Human Action Recognition Using Star Skeleton

Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen and Suh-Yin Lee
Department of Computer Science
National Chiao-Tung University
Hsinchu, Taiwan 300
{xschen, huatsung, ewchen, sylee}@csie.nctu.edu.tw

ABSTRACT

This paper presents a HMM-based methodology for action recognition using star skeleton as a representative descriptor of human posture. Star skeleton is a fast skeletonization technique by connecting from centroid of target object to contour extremes. To use star skeleton as feature for action recognition, we clearly define the feature as a five-dimensional vector in star fashion because the head and four limbs are usually local extremes of human shape. In our proposed method, an action is composed of a series of star skeletons over time. Therefore, time-sequential images expressing human action are transformed into a feature vector sequence. Then the feature vector sequence must be transformed into symbol sequence so that HMM can model the action. We design a posture codebook, which contains representative star skeletons of each action type and define a star distance to measure the similarity between feature vectors. Each feature vector of the sequence is matched against the codebook and is assigned to the symbol that is most similar. Consequently, the time-sequential images are converted to a symbol posture sequence. We use HMMs to model each action types to be recognized. In the training phase, the model parameters of the HMM of each category are optimized so as to best describe the training symbol sequences. For human action recognition, the model which best matches the observed symbol sequence is selected as the recognized category. We implement a system to automatically recognize ten different types of actions, and the system has been tested on real human action videos in two cases. One case is the classification of 100 video clips, each containing a single action type. A 98% recognition rate is obtained. The other case is a more realistic situation in which human takes a series of actions combined. An action-series recognition is achieved by referring a period of posture history using a sliding window scheme. The experimental results show promising performance.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis–*motion*.

I.4.9 [**Image Processing and Computer Vision**]: Applications

I.4.7 [**Image Processing and Computer Vision**]: Feature measurement–*Feature representation*.

I.5.1 [**Pattern Recognition**]: Models – *Statistical.*

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – *motion, video analysis.*

General Terms

Algorithms, Performance, Experimentation, Human Factors.

Keywords

Action recognition, star skeleton, star distance, HMM.

1. INTRODUCTION

Vision-based human action recognition is currently one of the most active research areas in the domain of computer vision. It is motivated by a great deal of applications, such as automated surveillance system, smart home application, video indexing and browsing, virtual reality, human-computer interface and analysis of sports events. Unlike gesture and sign language, there is no rigid syntax and well-defined structure that can be used for action recognition. This makes human action recognition a more challenging task.

Several human action recognition methods were proposed in the past few years [1-20, 22-26]. A detailed survey can be found in [3, 10]. Most of the previous methods can be classified into three classes: model-based methods, eigenspace technique and Hidden Markov Model.

It is natural that human recognize action using the structure of human posture. Model-based methods [1, 5, 14] achieves human action recognition by using estimated or recovered human posture. In [1], a learning-based method for recovering 3D human body pose from single images and monocular image sequences is presented. Recovering human posture is an efficient method for action recognition since the human action is related to its posture. However, a large amount of computation cost is required for pose estimation.

An eigenspace technique [17] is also used in the action recognition field [18]. An action given by successive video frames is expressed as a curve (called a motion curve) in an eigenspace, and, by adopting a similarity measure, it can be used in judging if an unknown action is similar to any of the memorized motion curves. In [20], two kinds of superposed images are used to represent a human action: a motion history image (MHI) and a superposed motion image (SMI). Employing these images, a human action is described in an eigenspace as a set of points, and each SMI plays a role of reference point. An unknown action image is transformed into the MHI and then a match is found with images described in the eigenspace to realize action recognition. The eigenspace technique achieves high speed human action recognition. However, the recognition rate can still be improved.

Hidden Markov Model (HMM) which has been used successfully in speech recognition is a training based recognition techniques. HMM transforms the problem of action recognition into the problem of pattern recognition. Yamato et al. [24] are the first researchers who applied HMM for action recognition. They use HMM to recognize six different tennis strokes among three players. Reference [24] generalizes the linear eigenspace methods to a nonlinear approach using a tensor algebraic framework. Some of the recent works [4, 6, 8, 12, 15, 16, 19, 26] have shown that HMM performs well in human action recognition as well. HMM is a kind of stochastic state transit model, and it models an action by training. To recognize an action, the HMM which best matches the action is chosen. It achieves high recognition rate by using distinguishable feature, and requires low process time.

In this paper, we propose an action recognition method based on HMM using star skeleton as recognition feature. In our proposed method, time-sequential images expressing human action are transformed to an image feature vector sequence by extracting a feature vector from each image using star skeleton. Each feature vector of the sequence is then assigned a symbol which corresponds to a codeword in the codebook by Vector Quantization [11]. Consequently, the time-sequential images are converted to a symbol sequence. In the training phase, the model parameters of the HMM of each category are optimized so as to best describe the training symbol sequences from the categories of human action to be recognized. For human recognition, the model which best matches the observed symbol sequence is selected as the recognized category.

The paper is organized as follows. In section 2, we introduce the concept of HMM and how it is used for recognition. In section 3 we present the proposed recognition method with a detailed description of each step and some examples. Then experimental results and discussion are reported in section 4. Finally conclusion and future work are outlined in section 5.

2. Hidden Markov Model (HMM)

HMMs are a kind of stochastic state transit model, which treat discrete time sequences as the output of a Markov process whose states cannot be directly observed. Each state of the HMM stochastically outputs a symbol during state transition. We can observe the output symbol sequence but we can not observe the HMM states.

An HMM which has N states Q= $\{q_1, q_2, \ldots, q_N\}$ and M output symbols V= $\{v_1, v_2, \ldots, v_M\}$, are completely specified by the three parameter set $\lambda = \{A, B, \pi\}$. A simple example of a model where N=3 and M=3 is shown in Figure 1. Let the state at time t be s_t. Now the NxN state transition matrix A is

A= $\{a_{ij} \mid a_{ij} = P_r(s_{t+1} = q_j \mid s_t = q_i)\}$, where a_{ij} is the probability of transiting from state q_i to state q_j.

The NxM state output probability matrix B is

B= $\{b_j(k) \mid b_j(k) = P_r(v_k \mid s_t = q_j)\}$, where $b_j(k)$ is the probability of output symbol v_k at state q_j.

The initial state distribution vector π is

$$\pi = \{\pi_i \mid \pi_i = P_r(s_1 = q_i)\}$$

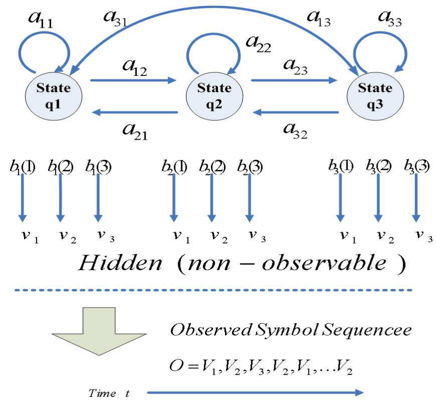


Figure 1. A full connected HMM with three states and three outputs

Recognition based on HMM involves two parts: training a model and computing $P_r(\lambda \mid O)$, the probability that the observation sequence O was generated by model λ. Each model is trained so that it is most likely to generate the symbol patterns for its training data. The training process is to optimize the model parameters (A, B, π), and this can be achieved by the standard Baum-Welch algorithm [21]. According to the Bayes rule, the problem of computing $P_r(\lambda \mid O)$ is to evaluate $P_r(O \mid \lambda)$, the probability that model λ generates the observation sequence O. The probability is calculated using the forward algorithm [21].

3. Proposed Method

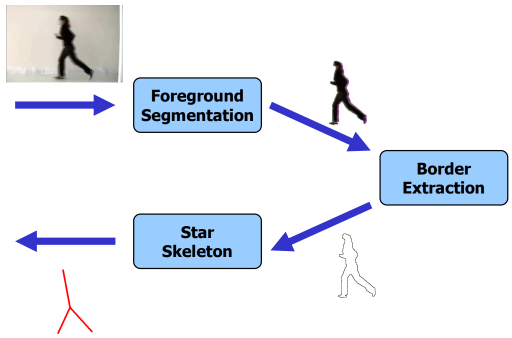
3.1 System Overview

The system architecture consists of three parts, including feature extraction, mapping features to symbols and action recognition as shown in Figure 2.

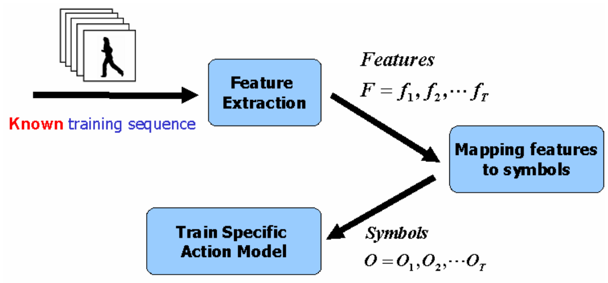
For feature extraction, we use background subtraction and threshold the difference between current frame and background image to segment the foreground object. After the foreground segmentation, we extract the posture contour from the human silhouette. As the last phase of feature extraction, a star skeleton technique is applied to describe the posture contour. The extracted star skeletons are denoted as feature vectors for latter action recognition. The process flow of feature extraction is shown in Figure 2 (a).

After the feature extraction, Vector Quantization (VQ) is used to map feature vectors to symbol sequence. We build a posture codebook which contains representative feature vectors of each action, and each feature vector in the codebook is assigned to a symbol codeword. An extracted feature vector is mapped to the symbol which is the codeword of the most similar (minimal distance) feature vector in the codebook. The output of mapping features to symbols module is thus a sequence of posture symbols.

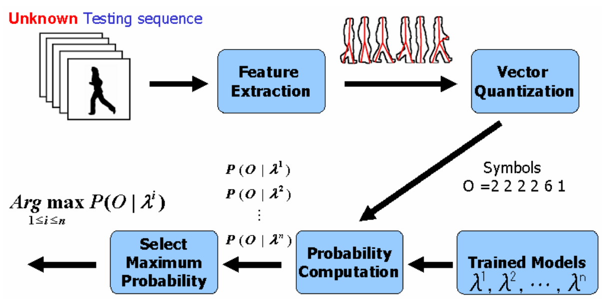
The action recognition module involves two phase: training and recognition. We use Hidden Markov Models to model different actions by training which optimizes model parameters for training data. Recognition is achieved by probability computation and selection of maximum probability. The process flows of both training and recognition are shown in Figure 2 (b) and (c).



(a) Process flow of feature extraction



(b) Process flow of training



(c) Process flow of recognition

Figure 2. Illustration of the system architecture

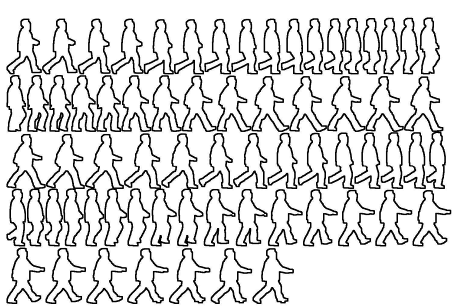


Figure 3. A walk action is a series of postures over time

3.2 Feature Extraction

Human action is composed of a series of postures over time as shown in Figure 3. A good way to represent a posture is to use its boundary shape. However, using the whole human contour to describe a human posture is inefficient since each border point is very similar to its neighbor points. Though techniques like Principle Component Analysis are used to reduce the redundancy, it is computational expensive due to matrix operations. On the other hand, simple information like human width and height may be rough to represent a posture. Consequently, representative features must be extracted to describe a posture. Human skeleton seems to be a good choice. There are many standard techniques for skeletonization such as thinning and distance transformation. However, these techniques are computationally expensive and moreover, are highly susceptible to noise in the target boundary. Therefore, a simple, real-time, robust techniques, called star skeleton [9] was used as features of our action recognition scheme.

3.2.1 Star Skeletonization

The concept of star skeleton is to connect from centroid to gross extremities of a human contour. To find the gross extremities of human contour, the distances from the centroid to each border point are processed in a clockwise or counter-clockwise order. Extremities can be located in representative local maximum of the distance function. Since noise increases the difficulty of locating gross extremes, the distance signal must be smoothed by using smoothing filter or low pass filter in the frequency domain. Local maximum are detected by finding zero-crossings of the smoothed difference function. The star skeleton is constructed by connecting these points to the target centroid. The star skeleton process flow of an example human contour is shown in Figure 4 and points A, B, C, D and E are local maximum of the distance function. The details of star skeleton are as follows:

Star skeleton Algorithm (as described in [9])

Input: Human contour

Output: A skeleton in star fashion

1. Determine the centroid of the target image border (x_c, y_c)

$$x_c = \frac{1}{N_b} \sum_{i=1}^{N_b} x_i$$

$$y_c = \frac{1}{N_b} \sum_{i=1}^{N_b} y_i$$

where N_b is the number of border pixels, and (x_c, y_c) is a pixel on the border of the target.

2. Calculate the distances d_i from the centroid (x_c, y_c) to each border point (x_i, y_i) $d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$

These are expressed as a one dimensional discrete function $d(i) = d_i$.

3. Smooth the distance signal $d(i)$ to $\hat{d}(i)$ for noise reduction by using linear smoothing filter or low pass filter in the frequency domain.

4. Take local maximum of $\hat{d}(i)$ as extremal points, and construct the star skeleton by connecting them to the centroid (x_c, y_c). Local maximum are detected by finding zero-crossings of the difference function

$$\delta(i) = \hat{d}(i) - \hat{d}(i-1)$$

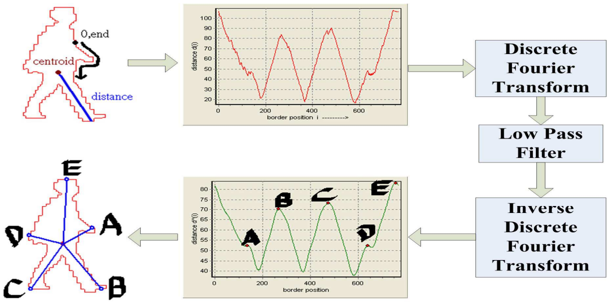


Figure 4. Process flow of star skeletonization

3.2.2 *Feature Definition*

One technique often used to analyze the action or gait of human is the motion of skeletal components. Therefore, we may want to find which part of body (e.g. head, hands, legs, etc) the five local maximum represent. In [9], angles between two legs are used to distinguish *walk* from *run*. However, some assumptions such as feet locate on lower extremes of star skeleton are made. These assumptions can not fit other different actions, for example, low extremes of crawl may be hands. Moreover, the number of extremal points of star skeleton varies with human shape and the low pass filter used. Gross extremes are not necessarily certain part of human body. Because of the difficulty in finding which part of body the five local maximum represent, we just use the distribution of star skeleton as features for action recognition.

As a feature, the dimension of the star skeleton must be fixed. The feature vector is then defined as a five dimensional vectors from centroid to shape extremes because head, two hands, two legs are usually local maximum. The final cut-off frequency of star skeleton is determined automatically. The cut-off frequency is first set to a higher frequency, and gradually decreases until the dimension of star skeleton is within five. For postures with more than five contour extremes, we adjust the low pass filter to lower the dimension of star skeleton to five. On the other hand, zero vectors are added for postures with less than five extremes.

Since the used feature is vector, its absolute value varies for people with different size and shape, normalization must be made to get relative distribution of the feature vector. This can be achieved by dividing vectors on x-coordinate by human width, vectors on y-coordinate by human height.

3.3 Mapping feature to symbol

To apply HMM to time-sequential video, the extracted feature sequence must be transformed into symbol sequence for latter action recognition. This is accomplished by a well-known technique, called Vector Quantization [11].

3.3.1 *Vector Quantization*

For vector quantization, codewords $g_j \in R^n$, which represent the centers of the clusters in the feature R^n space, are needed. Codeword g_j is assigned to symbol v_j. Consequently, the size of the code book equals the number of HMM output symbols. Each feature vector f_i is transformed into the symbol which is assigned to the codeword nearest to the vector in the feature space. This means f_i is transformed into symbol v_j if $j = \arg\min_j d(f_i, g_j)$ where $d(x, y)$ is the distance between vectors x and y.

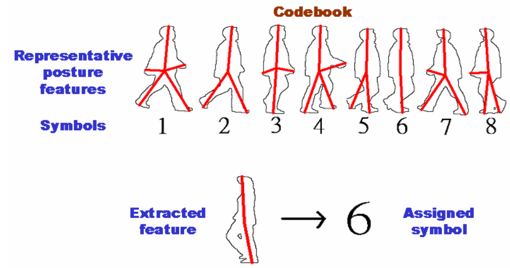


Figure 5. The concept of vector quantization in action recognition

For action recognition, we select m feature vectors of representative postures from each action as codewords in the codebook. And an extracted feature would be mapped to a symbol, which is the codeword of the most similar (minimal distance) feature vector in the codebook. The concept of the mapping process is shown in Figure 5. The codebook in the figure contains only some representative star skeletons of walk to explain the mapping concept. In the mapping process, similarity between feature vectors needs to be determined. Therefore we define distance between feature vectors, called star distance, to decide the similarity between feature vectors.

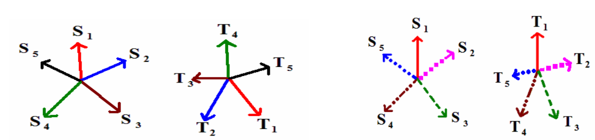
3.3.2 *Star Distance*

Since the star skeleton is made up of five sub-vectors, the star distance between two feature vectors S and T is first defined as the sum of the Euclidean distances of the five sub-vectors.

$$\text{Distance} = \sum_{i=1}^{5} (S_i - T_i)$$

However, consider the star skeletons S and T in Figure 6 (a). The two star skeletons are similar, but the distance between them is large due to mismatch. So we modify the distance measurement. Each sub-vector must find their closest mapping as shown in Figure 6 (b). The star distance is then defined as the sum of the Euclidean distance of the five sub-vectors under such greedy mapping. For simplicity, the star distance is obtained by minimal sum of the five sub-vectors in all permutation. Better algorithm to accelerate the star distance calculation can be found.

$$\text{Star Distance} = \underset{k=1}{\overset{5!}{Arg\min}} \sum_{i=1}^{5} (S_i - T_i)$$



(a) (b)

Figure 6. Illustration of star distance (a) Mismatch (b) Greedy Match

3.4 Action Recognition

The idea behind using the HMMs is to construct a model for each of the actions that we want to recognize. HMMs give a state based representation for each action. The number of states was empirieally determined. After training each action model, we calculate the probability $P(O \mid \lambda^i)$, the probability of model λ^i generating the observation posture sequence O, for each action model. We can then recognize the action as being the one, which is represented by the most probable model.

3.5 Recognition over a series of actions

What mentioned above are classification of single action. The following is a more complex situation. A man performs a series of actions, and we recognize what action he is performing now. One may want to recognize the action by classification of the posture at current time T. However, there is a problem. By observation we can classify postures into two classes, including key postures and transitional postures. Key postures uniquely belong to one action so that people can recognize the action from a single key posture. Transitional postures are interim between two actions, and even human cannot recognize the action from a single transitional posture. Therefore, human action can not be recognized from posture of a single frame due to transitional postures. So, we refer a period of posture history to find the action human is performing. A sliding-window scheme is applied for real-time action recognition as shown in Figure 7. At time current T, symbol subsequence between T-W and T, which is a period of posture history, is used to recognize the current action by computing the maximal likelihood, where W is the window size. In our implementation, W is set to thirty frames which is the average gait cycle of testing sequences. Here we recognize stand as walk. The unknown is due to not enough history. By the sliding window scheme, what action a man is performing can be realized.

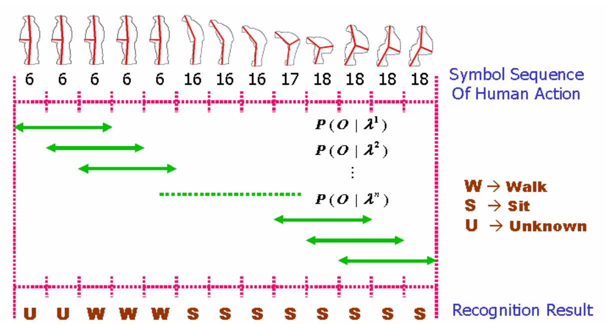


Figure 7. Sliding-window scheme for real-time action recognition

4. Experiments

To evaluate the performance of our approach, we implement a system to automatically recognize ten different types of actions. The system has been tested on real human action videos in two cases: (1) Single action recognition (2) Recognition over a series of actions. In (1), a confusion matrix was used to present the classification result. In (2), we would like to real-time recognize the action in current frame over a series of actions. Recognition of current frame was achieved by referring a period of posture history using a sliding window scheme. We compare the recognized action type and ground truth obtained by human in each frame to demonstrate the experimental results..

4.1 Single action recognition

The proposed system has been tested on real human action videos. For simplicity, we assumed a uniform background in order to extract better human contour. The categories to be recognized are ten types of human actions: ‘walk’, ‘sidewalk‘, ‘pickup’, ‘sit’, ‘jump 1’, ‘jump 2’, ‘push up’, ‘sit up’, ‘crawl 1’, and ‘crawl 2’. 5 persons performed each type of the 10 actions 3 times.

(a) walk

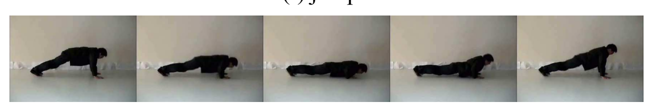
(b) sidewalk

(c) sit

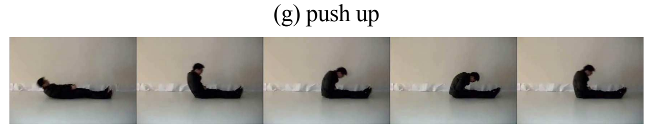
(d) pick up

(e) jump 1

(f) jump2

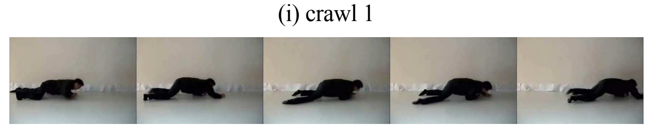


(g) push up



(h) sit up

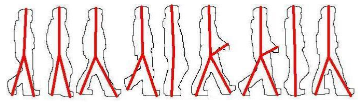
(i) crawl 1



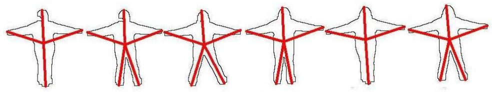
(j) crawl 2

Figure 8. Example clips of the ten action types

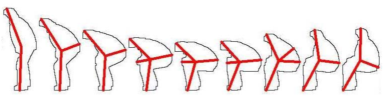
The video content was captured by a TV camera (NTSC, 30 frames / second) and digitized into 352x240 pixel resolution. The duration of each video clip was from 40 to 110 frames. This number of frames is chosen experimentally. Too short sequences do not provide enough information to characterize the action. On the other side, too long sequences make the learning phase very hard. Figure 8 showed some example video clips of the 10 types of human actions. In order to calculate the recognition rate, we used the leave out method. All data were separated into 3 categories, and each category contains 5 persons doing ten different actions one time. One category was used as training data for building a HMM for each action type, and the others were testing data.



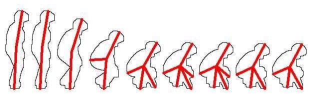
(a) walk



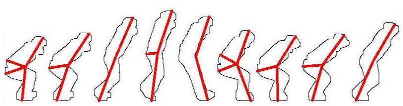
(b) sidewalk



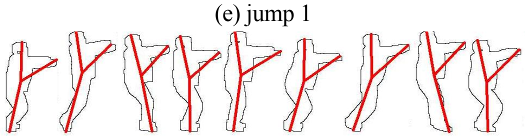
(c) sit



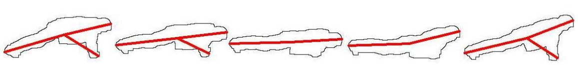
(d) pick up



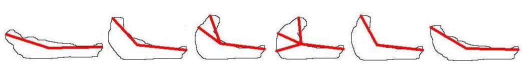
(e) jump 1



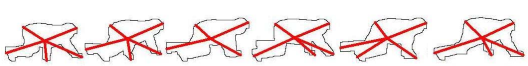
(f) jump 2



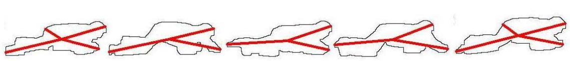
(g) push up



(h) sit up



(i) crawl 1



(j) crawl 2

Figure 9. Features of each action type using star skeleton

The features of each action type are extracted using star skeleton. Feature examples of each action are shown in Figure 9. For vector quantization, we manually selected m representative skeleton features for each action as codewords in the codebook for the experiment. m is set to five for simple actions like sidewalk and jump2. For other eight actions m is set to ten. Thus, the total number of HMM symbols was ninety. The codebook is built in one direction first, and then reversed representative feature vectors are added to recognize the actions in counter direction.

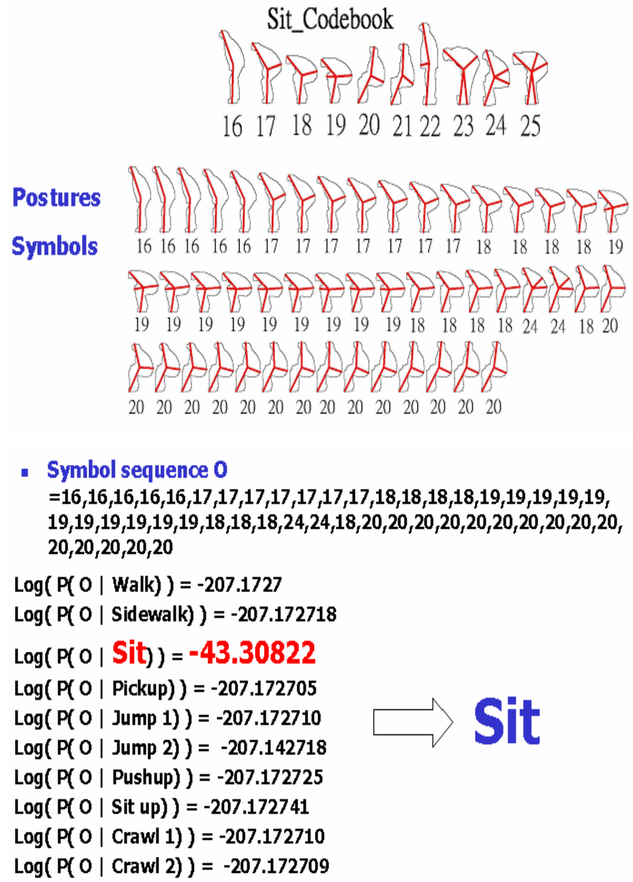


Figure 10. Complete recognition process of sit action

We take a sit action video as an example to explain the recognition process. The sit action is composed of a series of postures. Star skeleton are used for posture description, and map the sit action into feature sequence. The feature sequence is then transformed into symbol sequence O by Vector Quantization. Each trained action model computes the probability generating symbol sequence O, and the log scale of probabilities are shown in Figure 10. The sit model has the maximum probability, so the video is recognized as sit.

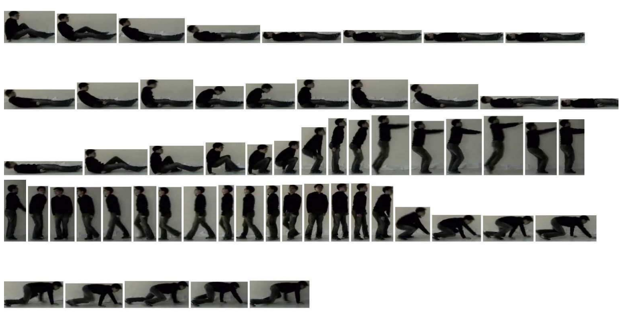
Finally, Table 1 demonstrates the confusion matrix of recognition of testing data. The left side is the ground truth of action types, and the upper side is the recognition action types. The number on the diagonal is the number of each action which are correctly classified. The numbers which are not on the diagonal are classification errors, and we can know which kind of action the system misclassify. From this table, we can find that most of the testing data were accurately classified. A great recognition rate of 98% was achieved by the proposed method. Only two confusions occurred between sit and pick up. We checked the two mistaken clips, and found a large portion of bending the body. The bending does not uniquely belong to sit or pickup so that the two action models get confused. In my opinion, a transitional action, bending, must be added to better distinguish *pickup* from *sit*.

Table 1. Confusion matrix for recognition of testing data

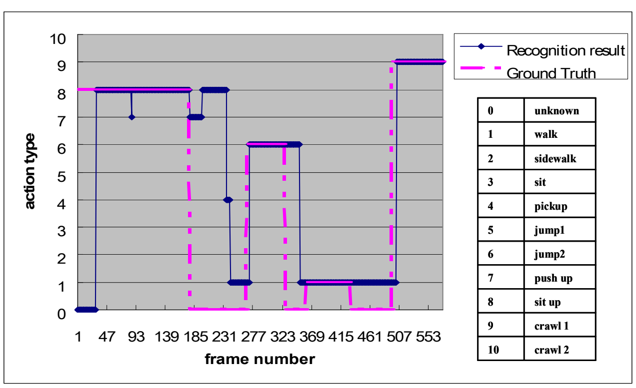
	Walk	Sidewalk	Sit	Pick up	Jump 1	Jump 2	Push up	Sit up	Crawl 1	Crawl 2
Walk	10	0	0	0	0	0	0	0	0	0
Sidewalk	0	10	0	0	0	0	0	0	0	0
Sit	0	0	9	1	0	0	0	0	0	0
Pick up	0	0	1	9	0	0	0	0	0	0
Jump 1	0	0	0	0	10	0	0	0	0	0
Jump 2	0	0	0	0	0	10	0	0	0	0
Push up	0	0	0	0	0	0	10	0	0	0
Sit up	0	0	0	0	0	0	0	10	0	0
Crawl 1	0	0	0	0	0	0	0	0	10	0
Crawl 2	0	0	0	0	0	0	0	0	0	10

4.2 Recognition over a series of actions

For more realistic application, we attempted to recognize successive action of "Sit up – get up – Jump 2 – turn about – Walk – turn about – Crawl 1", including four defined actions and two undefined actions. Figure11 (a) shows the original image sequence (it shows some selected frames), and (b) shows the recognition result. The x-coordinate of the graph is the frame number, and the y-coordinate indicates the recognized action. The purple dashed line is the ground truth defined by human observation, and the blue line is the recognized action types.



(a) Some original images of a series of action



(b) Recognition result

Figure 11. Recognition over a series of actions "Sit up-Jump 2-Walk-Crawl 1"

The unknown period is the time during which human performs actions that are not defined in the ten categories. The first period of unknown of ground truth is "get up", and the second and third period are "turn about". The unknown period of recognition result is due to the history of postures is not enough (smaller than the window size). We can find that the period human performs the defined actions can be correctly recognized. Some misunderstanding can be corrected by smoothing the recognition signal. A small recognition time delay occurs at the start of crawl due to not enough history for the sliding window scheme. However, the delay is very small that human can hardly feel. The time period human perform undefined action, the system chooses the most possible action from ten defined actions. Therefore, more actions need to be added to enhance the system.

5. Conclusions and Future Work

We have presented an efficient mechanism for human action recognition based on the shape information of the postures which are represented by star skeleton. We clearly define the extracted skeleton as a five-dimensional vector so that it can be used as recognition feature. A feature distance (star distance) is defined so that feature vectors can be mapped into symbols by Vector Quantization. Action recognition is achieved by HMM. The system is able to recognize ten different actions. For single action recognition, 98% recognition rate was achieved. The recognition accuracy could still be improved with intensive training. For recognition over a series of actions, the time human perform the defined ten actions can be correctly recognized.

Although we have achieved human action recognition with high recognition rate, we also confirm some restrictions of the proposed technique from the experimental results. One limitation is that the recognition is greatly affected by the extracted human silhouette. We used a uniform background to make the foreground segmentation easy in our experiments. To build a robust system, a strong mechanism of extracting correct foreground object contour must be developed. Second, the representative postures in the codebook during Vector Quantization are picked manually, clustering algorithms can be used so that they can be extracted automatically for a more convenient system. Third, the viewing direction is somewhat fixed. In real world, the view direction varied for different locations of the cameras. The proposed method should be improved because the human shape and extracted skeleton would change from different views.

6. REFERENCES

[1] A. Agarwal. and B. Triggs. "Recovering 3D Human Pose from Monocular Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 44-58, 2006.

[2] A. Ali and J. K. Aggarwal. "Segmentation and recognition of continuous human activity," Proceedings of International Workshop on Detection and Recognition of Events in Video, pp. 28-35, 2001

[3] J.K. Aggarwal and Q. Cai. "Human motion analysis: A review," Computer Vision Image Understanding, Vol.73, No.3, pp.428–440, March 1999.

[4] L. H. W. Aloysius, G. Dong, Z. Huang and T. Tan. "Human Posture Recognition in Video Sequence using Pseudo 2-D Hidden Markov Models," Proceedings of International Conference on Control, Automation, Robotics and Vision Conference, Vol. 1, pp. 712-716, 2004.

[5] C. W. Chu and I. Cohen. "Posture and Gesture Recogtion using 3D Body Shapes Decomposition," Proceedings of International Conference on Computer Vision and Pattern Recognition, pp. 69-77, 2005

[6] R. Cucchiara, A. Prati and R Vezzani. "Posture classification in a multi-camera indoor environment," Proceedings of International Conference on Image Processing, pp.725-728, 2005

[7] R. Cucchiara, C. Grana, A. Prati and R. Vezzani. "Probabilistic posture classification for Human-behavior analysis," IEEE transactions on Systems, Man and Cybernetics, Vol. 35, Issue 1, pp. 42-54, 2005

[8] X. Feng and P. Perona. "Human Action Recognition by Sequence of Movelet Codewords," Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission, pp. 717-721, 2002.

[9] H. Fujiyoshi and A. J. Lipton. "Real-Time Human Motion Analysis by Image Skeletonization." Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision, pp. 15-21, 1998.

[10] D.M. Gavrila. "The visual analysis of human movement: A survey," Computer Vision Image Understanding, Vol.73, No.1, pp.82–98, Jan. 1999.

[11] X. D. Huang, Y. Ariki, and M. A. Jack. "Hidden Markov Models for Speech Recognition". Edingurgh Univ. Press, 1990.

[12] V. Kellokumpu, M. Pietikäinen and J. Heikkilä. "Human Activity Recognition Using Sequences of Postures," Proceedings of IAPR Conference on Machine Vision Application, pp. 570-573, 2005

[13] M. W. Lee and I Cohen. "Human body tracking with auxiliary measurements," Proceedings of International Workshop on Analysis and Modeling of Faces and Gestures, pp. 112-119, 2003

[14] M.W. Lee, I Cohen. "Proposal maps driven MCMC for estimating human body pose in static images," Proceedings of International Conference on Computer Vision and Pattern Recognition, pp. 334-341, 2004

[15] M. Leo, T. D'Orazio, I. Gnoni, P. Spagnolo and A. Distante. "Complex Human Activity Recognition for Monitoring Wide Outdoor Environments," Proceedings of the 17th International Conference on Pattern Recognition, Vol.4, pp. 913-916, 2004.

[16] T. Mori, Y. Segawa, M. Shimosaka and T. Sato. "Hierarchical Recognition of Daily Human Actions Based on Continuous Hidden Markov Models," Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 779-784, 2004.

[17] H. Murase and S.K. Nayar. "Visual learning and recognition of 3-D objects from appearance," International Journal on Computer Vision, Vol.14, No.1, pp.5–24, Jan. 1995.

[18] H. Murase and R. Sakai. "Moving object recognition in eigenspace representation: Gait analysis and lip reading," Pattern Recognition Letter, pp.155–162, Feb. 1996.

[19] F, Niu and M. Abdel-Mottaleb. "View-Invariant Human Activity Recognition Based on Shape and Motion Features," Proceedings of IEEE Sixth International Symposium on Multimedia Software Engineering, pp. 546-556, 2004.

[20] T. Ogata, J. K. Tan and S. Ishikawa. "High-Speed Human Motion Recognition Based on a Motion History Image and an Eigenspace," IEICE Transactions on Information and Systems, pp. 281-289, 2006.

[21] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, pp. 257-286, 1989.

[22] Y. Sheikh, M. Sheikh and M. Shah. "Exploring the space of a human action," Proceedings of International Conference on Computer Vision, pp. 144-149, 2005

[23] M. A. O. Vasilescu. "Human motion signatures: analysis, synthesis, recognition" Proceedings of International Conference on Pattern Recognition, pp. 456-460, 2002

[24] J. Yamato, J. Ohya and K. Ishii. "Recognizing Human Action in Time-Sequential Images using Hidden Markov Model," Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 379-385, 1992.

[25] A. Yilma. and M. Shah. "Recognizing human actions in videos acquired by uncalibrated moving cameras," Proceedings of International Conference on Computer Vision, pp. 150-157, 2005

[26] R. Zhang, C. Vogler and D. Metaxas. "Human Gait Recognition," Proceedings of International Workshop on Computer Vision and Pattern Recognition, 2004.