstrated by interacting with an entertainment robot named AIBO and a humanoid robot Robovie.

Human-robot symbiotic systems have been studied extensively in recent years, considering that robots will play an important role in the future welfare society [Ueno, 2001]. The use of intelligent robots encourages the view of the machine as a partner in communication rather than as a tool. In the near future, robots will interact closely with a group of humans in their everyday environment in the field of entertainment, recreation, health-care, nursing, etc. In human-human interaction, multiple communication modals such as speech, gestures and body movements are frequently used. The standard input methods, such as text input via the keyboard and pointer/location information from a mouse, do not provide a natural, intuitive interaction between humans and robots. Therefore, it is essential to create models for natural and intuitive communication between humans and robots. Furthermore, for intuitive gesture-based interaction between human and robot, the robot should understand the meaning of gesture with respect to society and culture. The ability to understand hand gestures will improve the naturalness and efficiency of human interaction with robot, and allow the user to communicate in complex tasks without using tedious sets of detailed instructions.

This interactive system uses robot eye's cameras or CCD cameras to identify humans and recognize their gestures based on face and hand poses. Vision-based face recognition systems have three major components: image processing or extracting important clues (face pose and position), tracking the facial features (related position or motion of face and hand poses), and face recognition. Vision-based face recognition system varies along a number of

Source: Face Recognition, Book edited by: Kresimir Delac and Mislav Grgic, ISBN 978-3-902613-03-5, pp.558, I-Tech, Vienna, Austria, June 2007

dimensions: number of cameras, speed and latency (real-time or not), structural environment (restriction on lighting conditions and background), primary features (color, edge, regions, moments, etc.), etc. Multiple cameras can be used to overcome occlusion problems for image acquisition but this adds correspondence and integration problems.

The aim of this chapter is to present a vision-based face and hand gesture recognition method. The scope of this chapter is versatile. Segmentation of face and hand regions from the cluttered background, generation of eigenvectors and feature vectors in training phase, classification of face and hand poses, recognizes the user and gesture. In this chapter we present a method for recognizing face and gestures in real-time combining skin-color based segmentation and subspace-based patterns matching techniques. In this method three larger skin like regions are segmented from the input images using skin color information from YIQ color space, assuming face and two hands may present in the images at the same time. Segmented blocks are filtered and normalized to remove noises and to form fixed size images as training images. Subspace method is used for classifying hand poses and face from three skin-like regions. If the combination of three skin-like regions at a particular frame matches with the predefined gesture then corresponding gesture command is generated. Gesture commands are being sent to robots through TCP-IP network and their actions are being accomplished according to user's predefined action for that gesture. In this chapter we have also addressed multi directional face recognition system using subspace method. We have prepared training images in different illuminations to adapt our system with illumination variation.

This chapter is organized as follows. Section 2 focuses on the related research regarding person identification and gesture recognition. In section 3 we briefly describe skin like regions segmentation, filtering and normalization techniques. Section 4 describes subspace method for face and hand poses classification. Section 5 presents person identification and gesture recognition method. Section 6 focuses on human-robot interaction scenarios. Section 7 concludes this chapter and focuses on future research.

2. Related Work

This section briefly describes the related research on computer vision-based systems that include the related research on person identification and gesture recognition systems. Numbers of approaches have been applied for the visual interpretation of gestures to implement human-machine interaction [Pavlovic, 1997]. Major approaches are focused on hand tracking, hand poster estimation or hand pose classification. Some studies have been undertaken within the context of particular application: such as using a finger as a pointer to control TV, or manipulated Augmented desks. There are large numbers of household machine that can take benefit from the intuitive gesture understanding, such as: Microwave, TV, Telephone, Coffee maker, Vacuum cleaner, Refrigerator, etc. The aged/disable people can access such kind of machine if its have intuitive gesture understanding interfaces.

Computer vision supports a wide range of human tasks including, recognition, navigation, communication, etc. Using computer vision to sense and perceive the user in an HCI or HRI context is often called vision-based interaction or vision-based interface (VBI). In recent years, there has been increased research on practical vision-based interaction methods, due to availability of vision-based software, and inexpensive and fast enough computer vision related hardware components. As an example of VBI, hand pose or gesture recognition offers many promising approaches for human-machine interaction (HMI). The primary goal of the gesture recognition researches is to develop a system, which can recognize specific user and his/her gestures and use them to convey information or to control intelligent machine. Locating the faces and identifying the users is the core of any vision-based human-machine interface systems. To understand what gestures are, brief overviews of other gesturer researchers are useful.

2.1 Face Detection and Recognition

In the last few years, face detection and person identification attracts many researchers due to security concern; therefore, many interesting and useful research demonstrations and commercial applications have been developed. A first step of any face recognition or vision-based person identification system is to locate the face in the image. Figure 1 shows the example scenarios of face detection (partly of the images are taken from Rowley research paper [Rowley, 1997]). After locating the probable face, researchers use facial features (eyes, nose, nostrils, eyebrows, mouths, leaps, etc.) detection method to detect face accurately [Yang, 2000]. Face recognition or person identification compares an input face image or image features against a known face database or features databases and report match, if any. Following two subsections summarize promising past research works in the field of face detection and recognition.

2.1.1 Face Detection

Face detection from a single image or an image sequences is a difficult task due to variability in pose, size, orientation, color, expression, occlusion and lighting condition. To build a fully automated system that extracts information from images of human faces, it is essential to develop efficient algorithms to detect human faces. Visual detection of face has been studied extensively over the last decade. There are many approaches for face detection. Face detection researchers summarized the face detection work into four categories: template matching approaches, feature invariant approaches, appearance-based approaches and knowledge-based approaches [Yang, 2002]. Such approaches typically rely on a static background, so that human face can be detected using image differencing. Many researches also used skin color as a feature and leading remarkable face tracking as long as the lighting conditions do not varies too much [Dai, 1996], [Crowley, 1997].

Template Matching Approaches

In template matching methods, a standard template image data set using face images is manually defined. The input image is compared with the template images and calculated correlation coefficient or/and minimum distances (Manhattan distance, Euclidian distance, Mahalanobis distance, etc.). The existence of face is determined using the maximum correlation coefficient value and/or minimal distance. For exact matching correlation coefficient is one and minimum distance is zero. This approach is very simple and easy to implement. But recognition result depends on the template images size, pose, orientation, shape and intensity.

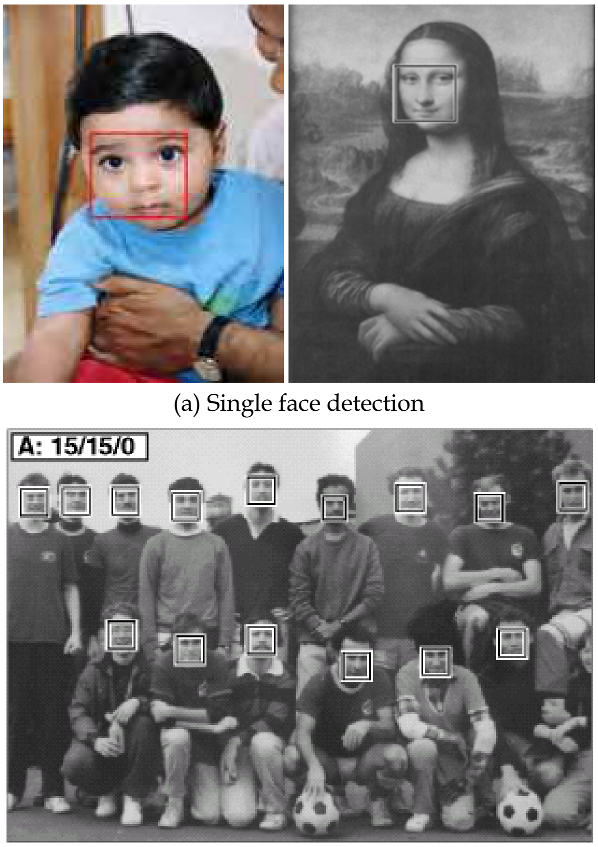
Sakai *et. al.* [Sakai, 1996] used several sub-templates for the eyes, nose, mouth and face contour to model a face which is defined in terms of line spaces. From the input images lines are extracted based on greatest gradient change and then matched against the sub-templates. The correlation between sub-images and contour templates are computed first to locate the probable location of faces. Then matching with the other sub-templates is performed at the probable face location.

Tsukamoto *et. al.* [Tsukamoto, 1994] presents a qualitative model for face [QMF]. In their model each sample image is divided into N blocks and qualitative features ('lightness' and 'edgeness') are estimated for each block. This blocked template is used to estimate "faceness" at every position of an input image. If the faceness measure is satisfied the predefined threshold then the face is detected.

We have developed a face detection method using the combination of correlation coefficient and Manhattan distance features, calculated from multiple face templates and test face image [Hasanuzzaman, 2004a]. In this method three larger skin-like regions are segmented first. Then segmented images are normalized to match with the size and type of the template images. Correlation coefficient is calculated using equation (1),

$$\alpha_t = M_t / P_t \tag{1}$$

where, M_t is total number of matched pixels (white pixels with white pixels and black pixels with black pixels) with the t^{th} template, P_t is total number of pixels in the t^{th} template and t, is a positive number. For exact matching α_t is 1, but for practical environment we have selected a threshold value for α_t ($0<\alpha_t\leq1$) through experiment considering optimal matching.



(a) Single face detection

(b) Multiple faces detection

Figure 1. Examples of face detection scenarios [Rowley, 1997]

Minimum distance can be calculated by using equation (2),

$$\delta_t = \{ \sum_{1}^{x \times y} | I - G_t | \} \tag{2}$$

where, I(x,y) is the input image and $G_1(x,y)$, $G_2(x,y)$,----------,$G_t(x,y)$ are template images. For exact matching δ_t is 0, but for practical environment we have selected a threshold value for δ_t through experiment considering optimal matching. If the maximum correlation coefficient and the minimum distance qualifier support corresponding specific threshold values then that segment is detected as face and the center position of the segment is use as the location of the face.

Miao *et. al.* [Miao, 1999] developed a hierarchical template matching method for multi-directional face detection. At the first stage, an input image is rotated from -20° to +20° in step of 5°. A multi-resolution image hierarchy is formed and edges are extracted using Laplacian operator. The face template consists of the edges produced by six facial components: two eyebrows, two eyes, nose and mouth. Finally, heuristics are applied to determine the existence of face.

Yuille *et. al.* [Yuille, 1992] used deformable template to model facial features that fit a priori elastic model to facial features. In this approach, facial features are described by parameterized template. An energy function is defined to link edges, peaks, and valleys in the input image to corresponding parameters in the template. The best fit of the elastic model is found by minimizing an energy function of the parameters.

Feature Invariant Approaches

There are many methods to detect facial features (mouth, eyes, eyebrows, lips, hair-line, etc.) individually and from their geometrical relations to detect the faces. Human face skin color and texture also used as features for face detection. The major limitations with these feature-based methods are that the image features are corrupted due to illumination, noise and occlusion problem.

Sirohey proposed a face localization method from a cluttered background using edge map (canny edge detector) and heuristics to remove and group edges so that only the ones on the face contour are preserved [Sirohey, 1993]. An ellipse is then fit to the boundary between the head region and the background.

Chetverikov *et. al.* [Chetverikov, 1993] presented face detection method using blobs and streaks. They used two black blobs and three light blobs to represent eyes, cheekbones and nose. The model uses streaks to represent the outlines of the faces, eyebrows and lips. Two triangular configurations are utilized to encode the spatial relationship among the blobs. A low resolution Laplacian image is generated to facilitate blob detection. Next, the image is scanned to find specific triangular occurrences as candidates. A face is detected if streaks are identified around the candidates.

Human faces have a distinct texture that can be separated them from other objects. Augusteijn *et. al.* [Augusteijn, 1993] developed a method that infers the presence of face thorough the identification of face like templates. Human skin color has been proven to be an effective feature for face detections, therefore many researchers has used this feature for probable face detection or localization [Dai 1996], [Bhuiyan, 2003], [Hasanuzzaman 2004b].

Recently, many researchers are combining multiple features for face localization and detection and those are more robust than single feature based approaches. Yang and Ahuja [Yang, 1998] proposed a face detection method based on color, structure and geometry.

Saber and Tekalp [Saber, 1998] presented a frontal view-face localization method based on color, shape and symmetry. Darrel *et. al.* [Darrel, 2000] integrated stereo, color and pattern detection method to track the person in real time.

Appearance-Based Approaches

Appearance-based methods use training images and learning approaches to learn from the known face images. These approaches rely on the statistical analysis and machine learning techniques to find the relevant characteristics of face and non-face images. There are many researchers using appearance-based methods.

Turk *et. al.* [Turk, 1991] applied principal component analysis to detect and recognize face. From the training face images they generated the eigenfaces. Face images and non-face images are projected onto the eigenspaces; form feature vectors and clustered the images based on separation distance. To detect the presence of a face from an image frame, the distance between the known face space and all location in the images are calculated. If the minimum distance satisfied the faceness threshold values then the location is identified as face. These approaches are widely used by the many researchers.

Knowledge-Based Approaches

These methods use the knowledge of the facial features in top down approaches. Rules are used to describe the facial features and their relations. For example, a face is always consists of two eyes, one nose and a mouth. The relationship is defined using relative distances and positions among them. For example, the center of two eyes are align on the same line, the center points of two eyes and mouth form a triangular. Yang and Huang [Yang, 1994] used hierarchical knowledge-based method to detect face. In this method they used three layers of rules. At the first level, all possible face candidates are found by scanning a mask window (face template) over the input images, and applying a set of rules at each location. At the second level, histogram equalization and edge detection is performed on candidate faces. At the third level, using rules facial feature are detected individually and using the pre-knowledge of their relation, detect the actual faces. Kotropoulous [Kotropoulous, 1997] and other also presented rule-based face localization method.

2.1.2 Face Recognition

During the last few years face recognition has received significant attention from the researchers [Zhao, 2003] [Chellappa, 1995]. Research on automatic machine- based face recognition has started in the 1970s [Kelly 1970]. Figure 2 shows an example of face recognition scenario. The test face image (preprocessed) is matched with the face images of known persons in the database. If the face is sufficient close (nearest and support predefined threshold) to any one of the face classes, then corresponding person is identified, otherwise the person is unknown. Zhao [Zhao, 2003] *et. al.* have summarized the past recent researches on face recognition methods with three categories: Holistic matching methods, Feature-based matching methods and Hybrid methods.

Holistic Methods

These methods use the whole face region as the raw input for the recognition unit. One of the most widely used representations of the face recognition is eigenfaces, which are based on principal component analysis (PCA). The eigenface algorithm uses the principal component analysis (PCA) for dimensionality reduction and to find the vectors those are best account for the distribution of face images within the entire face image spaces. Using

PCA many face recognition techniques have been developed [Turk, 1991], [Lee, 1999], [Chung, 1999], etc.

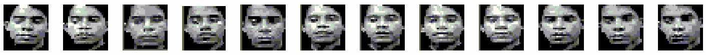
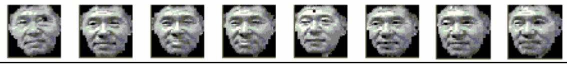
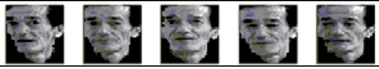
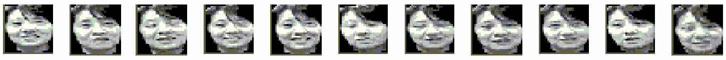
Known Face Images	**Test Image**	**Who is the person?**
		
		Person_4
		
		
		

Figure 2. Example of face recognition scenario

Turk and Pentland [Turk, 1991] first successfully used eigenfaces for face detection and person identification or face recognition. In this method from the known face images training image dataset is prepared. The face space is defined by the "eigenfaces" which are eigenvectors generated from the training face images. Face images are projected onto the feature space (or eigenfaces) that best encodes the variation among known face images. Recognition is performed by projecting a test image onto the "facespace" (spanned by the m number of eigenfaces) and then classified the face by comparing its position (Euclidian distance) in face space with the positions of known individuals. Figure 3 shows the example of 8 eigenfaces generated from 140 training face (frontal) images of 7 persons. In this example, the training faces are 60×60 gray images.

The purpose of PCA is to find out the appropriate vectors that can describe the distribution of face images in images spaces and form another face spaces. To form principal components m-numbers of eigenvectors are used based on the eigenvalues distribution. Eigenvectors and eigenvalues are obtained from the covariance matrix generated from training face images. The eigenvectors are sorted based on eigenvalues (higher-to-lower) and selected first m-number of eigenvectors to form principal components.

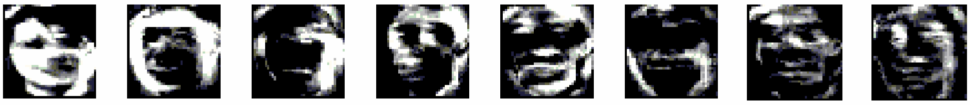


Figure 3. Example of eigenfaces

Figure 4 shows the example distribution of eigenvalues for 140 frontal face images. This graph explores the eigenvalues spectrum and how much variance the first m-vectors for. In most cases the number of eigenvectors that account for variance somewhere in the 65%-90% range.

Independent component analysis (ICA) is similar to PCA except that the distributions of the components are designed to be non-Gaussian. The ICA separates the high-order moments of the input in addition to the second order moments utilized in PCA. Bartlett *et. al.* [Bartlett, 1998] used ICA methods for face recognition and reported satisfactory recognition performance.

Face recognition system using Linear Discriminant Analysis (LDA) or Fisher Linear Discriminant Analysis (FDA) has also been very successful. In Fisherface algorithm by defining different classes with different statistics, the images in the learning set are divided in the corresponding classes [Belhumeur, 1997]. Then, the techniques similar to those used in eigenface algorithm are applied for face classification or person identification.

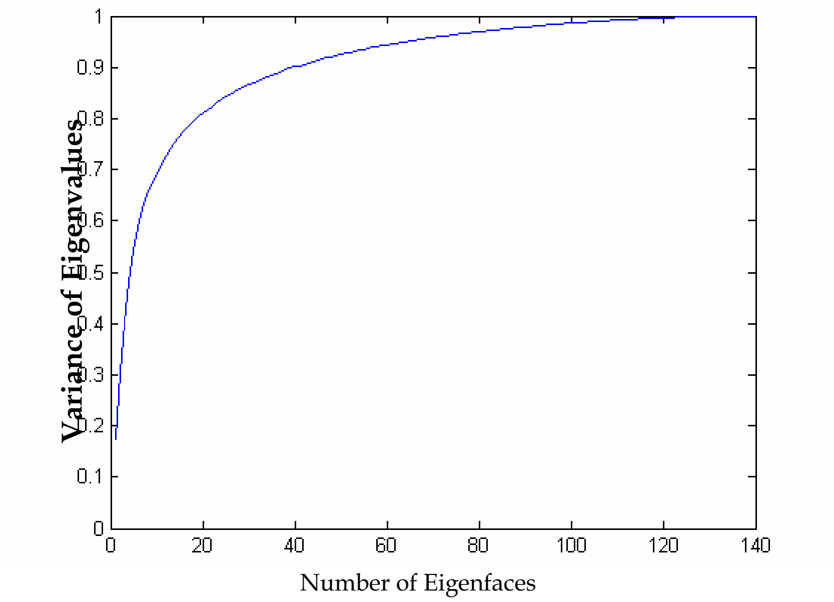


Figure 4. Example of eigenvectors spectrum for 140 eigenfaces

Feature-Based Matching Methods

In these methods facial features such as the eyes, lips, nose and mouth are extracted first and their locations and local statistics (geometric shape or appearance) are fed into a structural classifier. Kanade developed one of the earliest face recognition algorithms based on automatic facial feature detection [Kanade, 1977]. By localizing the corner of the eyes, nostrils, etc., in frontal views, that system compares parameters for each face, which were compared (using Euclidian distance metric) against the parameters of known person faces. One of the most successful of these methods is the Elastic Bunch Graph Matching (EBGM) system [Wiskott, 1997]. Other well-known methods in these systems are Hidden Markov Model (HMM) and convolution neural network [Rowley, 1997]. System based on EBGM approach have been applied to face detection and extraction, pose estimation, gender classification, sketch image based recognition and general object recognition.

Hybrid Approaches

These approaches use both holistic and features based approaches. These methods are very similar to human perception consider whole image and features individually at a time. Chung *et. al.* [Chung, 1999] combined Gabor Wavelet and PCA based approaches for face recognition and reported better accuracy than each of individual algorithm. Pentland *et. al.* [Pentland, 1994] have used both global eigenfaces and local eigenfeatures (eigeneyes, eigenmouth and eigennose) for face recognition. This method is robust against face images with multiple views.

2.2 Gesture Recognition and Gesture-Based Interface

Gestures are expressive meaningful body motions i.e., physical movements of the hands, arms, fingers, head, face or other parts of the body with the intent to convey information or interact with the environment [Turk, 2000]. People all over the world use their hands, head and other parts of the body to communicate expressively. The social anthropologists Edward T. Hall claims 60% of all our communications are nonverbal [Imai, 2004]. Gestures are used for everything from pointing at a person or an object to change the focus of attention, to conveying information. From the biological and sociological perspective, gestures are loosely defined, thus, researchers are free to visualize and classify gestures as these fit. Biologists define "gesture" broadly, stating, "the notion of gesture is to embrace all kinds of instances where an individual engages in movements whose communicative intent is paramount, manifest and openly acknowledged" [Nespoulous, 1986]. Gestures associated with speech are referred to as gesticulation. Gestures, which function independently of speech, are referred to as autonomous gestures. Autonomous gestures can be organized into their own communicative language, such as American Sign Language (ASL). Autonomous gesture can also represent motion commands to use in communication and machine control. Researchers are usually concerned with gestures those are directed toward the control of specific object or the communication with a specific person or group of people.

Gesture recognition is the process by which gestures made by the user are make known to the intelligence system. Approximately in the year 1992 the first attempts were made to recognize hand gestures from color video signals in real-time. It was the year, when the first frame grabbers for color video input were available, that could grab color images in real time. As color information improves segmentation and real time performance is a prerequisite for human-computer interaction, this obviously seems to be the start of development of gesture recognition. Two approaches are commonly used to recognize gestures, one is a gloved-base approach [Sturman, 1994] and another is a vision-based approach [Pavlovic, 1997].

2.2.1 Glove-Based Approaches

A common technique is to instrument the hand with a glove, which is equipped with a number of sensors, which provide information about hand position, orientation and flex of the fingers. The first commercially available hand tracker is the 'Dataglove' [Zimmerman, 1987]. The 'Dataglove' could measure each joint bend to an accuracy of 5 to 10 degrees, could classify hand pose correctly, but not the sideways movement of the fingers. The second hand tracker, 'CyberGlove' developed by Kramer [Kramer, 1989] uses strain gauges placed between the fingers to measure abduction as well as more accurate bend sensing.

Figure 5 shows the example of a 'CyberGlove' which has up to 22 sensors, including three bend sensors on each finger, four abduction sensors, plus sensors measuring thumb crossover, palm arch, wrist flexion and wrist abduction [Bllinghurst, 2002]. Once the gloves have captured hand pose data, gestures can be recognized using a number of different techniques. Neural network approaches or statistical template-matching approaches are commonly used to identify static hand posses [Fels, 1993]. Time dependent neural network and Hidden Markov Model (HMM) are commonly used for dynamic gesture recognition [Lee, 1996]. In this case gestures are typically recognized using pre-trained templates, however gloves can also be used to identify natural or untrained gestures. Glove-based approaches provide more accurate gesture recognition than vision-based approaches but they are expensive, encumbering and unnatural.

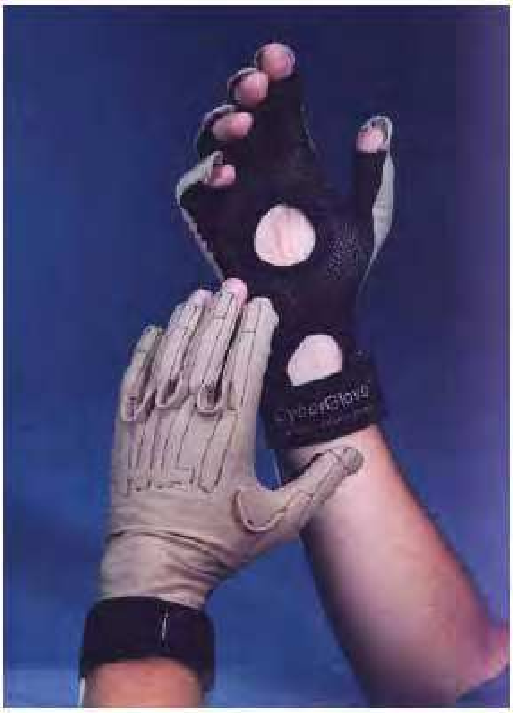


Figure 5. The 'CyberGlove' for hand gesture recognition [Bllinghurst, 2002]

2.2.2 Vision-Based Approaches

Vision-based gesture recognition systems can be divided into three main components: image processing or extracting important clues (hand shape and position, face or head position, etc.), tracking the gesture features (related position or motion of face or hand poses), and gesture interpretation (based on collected information that support predefined meaningful gesture). The first phase of gesture recognition task is to select a model of the gesture. The modeling of gesture depends on the intent-dent applications by the gesture.

There are two different approaches for vision-based modeling of gesture: Model based approach and Appearance based approach.

The Model based techniques are tried to create a 3D model of the user hand (parameters: Joint angles and palm position) [Rehg, 1994] or contour model of the hand [Shimada, 1996] [Lin, 2002] and use these for gesture recognition. The 3D models can be classified in two large groups: volumetric model and skeletal models. Volumetric models are meant to describe the 3D visual appearance of the human hands and arms.

Appearance based approaches use template images or features from the training images (images, image geometry parameters, image motion parameters, fingertip position, etc.) which use for gesture recognition [Birk, 1997]. The gestures are modeled by relating the appearance of any gesture to the appearance of the set of predefined template gestures. A different group of appearance-based model uses 2D hand image sequences as gesture templates. For each gestures number of images are used with little orientation variations [Hasanuzzaman, 2004a]. Images of finger can also be used as templates for finger tracking applications [O'Hagan, 1997]. Some researchers represent motion history as 2D image and use it as template images for different actions of gestures. The majority of appearance-based models, however, use parameters (image eigenvectors, image edges or contour, etc.) to form the template or training images. Appearance based approaches are generally computationally less expensive than model based approaches because its does not require translation time from 2D information to 3D model.

Once the model is selected, an image analysis stage is used to compute the model parameters from the image features that are extracted from single or multiple video input streams. Image analysis phase includes hand localization, hand tracking, and selection of suitable image features for computing the model parameters.

Two types of cues are often used for gesture or hand localization: color cues and motion cues. Color cue is useful because human skin color footprint is more distinctive from the color of the background and human cloths [Kjeldsen, 1996], [Hasanuzzaman, 2004d]. Color-based techniques are used to track objects defined by a set of colored pixels whose saturation and values (or chrominance values) are satisfied a range of thresholds. The major drawback of color-based localization methods is that skin color footprint is varied in different lighting conditions and also the human body colors. Infrared cameras are used to overcome the limitations of skin-color based segmentation method [Oka, 2002].

The motion-based segmentation is done just subtracting the images from background [Freeman, 1996]. The limitation of this method is considered the background or camera is static. Moving objects in the video stream can be detected by inter frame differences and optical flow [Cutler, 1998]. However such a system cannot detect a stationary hand or face. To overcome the individual shortcomings some researchers use fusion of color and motion cues [Azoz, 1998].

The computation of model parameters is the last step of the gesture analysis phase and it is followed by gesture recognition phase. The type of computation depends on both the model parameters and the features that were selected. In the recognition phase, parameters are classified and interpreted in the light of the accepted model or the rules specified for the gesture interpretation. Two tasks are commonly associated with the recognition process: optimal partitioning of the parameter space and implementation of the recognition procedure. The task of optimal partitioning is usually addresses through different learning-from-examples training procedures. The key concern in the implementation of the recognition procedure is computation efficiency. A recognition method usually determines confidence scores or probabilities that define how closely the image data fits each model. Gesture recognition methods are divided into two categories: static gesture or hand poster and dynamic gesture or motion gesture.

Static Gesture

Static gesture (or pose gesture) recognition can be accomplished by using template matching, eigenspaces or PCA, Elastic Graph Matching, neural network or other standard pattern recognition techniques. Template matching techniques are the simple pattern matching approaches. It is possible to find out the most likely hand postures from an image by computing the correlation coefficient or minimum distance metrics with template images.

Eigenspace or PCA is also used for hand pose classification similarly it used for face detection and recognition. Moghaddam and Pentland used eigenspaces (eigenhands) and principal component analysis not only to extract features, but also to estimate complete density functions for localization [Moghaddam, 1995]. In our previous research, we have used PCA for hand pose classification from three larger skin-like components that are segmented from the real-time images [Hasanuzzaman, 2004d].

Triesch *et. al.* [Triesch, 2002] employed the elastic graph matching techniques to classify hand posters against complex backgrounds. They represented hand posters by label graphs with an underlying two-dimensional topology. Attached to the nodes are jets, which are a sort of local image description based on Gabor filters. This approach can achieve scale-invariant and user invariant recognition and does not need hand segmentation. This approach is not view-independent, because it uses one graph for one hand posture. The major disadvantage of this algorithm is the high computational cost.

Dynamic Gesture

Dynamic gestures are considered as temporally consecutive sequences of hand or head or body postures in sequence of time frames. Dynamic gestures recognition is accomplished using Hidden Markov Models (HMMs), Dynamic Time Warping, Bayesian networks or other patterns recognition methods that can recognize sequences over time steps. Nam *et. al.* [Nam, 1996] used HMM methods for recognition of space-time hand-gestures. Darrel *et. al.* [Darrel, 1993] used Dynamic Time Warping method, a simplification of Hidden Markov Models (HMMs) to compare the sequences of images against previously trained sequences by adjusting the length of sequences appropriately. Cutler *et. al.* [Cutler, 1998] used a ruled-based system for gesture recognition in which image features are extracted by optical flow. Yang [Yang, 2000] recognizes hand gestures using motion trajectories. First they extract the two-dimensional motion in an image, and motion patterns are learned from the extracted trajectories using a time delay network.

2.2.3 Gesture-Based Interface

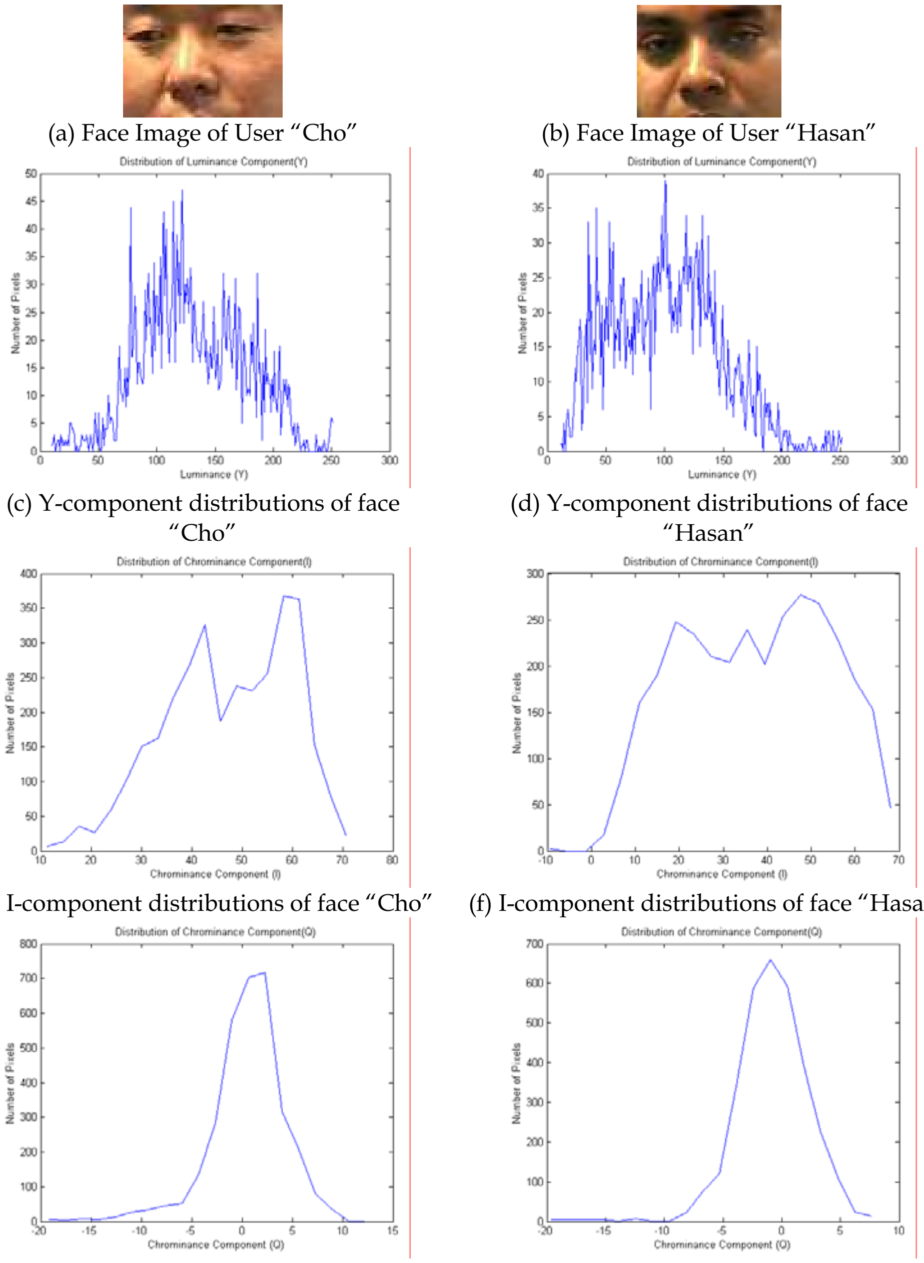
The first step in considering gesture-based interaction with intelligent machine is to understand the role of gesture in human-to-human communication. There are significant amount of researches on hand, arm and facial gesture recognition, to control robot or intelligent machine in recent years. This sub-section summarizes some promising existing gesture recognition system. Cohen *et. al.* [Cohen, 2001] described a vision-based hand gesture identifying and hand tracking system to control computer programs, such as browser of PowerPoint or any other applications. This method is based primarily on color matching and is performed in several distinct stages. After color-based segmentation, gestures are recognized using geometric configuration of the hand. Starner *et. al.* [Starner, 1998] proposed real-time American Sign Language (ASL) recognition using desk and wearable computer based video. The recognition method is based on the skin color information to extract hands poster (pose, orientation) and locate their position and motion. Using Hidden Markov Models (HMM) this system recognizes sign language words but vocabulary is limited to 40 words. Utsumi *et. al.* [Utsumi, 2002] detected predefined hand pose using hand shape model and tracked hand or face using extracted color and motion. Multiple cameras are used for data acquisition to reduce occlusion problem in their system. But in this process there incurs complexity in computations. Watanabe *et. al.* [Watanabe, 1998] used eigenspaces from multi-input image sequences for recognizing gesture. Single eigenspaces are used for different poses and only two directions are considered in their method. Hu [Hu, 2003] proposed hand gesture recognition for human-machine interface of robot teleoperation using edge features matching. Rigoll *et. al.* [Rigoll, 1997] used HMM-based approach for real-time gesture recognition. In their work, features are extracted from the differences between two consecutive images and target image is always assumed to be in the center of the input images. Practically it is difficult to maintain such condition. Stefan Waldherr *et. al.* proposed gesture-based interface for human and service robot interaction [Waldherr, 2000]. They combined template-based approach and Neural Network based approach for tracking a person and recognizing gestures involving arm motion. In their work they proposed illumination adaptation methods but did not consider user or hand pose adaptation. Torras has proposed robot adaptivity technique using neural learning algorithm [Torras, 1995]. This method is extremely time consuming in learning phase and has no way to encode prior knowledge about the environment to gain the efficiency.

3. Skin Color Region Segmentation and Normalization

Images containing faces and hand poses are essential for vision-based human-robot interaction. But still it is very difficult to segment face and hand poses in real time from the color images with cluttered background. Human skin color has been used and proven to be an effective feature in many application areas, from face detection to hand tracking. Since face and two hands may present in the images at a specific time in an image frame, three largest skins like regions are segmented from the input images using skin color information. Several color spaces have been utilized to label pixels as skin including RGB, HSV, YCrCb, YIQ, CIE-XYZ, CIE-LUV, etc. However, such skin color models are not effective where the spectrum of the light sources varies significantly. In this study YIQ (Y is luminance of the color and I, Q are chrominance of the color) color representation system is used for skin-like region segmentation because it is typically used in video coding and provides an effective use of chrominance information for modeling the human skin color [Bhuiyan, 2003], [Dai, 1996].

3.1 YIQ-Color Coordinate Based Skin-Region Segmentation

To detect human face or hand, it is assumed that the captured camera images are represented in the RGB color spaces. Each pixel in the images is represented by a triplet *P=F(R,G,B)*. The RGB images taken by the video camera are converted to YIQ color representation system (for detail please refer to Appendix A). Skin color region is determined by applying threshold values ((Y_Low<Y<Y_High) && (I_Low<I<I_High) && Q_Low<Q<Q_High)) [Hasanuzzaman, 2005b].



(a) Face Image of User "Cho"

(b) Face Image of User "Hasan"

(c) Y-component distributions of face "Cho"

(d) Y-component distributions of face "Hasan"

(e) I-component distributions of face "Cho"

(f) I-component distributions of face "Hasan"

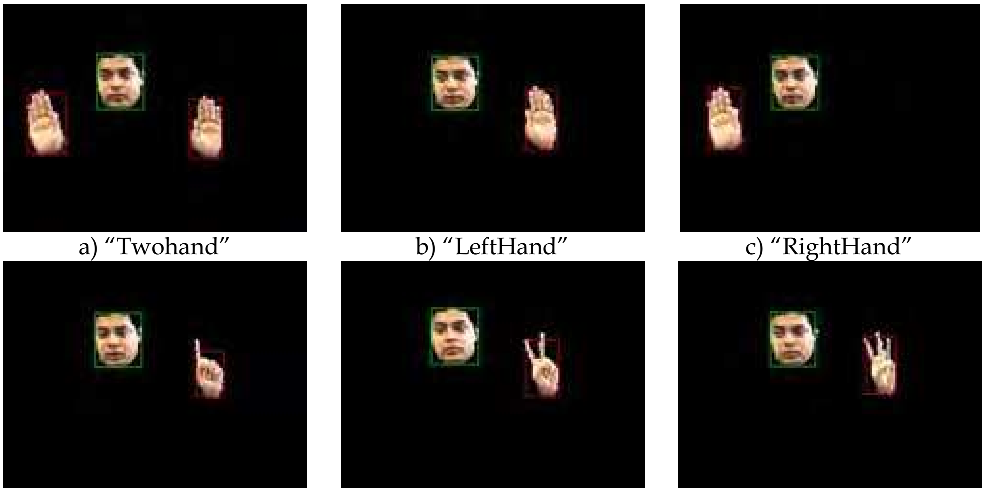
(g) Q-component distributions of face "Cho"

h) Q-component distributions of face "Hasan"

Figure 6. Histograms of Y, I, Q components for different person face images

Figure 6 shows example skin regions and its corresponding Y, I, Q components distributions for every pixels. Chrominance component I, play an important role to distinguish skin like regions from non-skin regions, because it is always positive for skin regions. Values of Y and I increases for more white people and decreases for black people. We have included an off line program to adjust the threshold values for Y, I, Q, if the person color or light intensity variation affect the segmentation output. For that reason we need to manually select small skin region and non-skin regions and run our threshold evaluation program, that will represent graphical view of Y, I, Q distributions. From those distinguishable graphs we can adjust our threshold values for Y, I, Q using heuristic approach.

Probable hands and face regions are segmented from the image with the three largest connected regions of skin-colored pixels. The notation of pixel connectivity describes a relation between two or more pixels. In order to consider two pixels to be connected, their pixel values must both be from the same set of values *V (for binary images V is 1, for gray images it may be specific gray value).* Generally, connectivity can either be based on *4- or 8-*connectivity. In the case 4-connectivity, it does not compare the diagonal pixels but 8-connectivity compares the diagonal positional pixels considering 3×3 matrix, and as a result, 8-connectivity component is more noise free than 4-connectivity component. In this system, 8-pixels neighborhood connectivity is employed [Hasanuzzaman, 2006].



a) "Twohand" b) "LeftHand" c) "RightHand"

d) "One" e) "Two" f) "Three"

Figure 7. Example outputs of skin-regions segmentation

In order to remove the false regions from the segmented blocks, smaller connected regions are assigned by the values of black-color (R=G=B=0). As a result, after thresholding the segmented image may contain some holes in the three largest skin-like regions. In order to remove noises and holes, segmented images are filtered by morphological dilation and erosion operations with a 3×3 structuring element. The dilation operation is used to fill the holes and the erosion operations are applied to the dilationed results to restore the shape.

After filtering, the segmented skin regions are bounded by rectangular box using height and width information of each segment: $(M_1 \times N_1)$, $(M_2 \times N_2)$, and $(M_3 \times N_3)$. Figure 7 shows the example outputs of skin like region segmentation method with restricted background. If the user shirt's color is similar to skin color then segmentation accuracy is very poor. If the user wears short sleeves or T-shirt then it needs to separate hand palm from arm. This system assumes the person wearing full shirt with non-skin color.

3.2 Normalization

Normalization is done to scale the image to match with the size of the training image and convert the scaled image to gray image [Hasanuzzaman, 2004a]. Segmented images are bounded by rectangular boxes using height and width information of each segment: $(M_1 \times N_1)$, $(M_2 \times N_2)$, and $(M_3 \times N_3)$. Each segment is scaled to be square images with (60×60) and converted it to as gray images (BMP image). Suppose, we have a segment of rectangle $P[(x^l, y^l) - (x^h, y^h)]$ we sample it to rectangle $Q[(0,0) - (60 \times 60)]$ using following expression,

$$Q(x^q, y^q) = P(x^l + \frac{(x^h - x^l)}{60} x^q, y^l + \frac{(y^h - y^l)}{60} y^q) \tag{3}$$

Each segment is converted as gray image (BMP image) and compared with template/training images to find the best match. Using the same segmentation and normalization methods training images and test images are prepared, that is why result of this matching approach is better than others who used different training/template image databases. Beside this, we have included training/template images creation functions in this system so that it can adapt with person and illumination changes. Figure 8 shows the examples of training images for five face poses and ten hand poses.

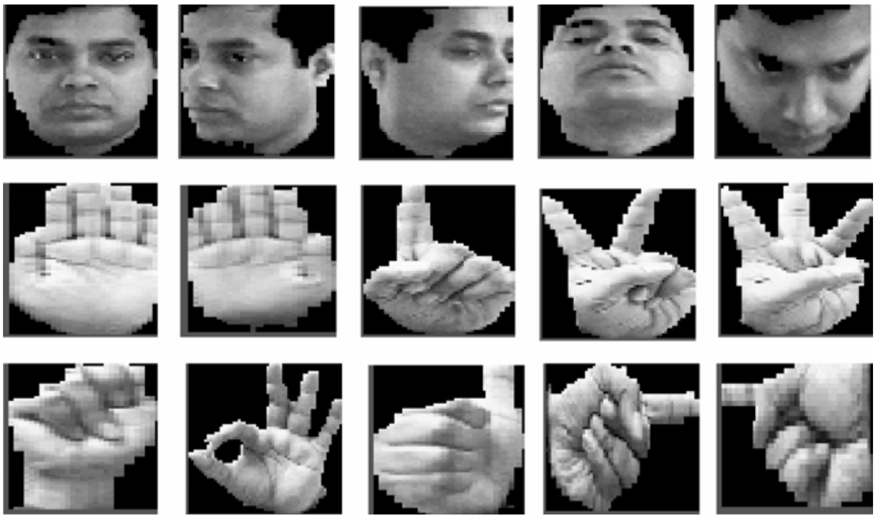


Figure 8. Examples of training images

4. Face and Hand Pose Classification by Subspace method

Three larger skin like regions are segmented from the input images considering that two hands and one face may present in the input image frame at a specific time. Segmented areas are filtered, normalized and then compared with the training images for finding the best matches using pattern-matching method. Principal component analysis (PCA) method is a standard pattern recognition approach and many researchers use it for face and hand pose classification [Hasanuzzaman, 2004d]. The main idea of the principal component analysis (PCA) method is to find the vectors that best account for the distribution of target images within the entire image space. In the general PCA method, eigenvectors are calculated from training images that include all the poses or classes. But for classification a large number of hand poses for a large number of users, need large number of training datasets from which eigenvectors generation is tedious and may not be feasible for a personal computer. Considering these difficulties we have proposed pose-specific subspace method that partition the comparison area based on each pose. In pose-specific subspace method, training images are grouped based on pose and eigenvectors for each pose are generated separately. In this method one PCA is used for each pose [Hasanuzzaman, 2005b] [Hasanuzzaman, 2004c]. In the following subsection we have described the algorithm of pose-specific subspace method for face and hand pose classification, which is very similar to general PCA based algorithm.

Symbols	Meanings
$T_j^{(i)}$	Training images for ith class
$u_m^{(i)}$	*m*th Eigenvectors for *i*th class
$\Omega_l^{(i)}$	Weight vector for *i*th class
$\omega_k^{(i)}$	Element of weight vector for *i*th class
Φ_i	Average image for *i*th class
$s_l^{(i)}$	*l*th Known image for *i*th class
ε	Euclidean distance among weight vectors
$\varepsilon_l^{(i)}$	Element of Euclidean distance among weight vectors for *i*th class

Table 1. List of symbols used in subspace method

Pose-Specific Subspace Method

Subspace method offers an economical representation and very fast classification for vectors with a high number of components. Only the statistically most relevant features of a class are retained in the subspace representation. The subspace method is based on the extraction of the most conspicuous properties of each class separately as represented by a set of prototype sample. The main idea of the subspace method is similar to principal component