Compressed Domain Action Classification Using HMM

R. Venkatesh Babu, B. Anantharaman, K.R. Ramakrishnan and S.H. Srinivasan
Department of Electrical Engineering, Indian Institute of Science,
Bangalore, Karnataka, India - 560 012.
e-mail: {rvbabu,ananth,krr}@ee.iisc.ernet.in, shs@mmsl.serc.iisc.ernet.in

Abstract

This paper proposes three techniques for person independent action classification in compressed MPEG video. The features used are based on motion vectors, obtained by partial decoding of the MPEG video. The features proposed are projected 1D, 2D polar and 2D Cartesian. 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.

1 Introduction

Identifying video clippings in terms of some action taking place in the video is very useful for video indexing and retrieval. As more and more video clips are available in MPEG compressed format, one needs technologies that use only motion encodings which can be easily obtained by partial frame decompression. Compressed domain techniques allow for rapid analysis of video content. There are a plethora of applications which calls for compressed domain processing of MPEG video. Segmentation of MPEG video, text extraction, object segmentation, event detection are some of the examples.

While there are a few research papers [5] [1] [4] which deal with pixel domain classification of action from image sequences to our knowledge there aren't any papers on the compressed domain classification of action sequences using the readily available motion data. The difficulty in the pixel domain action classification is that it is computationally expensive and hence may not be suitable for real time applications.

Human action is more characterized by motion than by any other spatial information. In the case of MPEG [2], motion is characterized by the motion vectors available from the inter coded frames (P and B). In this paper we propose three techniques for extracting features from the readily available motion vectors of MPEG video. The actions considered are: walk, run, jump, bend up, bend down, twist right, twist left. A discrete HMM for each action is trained with the corresponding training MPEG video sequence. The details about training phase is explained in Section 4.

The experimental results are provided in Section 5 and the concluding remarks in Section 6.

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. As shown in Fig. 1 the GOP structure of all MPEG video sequences considered was in $IB_1B_2P\ldots$ 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 of motion vectors between any I and P frame in the GOP.

For B_1 frame the scaled motion vector along x direction $mx = mx_{B1} \times 3$ where mx_{B1} is the motion vector obtained from B_1 frame and for B_2 frame the scaled motion vector $mx = mx_{B2} \times (-3)$, where mx_{B2} 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 above feature extraction methods are applied. Initially a binary image is obtained by assigning zero to the macroblocks having no motion and one to the

0-7695-1354-9/01 $17.00 © 2001 IEEE

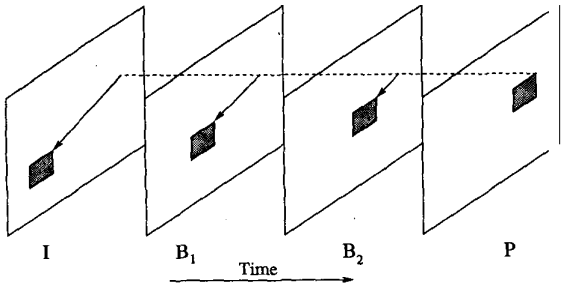

Figure 1. The relative motion vectors (forward) between an I frame and P frame under linearity assumption.

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). Fig. **4** shows the effectiveness of the noise removal.

Projected 1-D feature:

The motion vectors are projected to the x-axis and y-axis to get the x component m_x and the y component m_y of the motion vector. Here these two components are treated separately to get the feature vector.

Let $f_x(m_x; r_i, r_j)$ be the number of motion vectors in x-direction in the range r_i and r_j , with $(r_i < r_j)$. That is,

$$f_x(m_{x_k}; r_i, r_j) = \sum_{k=1}^{N} I_{[r_i,r_j]}(m_{x_k}) \qquad (1)$$

where, $I_{[r_i,r_j]}(m_{x_k})$ is the indicator function given by

$$I_{[r_i,r_j]}(m_{x_k}) = \begin{cases} 1 & if \quad r_i \le m_{x_k} < r_j \\ 0 & otherwise \end{cases}$$

N is the total number of motion vectors, m_{x_k} is the k^{th} motion vector along x-direction.

Similarly, let $f_y(m_y; r_i, r_j)$ be the number of motion vectors in the y direction in the range r_i and r_j with $(r_i < r_j)$.

$$f_y(m_{y_k}; r_i, r_j) = \sum_{k=1}^{N} I_{[r_i,r_j]}(m_{y_k}) \qquad (2)$$

where, m_{y_k} is the k^{th} motion vector along y direction.

Using different non-overlapping intervals $(r_i - r_j)$, an histogram is obtained for each direction for each inter coded frame.

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

$$f(\theta_{m_{x_k},m_{y_k}}; \theta_i, \theta_j, r_i, r_j) = \sum_{k=1}^{N} I_{[\theta_i,\theta_j,r_i,r_j]}(m_{x_k}, m_{y_k}) \qquad (3)$$

where,

$$I_{[\theta_i,\theta_j,r_i,r_j]}(m_{x_k}, m_{y_k}) = \begin{cases} 1 \; if \; \begin{cases} \theta_i \le \theta_{m_{x_k},m_{y_k}} < \theta_j \\ and \\ r_i \le \sqrt{m_{x_k}^2 + m_{y_k}^2} < r_j \end{cases} \\ 0 \qquad elsewhere \end{cases}$$

where, $\theta_{m_{x_k},m_{y_k}}$ is the direction angle of the motion vector with respect to common reference.

In all the above feature extraction methods (r_i, r_j) and (θ_i, θ_j) are chosen in such a way to cover the entire range of motion vectors and the angle ranging from $-\pi$ to π in a non-overlapping manner. Fig. **2** illustrates the method by which feature vectors are extracted by considering the direction(angle) and magnitude of motion vectors.

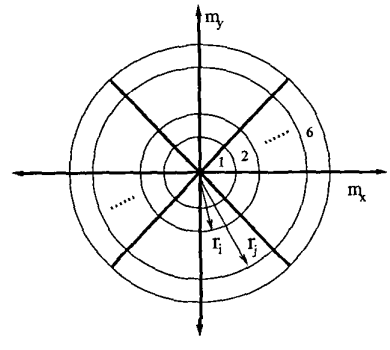

Figure 2. The number of motion vectors falling within each sector is counted to get the 2-D polar feature.

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 bins of not necessarily equal area.

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

$$f(m_{x_k}, m_{y_k}; p_i, p_j, q_i, q_j) = \sum_{k=1}^{N} I_{[p_i,p_j,q_i,q_j]}(m_{x_k}, m_{y_k}) \tag{4}$$

where,

$$I_{[p_i,p_j,q_i,,q_j]}(m_{x_k}, m_{y_k}) = \begin{cases} 1 \; if \begin{cases} p_i \le m_{x_k} < p_j \\ and \\ q_j \le m_{y_k} < q_j \end{cases} \\ 0 \qquad elsewhere \end{cases}$$

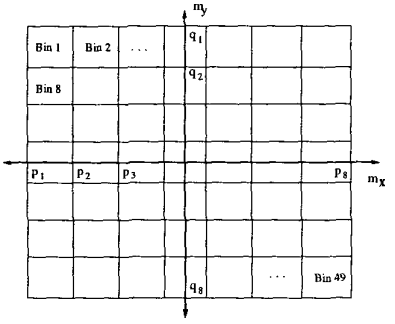

Figure 3. The number of motion vectors falling within each block is counted to get the 2-D Cartesian feature.

Preprocessing :

As the motion vectors obtained are noisy, preprocessing is done before the above feature extraction methods are applied.

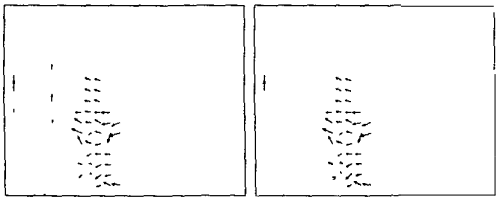

Figure 4. (a) The motion vectors before noise removal and (b) after morphologicalfiltering.

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). Fig. **4** shows the effectiveness of the noise removal.

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 $\lambda = (\mathbf{A},\mathbf{B},\Pi)$ is created for each action. More about HMM can be found in [3]. The parameters **A,** B and IS are determined during the training process. Here

$\mathbf{A} = \{a_{ij} | a_{ij} = P(s_{t+1} = q_j | s_t = q_i)\}$; state transition probability where

a_{ij} is the probability of transiting from state q_i to state q_j.

$s_t, t = 1, 2..T$ is the t^{th} state (unobservable); $T-$lengthof the observation sequence.

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

IS $= \{\mathbf{\Pi}_i \mid \mathbf{\Pi}_i = P(s_1 = q_i)\}$; Initial state probability.

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

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

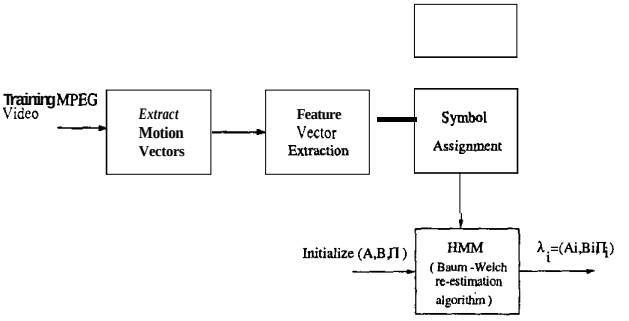

Figure 5. Illustration of training phase

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 for P and P+1 clusters is less than a threshold τ. The typical value of τ is chosen to be 5% to 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 $\{\mathbf{c}_1, \mathbf{c}_2, ...\mathbf{c}_M\}$ be the codebook vectors (cluster centers of all actions) and $\{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_M\}$ be its corresponding symbols. Let $\{\mathbf{I}_1, \mathbf{I}_2, ..\mathbf{I}_T\}$ be the frames of a MPEG sequence and $\{\mathbf{f}_1, \mathbf{f}_2, ...\mathbf{f}_T\}$ be the feature vectors. The symbol sequence $\{o_1, o_2, ...o_T\}$ is obtained as

$$o_i = v_j \left\{ \text{where, } j = \underset{3}{\operatorname{argmin}}\ d(\mathbf{f}_i, \mathbf{c}_j) \quad \begin{matrix} 1 \le i \le T \\ 1 \le j \le M \end{matrix} \right. \tag{5}$$

where, $d(\mathbf{f}_i, \mathbf{c}_j)$ is the Euclidean distance between the feature vector $\mathbf{f}_i$ and $\mathbf{c}_j$. The symbol sequence of an action from several sequences are used to train the HMM for that action. Initially the matrix $\mathbf{A}$ and $\mathbf{\Pi}$ is set to be equiprobable, and the matrix 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 likelihood for all HMM models are calculated. The given test sequence is declared to belong to the class which has the maximum log likelihood.

5 Results and Discussion

In this section , we detail the results obtained by using the features explained by **(1), (2), (3)** and **(4),** with recognition results shown in Table 1. The GOP structure of all video sequences considered was in *IBBP* ... format with 12 frames **per** GOP. All the actions considered are performed in front of a still camera and a fixed background.

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

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

The feature vector is given by,

$$\mathcal{F}_1 = [F_x\ F_y] \tag{6}$$

In all our experiments, five people performed an action three times for training. The number of states for all HMMs are set at four and the number of symbols according to k-means clustering was **4** for walk and run and three 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 likelihood for all actions performed by one person using all the three features are given in tables **2 , 3** and **4.**

Table 1. Test results obtained for all the proposed feature

Action	No of test	Classification Accuracy(%)		
	Sequences	1D feature	2D polar	2D 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 1D 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

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 $\frac{\pi}{4}$ and the motion vectors falling within each sub-sector is in **turn** grouped based on the magnitude (see Fig. **2).**

Let

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

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

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

Table 3. Log likelihoods for all actions of one person using 2D 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**

Table 4. Log likelihoods for all actions of one person using 2D 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**

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.

Experiment 3:

The entire range of x and y component 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 falling 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) is chosen in such a way to cover the entire range in the following intervals of $\{[-30,-15),[-15,-8),[-8,-2),[-2,2],(2,8],(8,15]$, (15,301) . The feature vector is given by,

$$\mathcal{F}_3 = [F_{p_i,p_j}],\ \forall (p_i,p_j) \text{ in the above range} \qquad (8)$$

The dimension of $\mathcal{F}_3$ is 49. Out of 93 sequences considered for recognition, the system could recognize 87 correctly. Fig. 8,7 and 9 shows that the discriminative property of all the proposed features. Tables **2,3** and 4 give the log likelihood of each input 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 histogram 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, $d(\mathcal{F}_{fe}) = \frac{1}{N}(\mathbf{L}_{fe}(c) - \mathbf{L}_{fe}(d))$, where, $\mathbf{L}_{fe}(c)$ is the log likelihood of the input action obtained by using the feature fe (one of the proposed features), where N is the number of blocks (30 frames per block), $c = \arg\max_i\{P(O \mid \lambda_i)\}, (i = 1,2...7)$ and $d = \arg\max_j\{P(O \mid \lambda_j)\}, \forall j \neq$ c . Table 5. gives the discriminative index of all the proposed features using the above mentioned measure for all the actions. From the table 5, it is found that 2-D polar feature performs better than other features for all actions except 'jump' , 2-D Cartesian feature performs better for the action 'jump' and shows poor performance for 'twist right'. The overall performance of 2-D polar feature is found to be superior, and 1-D projected feature performs better than 2-D Cartesian feature.

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

Action	1-D projected	2-Dpolar	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

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 performance of all the feature vectors are compared and the overall discriminating property of 2D 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

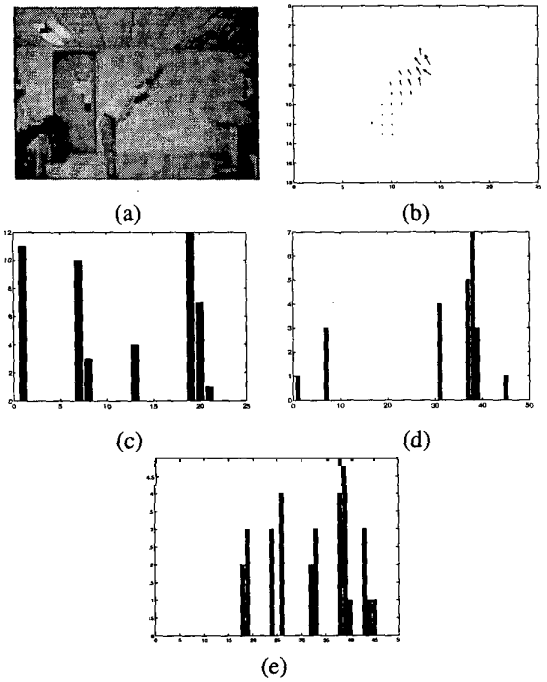

Figure 6. (a) A frame of bend down action and (b) the corresponding motion vectors. The histograms of the feature vectors extracted from the frame shown in (a) for (c) 1-D projected feature (d) 2-D polar feature (e) 2-D Cartesian feature

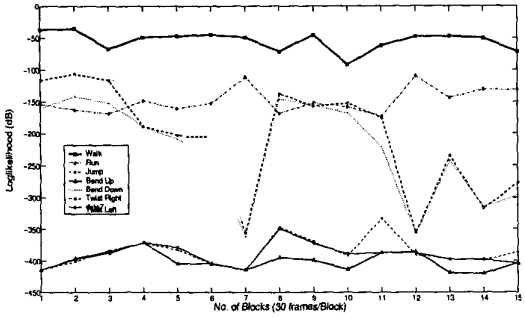

Figure 7. Log Likelihood of a test walk sequence using 1D projected feature.

of the subjects in the video by applying object segmentation methods.

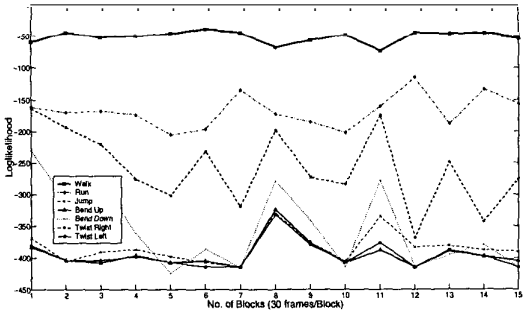

Figure 8. Log Likelihood of a test walk sequence using 2D polar feature.

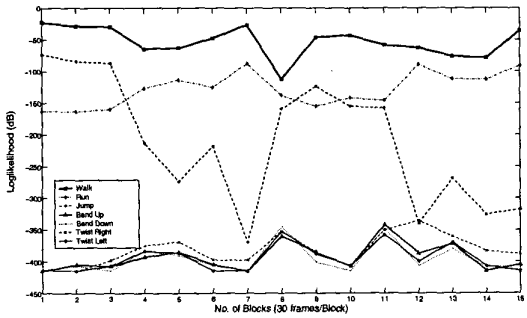

Figure 9. Log Likelihood of a test walk sequence using 2D Cartesian feature.

References

[1] C. Bregler. Learning and recognizing human dynamics in video sequences. In ***Proceedings of the*** *IEEE* ***CVPR,*** pages 568–574,1997.

[2] J.L. Mitchell, W.B. Pennebaker, C.E. Fogg, and D.J. LeGall. ***MPEG Video Compression Standard.*** International Thomson Publishing, 1996.

[3] L. Rabiner and B. H. Juang. ***Fundamentals of Speech Recognition.*** Prentice Hall PTR, 1993.

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

[5] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In ***Proceedings of*** *the IEEE* ***Conference on Computer Vision and Pattern Recognition,*** pages 379–385,1992.