

1998-12-04

Recognition of Human Action Using Moment-Based Feature

Rosales, Romer. "Recognition of Human Action Using Moment-Based Features", Technical Report BUCS-1998-020, Computer Science Department, Boston University, December 4, 1998.

[Available from: <http://hdl.handle.net/2144/1776>]

<https://hdl.handle.net/2144/1776>

Downloaded from DSpace Repository, DSpace Institution's institutional repository

Recognition of Human Action Using Moment-Based Features

Rómer Rosales
Computer Science Department
Boston University
Boston, MA 02215

Abstract

The performance of different classification approaches is evaluated using a view-based approach for motion representation. The view-based approach uses computer vision and image processing techniques to register and process the video sequence [6, 23]. Two motion representations called Motion Energy Images and Motion History Images [6] are then constructed. These representations collapse the temporal component in a way that no explicit temporal analysis or sequence matching is needed. Statistical descriptions are then computed using moment-based features and dimensionality reduction techniques. For these tests, we used 7 Hu moments, which are invariant to scale and translation. Principal Components Analysis is used to reduce the dimensionality of this representation. The system is trained using different subjects performing a set of examples of every action to be recognized. Given these samples, K-nearest neighbor, Gaussian, and Gaussian mixture classifiers are used to recognize new actions. Experiments are conducted using instances of eight human actions (i.e., eight classes) performed by seven different subjects. Comparisons in the performance among these classifiers under different conditions are analyzed and reported. Our main goals are to test this dimensionality-reduced representation of actions, and more importantly to use this representation to compare the advantages of different classification approaches in this recognition task.

1 Introduction and Motivation

Classifying the motion of non-rigid objects has been a very challenging and important problem in computer vision. When motion is described at a low level, using just image processing techniques for example, it is necessary to use a very high dimensional space to represent it. This is the source of many of the key difficulties encountered when analyzing visual motion. Methods to represent motion in low-dimensional spaces are therefore desirable. An important requirement is that these methods should be able to maintain, at maximum, the power of higher dimensional descriptions. It is then essential to count on experimental results that support their efficiency and demonstrate their power.

The importance of motion recognition problems is evidenced by the increasing attention they have received in recent years [24, 25, 1, 16, 20, 19]. One of the main areas of research is the analysis of humans in motion. There are numerous domains that motivate the research in this area: video surveillance, human-computer interaction, athletics, dance, robot motion, among others.

Motion of objects is much more semantically rich than static configurations of scene components, therefore the importance and necessity of its analysis. However, in this area it is not easy, very often, to limit the problem, establish categories, and evaluate the performance. This is mainly due to the difficulty to uniquely define or accurately describe actions without having to exhibit the action itself [5].

We deal with the general problem of non-rigid motion recognition. For this work we only focus on a small set of human actions (8 in total). In computer vision, it is very important to be able to give a high level interpretation of the motions and classify them using only the perceived motion from the video sequence. Our goals are to test a new representation of actions using dimensionality reduction, and more importantly to use this representation to compare the advantages of different classification approaches in this recognition task.

The classification approaches used here are: K-nearest neighbor with $K = 1, 3, 5$, Gaussian classifier, and Gaussian mixture classifiers. For the mixture classifier, besides using a hand-picked number of modes, the Minimum Description Length (MDL) of the classifier is also used to determine the *best* probability density representation. MDL is an information-theoretic approach based on a measure that determines the best balance between the number of parameters used and the performance achieved in classification.

It is clear that each classifier has a different level of complexity, but it is not necessarily true that the more complex it is, the better it is at classifying. The complexity of the task that each classifier can handle is related to the particular characteristics of the problem. Among other things, we expect to obtain a depiction of the source of the complexity of human action recognition tasks and the best approach (among those tested) that can be used for recognition.

The organization of this report is as follows: Sec. 2 introduces previous work done in the area and relevant approaches to the present work, Sec. 3 gives a brief overview of the method, Sec. 4 and 5 explains the way actions are represented, and how features are extracted from it. Sec. 6 gives a detailed exposition of the techniques used and motivation for our choices. Sec. 7 describes the general experimental settings. Sec. 8 shows the details of each experiment, the results, and a brief discussion about them. Sec. 9 presents some conclusions and discusses some possible extensions.

2 Background and Previous Work

One of the fundamental ideas in motion perception is the work of Johansson's moving light displays [14], where it was demonstrated that relatively little information (motion of a set of selected points on the object) is needed to perform motion recognition tasks. Perhaps one of the first approaches related to watching people in real environments is due to Hogg [12].

Among the seminal computer vision ideas toward motion recognition was the work of [17], whose data consisted of synthetic images and constraint satisfaction techniques were employed. Current technology in motion recognition can be divided into several groups of approaches. Here we described some of the most relevant to us. An excellent review on motion understanding is given by [24].

One of these groups is based on fitting explicit structural models such as [11, 22, 10, 21, 15]; 2D or 3D reconstruction is normally needed. There is an overlapping

group of techniques that achieve motion understanding by recognizing sequences of static configurations, although some of them may not require explicit reconstruction of a knowledge model, for example [5, 11, 21, 22].

The most relevant group of approaches related to ours, as in [6], are those that try to achieve recognition of the motion directly from the sequence of images. No explicit use of the static images is done besides what is needed to represent the motion [20, 2, 6]. We give a brief overview of their fundamentals.

According to [20], recognition of repetitive motion can be achieved on the basis of bottom up processing, without identifying specific parts or classification of the object. Their approach needs to estimate local velocity to undo the translation and looming to make the objects stationary (with respect to translation). They used a spatiotemporal template to match the motion. In order to recognize the motion (which is required to be periodic) they compute the feature vector of motion magnitudes and compare it with the patterns. Real-time implementations required 4 processors using 128x64 pixel images.

For [2], the main goal was to recognize human facial expressions as a dynamic system by considering the motion on particular patches of the face and using that motion directly to represent instances of facial expressions.

Davis [6] used a view-based technique to represent and recognize actions. Motion History Images (MHI) and Motion Energy Images (MEI) are used. The first represents the recency of motion using intensity, the second represents where the motion occurred. They presented a method for recognition of temporal templates. These templates are matched using a nearest neighbor approach against examples of given motions already learned.

The main restrictions of this method are the requirement that the objects do not undergo any global translational motion (as perceived in the image plane), its unsuitability for dealing with multiple objects and the insufficiency of the representation to discriminate among more or less similar motion representations of different actions or views. This work is the most relevant to ours in the sense that it uses MHI's, MEI's, and moment based features to represent and classify actions using one of the classification approaches here tested (1-nearest neighbor).

We avoid the object-stationarity requirement and also allow for multiple motions by using an object registration technique developed by the authors [23]. This technique allows for tracking and 3D trajectory recovery of multiple objects. We use the segmentation and object centered representations obtained using this technique as a front-end for motion analysis.

3 Overview and Description of the Approach

Using computer vision and image processing techniques[6, 23], we plan to recognize specific actions performed by people, given a set of predefined actions. Our settings assume a static background, a fixed camera, no major occlusion between people performing actions or between the subject and an external agent. Our last assumption is relaxed in Sec. 8.3 by the use of [23]. Due to space restrictions, this system is not described here.

A 2D view-based approach is going to be used. The different motions are going

to be described by templates in which the temporal component is embedded in the representation (as fully explained in next sections) in a way that no explicit temporal analysis or sequence matching is needed. In this way, the feature vector of every action is a function of their motion properties, but temporal properties have been partially dropped.

Because our approach is sensitive to view, (*i.e.*, different views of the same motion produce different feature vectors), we need to either 1) construct view-specific representations of the actions and recognize both view and action using a statistical model; or 2) constrain our implementation to deal with the problem of motion recognition by itself given a specific view (*i.e.*, we assume the view is known).

Because of the cost of training (we would need to have many cameras around the person performing the action and capture the 2D features of the motion at each view), and also because a single view is enough to test the potential of the different approaches, we have chosen to work mainly on 2). However, in order to test the discriminating power of the system when considering different views of the same action, one of the actions to be recognized is presented in 2 different views. Using many views is a straightforward extension of the single view implementation, although it may provide more accurate recognition at the cost of more detailed representations.

We now give a very brief overview of our approach. A set of human actions directly taken from video sequences are labeled by a user using the registration system designed for this work. For each action the user needs to specify its beginning, end, and the class it belongs to. Data from different subjects in different conditions were collected and labeled.

The system generates representations of the actions that consist of describing where the motion occurs and what are the temporal properties of it, this is done using two image-based representations: Motion Energy Images and Motion History Images, already used by [6] in a similar recognition task. These images are functions of the motion properties of the given actions where the temporal components have been embedded in a static vector representation.

This high dimensional representation is then processed using their statistical properties to generate a set of $d = 2 * 7$ moment based features that are invariant to translation, rotation and scaling. In [6] MHI's and MEI's along with a superset of its Hu-moments are used directly on recognition, here an extra processing step is done: Principal Components Analysis in the Hu-moments-space. Using Principal Components Analysis we reduce the dimensions of our space in a statistically optimal way assuming that classes are Gaussian distributed (*i.e.*, $P(z|\omega_i)$ is Gaussian, with \mathbf{z} to be the feature vector and ω_i the i -th class). The rotated vectors \mathbf{z} are the representation that we use to train the system and classify the actions.

For classification we use and compare different techniques: K-nearest neighbors, Gaussian classifier, and mixture of Gaussian classifiers. Results of different experiments are shown where the performance of these classifiers is described and compared.

4 Action Representation

In order to analyze the motion occurring in a sequence we first need a method that allows us to capture and represent it directly from video, an initial low-level represen-

tation.

We are interested on analyzing action: the motion occurring in a given window of time [6]. We will construct a view specific representation of the motion based on where motion has occurred and what are the temporal characteristics of it. The result of this space-time process will collapse the time dimension to a static depiction of the action.

These static representations are called Motion Energy Images (MEI) and Motion History Images (MHI) [3, 6]. They are functions of the observed motion properties at the corresponding spatial image location in the image sequence.

4.1 Motion Detection

Our input video consists of a sequence of color frames in RGB space. By using a simple linear standard operation, we perform a color space conversion to a gray scale color model.

$$I(x, y, t) = .30I_r(x, y, t) + .59I_g(x, y, t) + .11I_b(x, y, t), \quad (1)$$

where $I_c(x, y, t)$ is the channel c from the input frame at time t .

Our motion detection mechanism is based simply on change detection, the difference between two consecutive incoming frames. This difference is then thresholded to form a binary map that shows where there is a high likelihood of motion being present.

$$D(x, y, t) = |I(x, y, t) - I(x, y, t - 1)| \quad (2)$$

$$B(x, y, t) = \begin{cases} 1 & \text{if } D(x, y, t) > \Gamma \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where $I(x, y, t)$ is the processed frame, $D(x, y, t)$ is the difference image at location x, y at time t , and Γ is a selected threshold.

This low level processing does not necessarily guarantee that the captured motion will represent the motion in which we are particularly interested. In order not to complicate the data acquisition process, we assume a static background or the possibility to separate the motion of the object from that of the camera or other objects. There are computer vision techniques that partially solve these problems [16]. Here we will concentrate more on the recognition of the actions and will simplify the registration process to an acceptable level by assuming a static background.

4.2 Motion Energy Images

A MEI [6] is basically a cumulative motion image. It is a simple but probably useful representation of the observed motion. It indicates the spatial location where the motion occurred, but the time dimension has been fully dropped.

They are calculated as follows,

$$E_\tau(x, y, t) = \begin{cases} 0 & \text{if } B(x, y, t') = 0, t' \in \{t - \tau, \dots, t\} \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

where τ represent the duration of the time window used to capture the motion.



Figure 1: Sitting-on-chair action. Top row shows single frames of the action, middle and bottom row contain their respective MEI's and MHI's calculated up to the given frame. Notice that MEI's are just a thresholded version of the MHI's

4.3 Motion History Images

Temporal characteristics of the motion are obviously important when analyzing actions. Because MEI's drop the temporal component we need a different kind of representation that takes it into account. MHI's [6] characterize the temporal component of the action as follows :

$$H_{\tau}(x, y, t) = \begin{cases} \tau & \text{if } B(x, y, t) = 1 \\ \max(0, H_{\tau}(x, y, t - 1)) & \text{otherwise.} \end{cases} \quad (5)$$

The result is a function of the recency of the motion at every pixel. The brightness of a given pixel is proportional to how recently the intensity changed, presumably as a consequence of motion. An example of the MEI's and MHI's construction from the sitting on chair action (side view) computed from video can be seen in Fig. 1.

Using these images, motion over the sequences is described by a single image vector. A problem related to with this representation is that part of the motion may be lost due to self occlusion or overlapping of motion on the image plane. However it might be representative enough, typically humans could tell what action is being performed given the MHI.

5 Feature Extraction

Our representation space, so far, is very high dimensional, therefore performing recognition on the given space may not be a good decision. Given the representation discussed in the Sec. 4, we need to extract some useful features for classification. We have chosen to use moment-based features, specifically our choice are 7 Hu moments $h = (h_1, h_2, \dots, h_7)$ [13], which have been modified to achieve some useful proper-

ties. The modified set is represented by $m = (m_1, m_2, \dots, m_7)$. These features are computed from the statistical descriptions of the MHI and MEI given an action.

The Hu moments are based on the central moments, Eq. 6 shows how they were adapted to the image space. Reasons for this decision are explained below.

$$\eta_{pq} = \frac{1}{N} \sum_{i=1}^N (u_i I_i - \bar{u})^p (v_i I_i - \bar{v})^q, \quad (6)$$

where N is the number of pixels in the image, u_i and v_i are the row and column numbers for pixel i , I_i is the brightness of pixel i , $\bar{u} = \frac{1}{N} \sum_{i=1}^N (u_i I_i)$.

In a view based-approach some of the desirable properties of any descriptor are its invariability to translation (space in the image where object is represented), rotation, and scaling. The 7 Hu moments are invariant under image translation and rotation. To obtain scale invariance, the definition of the radius of gyration of a planar pattern is used [9]:

$$r = (\eta_{20} + \eta_{02})^{\frac{1}{2}}. \quad (7)$$

The radius is used to normalize h_2, \dots, h_7 to obtain the given vector m which has all the properties of h but now is also invariant to scaling in the camera plane. We compute these features in both the MHI and MEI to obtain \mathbf{x}_{MHI} and \mathbf{x}_{MEI} respectively.

Another advantage of this representation is that it is not too expensive computationally. A disadvantage is the fact that it is difficult to reason about intuitively [6]. The given feature vector will provide a reduction in dimensionality whose discriminating power has already been tested in computer vision to recognize shapes [9, 6].

Dimensionality reduction via Principal Component Analysis (PCA) Given our current description, we define a full 14-dimensional feature vector $\mathbf{x}_f = (\mathbf{x}_{\text{MEI}}, \mathbf{x}_{\text{MHI}})$. Even though the dimensionality of \mathbf{x}_f is very reduced compared to using MEI's and MHI's directly, we ran into empty-space related problems in our preliminary tests when estimating class distributions. Empty-spaces occur when during sampling a given space, according to some probability density function, certain regions do not generate any samples, not because their probability was zero but because of the discrete approximations introduced when sampling. This generally happens when the probability distribution of the given region is close to zero.

In the inverse problem, when estimating a probability density function from samples, empty-spaces decrease the estimation accuracy by indicating zero probability in areas where it is not the case. This problem can be reduced if more samples (data points) are used. We can think of at least two options: 1) if a 14-dimensional vector need to be used, more training data is a solution; 2) the space dimensionality can be reduced, thus decreasing the likelihood of empty-spaces. We propose the use of option 2 which slightly extends the representation proposed by [6].

To approach this problem we use Principal Components Analysis (PCA). The reduction is achieved by solving the well known eigenvalue decomposition problem.

$$\Lambda = \Phi^t \Sigma \Phi, \quad (8)$$

where Φ is the eigenvector matrix of the covariance of the data and Λ is the corresponding diagonal matrix of eigenvalues. Only M eigenvectors are kept corresponding to the M largest eigenvalues, obtaining the matrix Φ_R . Now we define our new feature vector $\mathbf{x} = \Phi_R^t \mathbf{x}_f$. In our experiments $M = 7$. This was determined by noticing that in our experiments with the training data we obtained 90% of the variance of the data using this choice of M .

Using PCA we can be sure of finding the minimum error reconstruction of our whole data based on the new sub-space assuming that the data is Gaussian distributed. Some of the numerical problems in our experiments were solved using this *transformed* space.

6 Recognition Approaches: Description and Discussion

We will evaluate three classification paradigms and some variations on them, namely K-nearest neighbors, Gaussian, and mixture of Gaussian classifiers. Discussions on the theoretical implications directly applied to our classification problem are going to be presented along with their descriptions and our motivations for choosing them.

6.1 Normalized K-nearest Neighbors

Our first approach to recognize actions is based on the k-nearest neighbor (KNN) algorithm. Nearest neighbors were used used in pattern recognition and statistical estimation at least since the beginning of the 70's. The KNN's are simply the k-closest samples from the training data to the new instance \mathbf{x} (according to some suitable metric). The basic idea is that it is reasonable to assume that observations which are close together (according to some appropriate metric) will have the same classification [9]. We will mention some of the reasons that motivated our choice.

An advantage of this technique is a consequence of its non-parametric nature, because of this, we do not need to make any assumptions on the parametric form of the underlying distribution of the classes. These distributions may be erroneous, this is often the case in high dimensional spaces. Therefore, a simpler classifier may give better results. The KNN directly tries to estimate the *a posteriori* probability $P(\omega_j|\mathbf{x})$. If the purpose is classification not probability estimation, it provides the decision function directly.

KNN also provides a simple solution for the problem of choosing the size of the window of data used for non-parametric estimation by letting it be a function of the data. The KNN classifier has the theoretical property that when the amount of training data $n \rightarrow \infty$, $P(\epsilon) < 2P^*(\epsilon)$, where $2P^*(\epsilon)$ is the Bayes error [8].

Another reason why we chose a KNN classifier is its likely good performance in classification even when there is not enough training data to reliably estimate the second order statistics (*i.e.*, means and covariances). This is particularly true in high dimensional feature spaces or when some instances of data useful for training occur rarely in real environments.

The features used (*i.e.*, represented by \mathbf{x}) have the property that their individual spread (variance) differs from each other. If we simply calculate the KNN using a Euclidean distance, then features with higher variance will tend to dominate the distance measure. In other words, features are going to be weighted differently in distance cal-

ulation. A simple solution for this is to normalize each feature by its corresponding variance. This distance measure is commonly called *Mahalanobis distance*:

$$\mathbf{d}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^t \Sigma^{-1} (\mathbf{x} - \mathbf{y}), \quad (9)$$

where statistics are computed from the whole training set:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (10)$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^t \quad (11)$$

6.2 Single Gaussian classifier

Our second method is based on a Bayesian classification criteria, and it is assumed that $P(\mathbf{x}|\omega_i)$ is normally distributed. No prior information is considered, therefore we can use $P(\omega_i) = \frac{1}{n}$, for any of the n classes. In this section, we discuss some of the reasons and implications of this choice.

Perhaps the main theoretical reason for considering a Gaussian distribution has its roots on the central limit theorem: the sum of independent identically distributed random variables has a Gaussian distribution. If the noise in our data is the result of the sum of contributions from a larger number of independent sources, then the central limit theorem allows us to model the total noise by a Gaussian distribution.

In our data, examples of this factors are: variation in lighting conditions, slight perturbations in the orientation of the camera with respect to the moving object, quantization error in the digital device, the possibility of confusing background and foreground when computing the difference images, camera sensor noise, etc. Note that only slight changes in the camera point of view and rotation can possibly be modeled as noise, major changes in general require a learned representation (view specific). Intuitively, we could think about our class as having a perfect feature vector description (e.g., one template), and its variability modeled as noise. It is possible argue against these assumptions, but its use in countless engineering applications has demonstrated its usefulness. In Sec. 6.3 we use a possibly better model to solve the inherent problems derived from this assumption.

Bayes decision theory is based on posing the decision problem in probabilistic terms. It is assumed that all of the relevant probability values are known [8]. The approach is supported by the Bayes' formula, where $P(\omega_j)$ describes the a priori probability of class j :

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(\mathbf{x})} \quad (12)$$

A Bayes classifier is naturally represented in terms of discriminant functions $g_i(\mathbf{x})$, $i = 1, \dots, n$ for n classes. We let $g_i(\mathbf{x}) = -R(\alpha_i|\mathbf{x})$, so that the maximum discriminant function corresponds to the minimum conditional risk. Because we are interested in obtaining the minimum error rate, we can simplify by making $g_i(\mathbf{x}) = P(\omega_i|\mathbf{x})$ [8],

(*i.e.*, the maximum a posteriori probability). We can also replace $g_i(\mathbf{x})$ by $f(g_i(\mathbf{x}))$, for any monotonically increasing function f . We omit the details and present the main results:

$$g_i(\mathbf{x}) = \log p(\mathbf{x}|\omega_i) + \log P(\omega_i) \quad (13)$$

Bayes decision rules are, by definition, optimal. However, if the models that are used turn out to be erroneous, it is sub-optimal, and other procedures may give better results. For our classifier we assumed our features to be mutually independent, yielding a diagonal covariance matrix Σ_i for class i . Different classes may have different distribution parameters given the feature vector \mathbf{x} . Using eq. 13 and the normal distribution equation, we obtain the following discriminant functions for the classifier:

$$g_i(\mathbf{x}) = -((\mathbf{x} - \mu_i)^t \Sigma_i^{-1} (\mathbf{x} - \mu_i) + \log |\Sigma_i|), \quad (14)$$

This resembles a which resembles a simply Mahalanobis distance measure with an extra shifting term. We just need to estimate μ_i and Σ_i based on the training data per class as given in eq. 11.

6.3 Mixture of Gaussian classifier

Our assumption of $P(\mathbf{x}|\omega_i)$ being normally distributed may not be sufficient to give a good approximation of the underlying density distributions. Therefore, we also considered the case for which $P(\mathbf{x}|\omega_i)$ is distributed as a mixture of Gaussians.

The theoretical foundations of this approach are similar to the one in Sec. 6.2, it is correct in a Bayesian sense. Notice that we do not have a verifiable reason to think of a mixture of Gaussian as being more accurate than the previous two approaches. Instead, this will be investigated in our experiments.

Intuitively, our motivation for using a mixture distribution is the fact that instead of every action being described with a single *perfect* temporal template, there may be changes that are not due to noise as defined in Sec. 6.2, but as changes in standard body configurations in humans, (*i.e.*, there is not a single temporal description, but many of them). These variability could be the result of anthropometric variations in the bodies used in our tests, and could possibly be extended to span the space of standard configurations for a given action. Obviously, we are only interested in finding a representative set of these modes. We use a model selection technique to choose the *best number* of parameters to be used.

6.3.1 Parameter estimation using the Expectation Maximization (EM) Algorithm

The EM algorithm [7] is a method for obtaining maximum likelihood parameter estimates when the observed data is incomplete. It is a technique that has been increasingly used in estimation. Here we discuss some of our implementation details.

In estimating the parameters for a Gaussian mixture distribution, we could think of the mixture modes as being unobserved. In our implementation, we assume that the Gaussian covariances are diagonal (*i.e.*, features are mutually independent).

The EM algorithm is based on increasing the expected likelihood of the complete data X given the observed data Y . In our case this can be represented as follows:

$$Q(\Theta|\Theta^{(P)}) = \sum_{i=1}^n \sum_{k=1}^m z_{ik} (\log \lambda_k - \log 2\pi - \log(\prod_{l=1}^d \sigma_{ll}) - \frac{1}{2} \sum_{l=1}^d (y_{il} - \mu_{kl})^2 \sigma_{ll}^{-2}) \quad (15)$$

where $z_{ik} = p(Z_i = k | y_i, \Theta^{(P)})$ and Z_i is an indicator of the mixture model. In the EM equations, n and m are the number of observations and the number of Gaussians used in the mixture respectively. The vector of parameters that define the given distribution is represented with Θ . For the E-step we find the expected likelihood of the complete data as a function of Θ . It basically reduces to finding z_{ik} on the E-step [7]:

$$z_{ik} = \frac{P(y_i | Z_i = k, \Theta^{(P)}) P(Z_i = k | \Theta^{(P)})}{\sum_{j=1}^m P(y_i | Z_i = j, \Theta^{(P)}) P(Z_i = j | \Theta^{(P)})} \quad (16)$$

The M-step re-estimates the parameters such that $Q(\Theta|\Theta^{(P)})$ is maximized: $\Theta^{(P+1)} = \arg \max_{\Theta} Q(\Theta|\Theta^{(P)})$. Due to space limitations, we omit the detailed derivation and give the final solution for our re-estimation step.

M-Step:

$$\mu_k^{(P+1)} = \frac{\sum_{i=1}^n z_{ik} y_i}{\sum_{i=1}^n z_{ik}} \quad (17)$$

$$\sigma_{kll}^{2(P+1)} = \frac{\sum_{i=1}^n z_{ik} (y_{il} - \mu_{kl})^2}{\sum_{i=1}^n z_{ik}} \quad (18)$$

$$\lambda_k^{(P+1)} = \frac{\sum_{i=1}^n z_{ik}}{\sum_{l=1}^m \sum_{i=1}^n z_{il}} \quad (19)$$

In order to select the model to use (*i.e.*, number of parameters) we use the MDL principle:

$$\text{MDL} : \arg \max_{\Theta, k} (\log P(\mathbf{x}|\Theta) - \frac{k}{2} \log n), \quad (20)$$

where k is the number of parameters in the model and n is the number of samples in the training data.

In the implementation of the EM algorithm, the stopping criterion was chosen to be 20 iterations and the initial values for Θ were chosen as follows:

- μ_k : chosen randomly using a uniform distribution in 7D with parameters given by the minimum and maximum values of the training data in every dimension
- σ_{kll}^2 : chosen to be the estimated covariance from the whole training set, the same for all mixtures, $\sigma_{kll}^2 = \frac{1}{n} \sum_{i=1}^n (y_{il} - \mu_l)^2$, where μ_l represents the l -th component of the population mean.
- λ_i : chosen to be uniformly distributed given the number of modes to be used in the EM algorithm.

7 Experimental settings

The classification techniques we will use for our tests have been defined in Sec. 6. Here we describe the experimental context that is common to all our tests.

First, we collect training samples of each action to be recognized. Training our system is an intensive labor, a training and registration system was designed so that actions were labeled by users more easily. We trained the system with 40 to 50 instances of each action performed by five to seven different subjects each. In our experiments we used seven or eight actions, therefore our test and training data is composed of approximately 300 labeled actions.

Some of the actions considered occur in a setting where there is a translational motion effect (*e.g.*, walking actions), which cannot be used by our training system. For these instances, we used the tracking system developed by [23] which is able to track and segment the subject in a given sequence. It also produces an object centered sequence of the action. This processed sequence was then the input to our training system when objects considered presented global translational motion besides the that of its components. Therefore, it allows actions that present a global translational component to be treated as actions whose motion is non-translational.

Our classification paradigms assume mutual independence among instances of examples in our training data set. Moreover, the training data set and the testing data set should also be independent. In order to improve the validity of our models, it is necessary to take into account these issues when collecting the training data. Related to these concerns, some of the considerations when collecting our data were:

- **Action performance independence:** Subjects did not watch the actions performed by other subjects before performing their actions. This may increase the independence among subject training data.
- **Actions from same subjects:** One could argue that different instances of the same action being performed by the same subject are not totally independent. In this case we would need to use a very large number of subjects performing every action. Instead we collected the data in different sessions, allowing for a higher independence of the actions.
- **Anthropometric variability:** To try to span a representative feature space, seven different subjects with more or less variable anthropometric characteristics were chosen. Therefore, our variability spans just the range of this variable presented in these subjects.
- **Environment variability:** One source of dependence could be created by the environment in which training takes place. We have chosen different scenarios (four) with different lighting conditions in different training sessions.
- **Subjects in training and test sets:** a effective way to guarantee independence between our training and test sets is to use different subjects in each set. In some of our experiments, we trained the system using one set of subjects and tested it on a disjoint set of subjects.

We will denote \mathcal{X}_d and \mathcal{X}_t to be our training and test sets respectively. The above is trying to guarantee $\mathcal{X}_d \cap \mathcal{X}_t = \phi$. Moreover, we increase our independence, generalization potential, and allowed variability by collecting \mathcal{X}_d and \mathcal{X}_t in a variety of conditions (by no means complete). The importance of these issues rely on the fact that this is the variability that our system will be prepared to handle.

The performance of our system is measured using simple error counting. Some results in classifying human motion were presented by [6], where 18 aerobic exercises were used as test domain using a similar representation. Using the first-nearest-neighbor approach, a probability of error of $P(e) = \frac{1}{6}$ was obtained, but the ranking of the correct move was often close to top.

Another result was presented by [4] on three different categories (running, walking, skipping). They based their classification approach on Hidden Markov Models (aided by other estimation techniques to obtain the data) and reported 85% - 93% accuracy. Using seven or eight actions, chance would give us a maximum probability of error of 0.857 and 0.875 respectively.

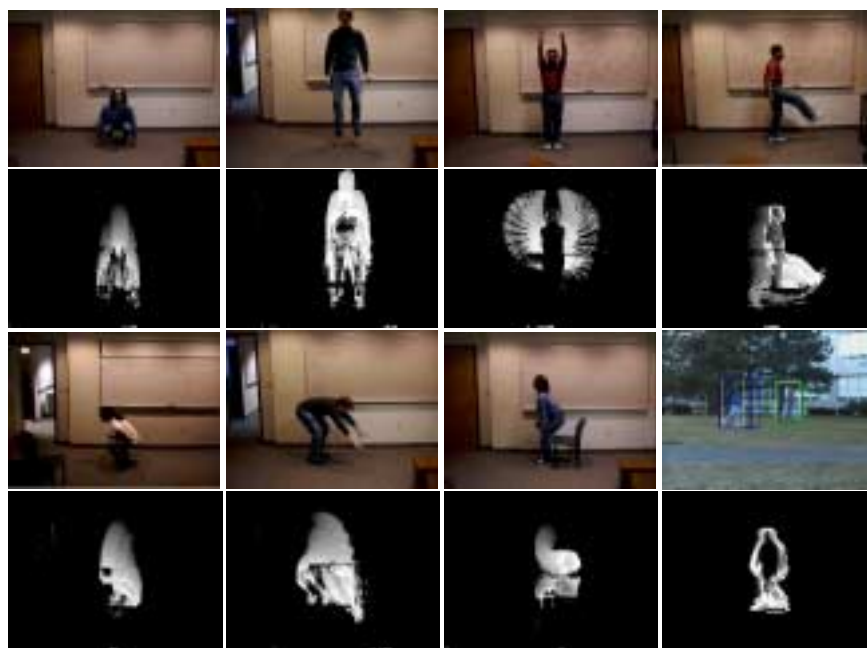


Figure 2: A single frame and MHI taken from the 8 actions used in our classification. First two rows show actions 1 to 4, last two rows show actions 5 to 8.

For all experiments we used either a consumer hand held video camera, or the standard SGI O2 uncalibrated camera recording at 10Hz (320x240 pixels color). The video is JPEG compressed, which causes a noticeable reduction in resolution (*e.g.*, box effect) at each frame.

Fig. 2 shows the complete set of actions/views that will be used in our test. Each action is illustrated with a single key frame of the video sequence and its respective

Actions	1-NN	3-NN	5-NN	Gaussian	Gauss. Mixture (2)	Gauss. Mixture (MDL)
A1	0.08	0.18	0.10	0.18	0.14	0.00
A2	0.06	0.06	0.06	0.00	0.06	0.00
A3	0.08	0.08	0.02	0.02	0.08	0.10
A4	0.04	0.12	0.20	0.26	0.00	0.05
A5	0.10	0.12	0.12	0.16	0.24	0.15
A6	0.16	0.18	0.40	0.70	0.44	0.25
A7	0.00	0.00	0.00	0.06	0.04	0.05
$P(e)$	0.0743	0.106	0.129	0.197	0.143	0.086

Table 1: $P(e)$ of the different classifiers using 50 rotations

Actions	A1	A2	A3	A4	A5	A6	A7
A1	-	0.00	0.00	0.00	0.00	0.08	0.00
A2	0.00	-	0.00	0.00	0.06	0.00	0.00
A3	0.00	0.02	-	0.06	0.00	0.00	0.00
A4	0.00	0.00	0.04	-	0.00	0.00	0.00
A5	0.00	0.00	0.00	0.00	-	0.10	0.00
A6	0.12	0.00	0.00	0.00	0.04	-	0.00
A7	0.00	0.00	0.00	0.00	0.00	0.00	-

Table 2: Confusion matrix indicating the $P(e)$ of the 1-NN classifier using 50 rotations

MHI. Notice that action 1 and action 5 both correspond to the same action, but the view angle is different. Actions are: 1) crouching-down, 2) jumping, 3) arm-waving, 4) kicking, 5) crouching-down (side-view), 6) leaning-over, 7) sitting on chair, and 8) walking.

Our classifiers are 1,3,5-nearest neighbor classifiers, a Gaussian classifier, a mixture of Gaussian classifier with 2 modes for all classes, and a mixture of Gaussian classifier that uses the MDL principle to find the best number or modes per class to be used in estimating the underlying densities.

8 Experimental results and discussion

The general settings of our experiments and description of our classification techniques were described in Sec. 7 and 6 respectively. Here we present the details of each experiment that complement our general settings along with their results. We conducted three different experiments using some variations of the training data collected as explained. Each experiment is used to evaluate certain properties of the potential of this approach.

8.1 Testing Using Data from All Subjects

For this experiment our goal is to classify the first seven actions shown in Fig. 2. We tested our classification techniques using 50 rotations of the training/test data sets. Each rotation is done as follows: from the set of approx. 300 actions, one action per class is chosen using a uniform pseudo-random number generator. Obviously, the selected 7 actions are not used for training. Notice that there is the possibility that one action is chosen more than once to be used as a test. At each rotation the system is trained using the rest of the actions. Tabs. 1, 2, 3, and 4 show the classification results obtained.

Actions	A1	A2	A3	A4	A5	A6	A7
A1	-	0.00	0.00	0.00	0.00	0.14	0.00
A2	0.00	-	0.00	0.00	0.06	0.00	0.00
A3	0.00	0.02	-	0.06	0.00	0.00	0.00
A4	0.00	0.00	0.00	-	0.00	0.00	0.00
A5	0.10	0.04	0.00	0.00	-	0.04	0.06
A6	0.34	0.02	0.00	0.00	0.02	-	0.06
A7	0.00	0.00	0.00	0.04	0.00	0.00	-

Table 3: Confusion matrix indicating the $P(e)$ of the Gaussian Mixture classifier

Actions	A1	A2	A3	A4	A5	A6	A7
A1	-	0.00	0.00	0.00	0.00	0.00	0.00
A2	0.00	-	0.00	0.00	0.00	0.00	0.00
A3	0.00	0.05	-	0.05	0.00	0.00	0.00
A4	0.00	0.00	0.05	-	0.00	0.00	0.00
A5	0.00	0.00	0.00	0.00	-	0.01	0.05
A6	0.20	0.00	0.00	0.00	0.00	-	0.05
A7	0.00	0.00	0.00	0.00	0.00	0.05	-

Table 4: Confusion matrix indicating the $P(e)$ of the Gaussian Mixture classifier using MDL (20 rotations)

Notice that by using this procedure, we can be sure that actions from the same subject on which the system is being tested, are part of the training data set. Intuitively this means that the performance of the system would not be strongly affected at least when using the KNN classifier and Gaussian mixture classifier. The Gaussian classifier would probably suffer because statistics are computed from the whole (non-homogeneous) class training set.

8.2 Using new subjects

This experiment differs from that in Sec. 8.1 in that we first train the system using a given set of subjects and then we test it using a set of subjects that does not overlap with the previous set.

Our goal is to measure the ability of the system to generalize the definition of the actions even when the subjects performing the actions are not in the training set. For this experiment, we chose the recorded actions of two subjects as our test data and the rest of the actions (from 5 subjects) as our training data. We then chose (pseudo-randomly) 5 moves of every action of the test data and classify them using the training data. This process was repeated 3 times using different subjects to be the training data. Tabs. 5 and 6 show the results of this experiment.

As expected the recognition performance degrades when actions performed by the test subject are not considered for training. The results indicate that the system could generalize with a small loss in performance. This result is mainly because in our experience, actions from the same subject are more likely to be similar to themselves than to same actions performed by other subjects.

Again we can see that the error rate is highly influenced by higher misclassification of classes 5 and 1. This occurs due to similarities in their statistics. The error decreases

Actions	1-NN	3-NN	5-NN	Gaussian	Gauss. Mixture (2)	Gauss. Mixture (MDL)
A1	0.10	0.20	0.20	0.24	0.16	0.00
A2	0.08	0.10	0.10	0.02	0.08	0.04
A3	0.08	0.06	0.08	0.04	0.08	0.08
A4	0.04	0.12	0.14	0.30	0.04	0.04
A5	0.12	0.12	0.14	0.18	0.24	0.18
A6	0.22	0.24	0.40	0.64	0.50	0.30
A7	0.02	0.02	0.04	0.10	0.06	0.06
$P(e)$	0.094	0.129	0.157	0.217	0.166	0.100

Table 5: $P(e)$ of the different classifiers in the *new subjects* experiment, using 3 rotations of 10 instances each. Actions of test subjects were not part of the training data.

Actions	A1	A2	A3	A4	A5	A6	A7
A1	-	0.00	0.00	0.00	0.02	0.08	0.00
A2	0.00	-	0.00	0.00	0.08	0.00	0.00
A3	0.00	0.04	-	0.04	0.00	0.00	0.00
A4	0.00	0.00	0.04	-	0.00	0.00	0.00
A5	0.04	0.02	0.00	0.00	-	0.06	0.00
A6	0.16	0.00	0.00	0.00	0.06	-	0.00
A7	0.00	0.00	0.00	0.00	0.02	0.00	-

Table 6: Confusion matrix for *new subjects* experiment indicating the $P(e)$ of the 1-NN classifier

when using more locally-based classifiers, but it may cause generalization problems if data is overfitted.

8.3 Translational motion tests

In this experiment we test recognition of the system on one more action. This action differs from the rest in that there is an extra component in the motion of the subjects, translational motion due to locomotion. We use the tracking system developed by [23] to obtain an object centered representation of the action. This is necessary because our representation relies on changes due to motion of the subject with respect to itself only. In the rest of the actions, subjects were always in the same spatial coordinates in the projected image. Therefore it is necessary to use a mechanism to undo the effect of translation as described by [23].

The processed sequence was then the input to our training system, as in the other actions. For this experiment we decided to use the same rotation mechanism used in Sec. 8.1. Classification results are shown in Tab 7 and Tab. 8.

The recognition rate is comparable with other classes, this example shows how it is

Actions	1-NN	3-NN	5-NN	Gaussian	Gauss. Mixture(2)	Gauss. Mixture (MDL)
A8	0.02	0.04	0.04	0.06	0.02	0.00
$P(e)$	0.068	0.098	0.118	0.18	0.128	0.075

Table 7: Performance $P(e)$ of the different classifiers in the *translational motion* experiment, using 50 rotations.

Actions	A1	A2	A3	A4	A5	A6	A7	A8
A1	-	0.00	0.00	0.00	0.00	0.08	0.00	0.00
A2	0.00	-	0.00	0.00	0.06	0.00	0.00	0.00
A3	0.00	0.02	-	0.04	0.00	0.00	0.00	0.00
A4	0.00	0.00	0.04	-	0.00	0.00	0.00	0.02
A5	0.00	0.00	0.00	0.00	-	0.08	0.00	0.00
A6	0.12	0.00	0.00	0.00	0.04	-	0.00	0.00
A7	0.00	0.00	0.00	0.00	0.00	0.00	-	0.00
A8	0.00	0.00	0.00	0.02	0.00	0.00	0.00	-

Table 8: Confusion matrix for *translational motion* experiment indicating the $P(e)$ of the 1-NN classifier using 50 rotations

possible to extend the ideas initially described to handle more complicated situations, namely multiple objects with translational motion.

9 Discussion

Experimental results indicate that the first-nearest neighbor approach performs best. However, very similar performance was obtained by the Gaussian mixture classifier with the MDL principle. This observation make it reasonable to think more closely about the specific advantages of one classifier over the other.

9.1 KNN and mixture of Gaussians

A problem derived from the use of the first-NN with a greater amount of training data is the possibility of overfitting. The generalization properties of the classifier would be reduced if classes present some *locality*. Another point against KNN classifiers is their weak data reduction properties. In its basic formulation, KNN's require to store and possibly search through all the data obtained during training. On the other hand, a mixture of Gaussian classifier reduces the description of the data distribution considerably. Moreover, the MDL principle puts constraints on the description length of the distribution and at the same time avoids overfitting using an information theoretic approach. Compression, data reduction and overfitting are thus some of the main reasons why a mixture classifier should be used.

On the other side, one of KNN's advantages is that it does not require a model of the form of the distribution. The risk of assuming the data distribution is considerably high in novel tasks. However, according to the results, a mixture of Gaussians could be a good model for the data distribution, its performance is almost as good as not assuming the form of the data distribution. These observations make us prefer the use of a Gaussian mixture classifier for further tests and applications.

9.2 Action Representation and Performance

In general, the recognition rate is limited by the similarities in the class descriptions given by the feature vector. This is the main source of error. The classifiers with the best performance had the tendency to concentrate their errors in misclassifying action 6 with action 1, and secondarily action 5 with 1 and 6, the similarities in their statistics are easily seen in Fig. 2.

Our second source of errors is a direct consequence of the simple motion detection mechanism we use (image differences). For example, motion is harder to detect when the object texture is low-frequency (imagine what the difference images would be if computed on a person wearing clothes of only one solid color). A more sophisticated motion detection technique would increase robustness.

The third source of error is due to the loss of accuracy obtained by representing sequences with MHI's and MEI's. Actions 1,5, or 6 are not really similar when seen in standard video. However, the representation used here makes them a lot more alike, mainly because of loss of motion details and self occlusion. This is one of the limitations of the representation of action chosen.

9.3 Human Actions and Form of the Distributions

Comparing the results of the classifiers in all of the experiments we could roughly infer the distribution of the classes given the data from all subjects. Some classes were represented by a compact set of points in the space (classes 2,3,5,7,8) but others are more disperse (classes 1,4,6). This could be a consequence of the differences in the way the same action is performed by different subjects. Note the performance differences obtained by the three different KNN classifiers here tested. This may be related to the insufficiency of the representation to establish clear boundaries among classes. A 1-NN classifier relies more strongly on local characteristics of the space. This supports our inference about the underlying distribution of the classes.

Besides the above reasons, the results obtained by the classifiers indicate that the distributions are not unimodal in general. A reliable estimate can be obtained using the MDL principle. Using MDL in a Gaussian mixture, we found that the average number of modes that best represent the underlying distribution was 3.97, indicating the better suitability of multimodal distributions to describe the data. We could see that the more multi-modal the estimated distribution, the better the classification results, but the risk of over-fitting our data is also higher. In the experiments, some classes were classified correctly even when unimodal densities were estimated. Class 2,3,7 in experiment 1 are examples of this observation. Increasing the modes of the estimated distributions does not affect the recognition rate in an appreciable positive way.

The average performance obtained in experiments 1 and 2 show that when testing on a given subject, the classifiers take advantage of the training data obtained from that given subject. When the test subject is not part of the training data, performance decreases slightly. It is natural to think that when an action performed by a given subject needs to be classified, the closest actions in the training set would be the ones performed by himself. This is a clear consequence of the interdependence of action and subject, (*i.e.*, different subjects perform the *same action* differently).

Our experiment with new subjects showed that even though there is no training data from actions of the subject being classified, the features chosen are good enough to generalize the actions. Experiment 3 shows that it is possible to extend our classification system to handle more complicated cases using an adequate registration mechanism. Occlusions, multiple subjects, and attention problems could be handled in this way. This is part of the future research directions we intend to explore.

References

- [1] A. Azarbayejani and A. Pentland. Real-time 3d tracking of the human body. In *Image Com*, 1996.
- [2] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motion using local parametric models of image motion. In *ICCV*, 1995.
- [3] A. Bobick and J. Davis. An appearance-based representation of action. In *ICPR*, 1996.
- [4] C. Bregler. Learning and recognizing human dynamics in video sequences. In *CVPR97*, 1997.
- [5] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *ICCV*, 1995.
- [6] J. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, 1997.
- [7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data. *Journal of the Royal Statistical Society (B)*, 39(1), 1977.
- [8] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1977.
- [9] S. Dudani, K. Breeding, and R. McGhee. Aircraft identification by moment invariants. *Trans. on Computers*, 26(1), 1977.
- [10] D. Gavrilu and L. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *Intl. Workshop on Automatic Face and Gesture Recognition*, 1995.
- [11] D. Hogg. Model-based vision: A paradigm to see a walking person. *Image and Vision Computing*, 1(1), 1983.
- [12] D. Hogg. *Interpreting Images of a Known Moving Object*. PhD thesis, University of Sussex, 1984.
- [13] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory*, IT(8), 1962.
- [14] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2): 210-211, 1973.
- [15] S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proc. Gesture Recognition*, 1996.
- [16] R. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7(1), 1991.
- [17] J. O'Rourke and N. I. Badler. Model-based image analysis of human motion using constraint propagation. *PAMI*, 2(6):522-536, 1980.
- [18] M. Ostendorff. *Expectation Maximization Algorithm Notes*. Boston University, College of Engineering, 1998.
- [19] A. Pentland and B. Horowitz. Recovery of non-rigid motion and structure. *PAMI*, 13(7):730-742, 1991.
- [20] R. Polana and R. Nelson. Low level recognition of human motion. In *Proc. IEEE Workshop on Nonrigid and Articulate Motion*, 1994.
- [21] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, 1995.
- [22] K. Rohr. Incremental recognition of pedestrians from image sequences. In *CVPR*, 1993.
- [23] R. Rosales and S. Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *CVPR Workshop on the Interpretation of Visual Motion*, 1998.
- [24] M. Shah and R. Jain. *Motion-Based Recognition*. Kluwer Academic, 1997.
- [25] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder: Real time tracking of the human body. Technical Report TR 353, MIT Media Lab, 1996.