

## Computer Based Human Gesture Recognition With Study Of Algorithms

Mr. Kaushik I Manavadariya, Mr. Ankit J Faldu, Mr. Parag C Shukla,  
Mr. Deven J Patel

<sup>1</sup>(Master of Computer Application, Atmiya Institute of Technology & Science, India)

<sup>2</sup>(Master of Computer Application, Atmiya Institute of Technology & Science, India)

<sup>3</sup>(Master of Computer Application, Atmiya Institute of Technology & Science, India)

<sup>4</sup>(Master of Computer Application, Atmiya Institute of Technology & Science, India)

---

**Abstract :** Gesture recognition is a technology of computer science with the goal of interpreting human gestures or human motion via mathematical algorithms. Gesture recognition begins from facial, voice, eye tracking, lip movement, ear etc. Also the identification and recognition of posture, walking style, and human behaviours is also the part of gesture recognition techniques. To interpret the sign language, the person is used a device like mobile, computer which is embedded with camera like Depth-aware cameras, stereo camera, single camera etc. It is a good way for building a connection between human and computer system. Gesture recognition enables humans to interface with the machine and interact naturally without any mechanical devices. There are two types of gesture: Online gesture, Offline gesture. Two different approaches in gesture recognition is related to 3D model and related to appearance-based. 3D model is used to provide information related the body parts like thumb, position of palm, joint angles , etc.. In short, Appearance-based systems use images or videos for direct interpretation.

**Keywords -** Computer Aided Design (CAD), Active Appearance Models (AAM), Hidden Markov Models (HMM), Sequence of the Most Informative Joints (SMIJ), Histogram-of-Motion Words (HMW), System for secure face identification (SCiFI), Point distribution model (PDM), Discrete cosine transform (DCT), Principal component analysis (PCA), Discrete wavelet transform (DWT).

---

### I. INTRODUCTION

Since the introduction of the most common computer based input devices not a lot have to change. This is probable because the existing devices are sufficient. It is also having been so tightly integrated with everyday routine life, that the new applications or software and hardware are constantly introduced. The meaning of communicating with computers at the moment is limited to keyboards, light pen, trackball, etc.

These devices have grown to be very popular but inherently limit the speed and naturalness with which we interact with the computer.

As the industry follows Moore's Law since middle Nineteen Sixties, powerful machines area unit engineered equipped with a lot of peripherals. Vision based mostly interfaces area unit possible and at this moment the computer is in a position to "see". Thence users' area unit allowed for richer and user-friendlier man-machine interaction. This will cause new interfaces that may permit the readying of latest commands that aren't doable with the present input devices. Lots of time is going to be saved moreover.

Computer recognition of hand gestures might offer a lot of natural-computer interface, permitting folks to purpose, or rotate a CAD model by rotating their hands. Hand gestures will be classified in 2 categories: static and dynamic. A static gesture could be an explicit hand configuration and create, portrayed by one image. A dynamic gesture could be a moving gesture, portrayed by a sequence of pictures. We are going to specialize in the popularity of static pictures.

### II. Spatial Gesture Algorithm

Spatial gesture algorithm mainly divided in two parts. 1) Appearance based, 2) 3D model based

#### 2.1 Appearance-based Gesture Recognition

Work in the field of perception-based gesture recognition usually first segments parts of the input images, for example the hand, and then uses features calculated from this segmented input like shape or motion. At present first "working" real-time gesture recognition system developed as sign language recognition systems was developed.

A person-independent gesture recognition system is presented in the system uses global motion features extracted from each difference image of the image sequence, and HMMs as a statistical classifier. A view-based approach to the representation and recognition of human movement using temporal templates is

presented. The authors develop a recognition method by matching temporal templates against stored instances of views of known actions.

The gesture recognition system represented in can recognize a vocabulary of 46 single-hand gestures of the American Sign Language finger spelling alphabet and digits in real time. Each and every video frame is processed independently, and dynamic gestures are replaced by static ones. The system was trained and tested using data of one person and thus is highly person dependent. A two-stage classification procedure is presented where an initial classification stage extracts a high-level description of hand shape and motion. A second stage of classification is then used to model the temporal transitions of individual signs using a classifier bank of Markov chains combined with Independent Component Analysis.

In a classification system using global features is presented. In contrast to the work presented here for the training data those are manually segmented and only the relevant part of the video, it means exactly the frames where a sign is gestured is taken into account. The authors studied and reported preliminary experiments on the fusion of two appearance-based approaches, PCA and LDA, for face verification and recognition.

Visual tracking of object in complex environment is currently one of the most challenging and intensively studied tasks in machine vision field. Various visual cues including optical flow, edge, color, and depth have been employed in tracking.

### **2.1.1 Active Appearance Models (AAM)**

AAM is a method to match deformable statistical models of shape and appearance to an image. The algorithm consists in fitting AAM model to an image minimizing the error between the input image and the closest deformable model. To build a model, a set of landmarks of enhanced points are selected with this set of landmarks, a mesh is wrapped to the image using Delaunay triangles. Using a training set of several images and PCA reduction, the model of the shape and the appearance is a linear combination of the shapes and appearance of all training images.

First Equation (1) is represents the shape and appearance models, where  $\bar{x}$  is the mean shape,  $\bar{g}$  is the mean normalized texture is the parameter controlling the shape and appearance and  $Q_s$  and  $Q_p$  are matrices representing the shape and appearance variations derived from the training image set. First, AAM models have been used to obtain a statistical representation of the human face. Extracting some important features of the mesh resulting of the AAM algorithm, like mouth and eyebrows shape, a neural network has been trained in conjunction with a global optimizer to distinguish between three different states, happy, sad, and normal (no emotion).

### **2.1.2 3D image Based Gesture Recognition**

Hand gesture recognition is a discipline that has been around for decades in the human-computer interaction research community. The implementation of gesture recognition typically requires the use of different imaging and tracking devices or gadgets. A lot of work has been done on this area based on computer vision. However there are many problems preventing vision-based hand gesture recognition system for everyday use such as hand tracking and view invariant gesture recognition.

Hand detection and tracking is a difficult issue within complex background and various lighting conditions. Typical hand gesture recognition systems use color gloves or detecting and tracking hand in static background to overcome this problem.

All hand gesture recognition systems using one camera have apparent handicaps. First, different gestures may share the same 2D route due to the projection, and the same gesture may have various projections from different viewpoints in one plane. Second thing is that the people solve viewpoint problems by adding more categories with different viewpoints while this increases the computation burden and decreases the recognition rate.

Here a method is worked with 3D image data from a range camera to achieve invariance to viewpoint. We proposed a volume motion template to overcome the viewpoint problem in a single-directional stereo camera environment. Another solution is using more than one camera.

There are 2 types of dynamic hand gestures: directional gestures [DG] and non-directional [NG] ones. DG is simple linear movements without changes of directions, while NG is nonlinear such as circle and triangle. More important thing is DG is dependent on viewpoint while the NG is independent on viewpoint. In most vital part is NG is almost in a plane, which we call principal gesture plane. There is rarely information loss when the 3D trajectory of gestures projected onto the principal gesture plane and gesture recognizing part is done using the projected trajectory. Various algorithms have been used in dynamic gesture recognition such as HMMs, TDNN, time-compressing templates and dynamic time wrapping.

Hidden Markov Models (HMMs) is one of the most successful and widely used tools for modeling signals with spatiotemporal variability. In human communication gestures are widely used to convey or emphasis information and the automatic recognition of gestures has therefore received much attention in many

areas of computer vision research. Reconstruction of humans and their exact pose has been a widely used approach but a current trend is to do recognition directly on image data.

All systems relying on information extracted from 2D images are faced with the common problem that the 2D image is a projection of the 3D gestures images. Some outcome of this by merely addressing gestures carried out in a plane parallel to the camera-plane. To handle different view-points a new class can be learned for each gesture for each view-point (given some resolution). This leads to even more tedious training and the risk that too many classes might lead to overlap in the feature-space, which of results again in reduced recognition rates. Furthermore, some gestures might be ambiguous in such systems. E.g., a “point right” gesture seen from the front is similar to a “point straight ahead” gesture seen from a side view. To overcome such problems recent methods have investigated the use of 3D data as opposed to 2D image data.

An additional versatile approach is to outline and acknowledge gestures by a collection of primitives. We have a tendency to represent gestures as Associate in Nursing ordered sequence of 3D motion primitives (temporal instances). We have a tendency to target arm gestures and so solely section the arms (when they move) and herewith suppress the remainder of the (irrelevant) body data. Concretely we have a tendency to use 3D double distinction pictures to extract the moving arms and represent this knowledge by their form Context. We have a tendency to build the primitives invariant to rotation round the vertical axis by re-representing the form Context mistreatment Spherical Harmonic basis functions, yielding a Harmonic form Context illustration. In every frame the primitive, if any, that best explains the determined knowledge is known resulting in a separate recognition drawback since a video sequence of vary knowledge are going to be born-again into a string containing a sequence of symbols, every representing a primitive.

### III. SPATIAL GESTURE RECOGNITION SUB ALGORITHM

3D images base algorithm and appearance base algorithm contain lots of sub algorithm

#### 3.1 Skeletal algorithm

The human body is an articulated system that can be represented by a hierarchy of joints that are connected with bones, forming the skeleton. Different joint configurations produce different skeletal poses and a time series of these poses yields the skeletal motion. An action can be simply described as a collection of time series of 3D positions of the joints in the skeleton hierarchy.

A better description is obtained by computing the joint angles between any two connected limbs and using the time series of joint angles as the skeletal motion data. Let  $a_i$  denote the joint angle time series of joint  $i$ , i.e. in equation (2).

Where in equation (2)  $[T]$  is the number of frames in an action sequence. An action sequence can then be seen as a collection of such time-series data from different joints, i.e.,  $A = [a_1 a_2 \dots a_j]$ , where  $j$  is the number of joints in the skeleton hierarchy. Hence,  $A$  is the  $T \times J$  matrix of joint angle time series representing an action sequence.

Common modeling methods such as LDS or HMM model the evolution of the time series of joint angles. However, instead of directly using the original joint angle time series data  $A$ , one can also extract various types of features from  $A$  such as the mean or variance of joint angle time series, or the maximum angular velocity of each joint. For the sake of generality, we will denote this operation with  $\mathcal{O}$ . Here  $\mathcal{O}(a): \mathbb{R}^{|a|} \rightarrow \mathbb{R}$  is a function that maps a time series of scalar values to a single scalar value. Furthermore, one can extract such features either across the entire action sequence or across smaller segments of the time series data. The former case describes an action sequence with its global statistics whereas the latter case emphasizes more the local temporal statistics of an action sequence.

Different actions require human to engage different joints of the skeleton at different intensity (energy) levels at different times. Hence, the ordering of joints based on their level of engagement across time should reveal significant information about the underlying dynamics.

In order to visualize this phenomenon, let consider the labeled joint angle configuration shown in Fig 3.1(a), and perform a simple analysis on Dataset 1, The analysis is based on the following steps: i) partition an action sequence into a number of congruent segments, ii) compute the variance of the joint angle time series of each joint over each temporal segment (note that  $\mathcal{O}$  is defined to be the variance operator in this particular case), iii) rank-order the joints within each segment based on their variance in descending order, iv) repeat the first three steps to get the orderings of joints for all the action sequences in the dataset. Below we investigate the resulting set of joint orderings for different actions. Fig 3.1(b), shows the distribution of the top-ranking joints for different actions.

Fig 3.2 shows the histogram of the top-ranking 5 joints for four different actions. While the differences in the distribution of 1st-, 2nd-, or 3rd-ranking joints, and so on, for actions 4 (punch) and 6 (wave one) are evident, actions 1 (jump) and 2 (jumping jacks) require closer look at the histograms. In short, different sets of joints reveal discriminative information about the underlying structure of the action.

The new feature representation, which we call Sequence of the Most Informative Joints (SMIJ), has two main components: i) the set of the most informative joints in each time segment, and ii) the temporal evolution of the set of the most informative joints over all of the time segments. To extract this representation from the time-series data, we first partition the action sequence into  $N_s$  temporal segments and compute  $\mathcal{O}$  over each segment. Let equation (3) be a segment of  $a^i$  where  $t_s^k$  and  $t_e^k$  denote the start and the end frames for the segment  $k$ , respectively. Then, an action sequence is written as a collection of features,  $F = \{f_k\}_{k=1, N_s}$  where  $f_k =$  equation (4).

The feature function,  $\mathcal{O}(a_k^i)$ , provides a measure of information (e.g. the mean or variance of joint angles, or the maximum angular velocity) of the joint  $i$  in the temporal segment  $ak$ . We then rank-order all the joints in  $f_k$  based on the value of  $\mathcal{O}$  and define SMIJ features as

$$SMIJ = \{ \{ \text{idof}(\text{sort}(f_k), n) \} \}_{k=1, \dots, N_s} \}_{n=1, \dots, N}$$

Where the sort operator sorts the joints based on their local  $\mathcal{O}$  score in descending order, the idof(.) operator returns the id of a joint that ranks  $n$ th in the joint ordering, and  $N$  specifies the number of top-ranking joints included in the representation. In other words, the SMIJ features represent an action sequence by encoding the set of  $N$  most informative joints at a specific time instant (by rank-ordering and keeping the top-ranking  $N$  joints) as well as the temporal evolution of the set of the most informative joints throughout the action sequence (by preserving the temporal order of the top-ranking  $N$  joints). The resulting feature descriptor is  $N_s \times N$  - dimensional.

Since the proposed representation is a set of sequences over a fixed alphabet - the joints, we use the Levenshtein distance,  $DL(S_i, S_j)$ , [10] for comparing the SMIJ features from two different sequences,  $S_i$  and  $S_j$ . The Levenshtein distance measures the amount of difference between two sequences of symbols such as strings. It is defined as the minimum number of operations required to transform one sequence into the other where the allowable operations are insertion, deletion, or substitution of a single symbol. We use a normalized version of the Levenshtein distance,

The size of the SMIJ feature depends on the number of segments  $N_s$ , which depends on how the action sequence is partitioned. The Levenshtein distance between two sequences is at least equal to the difference in lengths of the two sequences. Since we require a distance of zero when two actions have the same rank-ordering, irrespective of their actual temporal length, one natural choice is to fix  $N_s$  to a constant value for all the action sequences.

### Baseline Feature Representations

Instead of stacking the top-ranking  $N$  joints from all temporal segments into a single sequence of symbols, while keeping the temporal order of the joints intact, we create histograms separately for the 1st-ranking joints, 2nd-ranking joints and so on, from all temporal segments and then concatenate them as a feature descriptor, called Histograms of Most Informative Joints (HMIJ), to represent the action sequence in step, i.e.,

$$SMIJ = \{ \text{hist} \{ \text{idof}(\text{sort}(f_k), n) \} \}_{k=1, \dots, N_s} \}_{n=1, \dots, N}$$

Where the hist operator creates a  $j$ -bin  $l_1$ -normalized histogram from the input joint sequence, resulting in  $j \times N$  - dimensional feature descriptor. It is important to note that HMIJ features ignore the temporal order of the top-ranking  $N$  joints, and hence, will be used as a reference to assess the importance of preserving the temporal order information in the feature representation (as SMIJ features do).

Another popular method for representing an action sequence is based on the bag-of-motion words model. We first cluster the set of  $k$ s into  $K$  clusters (i.e., motion words) using  $k$ -means or  $k$ -medoids and then count the number of motion words that appear in a particular action sequence, yielding the Histogram-of-Motion Words (HMW) representation.

As mentioned earlier, one of the most common techniques to analyze human motion data is based on modeling the motion with a Linear Dynamical System over the entire sequence,

Even though we do not provide an exhaustive list of all possible feature representations, we believe that these three feature representations, i.e., HMIJ, HMW and LDSP, are comprehensive enough to demonstrate the power of the proposed SMIJ features in terms of discriminability and interpretability for human action recognition.

### Classification Method:

In this section we examine the quality of different feature representations by evaluating their classification performance using well-known methods such as 1 nearest neighbor (1-NN) and support vector machine (SVM). Since we are investigating feature descriptors with different characteristics, we need to select the distance metrics according to the feature representations. We use the Levenshtein distance as explained in Section 2 for classification based on SMIJ. We use the  $\chi^2$  distance for classification based on histogram Feature representation HMIJ and HMW. Finally, we use the Martin distance as a metric between dynamical systems for

classification based on LDSP. For SVM based classification, we follow one-vs-one classification scheme and use Gaussian kernel  $K(S_i, S_j) = e^{-\gamma D^2(S_i, S_j)}$  with an appropriate distance function  $D(S_i, S_j)$  depending on the feature type listed above. As for the SVM hyper parameters, we set the regularization parameter  $C$  to 1 and the Gaussian kernel function parameter  $\gamma$  to the inverse of the mean value of the distances between all training sequences as in.

### 3.2 Hidden Markov Model [HMM]

HMM statistical model which is including the system being modeled is assumed to be a Markov process with unobserved (hidden) states.

The ability of HM models to compensate time and amplitude variations has been proven for speech recognition, gesture recognition, sign language recognition and human action recognition. HMM were introduced in the mid 1990's, and quickly became the recognition method of choice, due to its implicit solution to the segmentation problem.

HMM is a collection of finite states connected by transitions probability. Each and every state is characterized by two sets of probabilities: a transition and either a discrete output probability distribution or a continuous output probability density function. HMM can be defined by: (1) a group of states Associate in Nursing initial state  $S_i$  and a final state  $S_f$ ; (2) The transition likelihood matrix,  $A = [a_{ij}]$ , where  $a_{ij}$  is that the transition likelihood of taking the transition from state  $i$  to state  $j$ ; (3) The output likelihood matrix  $B = [b_{ok}]$ , where  $b_{ok}$  represents distinct observation image. For an eternal HMM,  $B = [b_{xk}]$ , where  $x$  represents continuous observations of  $k$  dimensional random vectors. With the initial state distribution  $\Pi = [\pi_i]$ , the entire parameter set of the HMM are often expressed succinctly as  $\lambda = (A, B, \Pi)$ . HMM are often primarily based either on distinct observation likelihood distributions or continuous mixture likelihood density functions. Within the distinct HMM, the distinct likelihood distributions area unit sufficiently powerful to characterize random events with an inexpensive parameters. The principal advantage of continuous HMM is that the ability to directly model the parameters of an eternal signal. A semi continuous HMM provides a framework for unifying the distinct and continuous HMMs. In our analysis, we have a tendency to solely take into account a distinct HMM. There are a unit 3 basic issues that has got to be solved for the \$64000 application of HMM: 1) analysis, 2) secret writing, and 3) learning. The answer to those 3 issues area unit the Forward-Backward algorithmic program, the Viterbi algorithmic program, and also the Baum-Welch algorithmic program, If this will be calculated for all competitor models for associate degree observation sequence, then the model with the best likelihood are often designated because the first objective gesture model. In our analysis, the observation image zero is delineate because the sequence of  $O_i$ .

The initial topology for HMM can be determined by estimating how many different states are involved in specifying a gesture. While different topologies can be organized for each gesture, a 4-state HMM with skip transition is determined to be sufficient for our work as shown in Figure.

### 3.3 Databases

#### 3.3.1 LTI-Gesture Database

The LTI-Gesture database was created at the Chair of Technical Computer Science of the RWTH Aachen University. It contains fourteen dynamic gestures, one hundred and forty training and testing sequences. An error rate of 4.3% was achieved on this database. HMMs are required for recognition as some gestures can only be distinguished using motion. In particular, the gestures 'five', 'stop', and 'pause' have the same hand shape but differ in the moment of the hand.

#### 3.3.2 DUISBURG-Gesture Database

For the training and the testing of the system presented in video sequences of twenty four different dynamic gestures were recorded. The resolution of the video sequences was 96 by 72 grey-scale pixel and 16 frames per second. Figure shows some examples of the different gestures. The database consists of 336 image sequences that contain gestures of 12 different persons. With a leaving-one-person-out classification an error rate of 7.1% was achieved.

We recorded a database of finger spelling gestures of German Sign Language. The database contains 35 gestures with video sequences showing the signs 'A' to 'Z', 'SCH', the German umlauts "A", "O", "U", and the numbers '1' to '5'. HMMs are necessary for recognition as five of the gestures contain inherent motion ('J', 'Z', "A", "O", and "U"). The database consists of disjunction sets of 700 training sequences and 700 test sequences.

### 3.4 Face recognition algorithms and identification

The novel algorithm that we use in SCiFi performs well in the one-to-many identification task since it can generalize to unseen conditions. There are other face recognition algorithms which are robust to changes in

the environment in which photos are taken. The algorithm of SCiFI is unique, however, in that it lends itself easily to secure computation, which is inherently based on the usage of discrete mathematics. Other effective face recognition algorithms employ continuous face representations which are compared by complex measures of similarity that in some cases are not even metric. Such representations are not easily supported by cryptographic algorithms. A naive conversion from the continuous comparison methods used in face recognition to a discrete measure using and the effect to the accuracy of image recognition and result in degraded performance

**3.5 3-D face recognition using local appearance-based models**

Recorded point clouds may have different poses and expressions. In order to extract proper local information from corresponding local facial blocks, a precise peer-to-peer correspondence should be established. In the following subsections, the processing steps of the face registration and depth image generation are explained.

Real Time Gesture Recognition Using Eigen space from Multi Input Image Sequences

In interactive systems, a complicated gesture, which may have a self-occlusion and change of human part in the direction of image depth, must be recognized correctly. The degree information of gesture must be obtained correctly even if the gesture is complicated. In addition, the processing must be performed in real time.

**Human Part Segmentation and Linked Human Part Image**

It shows process constituting a linked human part image. In our experiment, we use two kinds of input image which are captured from a side view camera (left frontal) and a top view camera. In Fig 3.6, the binary input image 1 was captured from a side view camera and the input image 2 was captured from a top view camera.

**IV. EQUATIONS**

$$x = \bar{x} + Qsc \quad (1)$$

$$a_i = \begin{cases} a_t^i \\ t = 1 \end{cases} \quad (2)$$

$$a_k^i = \{a_k^i\}_{k=s}^t \quad (3)$$

$$fk = [(a_k^1) (a_k^2) (a_k^j)] \quad (4)$$

**V. FIGURES**

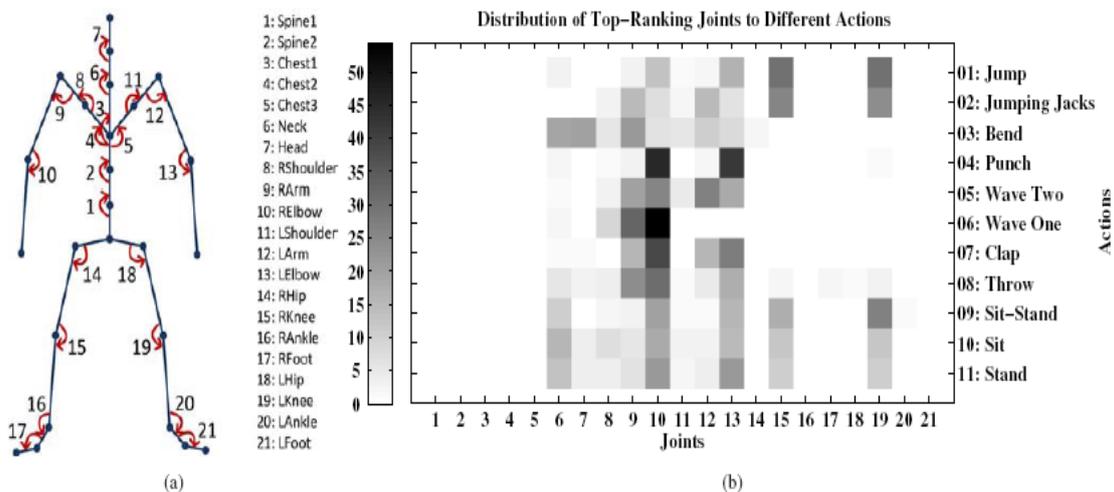
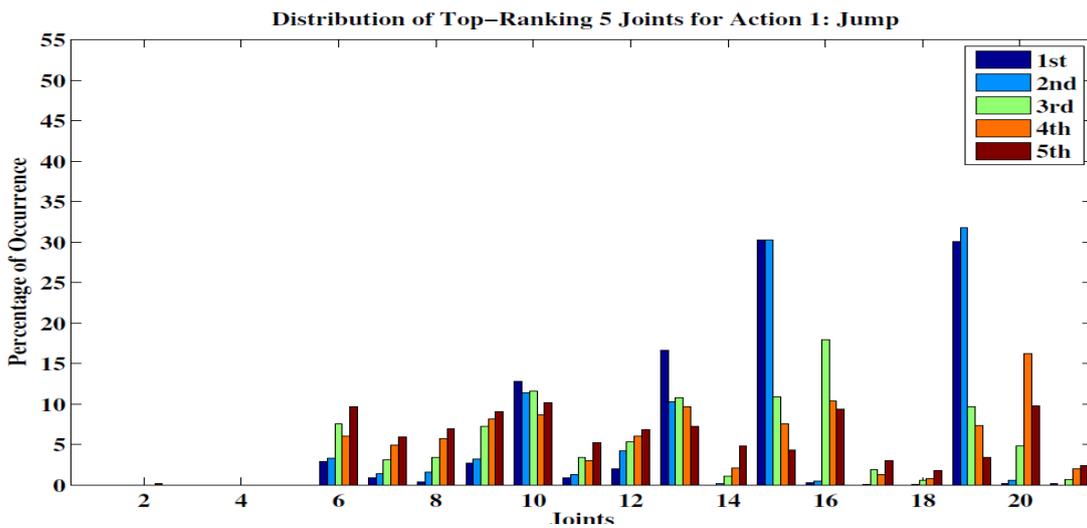
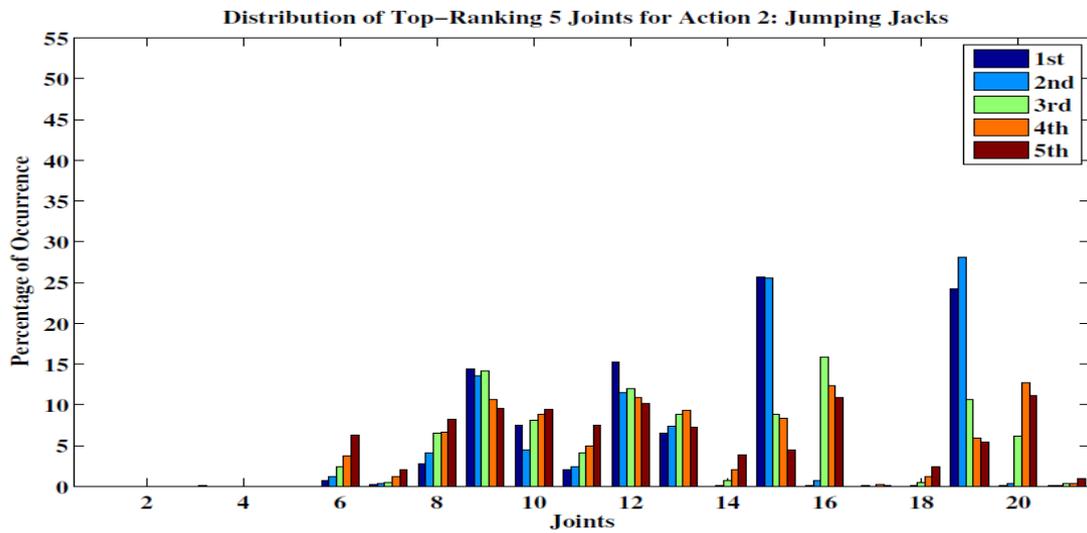
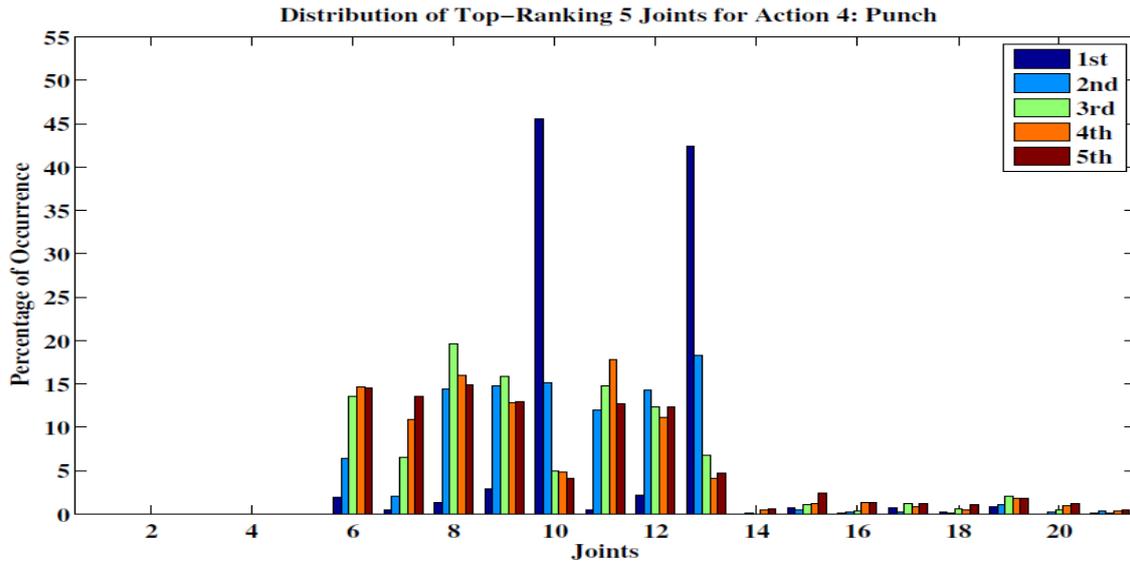


Figure 3.1: (a) the structure of the skeleton used in Dataset 1 and corresponding set of 21 joint angles computed from it. (b) Distribution of the top-ranking joints.



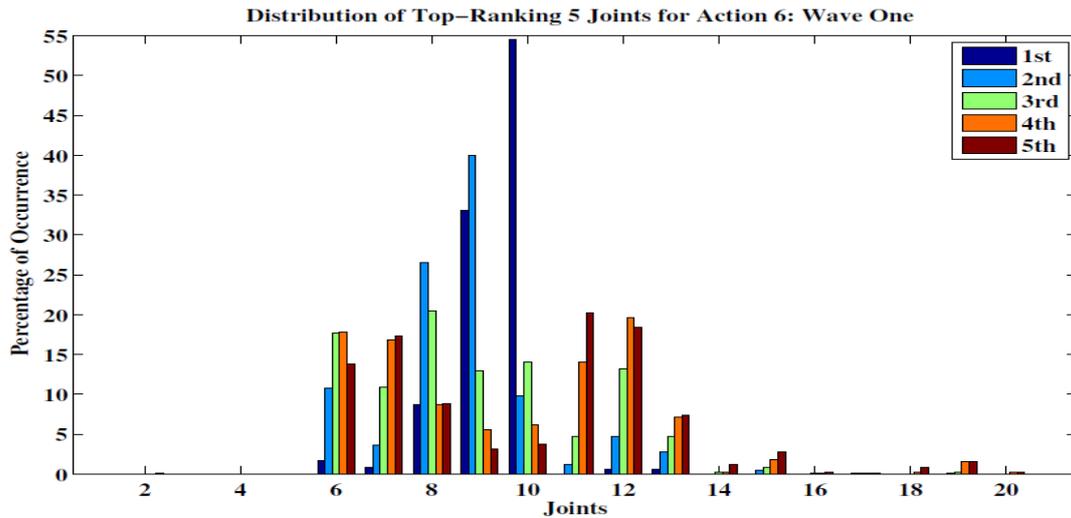


Fig. 3.1.1: Histogram distribution of the top-ranking 5 joints for four actions selected from Dataset.

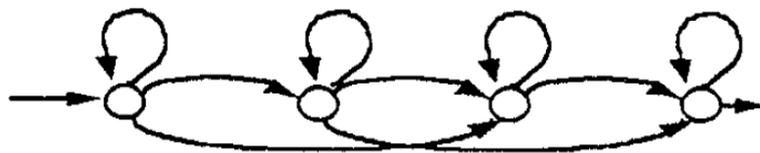


Figure 3.2: The 4-state HMM used for recognition



### 3.3.3 RWTH- Gesture Database

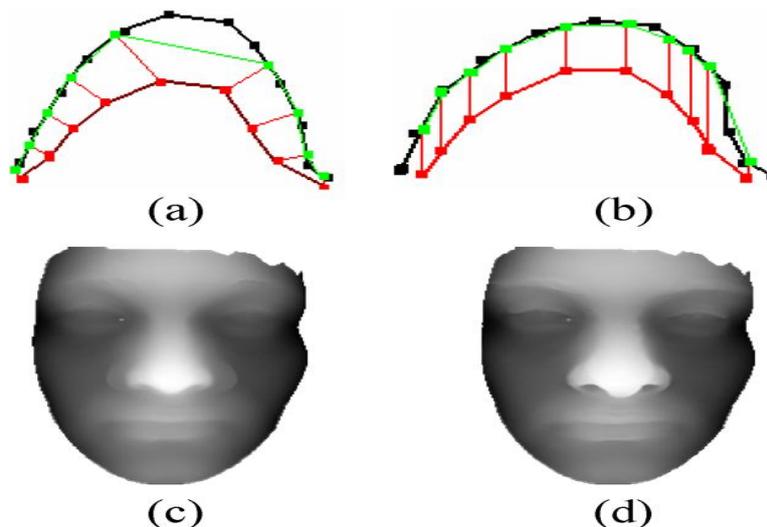


Fig 3.5.1 (a) Closest-point mapping, find the closest point on the target mesh for each base mesh vertex. (b) Ray-casting mapping, find the crossing point on target mesh for each ray-casting from the base mesh vertices. (c) Depth map constructed with closest-point mapping. (d) Depth map constructed with ray casting mapping.

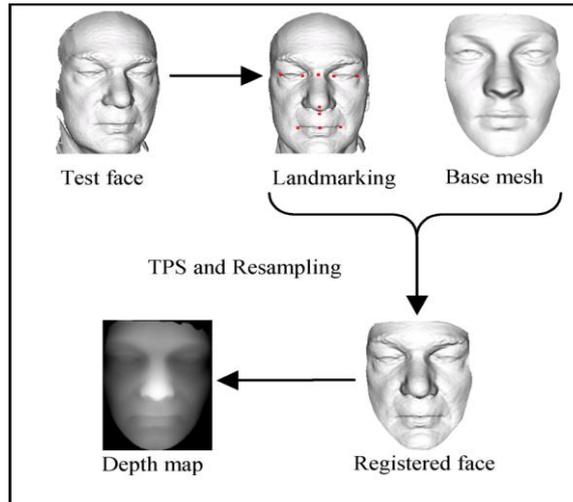


Fig 3.5.2 Face registration and depth map image generation.

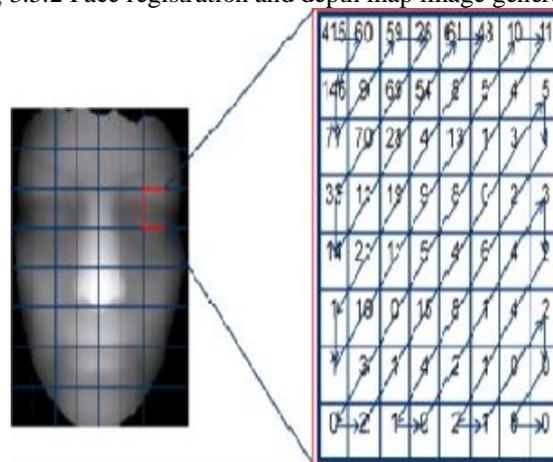


Fig3.5.3 Local blocks in depth image, DCT features are extracted using zig-zag scan

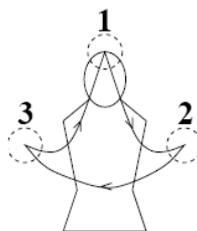


Figure 3.6 Arm movement of a triple time conducting gesture

## VI. CONCLUSION

During this dissertation we studied the algorithm of gesture recognition and we conclude of each algorithm is as under. Develop a real time gesture recognition method using the Eigen space from multi image sequences. Using the Eigen space, we can capture gesture in the approximate 3D space, and our method can recognize a gesture robustly and obtain the degree information of gesture correctly.

HMM is a very effective tool for hand gesture recognition. Although the test results prove this view, our method doesn't fully reflect the characteristics of the dynamic gesture such as the optical flow information, the hand posture, the inertia of moving object, momentum, eccentricity, velocity, angular velocity, curvature, and so on. Among these features, the hand posture, the optical flow, inertia, momentum, eccentricities are very sensitive to the segmentation of moving object and completely depend on the image pre-processing.

With the increasing popularity of real-time 3D acquisition systems with human skeleton extraction, we believe SMIJ is ideal for human activity recognition tasks in practical applications. Propose a depth image-based 3-D face recognition approach using local appearance-based models. Depth images were obtained by a base mesh-based registration and a resampling technique.

## REFERENCES

- [1] Blake, A., Isard, M.: *Active Contours*. Springer, Heidelberg (1998).
- [2] Chen, Y., Rui, Y., Huang, T.S.: Mode-based multi-hypothesis head tracking using parametric contours. In: *Proceedings of fifth International Conference on Automatic Face and Gesture Recognition*, Washington D.C (2002).
- [3] DeCarlo, D., Metaxas, D.: The integration of optical flow and deformable models with applications to human face shape and motion estimation. In: *Proceedings of CVPR '96*, pp. 231–238 (1996).
- [4] D.Weinland, R. Ronfard, and E. Boyer, "Free Viewpoint Action Recognition using Motion History Volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2, 2006.
- [5] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition. Proc. Third IEEE International Conference on*, pp 300–305, 1998
- [6] G. Rigoll, A. Kosmala, and S. Eickeler. High Performance Real-Time Gesture Recognition using Hidden Markov Models. In *International Gesture Workshop*, volume 1371, Springer-Verlag, Bielefeld, Germany, pages 69–80, Sep 1998.
- [7] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78:431–459, 1995.
- [8] Holte and T. Moeslund. "View invariant gesture recognition using 3D motion primitives," *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 4 2008, pp.797-800.
- [9] Hyeon-Kyu Lee, Jin H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.21, no.10, 1999. pp.961-973.
- [10] I. Cohen and H. Li, "Inference of Human Postures by Classification of 3D Human Body Shape," in *Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, Oct. 2003.
- [11] Isard, M., Blake, A.: Condensation—conditional density propagation for visual tracking. *Int. J. Comput. Vis.* 29, 5–28 (1998)
- [12] Jorge García Bueno, Miguel González-Fierro, Luis Moreno, Carlos Balaguer. Facial Gesture Recognition using Active Appearance Models based on Neural Evolution at Robotics Lab, 28911 Madrid, Spain. Page no 134, March 2012.
- [13] L.R.Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc.IEEE*, vol.77, no.2, Feb 1989. pp.257-286.
- [14] Marcialis GL, Roli F (2002) Fusion of LDA and PCA for face verification. In: Tistarelli M, Bigun J, Jain AK (eds) (2002) *Proceedings of the workshop on biometric authentication*, Copenhagen, Denmark, June 2002, Springer, Berlin Heidelberg New York, LNCS 2359, pp 30–37.