DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PONDICHERRY ENGINEERING COLLEGE

SEMINAR REPORT ON

GESTURE RECOGNITION

SUBMITTED BY

PRAKRUTHI.V (283175132)

PRATHIBHA ANNAPURNA.P (283175135)

SARANYA.S (283175174)

SUBMITTED TO

Dr. R. Manoharan, Asst. Professor,

Seminar Coordinator.

1. INTRODUCTION

1.1 INTRODUCTION TO GESTURE RECOGNITION

In the present day framework of interactive, intelligent computing, an efficient human–computer interaction is assuming utmost importance. Gesture recognition can be termed as an approach in this direction. It is the process by which the gestures made by the user are recognized by the receiver.

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of:

- conveying meaningful information or
- interacting with the environment.

They constitute one interesting small subspace of possible human motion. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver.

Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse. Gesture recognition enables humans to interface with the machine (HMI) and interact naturally without any mechanical devices.

Gesture recognition can be conducted with techniques from computer vision and image processing.

1.2 NEED FOR GESTURE RECOGNITION

The goal of virtual environments (VE) is to provide natural, efficient, powerful, and flexible interaction. Gesture as an input modality can help meet these requirements because Human

gestures are natural and flexible, and may be efficient and powerful, especially as compared with alternative interaction modes.

The traditional two-dimensional (2D), keyboard- and mouse-oriented graphical user interface (GUI) is not well suited for virtual environments. Synthetic environments provide the opportunity to utilize several different sensing modalities and technologies and to integrate them into the user experience. Devices which sense body position and orientation, direction of gaze, speech and sound, facial expression, galvanic skin response, and other aspects of human behavior or state can be used to mediate communication between the human and the environment. Combinations of communication modalities and sensing devices can produce a wide range of unimodal and multimodal interface techniques. The potential for these techniques to support natural and powerful interfaces for communication in VEs appears promising.

Gesture is used for control and navigation in CAVEs (Cave Automatic Virtual Environments) and in other VEs, such as smart rooms, virtual work environments, and performance spaces. In addition, gesture may be perceived by the environment in order to be transmitted elsewhere (e.g., as a compression technique, to be reconstructed at the receiver). Gesture recognition may also influence – intentionally or unintentionally – a system's model of the user's state. Gesture may also be used as a communication *backchannel* (i.e., visual or verbal behaviors such as nodding or saying something, or raising a finger to indicate the desire to interrupt) to indicate agreement, participation, attention, conversation turn taking, etc. Clearly the position and orientation of each body part – the parameters of an articulated body model – would be useful, as well as features that are derived from those measurements, such as velocity and acceleration. Facial expressions are very expressive. More subtle cues such as hand tension, overall muscle tension, locations of self-contact, and even pupil dilation may be of use.

1.3 CLASSIFICATION OF GESTURES

Gestures can be static(the user assumes a certain pose or configuration) or dynamic(with prestroke, stroke, and poststroke phases).

Some gestures have both static and dynamic elements, as in sign languages. The automatic recognition of natural continuous gestures requires their temporal segmentation. The start and end points of a gesture, in terms of the frames of movement, both in time and in space are to be specified. Sometimes a gesture is also affected by the context of preceding as well as following gestures. Moreover, gestures are often language- and culture-specific.

Gestures can broadly be of the following types:

- ***hand and arm gestures*:** recognition of hand poses, sign languages, and entertainment applications (allowing children to play and interact in virtual environments);
- ***head and face gestures*:** some examples are: a) nodding or shaking of head; b) direction of eye gaze; c) raising the eyebrows; d) opening the mouth to speak; e) winking, f) flaring the nostrils; and g) looks of surprise, happiness, disgust, fear, anger, sadness, contempt, etc.;
- ***body gestures*:** involvement of full body motion, as in: a) tracking movements of two people interacting outdoors; b) analyzing movements of a dancer for generatingmatching music and graphics; and c) recognizing human gaits for medical rehabilitation and athletic training.

There are many classification of gestures, such as

- *Intransitive gestures:* "the ones that have a universal language value especially for the expression of affective and aesthetic ideas. Such gestures can be indicative, exhortative, imperative, rejective, etc."

- *Transitive gestures:* "the ones that are part of an uninterrupted sequence of interconnected structured hand movements that

are adapted in time and space, with the aim of completing a program, such as prehension."

The classification can be based on gesture's functions as:

- *Semiotic* – to communicate meaningful information.
- *Ergotic* – to manipulate the environment.
- *Epistemic* – to discover the environment through tactile experience.

The different gestural devices can also be classified as *haptic* or *non-haptic* (haptic means relative to contact).

Typically, the meaning of a gesture can be dependent on the following:

- spatial information: where it occurs;
- pathic information: the path it takes;
- symbolic information: the sign it makes;
- affective information: its emotional quality.

1.4 REQUIREMENTS AND CHALLENGES

The main requirement for gesture interface is the tracking technology used to capture gesture inputs and process them. Gesture-only interfaces with a syntax of many gestures typically require precise pose tracking.

A common technique for hand pose tracking, 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 was the Dataglove. Although instrumented gloves provide very accurate results they are expensive and encumbering.

Computer vision and image based gesture recognition techniques can be used overcoming some of the limitations. There are two different approaches to vision based gesture recognition; model based techniques which try to create a three-dimensional model of the users pose and use this for recognition, and image

based techniques which calculate recognition features directly from the image of the pose.

Effective gesture interfaces can be developed which respond to natural gestures, especially dynamic motion. This system must respond to user position using two proximity sensors, one vertical, the other horizontal. There must be a direct mapping of the motion to continuous feedback, enabling the user to quickly build a mental model of how to use the device.

There are many challenges associated with the accuracy and usefulness of gesture recognition software. For image-based gesture recognition there are limitations on the equipment used and image noise. Images or video must be under consistent lighting, or in the same location. Items in the background or distinct features of the users should not make recognition difficult.

The variety of implementations for image-based gesture recognition may also cause issue for viability of the technology to general usage. For example, an algorithm calibrated for one camera may not work for a different camera. These criteria must be considered for viability of the technology. The amount of background noise which causes tracking and recognition difficulties, especially when occlusions (partial and full) occur must be minimized. Furthermore, the distance from the camera, and the camera's resolution and quality, which causes variations in recognition and accuracy, should be considered. In order to capture human gestures by visual sensors, robust computer vision methods are also required, for example for hand tracking and hand posture recognition or for capturing movements of the head, facial expressions or gaze direction.

1.5 BENEFITS OF GESTURE RECOGNITION

A human computer interface can be provided using gestures

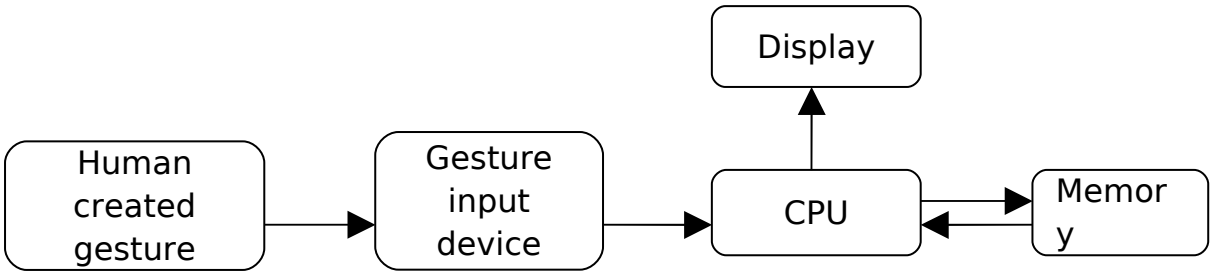
- Replace mouse and keyboard
- Pointing gestures
- Navigate in a virtual environment
- Pick up and manipulate virtual objects
- Interact with the 3D world
- No physical contact with computer
- Communicate at a distance

2. ARCHITECTURE AND DESIGN OF GESTURE RECOGNITION SYSTEM

2.1 ARCHITECTURE

A basic gesture input device is the word processing tablet. In the system, two dimensional hand gestures are sent via an input device to the computer's memory and appear on the computer monitor. These symbolic gestures are identified as editing commands through geometric modeling techniques. The commands are then executed, modifying the document stored in computer memory. Gestures are represented by a view-based approach, and stored patterns are matched to perceived gestures using dynamic time warping. View-based vision approaches also permit a wide range of gestural inputs when compared to the mouse and stylus input devices. Computation time is based on the number of view-based features used and the number of view models stored. Through the use of specialized hardware and a parallel architecture, processing time is less than 100 ms. Gestures are tracked through the use of a data-glove and converted into hypertext program commands. System identifies natural hand gestures unobtrusively with a data-glove, it is more intuitive to use than a standard mouse or stylus control system.

Fig.1: Block Diagram of Architecture for Gestural Control of Memory and Display



System Architecture Concepts for Gesture Recognition Systems

Based on the use of gestures by humans, the analysis of speech and handwriting recognizing systems and the analysis of other gesture recognition systems requirements for a gesture recognition system can be detailed.

Some requirements and tasks are:

- Choose gestures which fit a useful environment.
- Create a system which can recognize non-perfect human created gestures.
- Create a system which can use a both a gesture's static and dynamic information components.
- Perform gesture recognition with image data presented at field rate (or as fast as possible).
- Recognize the gesture as quickly as possible, even before the full gesture is completed.
- Use a recognition method which requires a small amount of computational time and memory.
- Create an expandable system which can recognize additional types of gestures.
- Pair gestures with appropriate responses (language definitions or device command responses).
- Create an environment which allows the use of gestures for remote control of devices.

2.2 DESIGN

The operation of the system proceeds in four basic steps:

1. Image input
2. background subtraction
3. image processing and data extraction
4. decision tree generation/parsing (initial training of the system requires the generation of a decision tree, however subsequent use of the system only requires the parsing of the decision tree to classify the image).

2.2.1 Image input

To input image data into the system, an IndyCam or videocam can be used with an image capture program used to take the picture. The camera, used should take first a background image(fig.2), and then take subsequent images of a person. A basic assumption of the system is that these images are fairly standard: a the image is assumed to be of a person's upper body, facing forward(fig. 3), with only one arm outstretched to a particular side.

Fig. 2: Background image Fig. 3: Foreground image

2.2.2 Background Subtraction

Once images are taken, the system performs a background subtraction of the image to isolate the person and create a mask. The background subtraction proceeds in two steps. First, each pixel from the background image is channel wise subtracted from the corresponding pixel from the foreground image. The resulting channel differences are summed, and if they are above a threshold, then the corresponding image of the mask is set white, otherwise it is set black.

Fig.4: A background Subtraction mask

The resulting image is a mask that outlines the body of the person (*fig.4*). Two important features of the image are the existence of a second right “arm”, which is the result of a shadow of the right arm falling on the wall behind the person, and the noise in the generated mask image. This phantom arm is the result of the poor image quality of the input image,but could be corrected for by space of the conversion of the color images and the use of another method of background subtraction. If the images were converted from RGB to HSB color space, than the subtracted values of the pixels (before being set to white or black, could be inspected, and those pixels that have a very low brightness could be discarded as well (set to black). Since a shadow tends to be very dark when compared to the body of a person (in an image), those pixels that have a low brightness can be inferred to be part of a shadow, and

therefore unimportant (discardable).The noise in the mask image For the GRS, I wrote a function that compares a pixel to the surrounding pixels (of a given radius), and sets that pixel to the value (black or white) of the majority of the other pixels. The GRS runs two such filters over the mask data, one with a radius of one pixel, and another of a radius of three pixels.

2.2.3 Image processing and data extraction

The final mask image is of good quality. There is no noise, and disconnected areas of the body (in this case it can be reduced significantly by running an averaging filter over the mask data. A hand that is separated from its arm (in *fig.4*) are reconnected (*fig.5*). Once a mask is generated, then that image can be processed for data to extract into a decision tree. Two strategies for extracting data from the image were tried. The first was to find the body and arms. Each column of pixels from the mask was summed, and the column with the highest sum was assumed to be the center of the body. This is a valid criteria for determining the body center, based on the assumptions of the input image. Arms are significantly thinner than the height of the main body of a person, and are so within the mask (even with the existence of shadow artifacts). The center of a person's head is the highest point on the body, and the torso of the body extends to the bottom of the image. In all of the samples this technique was tried on, it was successful in finding the center of the body, within a few pixels.

Fig. 5: an equalized background subtracted image

3. HAND AND ARM GESTURES

Hand gestures are the most expressive and the most frequently used gestures. This involves:

- **a posture:** static finger configuration without hand movement and
- **a gesture:** dynamic hand movement, with or without finger motion.

Gestures may be categorized as

- ***gesticulation*:** spontaneous movement of hands and arms, accompanying speech. These spontaneous movements constitute around 90% of human gestures. People gesticulate when they are on telephone, and even blind people regularly gesture when speaking to one another;
- ***language like gestures*:** gesticulation integrated into a spoken utterance, replacing a particular spoken word or phrase;
- ***pantomimes*:** gestures depicting objects or actions, with or without accompanying speech;
- ***emblems*:** familiar signs such as “V for victory,” or other culture-specific “rude” gestures;
- ***sign languages*:** well-defined linguistic systems. These carry the most semantic meaning and are more systematic, thereby being easier to model in a virtual environment.

3.1 HAND GESTURE RECOGNITION

General approaches to the hand gesture recognition are

- Appearance-based gesture classification
- 3D model-based hand pose and motion estimation

The two-level approach consists of

(i) hand posture recovery

(ii) recognition of hand gestures

3D model-based hand gesture recognition aims to recover the full kinematic structure of the hand, i.e. the pose of the palm, the angles of each fingers, etc.

3.2 REPRESENTATION OF HAND GESTURE

Representation of hand motion includes:

- Global configuration: six DOF of a frame attached to the wrist, representing the pose of the hand.
- Local configuration: the angular DOF of fingers

KINEMATIC MODEL OF A HUMAN HAND

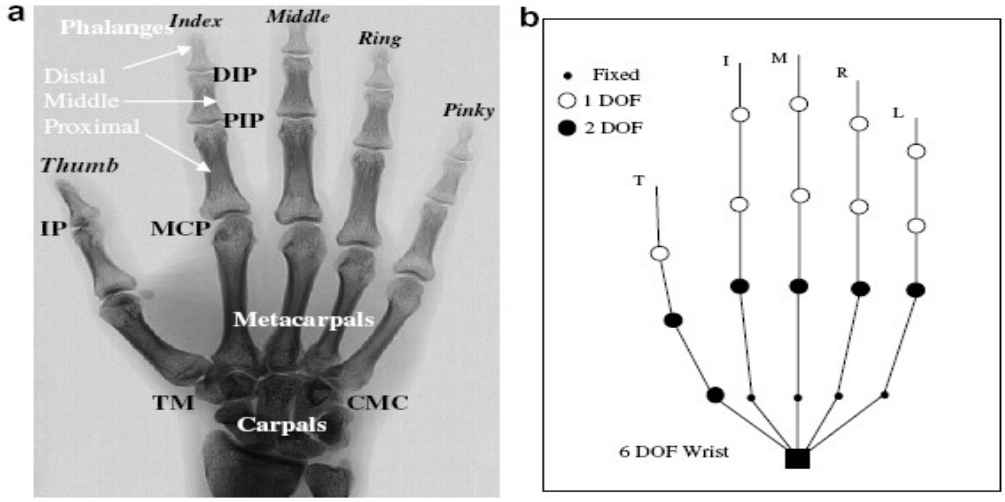


Fig. 2. Skeletal hand model: (a) Hand anatomy, (b) the kinematic model.

Kinematic model is augmented with shape information to generate appearances of a hand seen in 2D images. Hand pose or motion constraints are applied to reduce the search space for pose estimation. Calibrations are required to capture the distinct shape models for different persons in case of high-precision applications.

Different shape models may be used according to the desired applications. Primitive geometric shapes (cylinders, spheres, or ellipsoids) may be attached through joints of the hand skeleton. B-spline surfaces are employed for more realistic models. Different

hand models like cardboard model, quadrics-based hand model, B-spline surface model, etc.. are used.

3.3 HAND SHAPE (FEATURES) EXTRACTION AND SELECTION

3.3.1 FEATURE EXTRACTION

Feature extraction decides, in large extent, the robustness and processing efficiency of a tracking system. It has a huge impact on the overall system performance. Natural hand motion is highly constrained, which is not reflected in the kinematic model. These constraints need to be captured and applied.

- Static constraints: the range of each parameter
- Dynamic constraints: the joint angle dependencies

PCA is applied to reduce dimensionality. Seven DOF is reportedly achieved by PCA. High-level features include fingertips, fingers, joint locations, etc. They are very compact representation of the input, enabling faster computation, which are, very difficult to extract robustly. Low-level features include contours, edges, etc. Volumetric model is projected on the images. Point correspondences between model contours and image contours are established. Distance between corresponding points results in Matching Error. Combined edge orientation and chamfer matching, skin color model and combination of other features improve robustness in general. Silhouette (outline of the hand): Overlapping area of the model and hand silhouettes result in similarity. However, it requires a separate segmentation module.

3D features can also be extracted. Stereo cameras obtain a dense 3D reconstruction. The hand is segmented by

thresholding the depth map, which helps dealing with cluttered backgrounds. The depth map represents a surface, which is then matched against the model surface in 3D. Approximate reconstruction may be appropriate for fast processing, as an accurate 3D reconstruction is hard to get in real-time. It needs additional computational cost.

3.3.2 FEATURE SELECTION FOR OBJECT DESCRIPTION

Features are obtained from the input image sequence of hand gestures, they are further converted to symbols which are the basic elements of the HMM. Effective and accurate feature vectors play a crucial role in model generation of the HMM. For selecting good features, the following criteria are considered useful:

(1) Features should be preferably independent on rotation, translation and scaling.

(2) Features should be easily computable.

(3) Features should be chosen so that they do not replicate each other.

This criterion ensures efficient utilization of information content of the feature vector. The features obtainable from the image sequence of hand gesture are spatial and temporal features. To extract the shape features, we choose the FD to describe the hand shape, and to extract the temporal features, we use motion analysis to obtain the non-rigid motion characteristics of the gesture. These features should be invariant to the small hand shape and trajectory variations and it is also tolerant to small different gesture-speed.

3.3.2.1 Fourier descriptor

The objects may be described by their features in the frequency domain, rather than those in the spatial domain.

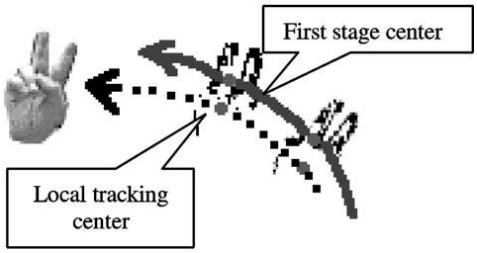


Fig. 14. Difference between the first stage center and the local tracking
center. The solid line is the trajectory of the first stage center, and dotted
line is the trajectory of the second stage center.

The local feature property of the node is represented by its Fourier Descriptors (FD) . Assume the hand-shape is described by external boundary points, then the FD representation may be used for boundary description.

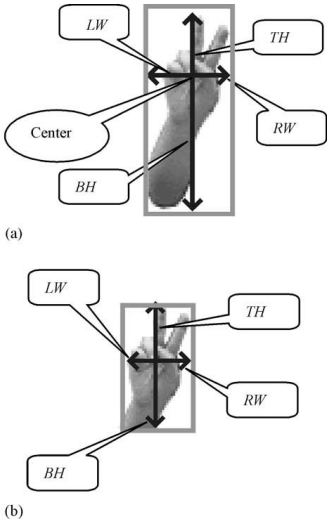


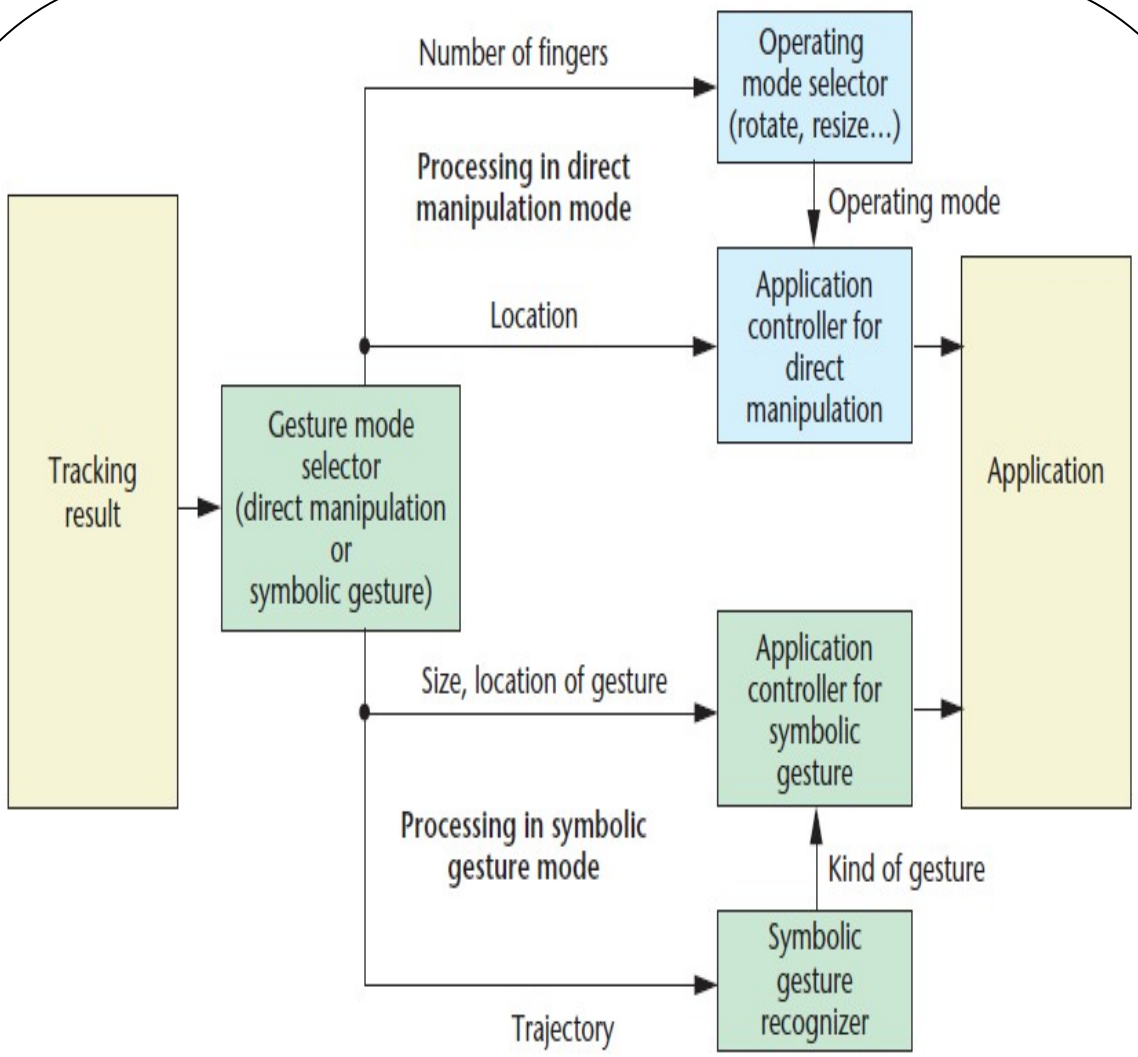
Fig. 15. (a) Four parameter of hand gesture bounding box,
(b) new hand gesture bounding box.

3.4 HAND GESTURE RECOGNITION AND TRACKING

Hand gesture recognition consists of *gesture spotting* that implies determining the start and end points of a meaningful gesture pattern from a continuous stream of input signals and, subsequently, segmenting the relevant gesture. This task is very difficult due to:

- the *segmentation ambiguity* and
- the *spatio–temporal variability* involved.

As the hand motion switches from one gesture to another, there occur intermediate movements as well. These transition motions are also likely to be segmented and matched with reference patterns, and need to be eliminated by the model. Moreover, the same gesture may dynamically vary in shape and duration even for the same gesturer.



Interaction based on direct manipulation and symbolic gestures

Orientation histograms can also be used as a feature vector for fast, simple hand gesture classification and interpolation. It is based on computing the Euclidean distance from the prototypes in terms of these features. The histogram of orientations, representing an orientation in terms of its angle, provides for translational invariance.

3.4.1 HAND GESTURE TRACKING SYSTEM

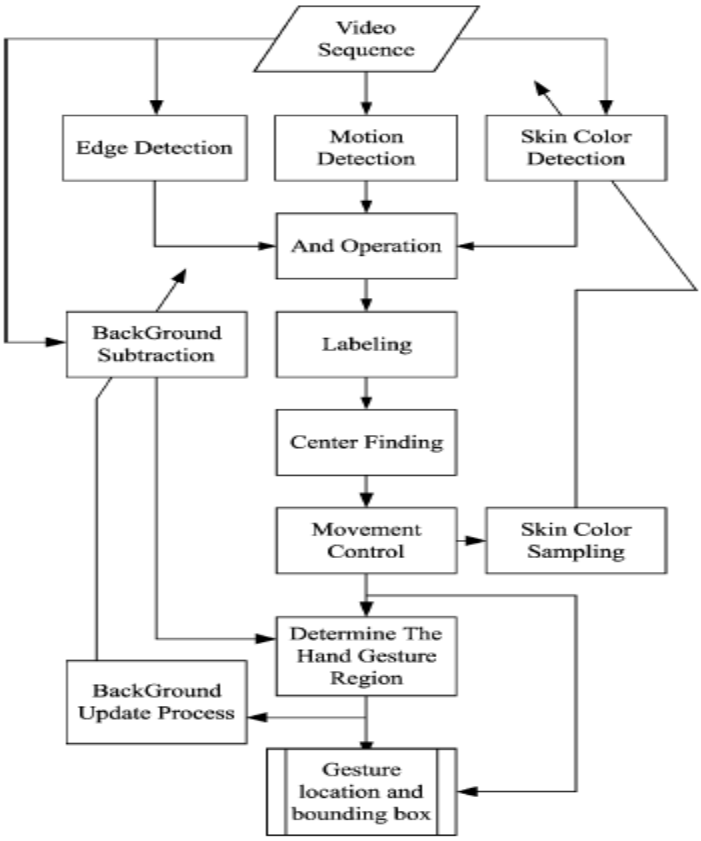


Fig. 8. The flow diagram of hand gesture tracking system.

3.4.1.1 Thresholding

Having extracted the moving object region, the thresholding on the frame difference can be applied to extract the possible moving region in complex background. Conventional thresholding methods, such as Ostu thresholding , are not suitable for the case of detecting motion difference. Instead, a simple thresholding technique can be used to extract moving regions. The threshold for motion detection is determined.

Fig. 2. (a) The origin frame, (b) apply our threshold, (c) apply Ostu thresholding.

The fig. shows that if there is no significant movement, Ostu thresholding method will generate a lot of noise.

3.4.1.2 Skin color detection

Skin can be easily detected by using the color information. First, we use the constraint, i.e. R . G . B, to find the skin color regions which may include a wide range of colors, such as red, pink, brown, and orange color. Therefore, we will find many regions other than the skin regions. However, those non-skin regions satisfy our constraint will be excluded due to there is no motion information, e.g. a region in orange color will not be misidentified as the hand region. Second, we may obtain some sample colors from the hand region. To find the skin regions, we compare the colors in the regions with the prestored sample color. If they are similar, then the region must be skin region. The hand region is obtained by the hand tracking process in the previous frame.

Fig. 3. (a) The origin frame, (b) extracted skin regions satisfying R . G . B, and (c) compare the colors of the extracted skin regions with the sample skin color.

Fig. 3 shows the skin detection results. The rectangular region is the hand region in the previous frame. Finally, some skin similar colors are eliminated, e.g. the orange color, and the skin color image is denoted.

3.4.1.3 Edge detection

Edge detection is applied to separate the arm region from the hand region. It is easy to find that there are fewer edges on the arm region than on the palm region. Here, we use a simple edge detection technique (e.g. Kirsch edge operator) to obtain different direction edges, and then choose the absolute maximum value of each pixel to form the edge image of ith frame.

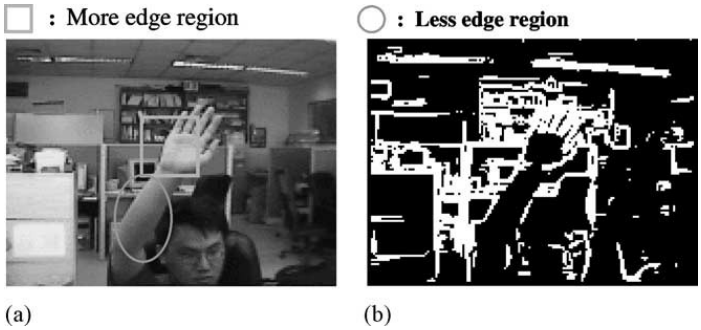


Fig. 4. (a) The origin frame, (b) the edge detection result.

Fig. 4 shows that the edges on the arm region are less than those on the palm region. We combine edge, motion, skin color region information to allocate the hand region.

3.4.1.4 Combination of motion, skin color, and edge detection

The hand gestures information consists of movement, skin color and edge feature. We use the logic ‘AND’ to combine these three types of information. The combined image has many features that can be extracted. Because the different image processing methods have extracted different kind of information. Each image consists of different characteristic regions such as motion regions, skin color regions and edge regions as shown in Fig. 5.

Fig. 5. The hand gesture information. (a) Original image (b) motion region (c) skin color region (d) edge region

Fig. 6. The combined region

The combined image consists of a large region in the palm area and some small regions in the arm area. These two regions are separated to allocate the hand region.

3.4.1.5 Region identification

A simple method for region identification is to label each region with a unique integer number which is called the labeling process. After labeling, the largest integer label indicates the

number of regions in the image. After the labeling process, the small regions can be treated as noise and then be removed.

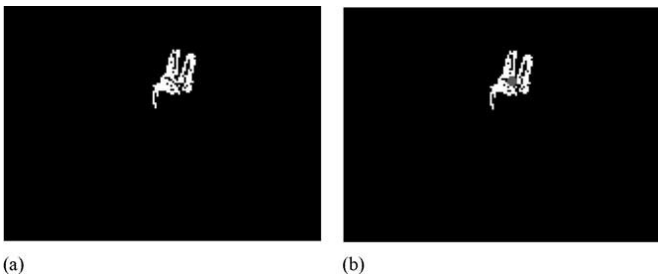


Fig. 7. (a) The labeling results (b) the correct center position of the hand region

3.4.1.6 Background subtraction

For gesture recognition process, more hand gesture information is needed.A simple background subtraction technique is used to obtain the hand gesture shape. The background model is created BGi by using the first frame.

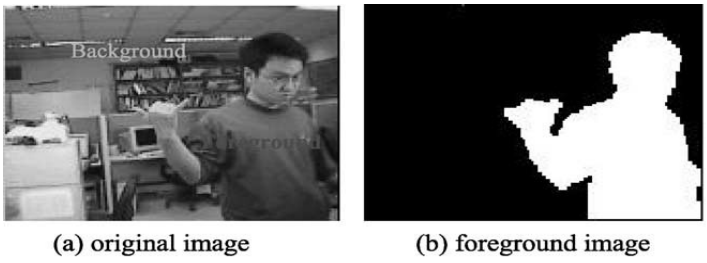


Fig. 10 The result of background subtraction.

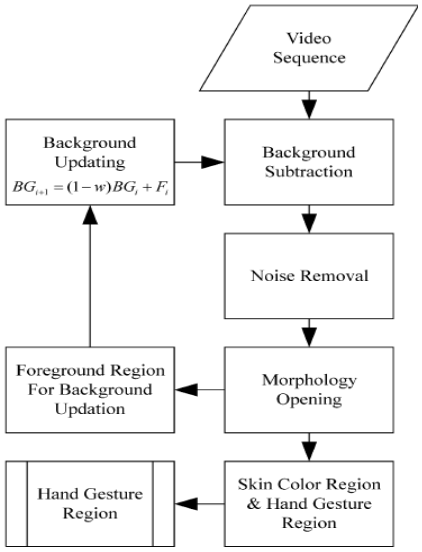


Fig. 11. Background subtraction process.

Fig. 11 Procedure to obtain the foreground.

To update our background model, the background model is updated by using the current frame Fi and foreground region FGi: We have generated two different types of foreground regions, one is FG1i = FGi; which is used to obtain the hand gesture region; and the other is FG2i(FG2i is obtained by dilating FG1i) which is applied for background updating process. FG1i has a compact shape, so that it can be used to obtain the hand region. Because there are small errors on the boundary of foreground and background, FG1i is not used to update the background. FG2i is generated for background updating. Only the background region is updated.

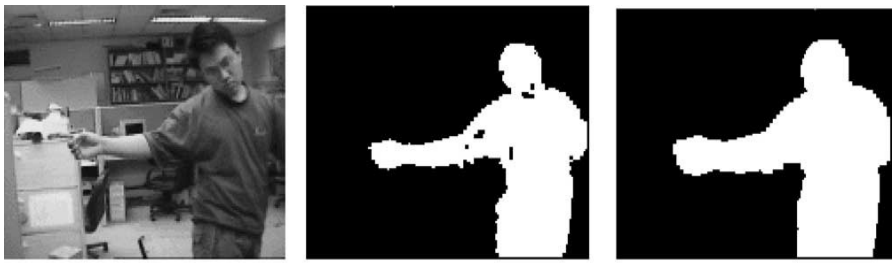


Fig. 12. Different type of foreground: (a) original image, (b) foreground FG1 for gesture tracking,
(c) foreground FG2 for updating the background.

The background is updated gradually, and the weighting factor w is 0.1. The updating process is more reliable for a smaller w. Finally, the foreground region does not really indicate the human hand. The skin color analysis and the hand region position tracking are to be applied to correctly extract the hand region.

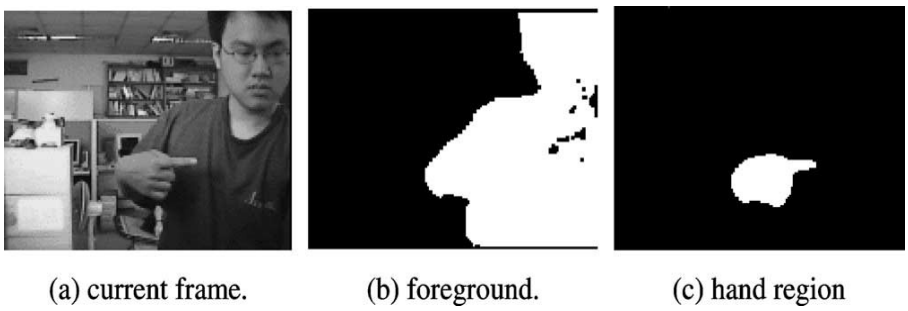


Fig. 13. Foreground region combining skin color and hand gesture position. It shows the results of hand gesture region extraction process.

3.5 HAND MOTION TRACKING TECHNIQUES

3.5.1. HMMs for Hand Gesture Recognition

HMM is a rich tool used for hand gesture recognition in diverse application domains.

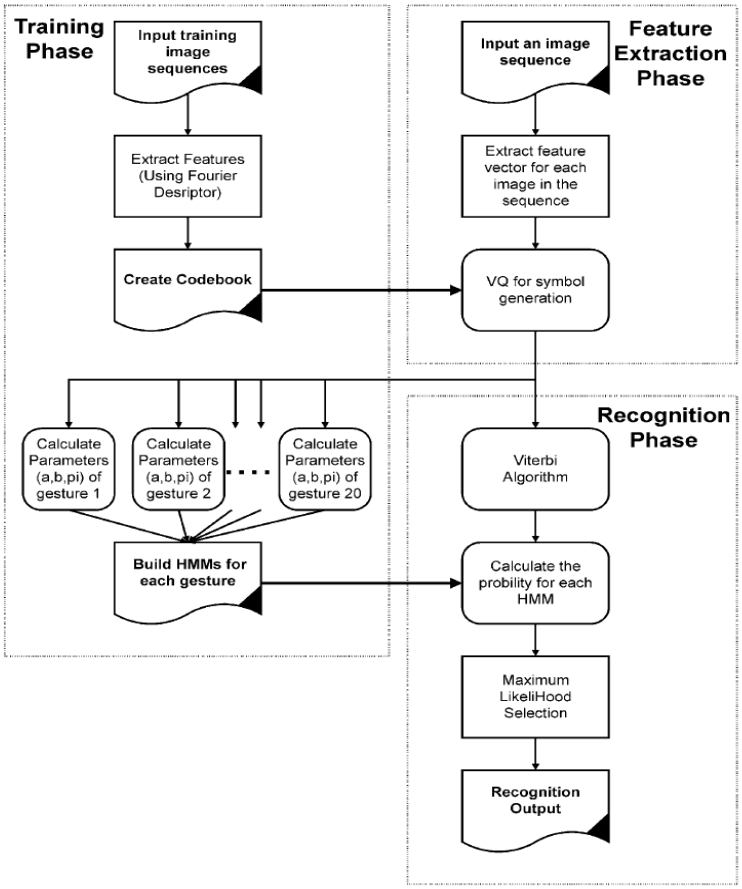


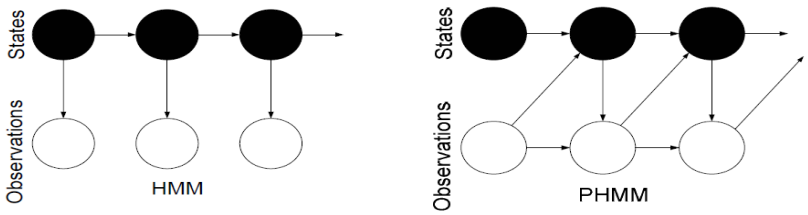
Fig. 1. The flow diagram of hand gesture recognition system.

Fig. 1 shows the flow diagram of hand gesture recognition system consisting of three phases: the feature extraction phase, the training phase, and the recognition phase. FD and motion features are combined as the feature vector to describe the moving object. Each feature vector is represented by a symbol. Each symbol corresponds to the designated partition generated through the vector quantization of the feature vectors of all possible hand-

shapes of the training gestures. For each feature vector, a symbol is assigned. In this system, the input image sequence is represented by a sequence of symbols. In training phase, we need to build a HMM for each gesture. In the recognition phase, a given input gesture is tested by every HMM with different model parameters. The outcome of the HMM with the maximum likelihood function is identified to recognize the gesture.

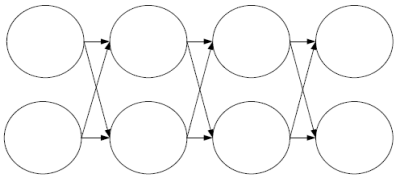
In this approach, before applying the HMM, the image sequence goes through several preprocessing steps such as low-pass filtering to reduce the noise, background subtraction to extract the moving objects, and binarization of the moving objects in order to generate blobs. The blobs roughly represent the poses of the human. The features are the amounts of object (black) pixels. These features are vector quantized, such that the image sequence becomes a sequence of vector quantized(VQ-labels), which are then processed by a discrete HMM.

HMMs are only piecewise stationary processes. But in gestures, all parts are transient. Hence, HMMs are not always suitable for gesture recognition. PHMM(Partly Hidden Markov Model)- a second order model was introduced.



If Markov condition is violated then HMM fails.Hence,Coupled Hidden Markov Models (CHMM) was introduced.

Coupling HMMs to model interactions between them

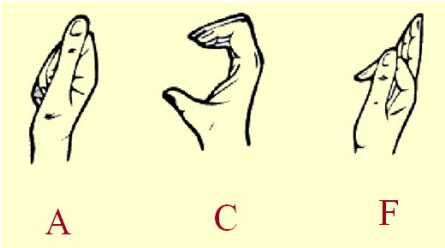


Several applications of hand gesture recognition have been developed based on HMMs.

(1) Sign Language Recognition

Sign language is a visual language. Usually, sign language consists of three major components:

i. finger-spelling: used to spell words letter by letter;
ii. word-level sign vocabulary: used for majority of communication; and
iii. nonmanual features: consisting of facial expression, position of tongue, mouth, and body.



Examples of sign language

(2) Graphic Editor Control

Another HMM-based model uses hand localization, hand tracking, and gesture spotting at preprocessing for hand gesture recognition. Hand candidate regions are located on the basis of skin color and motion. The centroids of the moving hand regions are connected to produce a hand trajectory, which is then divided into real and meaningless segments (categories). Input feature codes are extracted in terms of combined weighted location, angle, and velocity. This is followed by *c*-means clustering to generate the HMM codebook. Left-to-right HMM with ten states is used for recognizing hand gestures to control a graphic editor. The gestures modeled

include 12 graphic elements (circle, triangle, rectangle, arc, horizontal line, and vertical line) and 36 alphanumeric characters (ten Arabic numerals and 26 alphabets).

A charge-coupled device (CCD) camera placed in front of a monitor gives a sequence of gesture images from an image capture board. The *I* and *Q* components of the YIQ color system are used here to extract hand areas from input images. *A priori* knowledge about the hand location of a previous video image, the usual face location, and the size of the hand region are used to distinguish the hand region from multiple candidate regions.

The basic hand location algorithm is outlined as follows:

i. color system conversion from RGB to YIQ;
ii. estimation of similarity measures between model and input regions;
iii. thresholding similarity measures;
iv. noise removal and dilation;
v. detection of hand candidate regions;
vi. selection of hand region.

Garbage movements that come before and after a pure gesture are removed by using a spotting rule, whereby the user intentionally stops for a while (2–3 s) at the start and end of the gesture. An eight-connectivity counterclockwise directional chain code is used to convert the orientation angles into feature codes. The velocity component takes care of the fact that while a simple “circle" gesture may have an almost nonvarying speed, a complex “q” or “w” gesture generation can involve varying speeds of movement.

(3)Robot Control

A combination of static shape recognition, Kalman-filter-based hand tracking, and an HMM-based temporal characterization scheme is developed for reliable recognition of single-handed dynamic hand gestures. Here, the start and end of gesture sequences are automatically detected. Hand poses can undergo motion and discrete changes in between the gesture, thereby

enabling one to deal with a larger set of gestures. However, any continuous deformation of hand shapes is not allowed. The system is robust to background clutter, and uses skin color for static shape recognition and tracking.

Real time implementation of robot control:

The user is expected to bring the hand to a designated region for initiating a gestural action to be grabbed by a camera. The hand should be properly illuminated, and the user should avoid sudden jerky movements. When the user moves the hand away from the designated region, it signals the end of the gesture and the beginning of the recognition process. The grabber and tracker are operated as synchronized threads.

The five hand gestures and the corresponding instructions modeled for the robot are:

i. **closed to open forward:** move forward;
ii. **closed to open right:** move forward then right;
iii. **closed to open left:** move forward then left;
iv. **open to closed right:** move backward then right; and
v. **open to closed left:** move backward then left.

Static hand shapes are described by their contours, specified through mouse clicks on the boundary of the image, and subsequently fitting a B-spline curve. Translated, scaled, and rotated versions of these shapes are also added to the prior. For a test shape, a matching is made with these stored priors. The closest contour match is chosen for tracker initialization. A gesture is considered as a sequence of epochs, where each epoch is characterized by a motion of distinct hand shapes. Kalman filter is used for hand tracking, to obtain motion descriptors for the HMM. The moving hand is approximated as a planar rigid shape, under the assumption that the fingers are not being flexed, and the perspective effects are not significant. The left-to-right HMM, with four states and an out degree of three, proceeds by doing the following:

- extracting symbolic descriptors of the gesture at regular intervals from the tracker and hand shape classifier;
- training HMMs by the sequence of symbolic descriptors corresponding to each gesture;
- finding the model, which gives maximum probability of occurrence of the observation sequence generated by the test gesture.

The gesture recognition algorithm is outlined as follows.

i. Detect hand for boot-strapping the tracker.
ii. Recognize the starting hand shape, and initialize tracker with its template.
iii. *While* hand is in view repeat
    a) Track the hand and output encoded motion information *until* shape change is detected.
    b) Recognize the new shape and initialize the tracker with template of the recognized shape
iv. Using HMM, find the gesture, which gives the maximum probability of occurrence of observation sequence composed of shape templates and motion information.

3.5.2. Condensation Algorithm

The condensation algorithm was developed based on the principle of particle filtering.

In particle Filtering, approximate arbitrary distributions are present with a set of random samples. It deals with clutter and ambiguous situations more effectively, by depending on multiple hypotheses. Particles of the hand configuration distribution are updated in each frame. To reduce the number of samples: Semi-parametric particle filters and local search algorithms may be used.

Condensation algorithm was originally applied effectively in tracking rapid motion of objects in clutter . A mixed-state condensation algorithm has been extended to recognize a greater number of gestures based on their temporal trajectories. Here, one of the gesture models involves an augmented office white-board

with which a user can make simple hand gestures to grab regions of the board, print them, save them, etc. In this approach, compound models that are very like HMMs are allowed, with each state in the HMM being one of the defined trajectory models.

3.5.3. FSMs for Hand Gesture Recognition

A gesture can be modeled as an ordered sequence of states in a spatio–temporal configuration space in the FSM approach. This has been used to recognize hand gestures. A method to recognize human-hand gestures using a FSMmodel- based approach has been used. The state machine is used to model four qualitatively distinct phases of a generic gesture—static start position (static at least for three frames), smooth motion of the hand and fingers until the end of the gesture, static end position for at least three frames, and smooth motion of the hand back to the start position. The hand gestures are represented as a list of gesture vectors and are matched with the stored gesture vector models based on vector displacements. Another state-based approach to gesture learning and recognition is present. Here, each gesture is defined to be an ordered sequence of states, using spatial clustering and temporal alignment. The spatial information is first learned from a number of training images of the gestures. This information is used to build FSMs corresponding to each gesture. The FSM is then used to recognize gestures from an unknown input image sequence.

The system is highly reconfigurable, and no training concept is involved. The FSM has five states, viz., start (S), up (U), down (D), left (L), and right (R). The length of the signature pattern is independent of the duration overwhich the gesture is performed, but depends on the number of changes in the dominant direction of motion. Self-loops are essential to accommodate the idleness of hand movement while changing the direction of hand waving. Symbolic commands such as come closer, go far, move left, move right, and emergency stop are recognized. For example, “come closer” is modeled by repeatedly sweeping one hand toward the body and then slowly away (say, by S-D-U-U-D-U-D-D-UD). Again,

move right may be represented by moving the hand continuously toward the right direction and then the left (say, by S-L-R-R-L-L-R-L-R-L).

A lexicon is constructed to model the motion profile of the hand for each gesture. This knowledge is utilized for signature representation in the form of simple production rules, followed by their interpretation. These can be used as input to a robot programming language for generating machine-level instructions, in order to mimic the intended operation of the corresponding gesture.

3.5.4. Connectionist Approach to Hand Gesture Recognition

The gesture recognition system for American Sign Language(ASL) in this approach has been divided into two distinct steps. In the first step, multiscale motion segmentation is applied to track movements of objects between the frames. Regions between two consecutive frames are matched to obtain two-view correspondences. Then, *affine* transformations are computed to define pixel matches and recognize the motion regions or trajectories. In the second step, a TDNN is used to match the trajectory of movements to a given gesture model.

To recognize the sign language, only those object areas of motion where skin color is detected are determined first. Then, the detected regions of motion are merged until the shape of the merged region is either an ellipse or a rectangle. This is because sign languages are typically described by the relation between head (a large elliptical shape), palm (small elliptical shape), and (or) closed hand (a rectangular shape). The TDNN with two hidden layers is employed to classify the motion of various regions over time as a particular gesture (sign) in the sign language (ASL).

4.APPLICATIONS OF GESTURE RECOGNITION

Gesture recognition has wide-ranging applications such as the following:

- developing aids for the hearing impaired;
- enabling very young children to interact with computers;
- designing techniques for forensic identification;
- recognizing sign language;
- medically monitoring patients' emotional states or stress levels;
- lie detection;
- navigating and/or manipulating in virtual environments;
- communicating in video conferencing;
- distance learning/tele-teaching assistance;
- monitoring automobile drivers' alertness/drowsiness levels, etc.
- Public Display Screens: Information display screens in Supermarkets, Post Offices, Banks that allows control without having to touch the device.
- Robots: Controlling robots without any physical contact between human and computer.
- Graphic Editor Control: Controlling a graphic editor by recognizing hand gestures using HMM

5.CONCLUSION

The importance of gesture recognition lies in building efficient human–machine interaction. Its applications range from sign language recognition through medical rehabilitation to virtual reality. The major tools surveyed for this purpose include HMMs, particle filtering and condensation algorithm, FSMs, and ANNs. Facial expression modeling involves the use of eigenfaces, FACS, contour models, wavelets, optical flow, skin color modeling, as well as a generic, unified feature-extraction-based approach. A hybridization of HMMs and FSMs can increase the reliability and accuracy of gesture recognition systems.

Soft computing tools pose another promising application to static hand gesture identification. For large datasets, neural networks have been used for representing and learning the gesture information. Both recurrent and feed forward networks, with a complex preprocessing phase, have been used for recognizing static postures. The dynamic movement of hand has been modeled by HMMs and FSMs. The similarity of a test hand shape may be determined with respect to prototype hand contours, using fuzzy sets. TDNN and recurrent neural networks offer promise in capturing the temporal and contextual relations in dynamic hand gestures. Similarity-based matching of the retrieved images may be performed on clusters of databases, using concepts from approximate reasoning, searching, and learning. Thus, gesture recognition promises wide-ranging applications in fields from photojournalism through medical technology to biometrics.

6. REFERENCES

1. Sushmita Mitra and Tinku Acharya, “Gesture Recognition: A Survey”, IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 37, no. 3, may 2007
2. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.6021&rep=rep1&type=pdf
3. http://www.iit.demokritos.gr/IIT_SS/Presentations/Athitsos_demokritos.ppt
4. http://www.computing.dcu.ie/~alistair/CA107.ppt
5. http://129.97.86.45:88/redmine/attachments/22/Hand_Gesture_Recognition.ppt
6. Kenji Oka , Yoichi Sato and Hideki Koike,”Real-Time Fingertip Tracking And Gesture Recognition”,University of Tokyo