A Method For Human Action Recognition

Osama Masoud and Nikos Papanikolopoulos*

Department of Computer Science and Engineering

University of Minnesota

200 Union Street SE, 4-192 EE/CS Bldg.

Minneapolis, MN 55455

{masoud,npapas}@cs.umn.edu

February 2003

*Corresponding author.

Abstract

This paper deals with the problem of classification of human activities from video. Our approach uses motion features only that are computed very efficiently, and subsequently projected into a lower dimensional space where matching is performed. Each action is represented as a manifold in this lower dimensional space and matching is done by comparing these manifolds. To demonstrate the effectiveness of this approach, it was used on a large data set of similar actions, each performed by many different actors. Classification results were very accurate and show that this approach is robust to challenges such as variations in performers' physical attributes, color of clothing, and style of motion. An important result of this paper is that the recovery of the three-dimensional properties of a moving person, or even the two-dimensional tracking of the person's limbs need not precede action recognition.

Keywords: Motion recognition, human tracking, articulated motion.

1 Introduction

Recognition of human actions from video streams has many applications in the surveillance, entertainment, user interfaces, sports and video annotation domains. Given a number of predefined actions, the problem can be stated as that of classifying a new action into one of these actions. Normally, the set of actions has a meaning in a certain domain. In sign language for example, the set of actions corresponds to the set of possible words and letters that can be produced. In ballet, the actions are the step names in one of the ballet notation languages.

In psychophysics, the study of human body motion perception by the human visual system was made possible by the use of the so-called moving light displays (MLDs) first introduced by Johansson in 1973 [22,23]. Johansson devised a method to isolate the motion cue by constructing an image sequence where the only visible features are a set of moving lights corresponding to joints of the human body. Figure 1 shows an example. He found that when a subject was presented an MLD corresponding to an actor performing an activity such as walking, running, or stair climbing, the subject had no problem recognizing the activity in under 200 milliseconds. The subjects were not able to identify humans when the lights were stationary. Cutting and Kozlowski [11,25] demonstrated that the gender of the walking person and the gait of a friend can be identified from MLDs. It was later shown that subjects can identify more complex movements such as hammering, box lifting, ball bouncing, dancing, greeting, and boxing [14]. Two theories on how people recognize actions from MLDs have been suggested. In the first theory, the visual system performs shape-from-motion reconstruction of the object and then use that to recognize the action. In the second theory, the visual system utilizes motion information directly without performing reconstruction.

In this paper, we present a general method for human activity classification. Our method

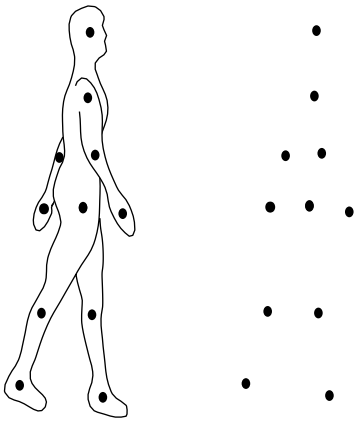


Figure 1 A moving light display with and without the human body outline.

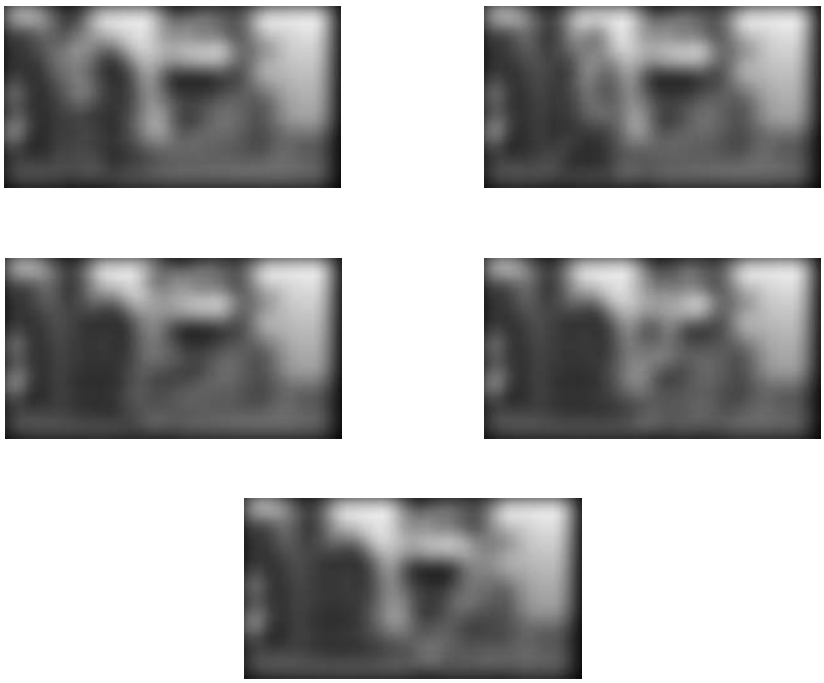


Figure 2 Snapshots from a motion sequence (of a person skipping) where the images have been blurred. Humans have no difficulty perceiving the action when they watch the movie.

uses motion information directly from the video sequence. The other alternative is to perform tracking in 2-D or in 3-D and then use the tracking information to do action classification. Although there has been a few successful attempts to perform limb tracking in 2D and 3D, tracking an articulated body like the human body remains a complex problem due to issues of self-occlusion and the effects of clothing on appearance. Our work is motivated by the need to investigate if it is possible to perform the task without having to perform limb tracking. It is also motivated by psychophysical evidence [12] where it was demonstrated that our visual capabilities allow us to perceive actions with ease even when presented with an extremely blurred image sequence of an action (see Figure 2). These experiments suggest that using motion alone to recognize actions may be favorable to reconstruction-based approaches.

Our method uses motion extracted directly from the image sequence. At each frame, motion information is represented by a *feature image*. Motion information is calculated efficiently using an Infinite Impulse Response (IIR) filter. A different method, though conceptually similar, was used by Davis and Bobick [12]. Unlike [12], an action is represented by several feature images rather than just one image. Actions can be complex and

repetitive, making it difficult to capture motion details in one feature image. Motion features were also used by Polana and Nelson [33]. In this case, several motion features were extracted throughout the action duration. The features were based on normal flow and were very small in size (a 4×4 matrix). Our choice of IIR filtering is motivated by the efficiency of this approach. The feature image used is not limited to a small size. Higher representation resolution can provide discriminatory power when there is a similarity among actions. Dimensionality reduction using principle component analysis (PCA) is then utilized at the recognition stage.

Our method is view dependent in that it assumes the actions are performed in a fronto-parallel fashion with respect to the camera. Some ways to overcome this limitation are discussed in Section 7. To evaluate our approach we perform classification experiments involving eight different action classes, each performed by 28 different people for a total of 232 samples, 168 of which were used as test samples.

The paper is organized as follows. Related work is presented in Section 2. After explaining the details of the feature image representation in Section 3, the recognition process is described in Section 4. The action data that was used for testing and its acquisition steps are presented in Section 5. Finally, results and analysis follow in Section 6.

2 Related Work

Human tracking and, to a lesser extent, human action recognition have received considerable attention in recent years. A good review can be found in [10,16]. The tracking problem was addressed as the problem of whole body tracking (e.g., [8,28]), limb tracking in 2D image space (e.g., [1,38]), and limb tracking in 3D space (e.g., [2,7,13,17,37]). Action recognition was also addressed in several ways. We can classify the different approaches into three categories based on the type of input used. The input can be either 2D tracking data, 3D tracking data, or motion features extracted directly from the image. The next three paragraphs describe some of the work done in these categories. The work described in this paper belongs to the third category.

2D tracking data in the form of MLDs was used by Goddard [18]. Guo *et al.* [19] used the parameters of 2D stick figures fitted to tracked silhouettes. Yacoob and Black [39] used 2D tracking data in the form of parameterized models of the tracked legs. The recovered parameters over the duration of the action were then compressed using PCA. Matching took place in eigenspace. They reported a recognition rate of 82% using four action classes. Pavlovic and Rehg [31] used tracked 2D limbs to learn motion dynamics using a class of learned dynamic models. Bregler [6] used tracked features on the human at the image level and propagated hypotheses probabilistically utilizing Hidden Markov Models (HMMs). Rangarajan *et al.* [34] matched motion trajectories using scale space. They used

speed and direction parameters rather than locations to achieve translation and rotation invariance. The input was a set of manually tracked points on several parts of the body performing the action. Given two speed signals, matching was performed by differencing the scale space images of the signals.

The second category methods use 3D body tracking information. Upon successful 3D tracking, motion recognition can make use of any of the recovered parameters such as joint coordinates and joint angles. Although there has been a tremendous amount of work in 3D limb tracking, work in action recognition that uses 3D tracking information been limited to inputs in the form of MLDs obtained by placing markers on various body joints which are tracked in 3D. The techniques by Campbell and Bobick [9] who used phase-space, and Gavrila and Davis [17] who used dynamic time warping belong to this category.

The third and final category of methods use image features directly. One such method is by BenAbdelkader *et al.* [4]. Their method is similar to ours in that it uses PCA to represent features but they targeted the problem of gait recognition (identification of individuals by the way they walk) as opposed to action classification. Foster *et al.* [15] and Huang *et al.* [21] also tackled the problem of gait recognition. They used silhouettes and area features and applied PCA techniques. Krahnstover *et al.* [26] described an interesting spatio-temporal approach that can not only recognize the action but track it as well. The features used were frame-to-frame differences. It would be interesting to see how their approach performs on a large set of action classes. Yamato *et al.* [40] used HMMs to distinguish different tennis strokes. The feature vector was formed for every frame based on spatial measurements of the foreground. Recognition was done simply by selecting the HMM that was most likely to generate the given sequence of feature vectors. The main advantage of such an approach is that adding a new action can be simply done by training a new HMM. The approach, however, was sensitive to the shape of the person performing the stroke. The use of motion features rather than spatial features may reduce this sensitivity. Davis and Bobick [12] used what they called *motion-history* images (MHIs). An MHI represents motion recency where locations of more recent motions are brighter than older motions. A single MHI is used to represent an action. A pattern classification technique using seven Hu moments of the image was then used for recognition. They presented results of recognizing aerobic exercises performed by two actors, one for training and one for testing. The choice of an appropriate duration parameter used in the MHI calculation is critical. Temporal segmentation was done by trying all possible parameters. The system was able to successfully classify three different actions: sitting, arm waving and crouching. Motion information extracted directly from the image sequence was also utilized by Polana and Nelson [33]. In their work, they used normal flow. The feature vector in this case was computed by temporally dividing the action into six sections and finding the normal flow in each. Furthermore, each division is spatially partitioned into 4 by 4 cells. The summation of the magnitude of the normal flow at each cell was used to make up the feature vector. Recognition was done by finding the most similar vector in the training set using the near-

est centroid algorithm. The duration of the action was determined by calculating a periodicity measure [32], which helps in correcting for temporal scale but not temporal translation (or phase). To overcome this problem, their technique matched the feature vector at every possible phase shift (six in this case). They tested their method using six different activities, each performed several times by the same person and one activity performed by a toy frog. The method worked fairly well, which shows the discriminatory power of the motion features used.

3 Feature Image Creation

An IIR filter is used to construct the feature image. In particular, we use the response of the filter as a measure of motion in the image. A slightly different formulation of this measurement has been used by Halevi and Weinshall [20]. The idea is to represent motion by its recency: recent motion is represented as brighter than older motion. This technique, also called *recursive filtering*, is simple and time-efficient. It is thus suitable for real-time applications. A weighted average at time i, M_i, is computed as

$$M_i = \alpha \times I_{i-1} + (1 - \alpha) \times M_{i-1}, \quad (1)$$

where I_i, is the image at time i, and α is a scalar in the range 0 to 1. The feature image at time i, F_i, is computed as follows: $F_i = |M_i - I_i|$. Figure 3 is a plot of the filter response to a step function with α set to $0.\dot{5}$. F can be described as an exponential decay function similar to that of a capacitor discharge. The rate of decay is controlled by the parameter α. An α equal to 0 causes the weighted average, M, to remain constant (equal to the background) and therefore F will be equal to the foreground. An α equal to 1 causes M to be equal to the previous frame. In this case, F becomes equivalent to image differencing. Between these two extremes, the feature image captures temporal changes (features) in the sequence. Moving objects produce in a fading trail behind them. The speed and direction of motion are implicit in this representation. The spread of the trail indicates the speed while the gradient of the region indicates direction. Figure 4 shows several frames from a motion sequence along with the extracted motion features using this technique. Note that it is the contrast of the gray level of the moving object which controls the magnitude of F, not the actual gray level value.

With the assumption that the height, h, of the person and his/her location in the image are known, feature images are sized and located accordingly. The feature image is computed in an box of dimensions $0.9h$ by $1.1h$ whose bottom is aligned with the base line and centered around the midline of the person. This is illustrated in Figure 5. The extra height is needed in case there are some actions that involve jumping. The width is large

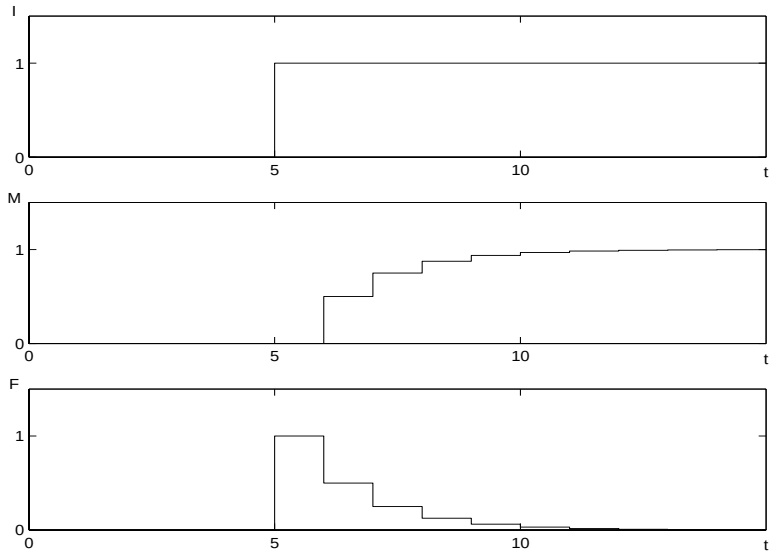


Figure 3 Filter response to a step signal. Given the input I (top), the filter response M (middle) is shown for $\alpha = 0.5$. The feature that we use, F (bottom), is computed as $|M - I|$.

enough to accommodate motion of the legs and the motion trails behind them. Details about how the person's height and location can be estimated will be explained in Section 5.2.

The feature image values are normalized to be in the range $[0, 1]$. They are also thresholded to remove noise and insignificant changes (a threshold of 0.05 was found appropriate). Finally, a low-pass filter is applied to remove additional noise.

4 Learning and Recognition

Our goal is to classify actions into one of several categories. We use the feature image representation calculated throughout the action duration. The idea is to compare the feature images with reference feature images of different learned actions and look for the best match. There are several issues to consider using this approach. Action duration is not necessarily fixed for the same action. Also, the method should be able to handle small speed increases or decreases. Even if we assume that actions are performed at constant speed, we cannot assume temporal alignment and therefore a frame-by-frame matching starting from the first frame should be avoided. The frame-to-frame matching process itself needs to be invariant to the actor's physical attributes such as height, size, color of clothing, etc. Moreover, since an action can be composed of a large number of frames, correlation-based

(a)

(b)

Figure 4 An example of a walking and a running motion sequence. (a) Original images. (b) Filtered images (feature images) with $\alpha = 0.3$.

methods for matching may not be appropriate due to their computationally intensive nature.

All these issues have been considered in the development of our recognition method. This section describes the details of our algorithm and addresses the issues mentioned above.

4.1 Magnitude and Size Normalization

As actions are represented as sequences of feature images, two types of normalization are performed on a feature image:

1. Magnitude normalization: Because of the way feature images are computed, a person

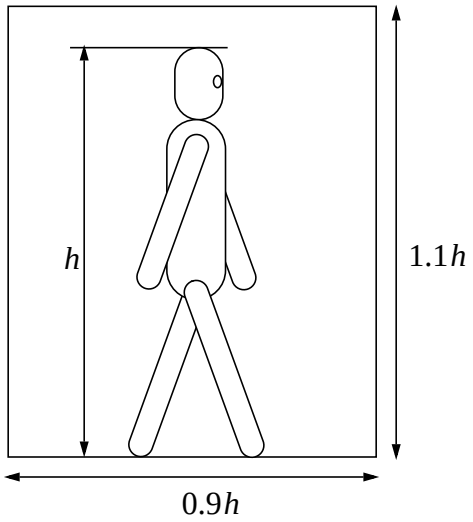


Figure 5 Feature image size selection.

wearing clothes similar to the background will produce low magnitude features. To adjust for this, we normalize the feature image by the 2-norm of the vector formed by concatenating all the values in all the feature images corresponding to the action. The values are then multiplied by the square root of the number of frames to provide invariance to action length (in number of frames).

2. Size normalization: The images are resized so that they are all of equal dimensions. Not only does this type of normalization work across different people but, it also corrects for changes in scale due to distance from the camera, for instance.

4.2 Principle Component Analysis

PCA has been successfully used in the field of face recognition [3,35,36]. It has also been used in gait and action recognition. BenAbdelkader *et al.* [4] used PCA to compress features for the purpose of gait recognition. Their features consisted of regions in a self-similarity plot constructed by comparing every pair of frames in the action. Huang *et al.* [21] also performed gait recognition and represented each person by the centroid of the projected feature images into eigenspace. Krahnstover *et al.* [26] used PCA on feature images computed by image differencing. The projected points were then used to train HMMs. Of a particular relevance to our work is the work of Yacoob and Black [39]. In their method, the features used were based on tracking five body parts using the work of Ju *et al.* [24]. Each tracked part provided eight temporal measurements. Thus, in total, 40 temporal curves are used to represent an action. Training data is composed of these curves for every example action. Each training sample is composed by concatenating all 40 curves. The training data is then compressed using a PCA technique. An action can now be represented in terms of coefficients of a few basis vectors. Given a new action, recogni-

tion is done by a search process which involves calculating the distance between the coefficients for this action and the coefficients of every example action and choosing the minimum distance. Their method handles temporal variation (temporal shift and temporal duration) by parameterizing this search process using an affine transformation.

Our method differs in that an action is not represented by a single point in eigenspace but rather a manifold whose points correspond to the different feature images the action goes through. This moves the burden of temporal alignment and duration adjustments from searching in the measurement space to searching in eigenspace. We see two main advantages for doing this:

1. Reduction in search complexity: Because the eigenspace has a much lower dimension than the measurement space, a more exhaustive search can be afforded.
2. Increased robustness: PCA is based on linear mapping. Action measurements are inherently nonlinear and this nonlinearity increases as these measurements are aggregated across the whole action. PCA can provide better discrimination if the action is not considered as one entity but a sequence of entities.

Nayar *et al.* [30] used parameterized eigenspace manifolds to recognize objects under different pose and lighting conditions.

In our method, the training set consists of a actions each performed a certain number of times, s. For each of the as samples, normalized feature images are computed throughout the action duration. Let the j-th sample of action i consist of T_{ij} feature images: $F_1^{ij}, F_2^{ij}, \ldots, F_{T_{ij}}^{ij}$. A corresponding set of column vectors $\mathbf{S}_{ij} = \left[\mathbf{f}_1^{ij}\ \mathbf{f}_2^{ij}\ \ldots\ \mathbf{f}_{T_{ij}}^{ij}\right]$ is constructed where each $\mathbf{f}$ is formed by stacking the columns of the corresponding feature image. To avoid bias in the training process, we use a fixed number L of $\mathbf{f}$'s since the number of feature images T_{ij} for a particular sample depends on the action and how the action was performed. From every set of $\mathbf{f}$'s, we select a subset consisting of L evenly spaced (in time) vectors $\mathbf{g}_1^{ij}, \mathbf{g}_2^{ij}, \ldots, \mathbf{g}_L^{ij}$. L should be small enough to accommodate the shortest action. To ensure that the selected feature images for the samples of one action correspond to similar postures, the samples for each action are assumed to be temporally aligned. This restriction is removed in the testing phase. The grand mean, μ, of these vectors ($\mathbf{g}$'s) over all i's and j's is computed. The grand mean is subtracted from each one of the $\mathbf{g}$'s and the resultant vectors are the columns of the matrix $\mathbf{X} = \left[\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_N\right]$, where $N = asL$ is the total number of columns. The number of rows of $\mathbf{X}$ is equal to the size of the feature image. The first m eigenvectors $\Phi = [\phi_1\phi_2\ldots\phi_m]$ (corresponding to the largest m eigenvalues) are then computed. Each sample $\mathbf{S}_{ij}$ is first updated by sub-

tracting μ from each column vector and then projected using these eigenvectors. Let $\tilde{\mathbf{S}}_{ij} = \left[\tilde{\mathbf{f}}_1^{ij}\ \tilde{\mathbf{f}}_2^{ij}\ \ldots\ \tilde{\mathbf{f}}_{T_{ij}}^{ij}\right]$ be such that $\tilde{\mathbf{f}}_k^{ij} = \mathbf{f}_k^{ij} - \mu$. The projection into eigenspace is computed as

$$\begin{aligned}\mathbf{Y}_{ij} &= \Phi^T \tilde{\mathbf{S}}_{ij} \\ &= \left[\mathbf{y}_1^{ij}\ \mathbf{y}_2^{ij}\ \ldots\ \mathbf{y}_{T_{ij}}^{ij}\right].\end{aligned} \tag{2}$$

Each $\mathbf{y}_k^{ij}$ is an m-dimensional column feature vector which represents a point in eigenspace (the values are coefficients of the eigenvectors). $\mathbf{Y}_{ij}$ is therefore a manifold representing a sample action. We will refer to the set of all the $\mathbf{Y}$'s as the reference manifolds. Recognition will be performed by comparing the manifold of the new action to the reference manifolds as will be explained in the next section.

4.3 Recognition

As mentioned earlier, recognition is done by comparing the manifold of the test action in eigenspace to the reference manifolds. The manifold of the test action is computed in the same way as described above using the computed eigenvectors at the training stage. In this section, we describe the distance measure that was used for comparison and explain how it is used for classification.

4.3.1 Distance Measure

The computed manifold depends on the duration and temporal shift of the action which should not have an effect on the comparison. Our distance measure can handle changes in duration and is invariant to temporal shifts. Given two manifolds $\mathbf{A} = \left[\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_l\right]$ and $\mathbf{B} = \left[\mathbf{b}_1\ \mathbf{b}_2\ \ldots\ \mathbf{b}_h\right]$, we define

$$d(\mathbf{A}, \mathbf{B}) = \frac{1}{l}\sum_{i=1}^{l} \min_{1 \le j \le h} \left\| \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|} - \frac{\mathbf{b}_j}{\|\mathbf{b}_j\|} \right\| \tag{3}$$

as a measure of the mean minimum distance between every normalized point in $\mathbf{A}$ and every normalized point in $\mathbf{B}$. To ensure symmetry, the distance measure which we use is

$$D(\mathbf{A}, \mathbf{B}) = d(\mathbf{A}, \mathbf{B}) + d(\mathbf{B}, \mathbf{A}). \tag{4}$$

This distance measure is a variant of the Hausdorff metric (we use the mean of minima rather than the maximum of minima) which still preserves metric properties. The invari-

ance to shifts is clear from the expression. In fact, $d(\ ,\ )$ is invariant to any permutation of points since there is no consideration for order at all. This flexibility comes at the cost of allowing actions which are not similar, but somehow have similar feature images in a different order, to be considered similar. The likelihood of this happening, however, is quite low. This approach is similar to phase space approaches where the time axis is collapsed [9]. The temporal order in our case is not completely lost, however. The feature image representation has an implicit locally temporal order specification. This measure also handles changes in the number of points as long as the points are more or less uniformly distributed on the manifold. The normalization of points in equation (3) is effectively an intensity normalization of feature images.

4.3.2 Classification

Using the distance measure equation (4), three different classifiers have been considered:

1. Minimum Distance (MD): The test manifold is classified as belonging to the same action class the nearest manifold belongs to, over all reference manifolds. This requires finding the distance to every reference manifold.
2. Minimum Average Distance (MAD): The mean distance to reference manifolds belonging to each action class is calculated; and the shortest distance decides classification. This also involves finding the distance to every reference manifold.
3. Minimum Distance to Average (MDA) (also called nearest centroid): For each action, the centroid of all reference manifolds belonging to that action is computed. This is also a manifold with a number of points equal to the average number of points in each reference manifold belonging to the action. We do not interpolate to compute this manifold. Instead, the nearest points (temporally) on the reference manifolds are averaged to compute the corresponding point on the centroid manifold. A test manifold is classified as belonging to the action class with the nearest centroid. Testing involves calculating a number of distances equal to the number of action classes.

5 Action Data

5.1 Data Selection

To evaluate our recognition method, we recorded video sequences of eight actions each performed by 29 different people. Several frames from one sample of each action are shown in Figures 6 and 7. The actions are named as follows: Walk, Run, Skip, Line-walk, Hop, March, Side-walk, Side-skip. There are several reasons for our choice of this particular data set:

1. Discrimination becomes more challenging when there is a high degree of similarity

Figure 6 Several frames from Walk, Run, Skip, and March actions.

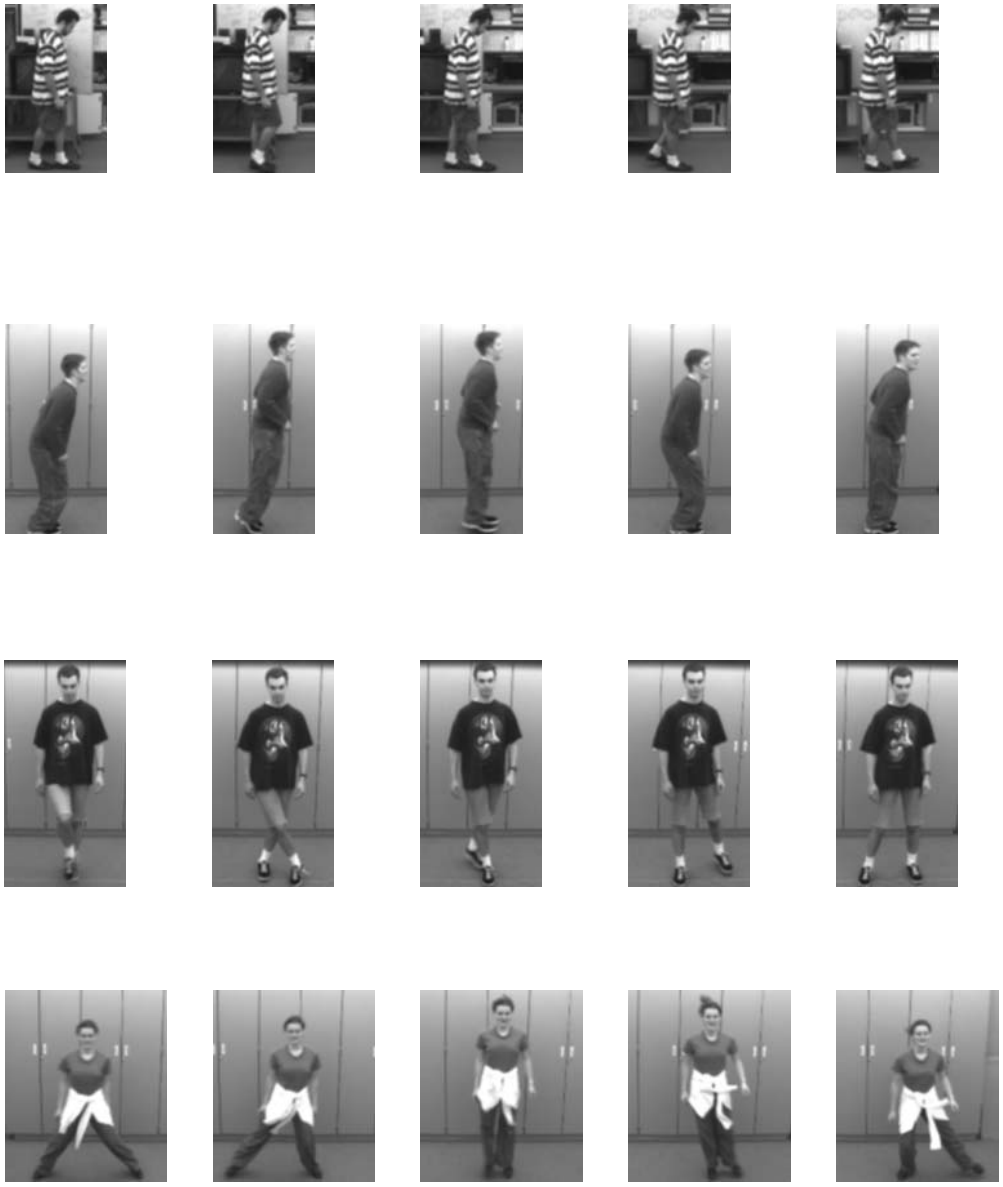


Figure 7 Several frames from Line-Walk, Hop, Side-walk, Side-skip actions.

among actions. Many of the actions we chose are very similar in the sense that the limbs have similar motion paths.

2. Rather than having a single person perform actions several times, we chose to have many different people. This provides more realistic data since, in addition to the fact that people have different physical characteristics, they also perform actions differently both in form and speed. Thus, it tests the versatility of our approach. It can be seen from Figures 6 and 7 that subject size and clothing are different. A few samples also had more complex backgrounds. Table 1 shows the variation in action performance speed throughout the data set. The table shows that the actions were performed at significantly varying speeds (more than double the speed in the case of Hop, for instance).

Action	Minimum Duration		Maximum Duration	
	sec.	# frames	sec.	# frames
Walk	0.93	28	1.77	53
Run	0.70	21	0.93	27
Skip	1.10	33	1.73	51
March	1.13	33	1.93	57
Line-walk	1.47	44	2.20	66
Hop	0.70	21	1.67	50
Side-walk	1.06	31	1.80	54
Side-skip	0.57	17	0.93	27

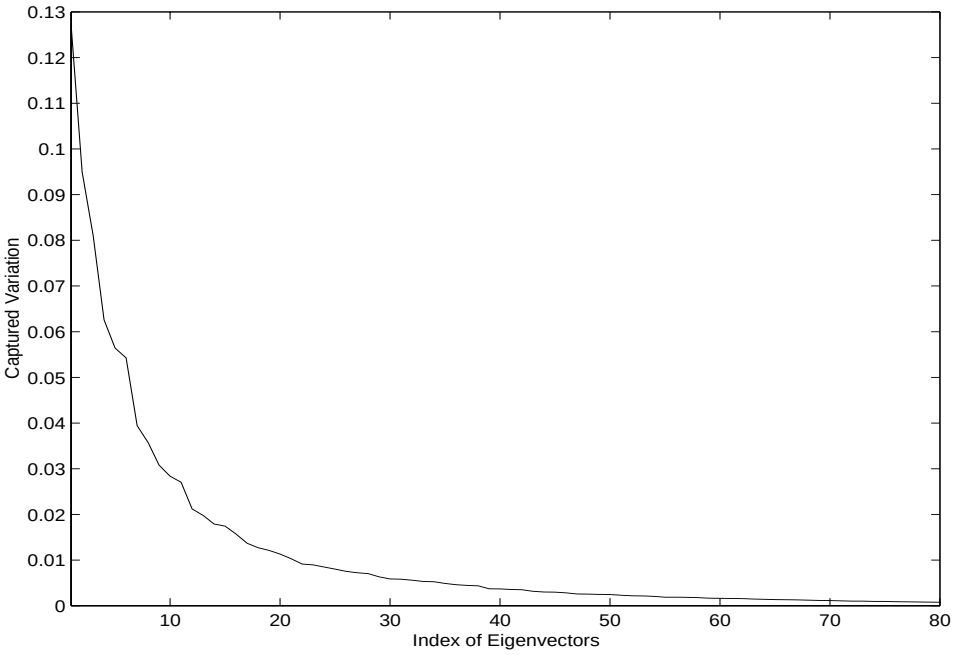
Table 1. Variation in the duration of one cycle for the data set.

3. Another consideration for a more realistic data set was that we avoided the use of a treadmill. Using a treadmill not only restricts speed variation but also simplifies the problem since the background is static relative to the actor.

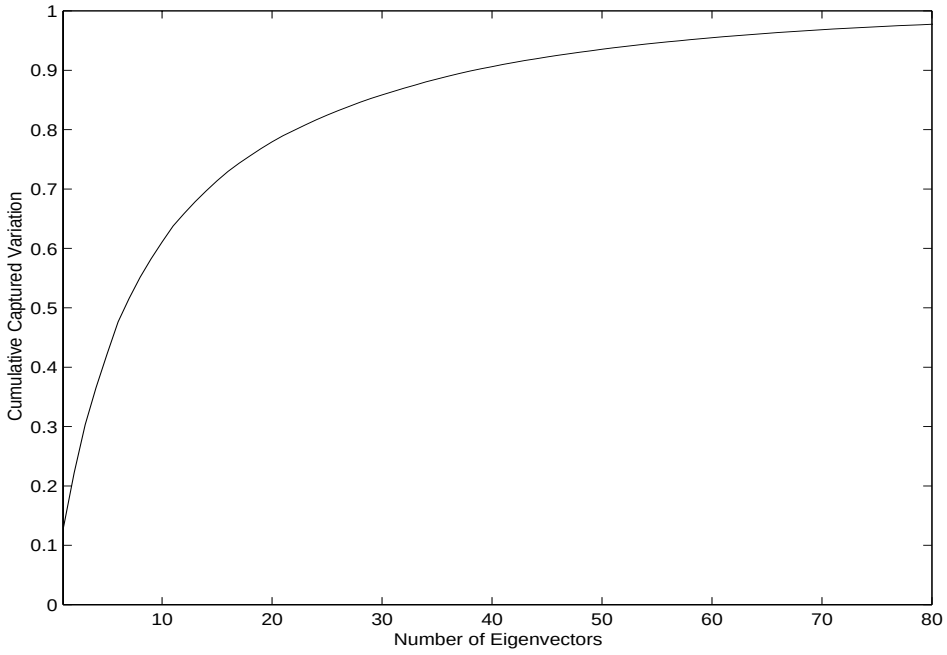
To our knowledge, this is one of the largest sets of action data ever used in terms of the number of subjects performing the actions multiplied by the number of actions.

5.2 Acquisition

The video sequences were recorded using a single stationary monochrome CCD camera mounted in such a way that the actions are performed parallel to the image plane.



(a)



(b)

Figure 8 Eigenvectors contribution to variation in data. (a) Individual contribution. (b) Cumulative contribution.

In our approach, we assumed that the height (in the image plane) and location of the person performing the action are known. Recovering location is necessary to ensure that the person is in the center of the feature images. Height is used for scaling the feature images to handle differences in subject size and distance from the camera. To attain the recovery of these parameters, we tracked the subjects as they performed the action. Background subtraction was used to isolate the subject. A simple frame-to-frame correlation was used to precisely locate the subject horizontally in every frame. A small template corresponding to the top third of the subject's body (where little shape variation is expected) was used. The height was recovered by calculating the maximum blob height across the sequence. For the general case, our tracking method in [27,28] can be used to locate the subject boundaries. Correlation can then be applied to find the exact displacement across frames. The computation of feature images as explained in Section 3 deals with the raw image data without any knowledge of the background. The information provided by the acquisition step is the location of the person throughout the sequence and the person's height.

6 Experimental Results

6.1 Classification Experiment

In our experiments, we used the data for eight of the 29 subjects for training (64 video sequences). This leaves a test data set of 168 video sequences performed by the remaining 21 subjects. The training instances were used to obtain the principle components. The number of selected frames (parameter L is described in Section 4.2) was arbitrarily set to 12. We will later show the effect of changing this parameter in further experiments. The resolution of feature images was also arbitrarily set to 25 horizontal pixels by 31 vertical pixels. Again, the effect of changing the resolution will be shown later. Decreasing the resolution has a computational advantage but reduces the amount of detail in the captured motion.

The training samples were organized in a matrix $\mathbf{X}$ as described in Section 4.2. The number of columns is $asL = 8 \times 8 \times 12 = 768$. The number of rows is equal to the image size ($n = 25 \times 31 = 775$). The eigenvectors are then computed for the covariance matrix of $\mathbf{X}$. Most of the 775 resulting eigenvectors do not contribute much to the variation of the data. The plot $\lambda_i / \left(\sum_{k=1}^{n} \lambda_k \right)$ in Figure 8(a) illustrates the contribution of each eigenvector. It can be seen that past the 50th eigenvector, the contribution is less than

0.5%. Figure 8(b) shows the cumulative contribution $\left(\sum_{k=1}^{i}\lambda_k\right)/\left(\sum_{k=1}^{n}\lambda_k\right)$. The curve increases rapidly during the first eigenvectors. The first ten eigenvectors alone capture more than 60% of the variation. The first 50 capture more than 90%. In Figure 9, the first

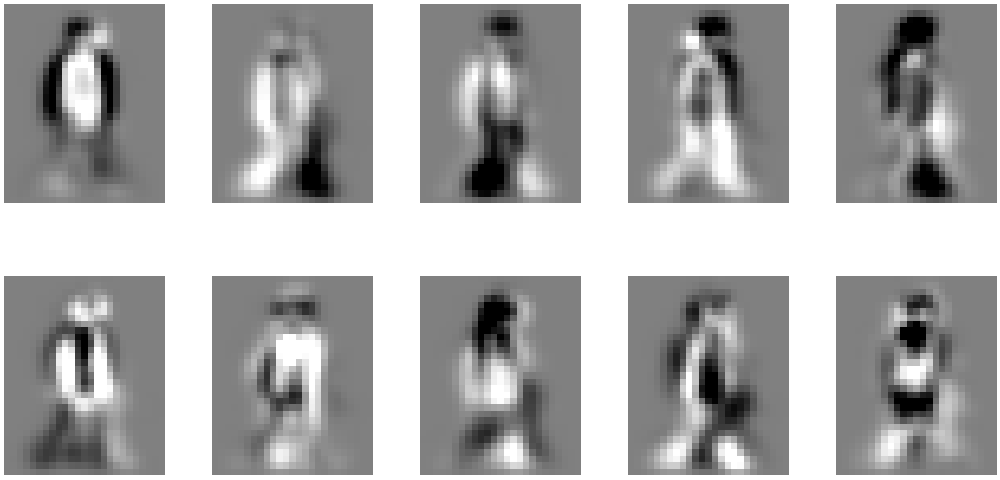


Figure 9 The first ten eigenvectors.

ten eigenvectors are shown. The gray region corresponds to the value of 0 while the darker and brighter regions correspond to negative and positive values, respectively. It can be seen from the figure that different eigenvectors are tuned to specific regions in the feature image.

In our experiments, the choice of m (the number of eigenvectors to be used) was varied from 1 to 50. Using a small m is computationally more efficient but may result in a low recognition rate. As m increases, the recognition rate is expected to improve and approach a certain level. Recognition was done on the 168 test sequences as described in Section 4.3 using all three classifiers (MD, MAD, MDA). Recognition rate was computed as the percentage of the number of samples classified correctly with respect to the total number samples. Figure 10 displays the recognition performance for the different classifiers as a function of m. It can be seen that the recognition rate rises rapidly during the first few values of m. At $m = 14$, the rate using MDA reaches over 91.6%. At $m = 50$, the rate is over 92.8% for MDA. MAD performance is slightly lower while MD is about 10% below. One explanation for this behavior is that some clusters are close to each other so that a point, which may be classified correctly using MDA, can be misclassified using MD. In

later experiments, only the MDA classifier will be shown.

Table 2 shows the confusion matrix for $m = 50$. Most actions had a perfect or near perfect classification except for the Skip action. Although the Skip action was classified correctly about 70% of the time, it was mistaken with Walk, March, and Hop actions numerous times. The 12 misclassified actions are shown in Figure 11. One person (number 15) had two actions misclassified while the remaining people had at most one misclassification. When the correct action class was allowed to be within the first two choices, the number of misclassified actions became five. All these five actions (mostly Skip actions) were either executed erroneously or had a very low color contrast.

To give an indication of the quality of classification, Figure 12 shows a confusion plot which represents the distance among test and reference actions averaged across all subjects. The larger the box size, the smaller the distance it represents. The diagonal in the figure stands out and very few other boxes come near the sizes of the boxes at the diagonal. However, it can be seen that there is mutual proximity in matching between Walk and Skip

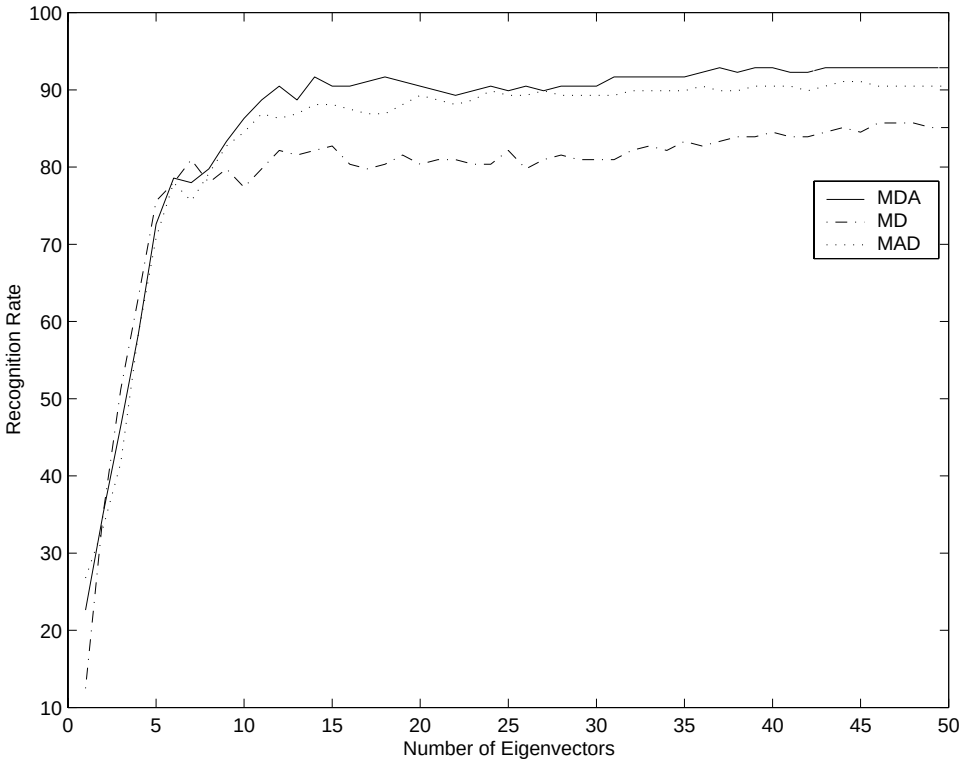


Figure 10 Recognition performance.

	Walk	Run	Skip	March	Line−walk	Hop	Side−walk	Side−skip
1				1				
2							8	
3			1					
4								
5	5							
6								
7			1					
8								
9			6					
10								
11			6					
12							5	
13								
14								
15		1	4					
16								
17								
18				3				
19								
20								
21			4					

Figure 11 Misclassified actions for each subject. The numbers indicate the actions that were chosen incorrectly (1=Walk, 8=Side-skip).

actions (a Walk action is close to a Skip action and vise-versa). This was expected due to the high degree of similarity between these two actions.

6.2 Parameter Selection

6.2.1 Resolution

The resolution of feature images decides the amount of motion detail captured. In size normalization of feature images, a certain resolution must be chosen. Figure 13 shows an example feature image and feature images normalized at different resolutions. The classification experiment was run with different resolutions to see if there is a resolution beyond which little or no improvement in performance is gained. Such a reduced resolution has computational benefits. It also gives an indication of the smallest “useful” resolution which can be used to decide the maximum distance from the camera at which action can take place (assuming the camera parameters are known). In Figure 14, the classification performance is shown for different resolutions. It can be seen from the figure that increas-

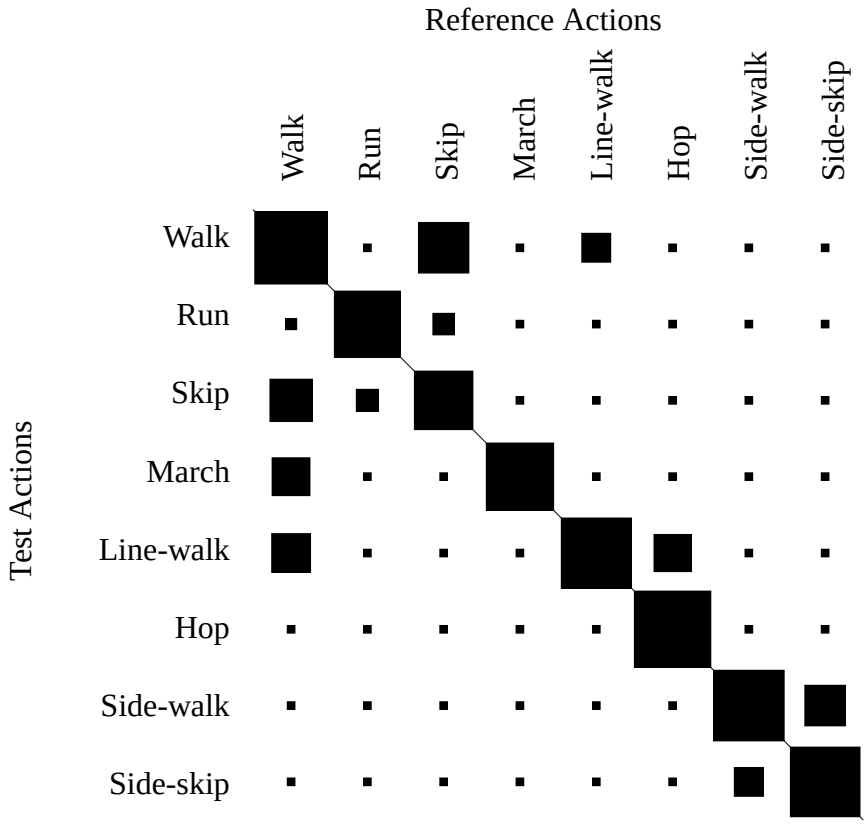


Figure 12 Confusion plot. The area of the squares indicates the distance using the distance measure in Section 4.3.1. The distances are averaged over all test samples.

ing the resolution beyond 25×31 does not produce any gain in performance.

6.2.2 Number of Training Images

The parameter L is used in the training process to select the same number of feature images from every training action sequence. Let us examine the effect of choosing different values for L on performance. Figure 15 shows the classification results for the values: 1, 2, 3, 4, 6, 12, 18, and 24. Values of 3 and above seem to have identical performance. This suggests that three feature images from an action sequence capture most of the variation in the different postures.

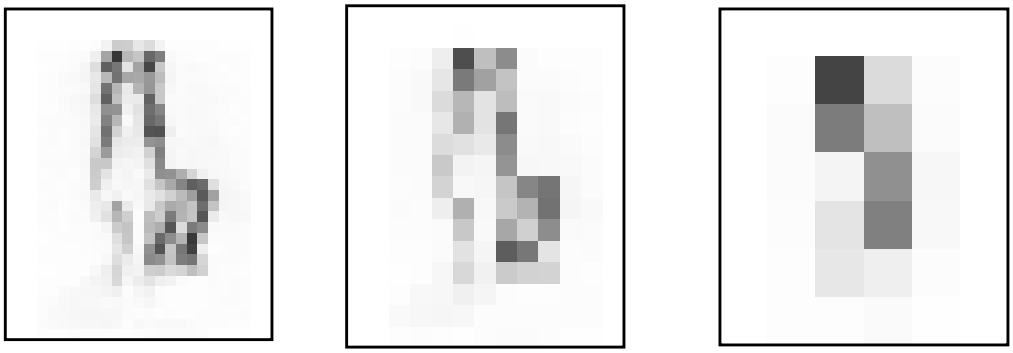


Figure 13 Original frame and its feature images at different resolutions.

6.3 Complexity

Testing an action involves computing feature images, projecting them in eigenspace, and comparing the resulting manifold with the reference manifolds. Computing feature images requires low level image processing steps (addition and scaling of images) which can be done efficiently. Let n be the number of pixels in the scaled feature image according to the selected resolution. Using m eigenvectors, projecting a feature requires an inner product operation with each eigenvector and thus, a complexity of $O(mn)$. If the action

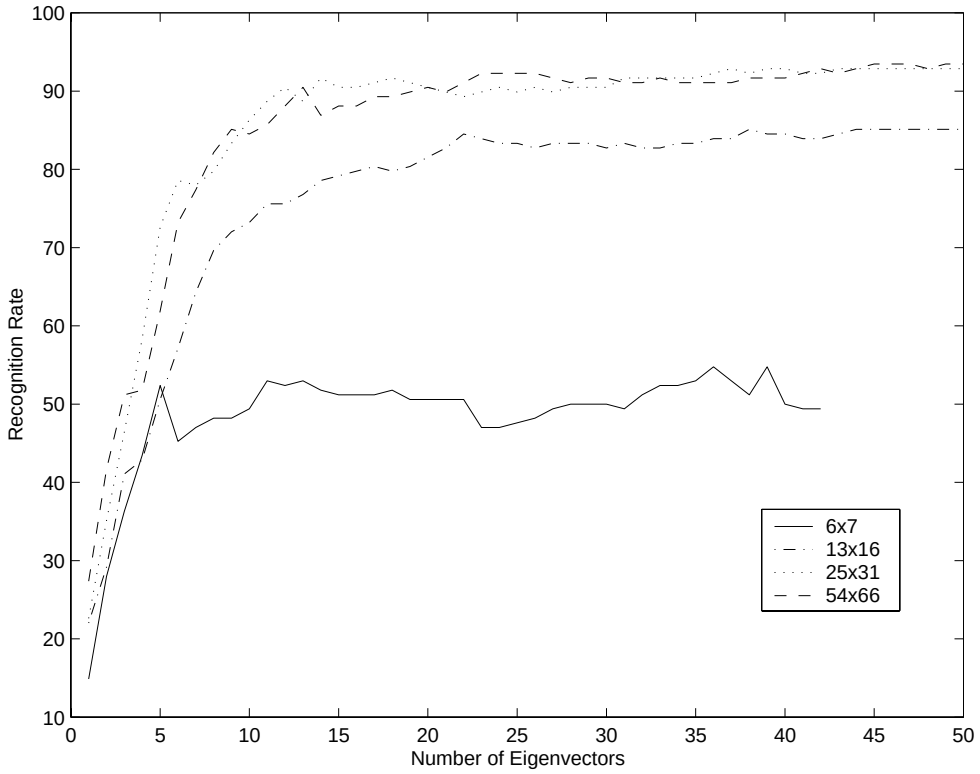


Figure 14 Effect of resolution on classification performance (L is fixed at 12).

has l frames, the time needed to compute the manifold is $O(lmn)$. Manifold comparison involves calculating the distance between every point on the action manifold and every point on every reference manifold. Assuming there are a action classes with s samples of each, and if the average length of the reference actions is T, there will be $asTl$ distance calculations in the case of MD and MAD, and aTl calculations in the case of MDA. Calculating a distance between two points in an m-dimensional eigenspace is $O(m)$. Therefore, recognizing an action using MD or MAD is $O(asTlm)$ while in the case of MDA, it is only $O(aTlm)$. In our experiments, $a = 8$, $s = 8$, $T \approx 37$, $m = 50$, and $n = 25 \times 31 = 775$.

The total complexity for MDA is therefore, $O(lmn) + O(aTlm)$, or $O(l)$ since the remaining variables are constant. This demonstrates the efficiency of this method and its suitability for a real-time implementation. On-line implementation is also possible where the distance measure is updated upon receiving new frames, requiring a small number of

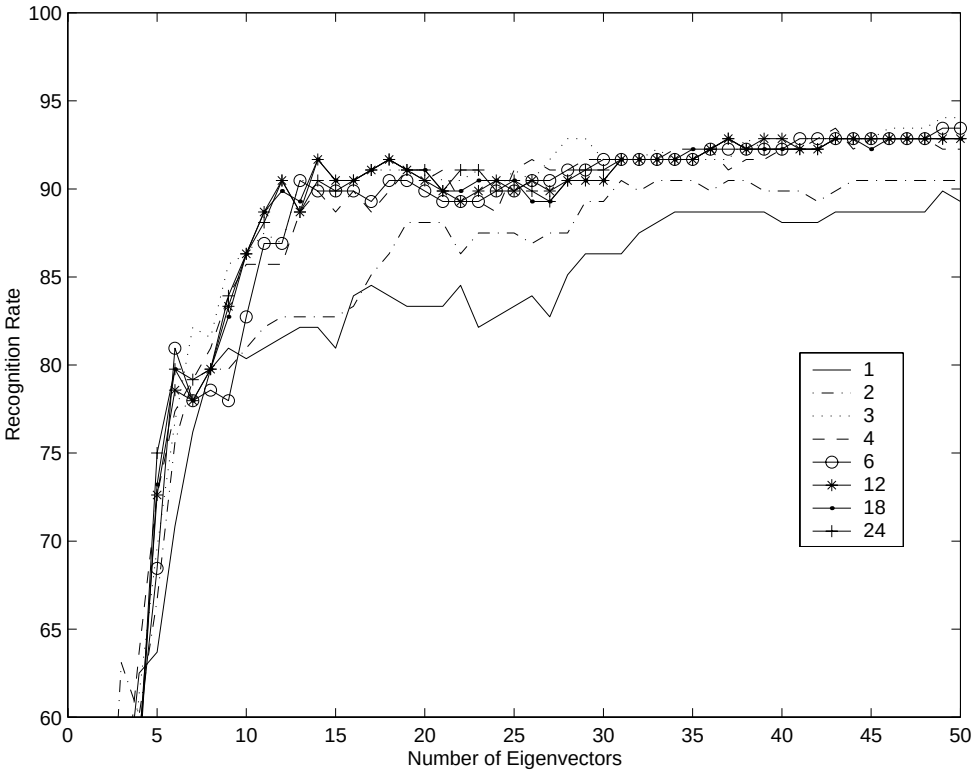


Figure 15 Classification performance for different L values (resolution is fixed at 25×31).

comparisons per frame. This allows incremental recognition such that certainty increases as more frames are available. The choice of the implementation approach depends on the application at hand. We left the actual real-time implementation as future work.

6.4 Other Choices for Feature Images

We have experimented using feature images computed in a different way than recursive filtering. Silhouettes, which are defined to be the binary mask of the foreground, were one choice. Classification results using silhouettes were approximately 20% lower than recursive filtering. When recursive filtering was applied to silhouettes, classification rates went up by about 10%. Our explanation for this behavior is that silhouettes alone do not carry any motion information, except for the spatial aspects of motion (e.g., the way a marching person should look like when his/her knee is at a right angle with his/her body). Recusively filtered silhouettes on the other hand encode some motion aspect but they miss others (e.g., the motion of an arm swinging in front of one's body). Our feature images do a

Action	Walk	Run	Skip	March	Line-walk	Hop	Side-walk	Side-skip
Walk	20	0	0	0	1	0	0	0
Run	1	20	0	0	0	0	0	0
Skip	2	0	15	2	0	2	0	0
March	1	0	1	19	0	0	0	0
Line-walk	0	0	0	0	21	0	0	0
Hop	0	0	0	0	0	21	0	0
Side-walk	0	0	0	0	1	0	19	1
Side-skip	0	0	0	0	0	0	0	21

Table 2. Confusion matrix.

better job than silhouettes because they encode even more motion specific information. Optical flow could probably provide an even better representation but we left that as future work.

7 Summary and Future Work

This paper describes an articulated motion recognition approach. The approach is based on low level motion features which can be efficiently computed using an IIR filter. Once computed, motion features at every frame, which we call feature images, are compressed using PCA to form points in eigenspace. An action sequence is thus mapped to a manifold in eigenspace. A distance measure was defined to test the similarity between two manifolds. Recognition is performed by calculating the distances to some reference manifolds representing the learned actions. Experimental results for a large data set (168 test sequences) were presented and recognition rates of over 92.8% have been achieved. Complexity was also analyzed. The results demonstrate the promise and efficiency of the proposed approach.

There are several possible future directions. One has to do with view independence. In particular, the effect of deviation from fronto-parallel views on performance needs to be tested. Our group is also investigating image-based rendering techniques to either produce novel views for training or to produce fronto-parallel views for testing.

Another direction is concerned with the nature of actions. In this work, we only worked with periodic actions. We plan to investigate the performance with non-periodic actions.

One difficulty with non-periodic actions is temporal segmentation. It is non-trivial to decide the start and end of such actions. In the case of periodic actions, temporal segmentation is possible [4] but temporal alignment (i.e., making sure that the extracted cycle starts at a specific phase) is also non-trivial. In our experiments, we assumed that only temporal segmentation is available (but not temporal alignment). For non-periodic actions, temporal segmentation and alignment become the same problem since there is no longer a concept of a cycle. One possible solution that will completely remove the temporal segmentation requirement for non-periodic as well as periodic actions is online recognition. Basically, at every time instant, we can consider the past m frames where m varies from 1 to some maximum number of frames. For every m, we can try to find a match and when a good match (above some threshold) is found, the system will output that match for that time instant. This is closely related to our interest in utilizing the efficiency of this approach to develop a real-time system that will classify actions as they are captured.

8 Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This work has been partially supported by the Minnesota Department of Transportation and the National Science Foundation through grants #CMS-0127893 and #IIS-0219863.

9 References

[1] Akita K, Image sequence analysis of real world human motion, Pattern Recognition, 17(1) (1984) 73-83.

[2] Azarbayejani A and Pentland A, Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features, in Proc. of International Conference on Pattern Recognition, Vienna (1996).

[3] Belhumeur P, Hespanha J, and Kriegman D, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," IEEE Transactions on Pattern Recognition and Machine Intelligence, 19(7) (1997) 711-720.

[4] BenAbdelKader C, Cutler R, and Davis L S, Motion-based recognition of people in eigengait space, 5th International Conference on Automatic Face and Gesture Rec-

ognition, 2002.

[5] Bobick A, Davis J, Intille S, Baid F, Campbell L, Ivanov Y, Pinhanez C, Schutte A, and Wilson A, KIDSROOM: Action recognition in an interactive story environment, MIT Media Lab Perceptual Computing Group Technical Report No. 398, MIT (December 1996).

[6] Bregler C, Learning and recognizing human dynamics in video sequences, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (June 1997)

[7] Bregler C and Mallik J, Tracking people with twists and exponential maps. in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (June 1998) 8-15.

[8] Cai Q and Aggarwal J K, Tracking human motion using multiple cameras, in Proc. of the 13th International Conference on Pattern Recognition (1996) 68-72.

[9] Campbell L and Bobick A, Recognition of human body motion using phase space constraints, in Proc. of International Conference on Computer Vision, Cambridge (1995) 624-630.

[10] Cedras C and Shah M, Motion-based recognition: a survey, Image and Vision Computing, vol. 13, no. 2, pp. 129-155, March 1995.

[11] Cutting J E and Kozlowski L T, Recognizing friends by their walk: Gait perception without familiarity cues, Bull. Psychonometric Soc., 9(5) (1977) 353-356.

[12] Davis J W and Bobick A F, The representation and recognition of human movement using temporal templates, in Proc. of IEEE Computer Vision and Pattern Recognition (1997) 928-934.

[13] DiFranco D E, Cham T J, and Rehg J M, Reconstruction of 3-D figure motion from 2-D correspondences, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (June 2001) 307-314

[14] Dittrich W H, Action categories and the perception of biological motion, Perception 22 (1993) 15-22.

[15] Foster J P, Nixon M S, and Prugel-Bennet A, New area based metrics for automatic

gate recognition, in Proc. BMVC (2001) 233-242.

[16] Gavrila D M, The visual analysis of human movement: a survey, Computer Vision and Image Understanding, vol. 73, no. 1, pp. 82-98, January 1999.

[17] Gavrila D M and Davis L S, 3-D model-based tracking of humans in action: a multi-view approach, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco (1996) 73-80.

[18] Goddard N, Incremental model-based discrimination of articulated movement direct from motion features, in Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin (1994) 89-94.

[19] Guo Y, Xu G, and Tsuji S, Understanding human motion patterns, in Proc. of the 12th IAPR International Conference on Pattern Recognition (1994) 325-329.

[20] Halevi G and Weinshall D, Motion of disturbances: detection and tracking of multi-body non-rigid motion, in Proc. of IEEE Conference Computer Vision and Pattern Recognition, Puerto Rico (June 1997) 897-902.

[21] Huang P S, Harris C J, and Nixon M S, Human gait recognition in canonical space using temporal templates, IEEE Proc. VISP 14(2) 1999 93-100.

[22] Johansson G, Visual perception of biological motion and a model for its analysis, Perception and Psychophysics 14(2) (June 1973) 201-211.

[23] Johansson G, Visual motion perception, Sci. Amer. 232 (June 1976) 75-88.

[24] Ju S, Black M, and Yacoob Y, Cardboard people: A parameterized model of articulated image motion, in Proc. of IEEE International Conference on Automatic Face and Gesture Recognition, Killington (1996) 38-44.

[25] Kozlowski L T and Cutting J E, Recognizing the sex of a walker from dynamic point–light displays, Perception and Psychophysics 21(6) (1977) 575-580.

[26] Krahnstover N, Yeasin M, and Sharma R, Towards a unified framework for tracking and analysis of human motion, in Proc. of IEEE Workshop on Detection and Recognition of Events in Video (2001) 47-54.

[27] Masoud O, Tracking and Analysis of Articulated Motion with an Application to Human Motion, Ph.D. Thesis, Department of Computer Science and Engineering, University of Minnesota (2000).

[28] Masoud O and Papanikolopoulos N, A novel method for tracking and counting pedestrians in real-time using a single camera, IEEE Transactions on Vehicular Technology 50(5) (2001) 1267-1278.

[29] Myers C, Rabiner L, and Rosenberg A, Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, IEEE Transactions on ASSP 28(6) (1980) 623-635.

[30] Nayar S K, Nene S A, and Murase H, Real-time 100 object recognition system, in Proc. of IEEE Conference on Robotics and Automation, 3, Minneapolis (1996) 2321-2325.

[31] Pavlovic V and Rehg J, Impact of dynamic model learning on classification of human motion, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (June 2000) 788-795

[32] Polana R and Nelson R, Detecting activities, Journal of Visual Communication and Image Representation 5(2) (1994) 172-180.

[33] Polana R and Nelson R, Detection and recognition of periodic, nonrigid motion, International Journal of Computer Vision 23(3) (1997) 261-282.

[34] K Rangarajan, Allen W, and Shah M, Matching motion trajectories using scale space, Pattern Recognition 26(4) (1993) 595-610.

[35] Swets D L and Weng J, Using discriminant eigenfeatures for image retrieval, IEEE Transactions on Pattern Recognition and Machine Intelligence 18(8) (1996) 831-836.

[36] Turk M and Pentland A, Eigenfaces for recognition, Journal of Cognitive Neuroscience 13(1) (1991) 71-86.

[37] Wang J, Lorette G, and Bouthemy P, Analysis of human motion: a model-based approach, in Proc. 7th Scandinavian Conference on Image Analysis, Aalborg

(1991).

[38] Wren C R, Azarbayejani A, Darrell T, and Pentland A, Pfinder: real-time tracking of the human body, in Proc. of the Second International Conference on Automatic Face and Gesture Recognition (October 1996) 51-56.

[39] Yacoob Y and Black M J, Parameterized modeling and recognition of activities, Journal of Computer Vision and Image Understanding 73(2) 232-247.

[40] Yamato J, Ohya J, and Ishii K, Recognizing human action in time sequential images using Hidden Markov Model, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (1992) 379-385.