
ELSEVIER

Pattern Recognition Letters 23 (2002) 1203–1213

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Compressed domain action classification using HMM

R. Venkatesh Babu, B. Anantharaman, K.R. Ramakrishnan *, S.H. Srinivasan

Department of Electrical Engineering, Multimedia Systems Laboratory, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, Karnataka 560 012, India

Received 8 August 2001; received in revised form 30 October 2001

Abstract

This paper proposes three techniques of feature extraction for person independent action classification in compressed MPEG video. The features used are extracted from motion vectors, obtained by partial decoding of the MPEG video. The feature vectors are fed to Hidden Markov Model (HMM) for classification of actions. Totally seven actions were trained with distinct HMM for classification. Recognition results of more than 90% have been achieved. This work is significant in the context of emerging MPEG-7 standard for video indexing and retrieval. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Action classification; Compressed domain; Content-based retrieval; Feature extraction; Indexing; MPEG-7

1. Introduction

Classification of human action from video sequences has evoked considerable interest in recent years among researchers in computer vision and multimedia (Yamato et al., 1992; Bregler, 1997; Rosales, 1998; Kuniyoshi and Inoue, 1993; Allmen and Dyer, 1993). A prominent reason for this is the utility of the outcome of such a classification in practical applications like automatic monitoring systems, video indexing, retrieval (Boykin and Merlino, 2000; Yoon et al., 2000; Davis and Shah, 1995; Eickeler et al., 2000; Lee and Kim, 1999) and segmentation (Shah and Jain, 1997; Starner and Pentland, 1995). Existing literature (Yamato et al., 1992; Bregler, 1997; Rosales, 1998) deals only with action classification in the pixel domain. In (Yamato et al., 1992), time sequential images expressing human actions are transformed to an image feature vector sequence by extracting mesh (Umeda, 1982) feature vector from each image. The mesh features are extracted from a binarized image obtained after subtracting the background image from original image by applying a suitable threshold. The drawbacks of this method are that it is sensitive to position displacement, noise, and also exhibits poor performance if the training and test subjects are different. In (Bregler, 1997), this classification problem has been approached from a statistical viewpoint. For each pixel in the image,

* Corresponding author. Tel.: +91-80-309-2441; fax: +91-80-360-0444.

E-mail addresses: rvbabu@ee.iisc.ernet.in (R. Venkatesh Babu), ananth@ee.iisc.ernet.in (B. Anantharaman), krr@ee.iisc.ernet.in (K.R. Ramakrishnan), SH_Srinivasan@satyam.com (S.H. Srinivasan).

PII: S0167-8655(02)00067-3

the spatio-temporal image gradient and the color values are represented as random variables. Then the blob hypothesis is used wherein each blob is represented with a probability distribution over coherent motion, color and spatial support regions. In (Rosales, 1998), the spatial location and the temporal properties of human actions are obtained from motion energy and motion history images (Davis and Bobick, 1997) from raw video sequences. From these motion energy and motion history images, a set of Hu-moment (Hu, 1962) features that are invariant to translation, rotation and scaling are generated. Using principal component analysis, the dimension of the Hu-moment space is reduced in a statistically optimal way. The drawback in the above methods of classification is that it is computationally expensive and hence may not be suitable for real time applications.

Most of the multimedia documents available nowadays are in the MPEG (Mitchell et al., 1996) compressed form to facilitate easy storage and transmission. Hence it would be efficient if the classification is performed in the MPEG compressed domain without having to completely decode the bit-stream and subsequently perform classification in the raw domain. This calls for techniques which can use information available in the compressed domain such as motion vectors and dct coefficients. The advantages of such an approach include the following:

1. Size of uncompressed video is large whereas MPEG which combines several compression techniques reduces the memory occupied by the same video. Processing time and memory requirements reduces to a greater extent if done in the compressed domain.
2. The computational time required for decoding all the pixels is high. With an approach that requires only *partial* decoding of readily available motion vectors reduces the computational time involved in the classification process.
3. With the availability of hardware encoders and decoders for MPEG, such an approach involving processing in the compressed-domain is amenable to hardware implementation, which can further reduce the computational time.

With the above in view, we present, in this paper, a successful – and, to our knowledge, the *first* – attempt towards classifying human actions with MPEG compressed video data. In this context, we may observe that:

1. Human action can be more effectively characterized by motion than by any other cue (such as color, depth and spatial features).
2. In MPEG, motion is quantified by the motion vectors available in the inter-coded frames (P and B frames).

In this paper, we propose three techniques for extracting features from the readily available motion vectors of MPEG video for action classification:

1. *Projected* 1-*D feature* corresponding to the horizontal and vertical components of the motion vectors;
2. 2-*D polar feature* corresponding to a polar tiling of the (horizontal and vertical components of the) motion vectors;
3. 2-*D Cartesian feature* corresponding to a Cartesian tiling of the (horizontal and vertical components of the) motion vectors.

The actions considered in this work are: walk, run, jump, bend up, bend down, twist right and twist left. A discrete Hidden Markov Model (HMM) for each action is trained with the corresponding MPEG video sequences. Classification is finally achieved by feeding a given (test) sequence to all the trained HMMs and employing a likelihood-based measure to declare the action performed in the video sequence.

The paper is organized as follows: Section 2 deals with the procedures for (i) normalizing the motion vectors obtained from inter-coded (B and P) frames; and (ii) extracting the above mentioned features from these normalized motion vectors. A brief description of the HMM is presented in Section 3. Section 4 brings out the details of the training and testing phases of the proposed approach. Section 5 gives illustrative experimental results bringing out the efficacy of the aforementioned features and Section 6 concludes the paper.

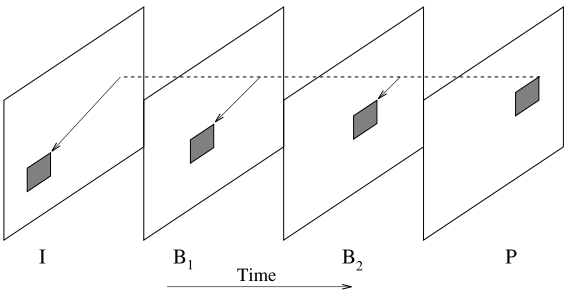

Fig. 1. The relative motion vectors (forward) between an I frame and P frame under linearity assumption.

2. Feature extraction

In this section, we describe the three proposed features derived from motion vectors. Prior to feature extraction the motion vectors obtained from different types of frames are normalized according to the structure of groups of pictures (GOP) as explained. The GOP structure of all MPEG video sequences considered was in $\mathrm{I\,B_1\,B_2\,P}\ldots$ format with 12 frames per GOP. The motion vectors obtained from the B frames are normalized with respect to the following P frame. Though B frames have both forward and backward motion vectors we consider only either of them based on its location. This is done in order to increase the reliability of the motion vectors. In our GOP sequence, forward motion vectors are considered for B_1 and backward motion vectors for B_2. These motion vectors are subsequently scaled with respect to the motion vectors obtained from P frame. The following normalization is done for all B frames by assuming linearity [1] of motion vectors between any I and P frame in the GOP (Fig. 1). This is done to make the value of the motion vectors independent of the frame type. For the above mentioned GOP structure, the motion vector for a P frame is estimated from the immediate I or P frame, the time interval between them being three units (the temporal distance between any two successive frames is considered as one unit). For B_1 frame the forward motion vectors are obtained from the immediately previous I or P frame whose time interval is one unit.

Hence, from the above discussion, for B_1 frame the scaled motion vector along x direction $mx = mx_{B_1} \times 3$, where mx_{B_1} is the motion vector obtained from B_1 frame and for B_2 frame the scaled motion vector $mx = mx_{B_2} \times (-3)$, where mx_{B_2} is the motion vector obtained from B_2 frame. Similar method is followed for computing the motion vectors along y direction.

Further, as the motion vectors obtained are noisy, preprocessing is done before the feature extraction methods are applied. Initially a binary image is obtained by assigning zero to the macroblocks having no motion and one to the remaining macroblocks. Then the binary image is subjected to a binary morphological operation ('clean') to remove isolated motion vectors (1's surrounded by 0's). The impulse kind of noise present in the motion vectors are removed in this process. Since the moving human object occupies several connected macroblocks in a frame, most of the isolated motion vectors do not belong to the moving object. Hence, to increase the reliability, these isolated motion vectors are removed before extracting the feature vectors. Fig. 2 shows the effectiveness of the noise removal.

2.1. Projected 1-D feature

Let a motion vector corresponding to macroblock (k,l) be represented by (m_x^{kl}, m_y^{kl}) and let $f_x(m_x; r_i, r_j)$ be the number of horizontal components of motion vectors in the ranges r_i and r_j with $(r_i < r_j)$. That is,

$$f_x(m_x; r_i, r_j) = \#\left\{(k,l) : r_i \leqslant m_x^{kl} \leqslant r_j\right\}, \tag{1}$$

where # represents the cardinality of its argument (which is a set). Similarly, $f_y(m_y; r_i, r_j)$ is given as

$$f_y(m_y; r_i, r_j) = \#\left\{(k,l) : r_i \leqslant m_y^{kl} \leqslant r_j\right\}. \tag{2}$$

Using different non-overlapping intervals (r_i, r_j), a histogram is obtained for each direction (horizontal and vertical) for each inter-coded

[1] The linearity assumption is valid only for the two B frames lying between any successive I/P or P/P frames. The time interval between any two such frames is around 3/25 of a second for a frame rate of 25 fps. The dynamics of human action sequences do not change considerably within this small time duration.

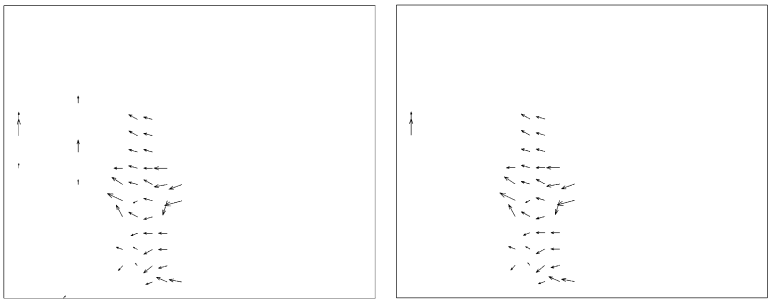

Fig. 2. (a) The motion vectors before noise removal and (b) after morphological filtering.

frame. The combination (f_x, f_y) forms the feature vector.

2.2. 2-D polar feature

Unlike the above, here the horizontal and vertical components are not treated separately. The motion vector direction and magnitude for each macroblock is obtained from both horizontal and vertical components of the corresponding motion vector.

The number of motion vectors falling between the angle range θ_i and θ_j and having magnitude within the range r_i and r_j can be expressed as

$$f_\theta(m_\theta; r_i, r_j, \theta_i, \theta_j) = \#\{(k,l) : r_i \leqslant |m_\theta| \leqslant r_j \text{ and } \theta_i \leqslant \angle m_\theta \leqslant \theta_j\}, \quad (3)$$

where m_θ is the motion vector (m_x, m_y) in polar coordinates with $|m_\theta| = \sqrt{m_x^2 + m_y^2}$ and $\angle m_\theta = \tan^{-1}(m_y/m_x)$.

Here (r_i, r_j) and (θ_i, θ_j) are chosen in such a way as to cover the entire range of motion vectors and the angle ranging from $-\pi$ to π in a non-overlapping manner. Fig. 3 illustrates the method by which feature vectors are extracted by considering the direction (angle) and magnitude of motion vectors.

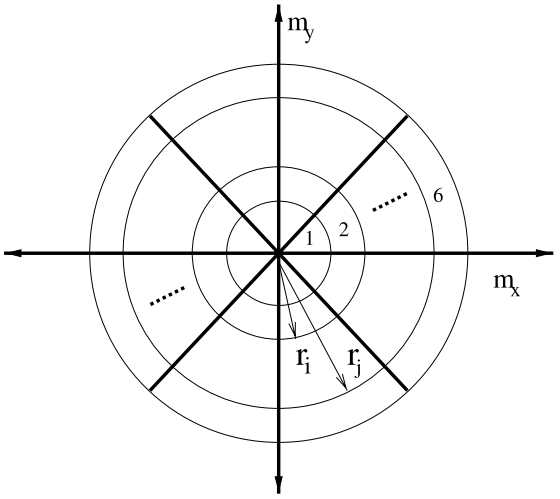

Fig. 3. The number of motion vectors falling within each sector is counted to get the 2-D polar feature.

2.3. 2-D Cartesian feature

In this method, the feature vectors are obtained from 2-D histogram of the x and y components of motion vectors by dividing the range of x and y motion vectors into rectangular bins. The dimensions of the rectangular bin are not equal throughout. These dimensions for a given range are selected depending upon the density of motion vectors in that range. It is generally found that the density of motion vectors in the central ranges is high, where the dimensions are less.

The number of x component motion vectors falling between the range p_i and p_j and having y component within the range q_i and q_j can be written as

$$f(m_x, m_y; p_i, p_j, q_i, q_j) = \#\{(k,l) : p_i \leqslant m_x \leqslant p_j \text{ and } q_i \leqslant m_y \leqslant q_j\}. \quad (4)$$

Fig. 4 illustrates the method of extracting feature vectors that are extracted from the rectangular bins.

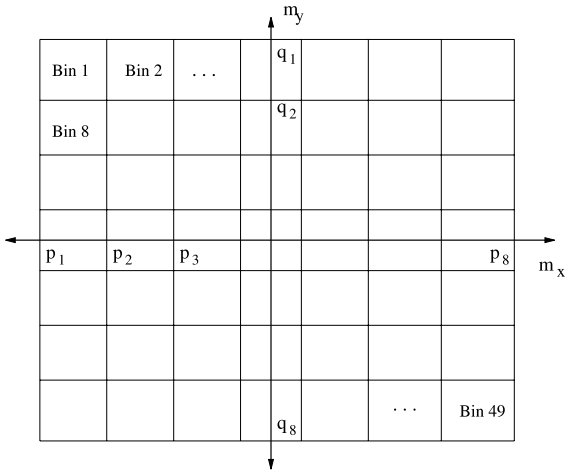

Fig. 4. The number of motion vectors falling within each rectangular block is counted to get the 2-D Cartesian feature.

3. HMM-based classification

Totally seven actions were considered for classification (walk, run, jump, bend up, bend down, twist right and twist left). A HMM model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$ is created for each action. More about HMM can be found in (Rabiner and Juang, 1993). The parameters $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\Pi}$ are determined during the training process. Here

- $\boldsymbol{A} = \{a_{ij} \,|\, a_{ij} = P(s_{t+1} = q_j \,|\, s_t = q_i)\}$; state transition probability where a_{ij} is the probability of transiting from state q_i to state q_j, s_t, $t = 1, 2, \ldots, T$, is the tth state (unobservable) and T is the length of the observation sequence.
- $\boldsymbol{B} = \{b_j(k) \,|\, b_j(k) = P(v_k | s_t = q_j)\}$; symbol output probability where $b_j(k)$ is the probability of output symbol v_k at state q_j.
- $\boldsymbol{\Pi} = \{\boldsymbol{\Pi}_i \,|\, \boldsymbol{\Pi}_i = P(s_1 = q_i)\}$; initial state probability.

For a classifier of seven categories, we choose the model which best matches the observation from seven HMMs, i.e $\lambda_i = (\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{\Pi}_i)$ $(i = 1, 2, \ldots, 7)$. For a observation sequence of length T, $O = (o_1, o_2, \ldots, o_T)$, we calculate $P(O|\lambda_i)$ $(1 \leqslant i \leqslant 7)$ and select the class λ_c such that

$$\lambda_c = \arg\max_i \{P(O|\lambda_i)\}.$$

4. Applying HMM to MPEG video

The motion vectors from each inter-coded frame is decoded and the feature vectors are obtained as explained in Section 2, which are subsequently transformed into symbol sequences using a codebook. The codebook is generated by clustering several training sequences of the same action by means of k-means clustering technique. The number of appropriate symbols (cluster centers) for each action is decided to be P, if the difference in MSE [2] for P and $P + 1$ clusters is less than a threshold τ. The typical value of τ is chosen to be 5–10% of the maximum error which corresponds to having one cluster. The cluster centers for all actions obtained by using the above method are collected together to form a codebook. Let $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M\}$ be the codebook vectors (cluster centers of all actions) and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$ be its corresponding symbols. Let $\{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_T\}$ be the frames of an MPEG sequence and $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_T\}$ be the feature vectors. The symbol sequence $\{o_1, o_2, \ldots, o_T\}$ is obtained as

$$o_i = v_k, \quad \begin{cases} k = \arg\min_j d(\boldsymbol{f}_i, \boldsymbol{c}_j) & 1 \leqslant i \leqslant T, \\ & 1 \leqslant j \leqslant M, \end{cases} \tag{5}$$

where $d(\boldsymbol{f}_i, \boldsymbol{c}_j)$ is the Euclidean distance between the feature vector $\boldsymbol{f}_i$ and $\boldsymbol{c}_j$. The symbol sequence of an action from several sequences is used to train the HMM for that action. Initially the matrices $\boldsymbol{A}$ and $\boldsymbol{\Pi}$ are set to be equi-probable, and the matrix $\boldsymbol{B}$ is initialized by manually segmenting the sequence and finding the proportion of symbol sequences that is emitted from each of the states. Fig. 5 shows the various modules involved in the training phase. In the testing phase, the symbol sequence for a given test sequence is obtained as discussed, and the log likelihoods for all HMM models are calculated. The given test sequence is declared to belong to the class which has the maximum log likelihood.

[2] Since the clustering converges to a local minima, we use multiple restart for each number of clusters with randomly chosen cluster centers and pick the minimum value of MSE.

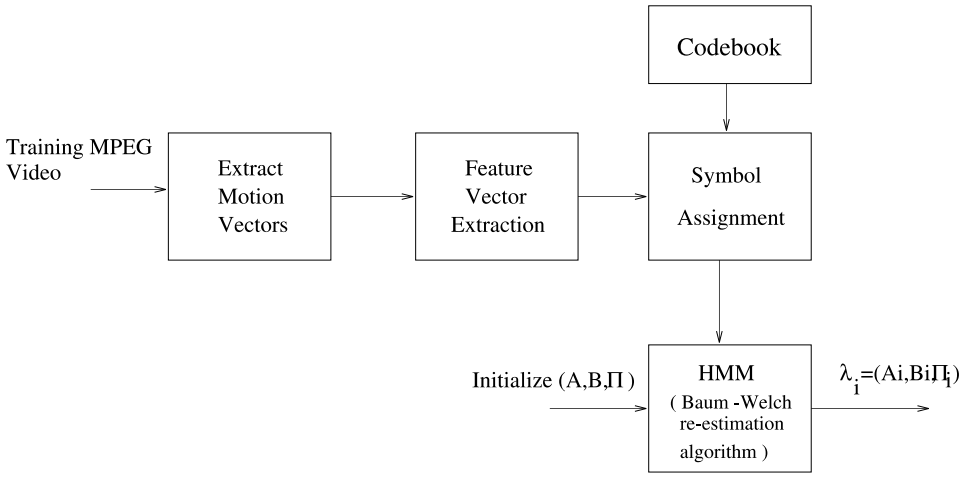

Fig. 5. Illustration of training phase.

5. Results and discussion

In this section , we detail the results obtained by using the features explained by (1)–(4), with recognition results shown in Table 1.

5.1. Experiment 1

The motion vector components obtained were in the range of −30 to 30 with half-pel accuracy and the interval $[r_i, r_j]$ in (1) and (2) is chosen to be

Table 1
Test results obtained for all the proposed feature

Action	No. of test Sequences	Classification accuracy (%)		
		1-D feature	2-D polar	2-D Cartesian
Walk	15	100	100	100
Run	8	100	100	100
Jump	11	100	100	100
Bend Up	15	100	100	100
Bend Down	17	100	100	100
Twist Right	14	86	100	64
Twist Left	13	85	85	92

Table 2
Log likelihoods for all actions of one person using 1-D projected feature

Action	Walk	Run	Jump	BnU	BnD	TwR	TwL
Walk	**−106**	−494	−1177	−1067	−434	−317	−1171
Run	−124	**−86**	−473	−605	−554	−620	−576
Jump	−469	−412	**−55**	−234	−273	−376	−352
BnU	−851	−839	−206	**−102**	−717	−296	−294
BnD	−511	−264	−141	−491	**−46**	−145	−142
TwR	−441	−403	−338	−276	−328	**−128**	−288
TwL	−1128	−1086	−325	−377	−378	−338	**−166**

Table 3
Log likelihoods for all actions of one person using 2-D polar feature

Action	Walk	Run	Jump	BnU	BnD	TwR	TwL
Walk	**−144**	−491	−1146	−1165	−802	−559	−1161
Run	−285	**−107**	−445	−587	−615	−620	−589
Jump	−470	−385	**−75**	−217	−290	−293	−297
BnU	−849	−603	−412	**−69**	−744	−189	−257
BnD	−536	−428	−142	−395	**−65**	−250	−139
TwR	−418	−460	−404	−312	−301	**−127**	−267
TwL	−1068	−684	−411	−407	−357	−446	**−176**

non-overlapping intervals of length five leading to a 24-dimensional feature vector:

$$F_x = [f_x(m_x, -30, -25) \ldots f_x(m_x, 25, 30)],$$
$$F_y = [f_y(m_y, -30, -25) \ldots f_y(m_y, 25, 30)].$$

The feature vector is given by

$$\mathscr{F}_1 = [F_x \; F_y]. \tag{6}$$

In all our experiments, five people performed an action three times for training. The number of states for all HMMs was set at 4 and the number of symbols according to k-means clustering was 4 for walk and run and 3 for rest of the actions. The subjects in the test sequence were different from the one used for training. Totally three subjects performing all the actions were tested for recognition. Out of 93 test sequences considered for recognition the system could recognize 89 correctly. The log likelihoods for all actions performed by one person using all the three features are given in Tables 2–4.

5.2. Experiment 2

In this experiment results are obtained using the feature vectors as explained by (3). The complete angle range from $-\pi$ to π is divided into non-overlapping sectors of $\pi/4$ and the motion vectors falling within each sub-sector are in turn grouped based on the magnitude (see Fig. 3).

Let

$$F_{\theta_i,\theta_j} = [f(\theta, 0, 5) \quad f(\theta, 5, 10) \quad \cdots \quad f(\theta, 25, 30)],$$

where $f(\theta, r_i, r_j)$ is explained by (3). The feature vector is given by

$$\mathscr{F}_2 = [F_{-\pi,-\frac{3\pi}{4}} \quad F_{-\frac{3\pi}{4},-\frac{\pi}{2}} \quad \cdots \quad F_{\frac{3\pi}{4},\pi}]. \tag{7}$$

Here the dimension of feature vector is 48. The training and the testing procedure for this experiment is same as described for Experiment 1. Out of 93 sequences considered for recognition, the system could recognize 91 correctly.

5.3. Experiment 3

The entire ranges of x and y components of motion vectors are divided into non-overlapping bins (not necessarily of equal area).

Let $F_{p_i,p_j} = [f(m_x; m_y, q_i, q_j)]$ be the number of motion vectors whose x component fall within the range (p_i, p_j) and y component within (q_i, q_j). The ranges of (p_i, p_j) and (q_i, q_j) are chosen in such a way as to cover the entire range in the following

Table 4
Log likelihoods for all actions of one person using 2-D Cartesian feature

Action	Walk	Run	Jump	BnU	BnD	TwR	TwL
Walk	**−73**	−478	−1207	−1198	−1208	−227	−1209
Run	−340	**−122**	−414	−534	−560	−508	−525
Jump	−380	−277	**−54**	−204	−255	−337	−249
BnU	−712	−404	−225	**−119**	−405	−193	−184
BnD	−443	−235	−146	−191	**−50**	−221	−102
TwR	−464	−357	−417	−192	−231	**−112**	−300
TwL	−1038	−587	−374	−271	−330	−436	**−164**

intervals of $\{[-30,-15),[-15,-8),\ [-8,-2),[-2,2],(2,8],(8,15],(15,30]\}$. The feature vector is given by

$$\mathscr{F}_3 = [F_{p_i,p_j}] \quad \forall (p_i,p_j) \text{ in the above range.} \tag{8}$$

The dimension of $\mathscr{F}_3$ is 49. Out of 93 sequences considered for recognition, the system could recognize 87 correctly. Figs. 7–9 show the discriminative property of all the proposed features. Tables 2–4 give the log likelihood of each input

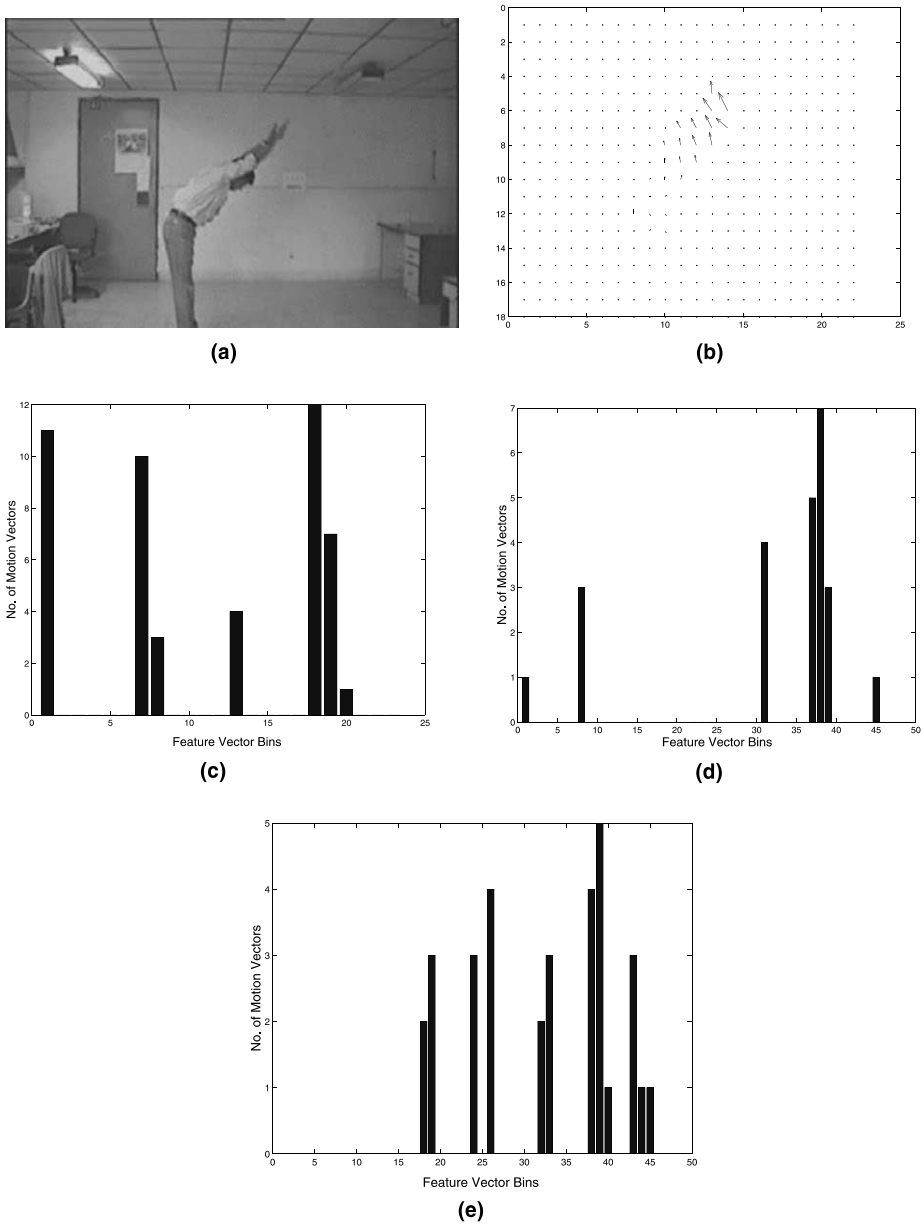

Fig. 6. (a) A frame of bend down action and (b) the corresponding motion vectors. The histograms of the feature vectors extracted from the frame shown in (a) for (c) 1-D projected feature (d) 2-D polar feature (e) 2-D Cartesian feature.

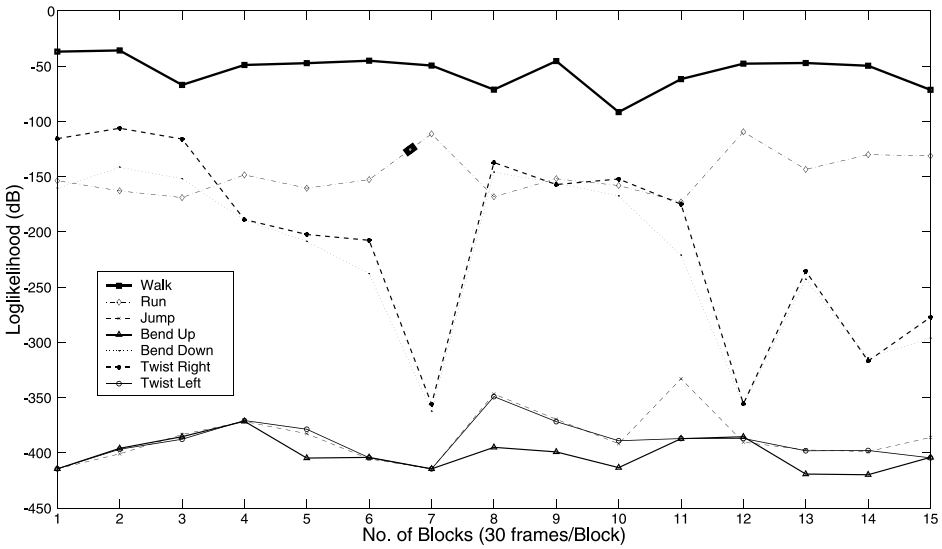

Fig. 7. Log likelihood of a test walk sequence using 1-D projected feature.

action tested against the models of all actions. The action corresponding to the number in bold letters indicates the result of the classifier. Fig. 6 shows the histograms of all the feature vectors extracted from a frame of bend down sequence.

To evaluate the discriminative property of all the proposed features, we used the following measure:

$$d(\mathscr{F}_{\mathrm{fe}}) = \frac{1}{N}(\boldsymbol{L}_{\mathrm{fe}}(c) - \boldsymbol{L}_{\mathrm{fe}}(d)),$$

where $\boldsymbol{L}_{\mathrm{fe}}(c)$ is the log likelihood of the input action obtained by using the feature *fe* (one of the proposed features), N is the number of blocks (30 frames per block), $c = \arg\max_i \{P(O|\lambda_i)\}$ $(i = 1, 2, \ldots, 7)$ and $d = \arg\max_j \{P(O|\lambda_j)\}$ $\forall j \neq c$.

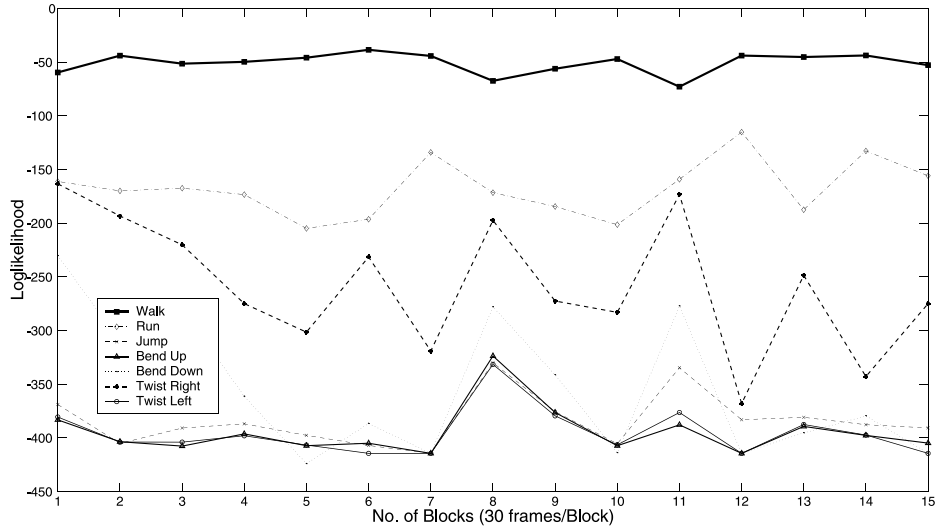

Fig. 8. Log likelihood of a test walk sequence using 2-D polar feature.

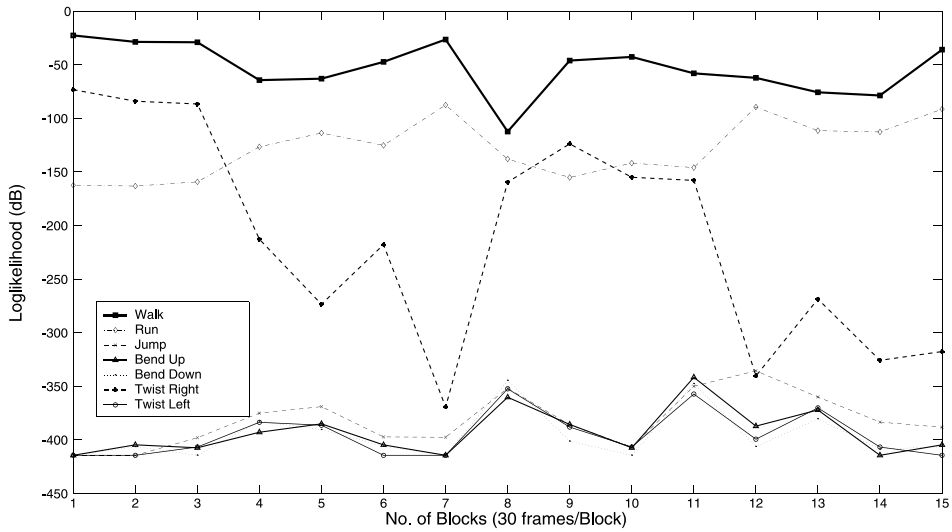

Fig. 9. Log likelihood of a test walk sequence using 2-D Cartesian feature.

Table 5
The discriminative property of the proposed features (in dB)

Action	1-D projected	2-D polar	2-D Cartesian
Walk	81.53	116.80	57.22
Run	34.04	137.14	126.72
Jump	143.52	140.34	146.38
Bend Up	47.36	59.04	30.69
Bend Down	40.89	43.13	29.94
Twist Right	12.09	17.93	4.34
Twist Left	19.29	21.36	21.39

Table 5 gives the discriminative index of all the proposed features using the above mentioned measure for all the actions. From Table 5, though the overall performance of the 2-D polar feature is better than the other features, for some specific actions like 'Jump' and 'Twist Left' the performance of the 2-D Cartesian feature is superior. Moreover the size of the feature vector is directly proportional to the computational complexity. Though the decrease in performance of 1-D projected feature is insignificant, a considerable computational saving is achieved due to its reduced feature vector size (half the size of 2-D polar feature).

Apart from the above method of codebook generation, we constructed a codebook by clustering the feature vectors of all the action sequences into 23 clusters (same as the total number of clusters used in the previous method). All HMMs were trained and tested using this codebook. It was observed that the performance of this global clustering method was not as good as the results obtained by the method proposed in Section 4.

6. Conclusion

In this paper we have proposed a system for classification of various human actions from a partially decompressed MPEG video using HMM. Three types of features obtained from the motion vectors are described and the results are compared. The performances of all the feature vectors are compared and the overall discriminating property of 2-D polar feature is found to be better than

other features. The system performance can be further improved by increasing the training data. This work can be extended to classify a video containing more than one subject by separating each of the subjects in the video by applying object segmentation methods.

Acknowledgements

The authors wish to express grateful thanks to the anonymous referees for their useful comments and suggestions.

References

Allmen, M., Dyer, C., 1993. Toward human action recognition using dynamic perceptual organization.

Boykin, S., Merlino, A., 2000. Machine learning of event segmentation for news on demand. Comm. ACM 43 (2), 35–41.

Bregler, C., 1997. Learning and recognizing human dynamics in video sequences. In: Proc. IEEE CVPR, pp. 568–574.

Davis, J., Bobick, A., 1997. The representation and recognition of human movements using temporal templates. In: Proc. IEEE CVPR, pp. 928–934.

Davis, J., Shah, M., 1995. Motion based recognition: a survey. Image Vision Comput. 13 (2), 129–155.

Eickeler, S., Muller, S., Rigoll, G., 2000. Recognition of jepg compressed face images based on statistical methods. Image Vision Comput. 18 (4), 279–287.

Hu, M., 1962. Visual pattern recognition by moment invariants. IRE Trans. Inform. Theory 8 (2), 179–187.

Kuniyoshi, Y., Inoue, H., 1993. Qualitative recognition of ongoing human action sequences. In: IJCAI, pp. 1600–1609.

Lee, H.K., Kim, J.H., 1999. An HMM-based threshold model approach for gesture recognition. IEEE Trans. Pattern Anal. Machine Intell. 21 (10), 961–973.

Mitchell, J., Pennebaker, W., Fogg, C., LeGall, D., 1996. MPEG Video Compression Standard. International Thomson Publishing.

Rabiner, L., Juang, B.H., 1993. Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ.

Rosales, R., 1998. Recognition of human action based on moment based features. Technical Report BU 98-020, Boston University, Computer Science.

Shah, M., Jain, R., 1997. Motion based Recognition. Kluwer Academic, Dordrecht.

Starner, T., Pentland, A., 1995. Real-time American sign language recognition from video using hidden Markov models. Technical Report TR-375, MIT Media Lab.

Umeda, M., 1982. Recognition of multi-font printed chinese character. In: Proc. 6th Computer Vision and Pattern Recognition, pp. 793–796.

Yamato, J., Ohya, J., Ishii, K., 1992. Recognizing human action in time-sequential images using hidden Markov model. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 379–385.

Yoon, K., DeMenthon, D., Doermann, D., 2000. Event detection from mpeg video in the compressed domain. In: Internat. Conf. on Pattern Recognition, Barcelona, Spain.