Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition

Nikolaos Gkalelis, Anastasios Tefas, *Member, IEEE,* and Ioannis Pitas, *Fellow, IEEE*

Abstract

In this paper, a novel method for continuous human movement recognition based on fuzzy vector quantization (FVQ) and linear discriminant analysis (LDA) is proposed. We regard a movement as a unique combination of basic movement patterns, the so-called dynemes. The proposed algorithm combines FVQ and LDA to discover the most discriminative dynemes as well as represent and discriminate the different human movements in terms of these dynemes. This method allows for simple Mahalanobis or cosine distance comparison of not aligned human movements, taking into account implicitly time shifts and internal speed variations, and, thus, aiding the design of a real-time continuous human movement recognition algorithm. The effectiveness and robustness of this method is shown by experimental results on a standard dataset with videos captured under real conditions, and on a new video dataset created using motion capture data.

Index Terms

Real-time continuous human movement recognition, fuzzy vector quantization, linear discriminant analysis.

I. Introduction

The detection and analysis of events in video sequences is a very important task for a number of applications in several areas, such as video annotation, surveillance, sports and car industry, due to the abundance of low-cost video recording devices as well as the fact that this technology offers the only non-invasive solution for complex event analysis tasks. In particular, human behavior understanding from video sources is currently at its infancy. This task encompasses the recognition of several types of human motion, for instance, running and playing soccer (which may include running, walking, kicking the ball, etc.). To formally describe human motion patterns, many human motion taxonomies have been proposed, e.g., [1], [2]. Here we use a human motion taxonomy similar to

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 211471 (i3DPost).

The authors are with the Department of Informatics, Aristotle University of Thessaloniki, Greece (email: {*galelis,tefas,pitas*}*@aiia.csd.auth.gr*). I. Pitas and N. Gkalelis are also with the Informatics and Telematics Institute, CERTH, Greece.

the one proposed in [2], which is inspired from the granularity of the human language and relevant work in speech recognition community. In the bottom of the hierarchy, *dyneme*, is defined as the smallest constructive unit of human motion, while one level above, *movement*, is perceived as a sequence of dynemes with clearly defined temporal boundaries and conceptual meaning, e.g., a period of walk constitutes the movement of walk. This paper deals with the recognition of movements using the dynemes. In Figure 1 we show image sequences of the movements walk and run, as well as the associated dynemes (which we describe in detail in section II).

Researchers in the field of human movement recognition exploit either the *local* or the *global* motion information within a sequence of human (body) posture images in order to represent a movement. Local motion information is derived by observing the spatial variation of the human body reference points over time. Reference point correspondences are acquired explicitly by feature tracking [3] - [6], or implicitly by optical flow [7] - [9]. Global motion refers to the shape configurations that the human body receives through the course of a movement. Consequently, a movement is represented by a sequence of posture images [10] - [14], without taking into account any point correspondences. However, it should not be disregarded that human body reference points contain weak human body shape information as well. Upon this observation recent tests in psychophysics [15], [16], have suggested that global motion information is mostly responsible for the perception of motion by the human visual system, while local motion has mainly supportive role, (e.g., for object segmentation and tracking). From a practical point of view, reliable feature tracking and optical flow calculation require computationally demanding algorithms making their use expensive for real time applications. On the other hand, the estimation/localization of a filled human silhouette (also called posture mask, a human body posture contour filled with uniform color), for representing a human posture, is, in general, a relatively easier task, particularly in cases of static/constant background. Motivated from the above discussion, we represent a human posture image with the respective binary posture mask and, hence, a human movement as a sequence of binary posture masks.

Human motion analysis is a broad topic consisting of several operational levels [17], [18]. In this paper we are interested in human movement recognition alone and, therefore, we review related approaches on this field. Following [19], we divide the existing work into three major categories: template matching, statistical classification and neural networks.

1) Template matching: In [5], motion tracking information is used to represent each posture frame and the optimal reconstruction coefficients after PCA analysis are used to represent a video in the feature space. Test videos are classified with the nearest centroid classifier. Similarly, the optical flow is computed in [8] and PCA is used to represent a movement video in the feature space. The optical flow is used as well in [7], to represent a human movement and classification of movements at a distance is done using normalized spatiotemporal correlation. In [20], vectors of shape context are clustered and based on these clusters, the feature vectors are quantized and aggregated to represent a movement video. In [13], an SVM-based algorithm is used to classify a test video by majority voting. In [14], the periodic content of periodic human movements is used to represent and classify them.

Recently, approaches that represent a movement with a sequence of human posture images and use space-time template matching classifiers, are receiving increasing attention. In [10], the locality preserving projections (LPP) method is used to learn a low-dimensional manifold for human movement recognition and the Hausdorff distance or the normalized spatiotemporal correlation is computed to classify a test video within a nearest neighbor framework. Similarly, in [11], posture mask sequences are represented with space-time shape features by solving the Poisson equation. In [21], space-time events are represented as spatiotemporal volumes and shape contours are extracted using an unsupervised algorithm. A flow-based correlation metric and a novel shape matching metric are combined to recognize an event in a cluttered or crowded environment.

2) Statistical approaches: In [2], [22], tracking information is exploited to form motion vectors for each video frame, train the HMMs and recognize a variety of human movements. In [6], a codebook is produced for each body part and used to represent body posture images. Then trained HMMs are used to recognize a test image sequence. Affine invariant Fourier descriptors of the human posture contours are computed in [12] and SVMs are combined with HMMs to recognize an unknown movement. In [1], a movement is expressed as a motion image energy and history template and several moments are computed to represent a human movement in the feature space. The motion information of major body parts is used in [3] to represent each frame, and a Mahalanobis-based majority voting criterion is used to classify a test video. In [4], decomposable triangulated graphs and dynamic programming are used to build movement prototypes and the maximum likelihood is applied to detect a learned movement.

3) Neural Networks: In [23], a three-layer feed forward neural network is employed to recognize among four different movements, while in [24], the DFT of silhouette histograms is calculated to represent a posture image, and a fuzzy neural network is trained to detect unknown static body postures.

The majority of the above methods model either the time sequential structure of the movements, e.g., by HMMs, which is quite expensive in terms of computational resources, or use an exhaustive comparison metric to compare variable length videos and account for internal speed variations. Moreover, most of the above methods require videos depicting one person executing only one movement type, and thus, are not suitable for continuous human movement recognition. On the other hand, recent studies in psychophysics, e.g., [16], have suggested that perception of a walking figure from the human visual system may occur from the posture resulting from the integration of a few consecutive postures of the movement, which is closely related with the dyneme described here. Additionally, humans can frequently recognize a specific movement type from a few consecutive postures or equivalently one dyneme, while for other movement types more dynemes are needed. This functionality suggests, that the human brain may be using multimodal densities to model human movement types, where some modes are confused among different movements and some modes are unique for a specific movement.

Motivated from the above studies, we consider each movement as a mixture density (called hereafter *movement density*), where the mixture components (called hereafter *dyneme densities*) are presented by their centers (i.e., the dynemes). In order to illustrate the rational of the proposed approach we have used binary mask sequences, of the movements walk and run, retrieved from the database described in [11]. Each binary mask was preprocessed and vectorized according to the method described in section III-B, and principal component analysis (PCA) was used to take an optimal (in terms of signal reconstruction) two dimensional representation of each vector. Then, the fuzzy c-mean algorithm (FCM) was applied assuming two or three clusters. The clustering results are shown in Figure 2.

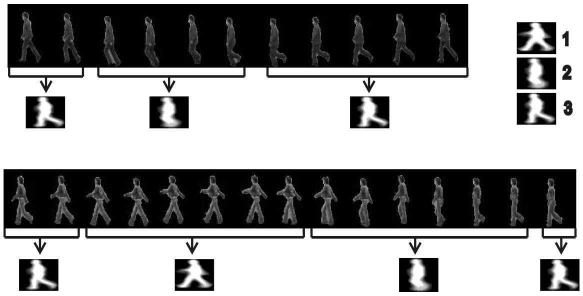
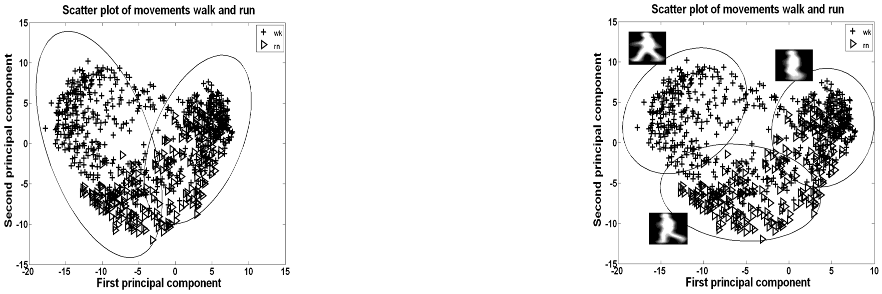


Fig. 1. *Movements of run (top) and walk (bottom) as well as the associated dynemes.*

We can see that the movements clearly overlap, and modelling them as unimodal densities, as in Figure 2a, is not an adequate representation. In contrast, using three dyneme densities as mixture components, as shown in Figure 2b, the movements are confused in dynemes 2 and 3, but dyneme 1 contains only samples of the movement walk. Moreover, the samples of dyneme density 1 correspond always in a specific part of the movement walk as we show in Figure 1, and, thus, these dyneme densities can be used to model and discriminate the two different movements. Based on this observation, we devise an algorithm that first identifies the dyneme densities, and then uses them to model and discriminate a number of different movements.



(a)

(b)

Fig. 2. *Clustering results of the movements walk and run using the FCM algorithm for* ***(a)*** *2 clusters and* ***(b)*** *3 clusters.*

In the following we summarize the main contributions of this paper:

- A novel approach that combines FVQ and LDA for modelling a movement as a mixture density and discriminating the different movements, is proposed. The robustness and recognition rates achieved with this method on a publicly available database are at the same level with the best reported rates in the field. Yet, the proposed technique is much faster than other state of the art approaches.
- The proposed algorithm can well recognize the dominant movement within an image sequence, and, thus, a windowing function is used to extend the algorithm for continuous human movement recognition.
- An analysis of the confusion of the postures among the different movements using the proposed algorithm, is presented.

A detailed analysis of the various steps of the method is provided in section II, while several experimental results are given in section III. Finally, conclusions regarding the proposed approach are drawn in section IV.

II. Fuzzy movement representation and recognition

A real-world movement is a continuous sequence of postures. In video-based action recognition, a movement is represented by a discrete temporal sequence of frames. Each frame in the sequence represents a specific posture of the movement. At this point we assume that binary human body masks of the postures at each frame are provided, possible from a tracking/segmentation subsystem. These posture masks are further preprocessed to create regions of interest (ROIs), which have the same dimensions, contain as much foreground as possible and are centered with respect to the centroid of the body posture. A posture mask ROI is scanned column-wise to produce the so-called *posture vector* $\mathbf{x} \in \Re^F$, where $F = W \times H$ is equal to the ROI size (in pixels). Thus, a human movement is represented in the input space $\Re^F$ with a spatiotemporal trajectory $\{\mathbf{x}_\imath\}$, which is called *movement sequence*.

Our target is to devise a fast algorithm that can learn and classify R simple everyday life movements in terms of dynemes, and in more specific from C basic posture-like vectors, the so-called *dyneme vectors* $\mathbf{v}_c \in \Re^F$, $c = 1, \dots, C$, which express the actual structure of the space of the particular movements. We compute the dyneme vectors with the fuzzy c-mean (FCM) algorithm and then use fuzzy vector quantization (FVQ) to map a posture vector $\mathbf{x}_\imath$ in a new space called *dyneme space*, whose coordinates are the normalized distances of vector $\mathbf{x}_\imath$ to the dyneme vectors:

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x}_\imath) &= [\phi(\mathbf{x}_\imath|\mathbf{v}_1, m), \dots, \phi(\mathbf{x}_\imath|\mathbf{v}_C, m)]^T \text{ or} \\ \boldsymbol{\phi}_\imath &= [\phi_{1,\imath}, \dots, \phi_{C,\imath}]^T \,, \end{aligned} \tag{1}$$

where m is the fuzzification parameter and $\boldsymbol{\phi}_\imath \in \Re^C$ is the quantized posture $\mathbf{x}_\imath$. The basis function $\phi(\mathbf{x}_\imath|\mathbf{v}_c, m)$ is related with the c-th dyneme and denotes the ability of this dyneme to represent posture vector $\mathbf{x}_\imath$.

The arithmetic mean of the postures of a movement sequence in the dyneme space, is called *movement vector*, $\mathbf{s} \in \Re^C$, and is used as the movement representation. The amount of each dyneme in the movement sequence will be encoded in the corresponding component of $\mathbf{s}$. Making the assumption that movements of the same type are characterized uniquely from specific dynemes, (i.e., movements of different types differ in at least one dyneme), movement vectors of different types will lay in a different subspace of $\Re^C$. This assumption can be further relaxed by allowing movements of different types to contain the same dynemes but in different quantities.

In order to discriminate the movements, LDA can be used for projecting the movement vectors to a discriminant subspace. In this space, movement types can be modelled with their sample mean, and test movements can be efficiently classified according to the Mahalanobis or cosine distance from the movement prototypes.

The above steps are encapsulated in a single algorithm, and the leave-one-out-cross-validation (LOOCV) approach, combined with the global-to-local search strategy, [25]–[27], is used to identify the number of dynemes C and the fuzzification parameter m. Each step of the algorithm is explained in detail in the following sections.

A. Computation of dynemes by FCM

Given a set of unlabelled posture vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, the number of dynemes C, and the fuzzification parameter m, the dyneme vectors are identified using the FCM algorithm in the input space. FCM [28] is based on the minimization of the following objective function:

$$J_{FCM}(\mathbf{\Phi}, \mathbf{V}) = \sum_{c=1}^{C} \sum_{\imath=1}^{N} (\phi_{c,\imath})^m (\| \mathbf{x}_\imath - \mathbf{v}_c \|_2)^2 \ , \tag{2}$$

where, N, C, are the number of samples and centroids respectively, $\mathbf{x}_\imath \in \Re^F$ is the $\imath$-th sample in the training data set, $\mathbf{V} = [v_{\jmath,c}] = [\mathbf{v}_1, \dots, \mathbf{v}_C] \in \Re^{F \times C}$ is the matrix of cluster prototypes, $\mathbf{\Phi} = [\phi_{c,\imath}] \in \Re^{C \times N}$ is the partition matrix with $\phi_{c,\imath} \in [0,1]$ being the degree that the $\imath$-th sample belongs to the c-th cluster, $m > 1$ is the fuzzification parameter and $\| \ \|_p$ is the p-norm of a vector. The FCM criterion (2) is subjected on producing non-degenerate fuzzy C-partitions of the training data, belonging to the set, $\{\mathbf{\Phi} \in \Re^{C \times N} \mid \sum_{c=1}^{C} \phi_{c,\imath} = 1, \ \forall \imath; \ 0 < \sum_{\imath=1}^{N} \phi_{c,\imath} < N, \ \forall c; \ 0 \leq \phi_{c,\imath} \leq 1\}$. The computation of the cluster centers and partition matrix is carried out through iterative optimization of (2), with the update of membership matrix and cluster centers at each step given by

$$\phi_{c,\imath} = \frac{(\| \mathbf{x}_\imath - \mathbf{v}_c \|_2)^{\frac{2}{1-m}}}{\sum_{\jmath=1}^{C} (\| \mathbf{x}_\imath - \mathbf{v}_\jmath \|_2)^{\frac{2}{1-m}}} \tag{3}$$

$$\mathbf{v}_c = \frac{\sum_{\imath=1}^{N} \phi_{c,\imath}^m \mathbf{x}_\imath}{\sum_{\imath=1}^{N} \phi_{c,\imath}^m} \tag{4}$$

The iteration is initialized with a random initial estimate of matrix $\mathbf{V}$ or $\mathbf{\Phi}$ and terminates when the difference of the estimated matrix between two iterations is smaller than a specified tolerance ϵ.

B. Fuzzy vector quantization

Using the dyneme vectors identified with FCM above, and the respective fuzzification parameter, FVQ [29] is used to map a posture vector $\mathbf{x}_\imath$ from the input space to the dyneme space. This transformation comprises two steps, the fuzzification and the normalization step. In the fuzzification step, the membership vector $\mathbf{u}_\imath$ is computed

$$\mathbf{u}_\imath = [u_{1,\imath}, \dots, u_{C,\imath}]^T \ ,$$

$$u_{c,\imath} = (\| \mathbf{x}_\imath - \mathbf{v}_c \|_2)^{\frac{2}{1-m}} \ .$$

The membership vector is normalized to produce the final representation of the posture in the dyneme space

$$\boldsymbol{\phi}_\imath = \frac{\mathbf{u}_\imath}{\| \mathbf{u}_\imath \|_1} \ . \tag{5}$$

C. Fuzzy movement representation

As discussed in the beginning of section II the movement vector $\mathbf{s} \in \Re^{C}$, i.e., the arithmetic mean of the comprising postures of a movement in the dyneme space is an adequate representation of a movement

$$\mathbf{s} = \frac{1}{L}\sum_{\imath=1}^{L} \boldsymbol{\phi}_{\imath} \ , \tag{6}$$

$\mathbf{s} = [s_1, \dots , s_C]^T$, where $\boldsymbol{\phi}_{\imath}$, given in (1), is the representation of the $\imath$-th posture of the movement in the dyneme space. Note that the sum of the components of $\mathbf{s}$ is one. In the following an equivalent probabilistic model is given.

Let $\omega_1, \dots, \omega_C$, be the C dyneme classes, with the associated class-conditional component densities $p(\mathbf{x}|\omega_c, \boldsymbol{\theta}_c)$ (also called dyneme densities), where $\boldsymbol{\theta}_c$ is the parameter vector of the c-th conditional density. We model each movement type r as a mixture density of the dyneme densities

$$p_r(\mathbf{x}|\boldsymbol{\theta}_r) = \sum_{c=1}^{C} p(\mathbf{x}|\omega_c, \boldsymbol{\theta}_c) P_r(\omega_c), \quad r = 1, \dots, R \, ,$$

where $\boldsymbol{\theta}_r$ is the parameter vector of the r-th mixture density. In this model, we see that the r-th movement mixture density differs from the other movement densities only in the mixing parameters $P_r(\omega_c), \ c = 1, \dots, C$, which must also satisfy

$$P_r(\omega_c) \geq 0 \ , \qquad \sum_{c=1}^{C} P_r(\omega_c) = 1 \, . \tag{7}$$

Given a set of L posture vectors of the r-rh movement type, $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$, and assuming that they are drawn independently, we can form the log-likelihood function of the observed posture vectors

$$\ln p(\mathcal{X}|\boldsymbol{\theta}_r) = \sum_{\imath=1}^{L} \ln\{\sum_{c=1}^{C} p(\mathbf{x}_{\imath}|\omega_c, \boldsymbol{\theta}_c) P_r(\omega_c)\} \, .$$

We can obtain an estimate of the mixing parameters by using the method of lagrange multipliers to maximize the log-likelihood function subject to the constraints of (7) (e.g., similar to [30], p. 518). The lagrangian function is

$$\ell = \sum_{\imath=1}^{L} \ln\{\sum_{c=1}^{C} p(\mathbf{x}_{\imath}|\omega_c, \boldsymbol{\theta}_c) P_r(\omega_c)\} + \lambda(\sum_{c=1}^{C} P_r(\omega_c) - 1).$$

Maximization of the lagrangian function in respect to $P_r(\omega_{\jmath})$ gives

$$\sum_{\imath=1}^{L} \frac{p(\mathbf{x}_{\imath}|\omega_{\jmath}, \boldsymbol{\theta}_{\jmath})}{\sum_{c=1}^{C} p(\mathbf{x}_{\imath}|\omega_c, \boldsymbol{\theta}_c) P_r(\omega_c)} + \lambda = 0 \, .$$

Assuming that $\hat{P}_r(\omega_{\jmath}) \neq 0$, and using the second constraint in (7) along with the posterior probability given by

$$P_r(\omega_{\jmath}|\mathbf{x}_{\imath}, \boldsymbol{\theta}_r) = \frac{p(\mathbf{x}_{\imath}|\omega_{\jmath}, \boldsymbol{\theta}_{\jmath}) P_r(\omega_{\jmath})}{\sum_{c=1}^{C} p(\mathbf{x}_{\imath}|\omega_c, \boldsymbol{\theta}_c) P_r(\omega_c)} \ ,$$

we obtain the maximum-likelihood (ML) estimate of the mixing parameters

$$\hat{P}_r(\omega_\jmath) = \frac{1}{L}\sum_{\imath=1}^{L} \hat{P}_r(\omega_\jmath|\mathbf{x}_\imath, \hat{\boldsymbol{\theta}}_r)\,,$$

where $\hat{\boldsymbol{\theta}}_r$ is the ML estimate of the parameter vector of the r-th mixture vector, and $\hat{P}_r(\omega_\jmath|\mathbf{x}_\imath, \hat{\boldsymbol{\theta}}_r)$ is the respective estimate for the posterior probability given by

$$\hat{P}_r(\omega_\jmath|\mathbf{x}_\imath, \hat{\boldsymbol{\theta}}_r) = \frac{p(\mathbf{x}_\imath|\omega_\jmath, \hat{\boldsymbol{\theta}}_\jmath)\hat{P}_r(\omega_\jmath)}{\sum_{c=1}^{C} p(\mathbf{x}_\imath|\omega_c, \hat{\boldsymbol{\theta}}_c)\hat{P}_r(\omega_c)}\,. \tag{8}$$

In our approach we have assumed that the overlap between the dyneme densities $p(\mathbf{x}|\omega_c, \boldsymbol{\theta}_c)$ is small. In this case (e.g., see [30], p. 548) the ML approach and the FCM procedure are expected to give similar results, and the ML estimate of the posterior probabilities in (8) is equivalent to the membership degrees in (4), i.e., $\phi_{c,\imath}^{(r)} = \hat{P}_r(\omega_c|\mathbf{x}_\imath, \hat{\boldsymbol{\theta}}_r)$. Then the c-th component of the movement vector in (6) is equal to the ML estimate of the prior probability of the c-th dyneme in this movement type

$$s_c^{(r)} = \frac{1}{L}\sum_{\imath=1}^{L}\phi_{c,\imath}^{(r)} = \frac{1}{L}\sum_{\imath=1}^{L}\hat{P}_r(\omega_c|\mathbf{x}_i, \hat{\boldsymbol{\theta}}_r) = \hat{P}_r(\omega_c)\,.$$

Therefore, each component of the movement vector depicts the ”amount” of the respective dyneme in the movement. Assuming that any two movement types differ in at least one dyneme, movement vectors resulting from different movement types will exhibit a different pattern, while movement vectors of the same movement type will exhibit a similar pattern, which will still not be exactly the same pattern as different people execute the same movement with their own style. This randomness in style when people execute the same movement allows for perceiving the movement vectors as random vectors and using them to model the different movement types as unimodal multivariate distributions, which are separable by the scatter of their means.

D. Analysis of the confusion between different movements

Let $\mathcal{U}$ be database of movement sequences, where each sequence belongs to one of R different movement classes. Ignoring the sequential information, we represent the n-th sequence of the r-th class with length L_n as a set of posture vectors $\{\mathbf{x}_{n,1}^{(r)}, \ldots, \mathbf{x}_{n,L_n}^{(r)}\}$. If N_r is the number of sequences in the r-th class, then the total number of movement sequences in the database is $N = \sum_{r=1}^{R} N_r$. Assuming that the dyneme vectors, and the fuzzification parameter, have been identified, we can map the posture vectors in the dyneme space using (5) and we can get the movement vectors in the dyneme space, $\{\mathbf{s}_1^{(1)}, \ldots, \mathbf{s}_{N_1}^{(1)}, \ldots, \mathbf{s}_{N_R}^{(R)}\}$ using (6). The arithmetic mean of the movement vectors belonging to the r-th class can be directly used to represent a movement type in the dyneme space

$$\boldsymbol{\mu}^{(r)} = \frac{1}{N_r}\sum_{n=1}^{N_r}\mathbf{s}_n^{(r)}\,, \tag{9}$$

$\boldsymbol{\mu}^{(r)} = [\mu_1^{(r)}, \ldots, \mu_C^{(r)}]^T$. This vector is also fuzzy in the sense that its components express membership degree and sum to one ($\sum_{c=1}^{C}\mu_c^{(r)} = \frac{1}{N_r}\sum_{n=1}^{N_r}\sum_{c=1}^{C} s_{c,n}^{(r)}$). The representation of each movement type with a fuzzy vector can be used to identify the dynemes that mostly represent each movement type. That is, large values for $\mu_c^{(r)}$ denote that movement r is well represented by dyneme c. Alternatively, one can use this interpretation to find the movements that are mostly represented from a specific dyneme c, and to reveal overlaps between different movement types. This can be done by forming a matrix $\mathbf{M} \in \Re^{C \times R}$ whose columns are the fuzzy vectors

$$\mathbf{M} = [\boldsymbol{\mu}^{(1)}, \dots , \boldsymbol{\mu}^{(R)}] \, . \tag{10}$$

Each component of the c-th row of $\mathbf{M}$ depicts the degree that the respective movement is expressed by the c-th dyneme. This analysis can reveal which dynemes actually represent a movement class and which are confused between different movement classes. For instance, let $[\mu_c^{(1)}, \dots , \mu_c^{(R)}]$ be the c-th row of $\mathbf{M}$. Then the c-th dyneme mostly represents the ρ-th movement class given by

$$\rho = \underset{r \in [1, \dots , R]}{\text{argmax}} \; (\mu_c^{(r)}) \; . \tag{11}$$

If $\mu_c^{(\rho)}$ is considerably larger than all other elements in the c-th row, then the c-th dyneme strongly expresses the ρ-th movement. In contrary, if there are more than one elements with relatively high values, this dyneme reveals the confusion of the respective movements. Such an analysis is carried out in section III-C2.

E. LDA projection

The number of the dynemes C is always larger than the number of the movement types R, thus, the movement vectors $\mathbf{s} \in \Re^{C}$ can be further projected using a subspace method, e.g., linear discriminant analysis (LDA), [30], [31]. Most LDA algorithms seek for the linear projection $\boldsymbol{\Psi}_{opt} \in \Re^{C \times R-1}$, that maximizes the criterion $J_{LDA}(\boldsymbol{\Psi}) = \frac{|\boldsymbol{\Psi}^T \boldsymbol{S}_b \boldsymbol{\Psi}|}{|\boldsymbol{\Psi}^T \boldsymbol{S}_w \boldsymbol{\Psi}|}$, i.e.,

$$\boldsymbol{\Psi}_{opt} = \underset{\boldsymbol{\Psi}}{\text{argmax}}(J_{LDA}(\boldsymbol{\Psi})) \; . \tag{12}$$

The matrix $\boldsymbol{\Psi}$ represents a linear transformation, and $\mathbf{S}_w$, $\mathbf{S}_b \in \Re^{C \times C}$, are the within and between scatter matrices respectively. The rank of $\mathbf{S}_w$ is at most $N - C$, and thus, is invertible if the number of training videos N is adequately larger than the number of the dynemes C. Then, the optimum matrix in (12) is formed by the $R - 1$ generalized eigenvectors that correspond to the largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$, and the projection of the n-th movement vector is given by

$$\mathbf{y}_n^{(r)} = \boldsymbol{\Psi}_{opt}^T \mathbf{s}_n^{(r)} \; . \tag{13}$$

As we have assumed that the movement vectors are derived from unimodal multivariate distributions, which are separable by the scatter of their means, the criterion $(\mathbf{S}_w^{-1}\mathbf{S}_b)$, can well measure the movement type separability, and, thus, conventional LDA can well separate the different movement types (e.g., see [31], p. 452). If $\mathbf{S}_w$ is not invertible, i.e., when $N < C$, an appropriate LDA variant can be used [25] - [34]. We should also note that before applying LDA, the movement vectors are standardized using the mean and the standard deviation along the training set.

F. Classification

The r-th movement is modelled by the respective prototype

$$\boldsymbol{\zeta}^{(r)} = \frac{1}{N_r} \sum_{n=1}^{N_r} \mathbf{y}_n^{(r)},\ r = 1, \ldots, R\ . \tag{14}$$

A test movement sequence $\{\mathbf{z}_1^{(t)}, \ldots, \mathbf{z}_{L_t}^{(t)}\}$ is represented in the dyneme space using (5), (6), standardized and projected using (13), to give a single point $\boldsymbol{\tau}^{(t)}$. The class label t of the test sequence is given by

$$t = \underset{r \in [1, \ldots, R]}{\text{argmin}}\ (\, g_r(\boldsymbol{\tau})\,)\ , \tag{15}$$

where $g_r(\boldsymbol{\tau}),\ r = 1, \ldots, R$, are the discriminant functions. Throughout our experiments we have used discriminant functions based on Mahalanobis or cosine distance [30].

G. Dyneme vectors and fuzzification parameter estimation

LOOCV procedure, e.g., [27], is utilized to determine the number of dyneme C and the fuzzification parameter m. The database may contain more than one instances of the same person performing the same movement. Therefore, at each validation cycle, the movement sequences of a person, e.g., person o, performing the movement type, e.g., movement α, are removed from the training set in order to form the test set, while the rest of the movement sequences of person o (them that do not refer to the movement α) remain in the training set. Thus, a test movement sequence would be classified correctly if it exhibits high similarity with the training movement sequences of other persons performing the same movement type α and not be confused with different movements performed by the test person. To determine the optimum parameters, the LOOCV procedure is combined with the global-to-local search strategy similar to [25], [26]. That is, after globally searching over a wide range of the parameter space, we find a candidate interval where the optimal parameters might exist. Then we try to find the optimal parameters within this interval. The algorithm is summarized in Figure 3.

H. Continuous human movement recognition

When a novel movement sequence is expressed in the dyneme space, most of its feature vector $\mathbf{s}$ components will be close to zero, except of those that correspond to the dynemes of the movement type that the sequence belongs to. This attribute allows the use of a windowing function for devising an algorithm for continuous human movement recognition.

Continuous movement recognition is performed in a sliding window using the optimum parameters, C, m, and the respective dyneme vectors and movement prototypes identified using the training set. Every new frame at time κ is scanned and preprocessed to provide the respective posture vector $\mathbf{x}_\kappa$. Then, $\mathbf{x}_\kappa$ is mapped to the dyneme space

```
Initialize: $O = \textit{number of objects},\ A = \textit{number of movements}$.
**LOOCV**: Set $C = C',\ m = m',\ \Upsilon_{c,m} = 0$.
**for** $a = 1$ to $A$;   **for** $o = 1$ to $O$
    Extract movement sequences of object $o$ performing
    action $a$ to form the test set: $\{\mathbf{z}^{(t)}_{1,1}, \ldots, \mathbf{z}^{(t)}_{N_z, L_{N_z}}\}$ .
    ***Training***
      1) Compute dyneme vectors by FCM (3), (4).
      2) Compute movement vector for each training video (5), (6).
      3) Standardize movement vectors and apply LDA (13).
      4) Compute movement prototypes (14).
    ***Testing***
    Compute test movement vectors (5), (6), (13) $\{\boldsymbol{\tau}^{(t)}_1, \ldots, \boldsymbol{\tau}^{(t)}_{N_z}\}$.
    **for** $\imath = 1$ to $N_z$
        Classify test movement vector $\boldsymbol{\tau}^{(t)}_\imath$ using (15).
        **if** $z_\imath$ is classified correctly **then**  $\Upsilon_{c,m}++$  **end if**
    **end for** $(\imath)$
**end for** $(o)$;   **end for** $(a)$
Perform the LOOCV procedure for different values $C', m'$.
Get optimal parameters : $(C_{opt}, m_{opt}) = \underset{c,m}{\text{argmax}}\ \Upsilon_{c,m}$.
```

Fig. 3. *Dyneme vectors and fuzzification parameter estimation.*

using (5), giving $\boldsymbol{\phi}_\kappa$. The weighted average of the posture representations in the dyneme space within a sliding window $\mathbf{w}$ is taken in order to provide the movement vector at frame κ

$$\mathbf{s}_\kappa = \sum_{\imath=-\ell_1}^{\ell_2} w_\imath\, \boldsymbol{\phi}_{\kappa-\imath}\ , \tag{16}$$

where the weights $w_\kappa,\ \kappa \in [-\ell_1, \ell_2],\ \ell_1, \ell_2 \geq 0$, are the coefficients of the sliding window, and $\ell_2,\ \ell_1$, determine the number of the past and future frames respectively to account for the computation of the movement vector. When $\ell_1 \neq 0$, i.e. future frames are used in (16), a respective delay is introduced in the computation of the movement vector. The movement vector is then projected to the discriminant subspace using (13) to yield $\boldsymbol{\tau}_\kappa$, which is then classified with (15), giving the resulting recognized movement class as output. The length of the window should be appropriate in order to conveniently capture an appropriate number of dynemes for the current movement that allows discrimination of the current movement from the other movements within the window.

III. Experimental results

In this section we present experimental results on two databases. The first, called ”Weizmann” database [11], is one of the very few adequately sized, in terms of persons and actions, publicly available databases for view-based human action recognition. The second database has been created using the motion capture CMU database [35].

A. ”Weizmann” database description

The ”Weizmann” database contains 93 low-resolution videos ($180 \times 144 \times 25$ fps) of 9 persons performing 10 movements (3 persons perform the same movement 2 times), namely, walk (wk), run (rn), skip (sk), gallop sideways (sd or side), jump jack (jk), jump (jp), jump in place (pj or pjump), bend (bd), wave with one hand (wo or wave1) and wave with two hands (wt or wave2). In addition, contains 20 videos, showing one person walking under several challenging conditions, which we describe in detail in section III-C3. The videos are captured under real conditions, and, thus, the posture masks, produced from background substraction, are imperfect. A few masks from several movements are depicted in Figure 4.

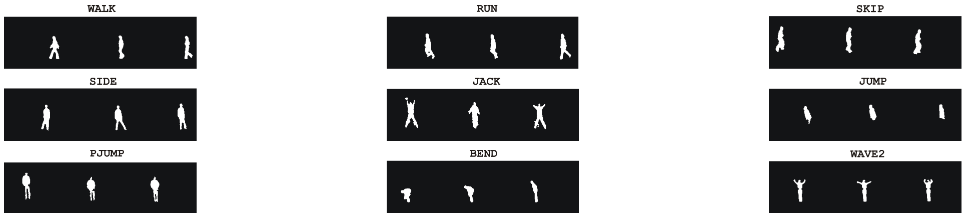


Fig. 4. *Three binary body masks for nine human movements.*

B. Preprocessing

In our computations, the binary masks provided in [11] are directly employed. For non-stationary movements, the mask sequences are transformed to show persons moving in the same direction, either towards left or right. This is done by first deciding the direction, and then mirroring the frames of the movement videos that show a person moving to an opposite direction from the chosen one.

The binary masks are preprocessed according to the procedure described in section I. The resulted posture mask ROIs are scaled to size 64×48 pixels using bicubic interpolation (as in [10]) and then scanned column-wise to form 3072-dimensional posture vectors.

C. Off-line analysis

Off-line analysis of a number of movement types using the proposed algorithm, includes the estimation of the parameters, C, m, the assessment of the robustness and classification performance of the algorithm, and the analysis of the confusion of the specific movement types.

1) Algorithm optimization: The classification dataset is used to optimize the proposed algorithm using the procedure shown in Figure 3. This procedure assumes that each video contains only a single instance of a human movement. However, some videos show a person executing several cycles of a periodic activity. We brake such videos to their constituting single period movements, and, thus, we produce a database of 230 *movement videos* as shown in Table I. We see that the movement videos in the database have variable inter- and intra-class length,

movement	#	av. len.	min.	max.
walk	42	13.5	12	15
jump	29	12.5	11	16
side	22	15	13	18
run	28	10	9	12
skip	29	12	11	14
jack	18	27	24	31
pjump	25	15.5	13	18
bend	9	62	57	66
wave1	14	28.5	24	33
wave2	14	28.5	24	33

TABLE I
The average, minimum and maximum length (in frames), of human movement videos.

therefore an algorithm should be able to compare variable length videos and account for internal speed variations effectively.

The movement videos are preprocessed to provide a set of movement sequences, which are directly imported in the procedure of Figure 3 in order to identify the optimal parameters C_{opt}, m_{opt}, for the ten movement types in the database. The plot in Figure 5 depicts the recognition rate of the proposed algorithm, over different number of dynemes C, while the fuzzification parameter is $m_{opt} = 1.1385$. The optimum recognition rate, 96%, was attained with $C_{opt} = 43$ dynemes, where only nine movement sequences were misclassified. The respective confusion matrix is given in Table II. For dynemes $C > 20$, the recognition rate is always above 90%, although there is a decline in performance when we start using too many dynemes.

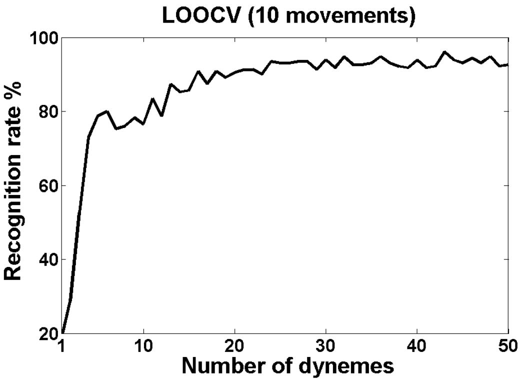


Fig. 5. *Human movement recognition rate vs the number of dynemes.*

2) Human movement analysis: The proposed algorithm can also be used as a qualitative analysis tool, i.e., to identify the degree to which different dynemes express each movement type as well as to assess the confusion between the different movement types. We do this using the optimal values $C = 43, m = 1.1385$ to compute the matrix **M**, as described in section II-D.

For illustration, for each of nine movements we depict the three dynemes, mostly associated with a movement

	bd	jk	jp	pj	rn	sd	sp	wk	wo	wt
bd	9									
jk		18								
jp			26	1			2			
pj			1	24						
rn					28					
sd				1		21				
sp			2				27			
wk								42		
wo									12	2
wt										14

TABLE II
Confusion matrix between ten human movements.

in Figure 6, as well as we plot the respective three rows of **M** in the corresponding subplot of Figure 7. Intuitively, we could say that some dynemes uniquely characterize the movement type they are assigned to, while others are confused among different movement types. For instance, dyneme 22, which is assigned to movement skip, is also very similar to some postures of movement run. These intuitive conclusions can be formally drawn from the plots in Figure 7, which explicitly depict an estimate of the percentage of the dyneme in the respective movement type, for each of the dynemes in Figure 6. That is, for each movement (e.g. SKIP, RUN, ..., etc.) the three representative dynemes (in terms of mixing strength) are plot along with the representation ability for other movements (e.g. wk, jp, rn, ..., etc.). For example, we see that the percentage of dyneme 22 in the movement SKIP is slightly larger than its percentage to the movement of run (rn in x-axis). On the other hand, some dynemes are found only on specific movement types. This is the case, e.g., for dynemes $14, 6, 43$, which are mostly found in the movement BEND.

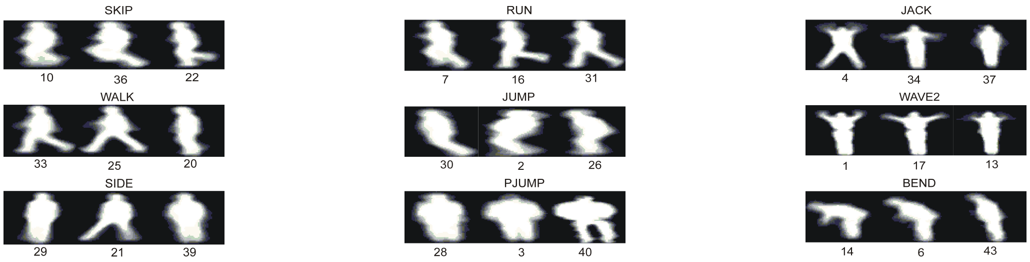


Fig. 6. *For each of nine movements three dynemes are depicted.*

In general, we see that movements of skip, walk and run exhibit high degree of confusion with other movement types. Among them, only walk bears some similarity with a non periodic movement, i.e., bend (bd). The latter appears to be the most distinct among all movements. Finally, some small confusion exists between gallop sideways, jump in place and jump movements as well as between the wave with two hands, wave with one hand and jack movements.

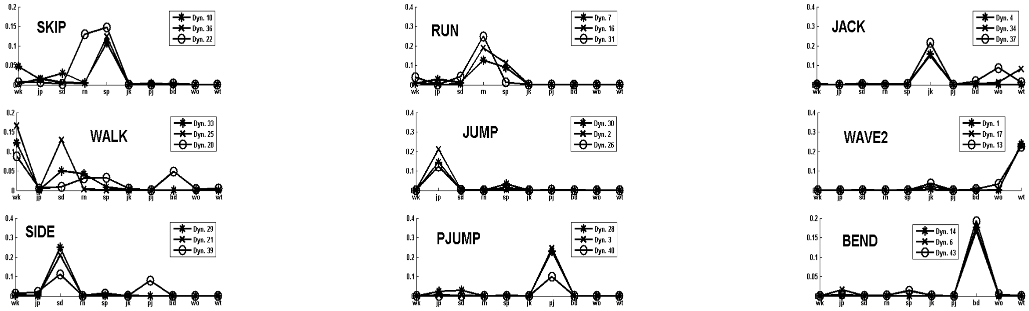


Fig. 7. *Confusion between dynemes: Each plot shows the degree that the r-th movement ($r =$ jp, sd, rn, sp, jk, pj, bd, wo, wt) belongs to the c-th dyneme.*

3) Robustness tests: The robustness of the algorithm was assessed using the ”robustness” database reported in [11] containing videos of various lengths (e.g., the largest video has 106 frames, while the smallest only 42), depicting a person walking under various scenarios. The videos can be roughly separated in two sets. The first set consists of ten videos that describe a person walking along various directions ($0° - 90°$) with respect to the horizontal camera axis. This set is suitable for testing the tolerance of the proposed algorithm against viewpoint changes. The other set consists of videos depicting a person walking in side view under varying conditions, e.g., under partial occlusion (passing behind a pole, walking next to a dog, walking while feet are occluded), wearing different clothing (wearing skirt, wearing a pair of trousers), executing different types of walk (normal walk, moonwalk, limp walk, walking with knees up), and, finally, walking while carrying an object (a bag or a briefcase).

The parameters ($C_{opt} = 43$, $m_{opt} = 1.1385$), the dyneme vectors, and the movement prototypes, identified in section III-C1, are directly used to test the robustness of the algorithm. That is, we do not perform any additional training, and all the robustness videos are used as test videos (even though, in reality, a test video consists of a sequence of walk movements). Each of the 20 ”robustness” videos is preprocessed as described in section III-B. Afterwards, all the posture vectors of the video in the input space are mapped in the dyneme space using (5), the movement vector for each video is computed using (6) and projected with LDA using (13). Then, the Mahalanobis distance is taken over all movement prototypes in the projection space to produce a set of 10 values, one for each movement type. The movement of the ”robustness” video is recognized as the movement type which produced the smallest value. Moreover, the set of the distance values is used to compute the median distance, i.e., the distance values are perceived as realizations of a distribution and the median distance is taken as the value from which half of the distances are above it and half of the distances are below it. The ratio of the smallest distance with the median distance can be used as a measure of the confidence of the recognition.

The classification results for both ”robustness” video sets are shown in Table III. In Table IIIa we see that walk videos captured under viewpoints smaller than $45°$ are recognized correctly, with high confidence over the median distance. For larger viewpoint angles, the walk movement is always recognized as the second best. Table IIIb on the same figure, depicts the classification results over the second ”robustness” video set, i.e., videos comprising different walk scenarios. The proposed algorithm classified correctly seven out of the ten walk videos, with high confidence over the median distance, while for the three misclassified videos, walk provided the second best match.

Video	**1^{st} best**	**2^{nd} best**	**Med.**
walk 0°	walk 0.24	side 13.8	15.6
walk 9°	walk 0.33	side 11.7	14.5
walk 18°	walk 1.15	side 8.8	15.4
walk 27°	walk 1.53	side 7.3	14.5
walk 36°	walk 2.34	side 6.1	15.2
walk 45°	side 3.10	walk 5.1	15.1
walk 54°	wave1 5.45	walk 7.4	13
walk 63°	wave1 3.27	walk 12.1	14.7
walk 72°	wave1 1.52	walk 14.9	18.1
walk 81°	wave1 1.42	walk 15.2	18.3

(a)

Video	**1^{st} best**	**2^{nd} best**	**Med.**
Occluded by pole	walk 3.1	side 9.7	16.2
Walk with dog	run 2	walk 8.3	17.9
Occluded feet	walk 2.8	side 5.1	12.8
Walk in skirt	walk 0.8	side 12.2	17.3
Normal walk	walk 0.78	side 12.6	17.5
Moonwalk	jack 7.3	walk 11.9	18.9
Limp Walk	walk 0.7	side 9.6	12.3
Knees up	side 3.6	walk 6.9	11.8
Carry bag	walk 1.5	side 9	14.4
Carry briefcase	walk 1.6	side 7.2	13.2

(b)

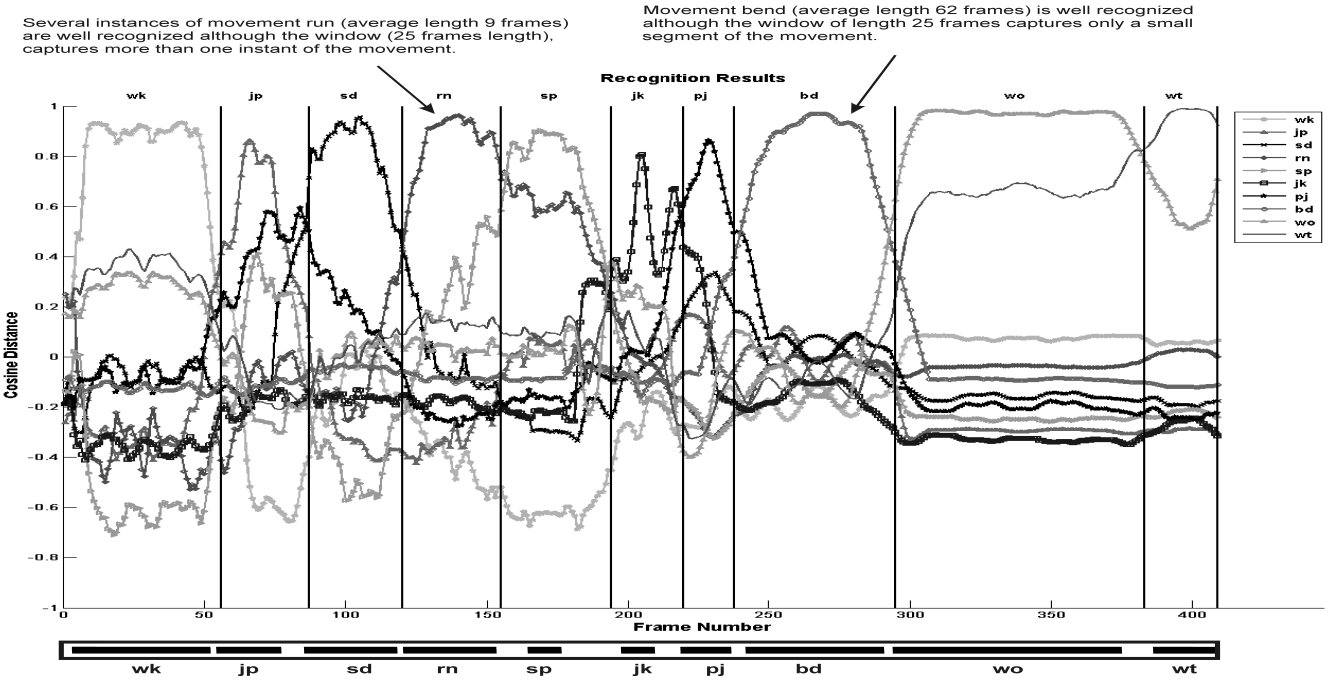
TABLE III

*Experimental results of robustness tests. The leftmost column describe the movement performed in the test video, while the first and second closest movement along with the corresponding distances, are shown in the second and third columns respectively. The median distance of the test movement from all the class movements is shown in the rightmost column. Table **(a)** depicts results on videos presenting a person walking on different directions, and Table **(b)** on videos showing a person walking in side view under varying conditions.*

D. Continuous movement recognition

We use a variant of the LOOCV procedure to evaluate the continuous human movement recognition algorithm, described in section II-H. The database contains videos of 9 persons performing 10 different movements. At each LOOCV cycle we remove from the database all the videos of the test person and concatenate the frames of these videos to form one test video. Thus, the test video depicts a person performing all the 10 movements, one after the other, e.g., first walk, then jump and so on. The training set consists of all the other persons' videos, i.e., videos showing 8 persons performing the 10 different movements. We break the training set videos to their constituting single period movements, as described in section III-C1, to produce the respective training movement videos. These videos are used to compute the movement prototypes, using the equations, (5), (6), (13), (14), with values $m_{opt} = 1.1385, C_{opt} = 43$.

At the test phase of each validation cycle we use equation (16), with $\ell_1 = \ell_2 = 12$, i.e., a window of size 25, and uniform weights, $w_\kappa = 0.7$, $\kappa = -12, \dots, 12$, to recognize the 10 movement types within the test video. At each frame instance the cosine similarity value between the resulting movement vector $\boldsymbol{\tau}^{(t)}$ and each of the R movement prototypes is computed, and all the values are normalized to yield R similarity values, $f_1(\boldsymbol{\tau}), \dots, f_R(\boldsymbol{\tau})$. The movement type at this frame is then recognized using, $t = \underset{r\in[1,\dots,R]}{\text{argmax}}\ (\, f_r(\boldsymbol{\tau})\,)$. In our experiments, in order to eliminate spurious detections and robustify the algorithm, we additionally used the following heuristic rules before accepting a movement type detection. Let $f_{r_1}(\boldsymbol{\tau})$ and $f_{r_2}(\boldsymbol{\tau})$ be the largest and second largest cosine similarity values respectively. The candidate label r_1 for the unknown movement at the current frame is accepted if $g_{r_1}(\boldsymbol{\tau}) > \alpha_1$ and $g_{r_1}(\boldsymbol{\tau}) - g_{r_2}(\boldsymbol{\tau}) > \alpha_2$. Moreover, the same movement type should be detected for more than α_3 frames continuously in order to consider the detection of this movement type as valid. The values used in our experiments are $\alpha_1 = 0.4, \alpha_2 = 0.15$ and $\alpha_3 = 6$.



Several instances of movement run (average length 9 frames) are well recognized although the window (25 frames length), captures more than one instant of the movement.

Movement bend (average length 62 frames) is well recognized although the window of length 25 frames captures only a small segment of the movement.

Fig. 8. *Recognition results for one validation cycle of the real time algorithm. The test video contains several instances of ten different movement types. The start and the end of each movement is not known, i.e., movements are not segmented a priori. The vertical lines through the plot indicate the actual end of a particular sequence of movements (e.g. walk), and the begin of another sequence of movements (e.g. jump). The horizontal bar at the bottom of the figure depicts the recognized movement type as computed by the proposed algorithm. The black color indicates that a movement type is recognized in the particular frame, while the white color depicts that a movement has not been detected.*

In total, the LOOCV procedure consists of 9 cycles, one for each person, where, at each cycle, the 10 movement types within the test video should be recognized. With the above settings, 86 movements out of 90 were recognized correctly, while the rest 4, namely, 2 skip movements, 1 gallop sideways and 1 wave with two hands, were considered undetected due to the above heuristics. Thus, the attained recognition rate was 95,6%. We should note that the estimate is performed with a delay of $\ell_1 = 12$ frames, which corresponds to 0.5 seconds. However, the proposed window method can be also considered to use only past frames for the movement detection. In this case, ℓ_1 is set to zero and the performance is slightly reduced.

In Figure 8 we show the results of one validation cycle, where all movements have been recognized correctly. The test video contains 409 frames and the movements appear with the following succession: walk, jump, gallop sideways, run, skip, jack, jump in place, bend, wave with one hand and wave with two hands. We can see that although the length of the movements varies considerable, from 9 frames to 66 frames, as shown in Table I, the algorithm can still effectively recognize them utilizing a sliding window of constant length, without the need of movement segmentation, time alignment, or identification of the start and the end frame of the movement. This is possible because movement types are different not only as a whole, but also if seen partially, and this difference is captured effectively by the respective dynemes and exploited by the proposed algorithm.

E. *Experiments on the CMU motion capture database*

The CMU motion capture database [35] has been used by several researchers in the field of human movement recognition, e.g., [36], [37]. However, we can not directly use this database as our method requires human movement videos. Thus, in order to create a video database of everyday life movements we devised 10 different male/female human figures casual dressed using Poser software [38], and then imported motion capture data, obtained from the CMU database, from the category of ”locomotion”. In particular, 10 trials of walk, 10 trials of run, 9 trials of jump and 8 trials of jump in place were used to finally generate 37 videos ($490 \times 500 \times 25$ fps). Each of the videos of run and walk show a person executing several cycles of the movement, and therefore we further broke these videos to produce a database of 92 movement videos and the respective binary mask sequences. The binary mask sequences are further preprocessed as described in section III-B to finally produce movement sequences consisting of 3072-dimensional posture vectors. Using the procedure described in Figure 3, we identified the optimum parameters, $C_{opt} = 8$, $m_{opt} = 1.1385$. For these parameters, only one movement sequence was wrongly classified, giving a classification rate of 98.9%. The misclassified sequence was a sequence of a person jogging very slowly and thus perceived as a movement of walk.

F. *Comparison with other methods*

The ”Weizmann” database reported in [11], has been recently used to evaluate the algorithms of several other state of the art methods, as shown in Table IV. The classification performance attained here is slightly lower than the best reported on this database [10], [11]. However, these methods, such as the most of the methods currently applied in human movement recognition, assume that the test video contains instances of only one movement type, and, thus, are not directly applicable on videos that contain instances of different movement types. Moreover, both the aforementioned methods during the on-line operation use nearest neighbor classification combined with the Hausdorff distance comparison metric (given by $D_H = \underset{\imath}{\text{median}}(\underset{\jmath}{\min} \parallel \mathbf{x}_\imath - \mathbf{y}_\jmath \parallel)$, where, $\{\mathbf{x}_\imath\}, \{\mathbf{y}_\jmath\}$ are movement sequences in some space) in order to align the sequences in time, and allow comparison of different length movements. Therefore, the on-line computational complexity of the proposed algorithm, which uses simple nearest centroid cosine distance, is lower. For instance, in [11], the reported processing time in Matlab (solving the

Method	[39] CVPR ‘07	[10] TIP ‘07	[20] ICIP ‘07	[40] MMC ‘07
Accuracy	72.8 %	100 %	88.9 %	82.6 %
Method	[11] PAMI ‘07	[41] WMVC ‘08	Proposed	
Accuracy	97.5 %	82.7 %	96 %	

TABLE IV
Correct classification rates on the ”Weizmann” database.

Poisson equation and computing moments of the space-time cubes), is around 30 seconds, on a Pentium 4, 3.0 GHz, for a pre-segmented video of size $110 \times 70 \times 50$. We used a video of the same dimensions to evaluate the efficiency of our algorithm using an unoptimized Matlab implementation, on a Pentium 4, 1.6 GHz. The preprocessing time (section III-B) was around 5 seconds, while FVQ and classification steps (test mode part of the algorithm depicted in Figure 3) needed less than 1/2 second.

IV. Conclusions

In this paper, inspired by the way that humans recognize movements, movement types are modelled as mixture densities. With the proposed method, a movement sequence of any length can be compactly expressed as a single low dimensional vector. This principle, allows fast nearest-centroid classification, and an overall low computational processing time. However, the major advantage of the method, as shown in the experimental results section, is its applicability for continuous human movement recognition. The algorithm has been evaluated on two databases providing good classification rates and exhibiting adequate robustness against partial occlusion, different styles of movement execution, viewpoint changes, male/female silhouettes, clothing conditions and other challenging factors.

References

[1] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[2] R. Green and L. Guan, “Quantifying and recognizing human movement patterns from monocular video images-part I: A new framework for modeling human motion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 179–190, Feb. 2004.

[3] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, “Human activity recognition using multidimensional indexing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1091–1104, Aug. 2002.

[4] Y. Song, L. Goncalves, and P. Perona, “Unsupervised learning of human motion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 814–827, Jul. 2003.

[5] Y. Yacoob and M. Black, “Parameterized modeling and recognition of activities,” in *Proc. 6th Int. Conf. Comput. Vis.*, Bombay, India, Jan. 1998, pp. 120–127.

[6] X. Feng and P. Perona, “Human action recognition by sequence of movelet codewords,” in *Proc. 1st Int. Symp. 3D Data Processing Visualization and Transmission*, Padova, Italy, Jun. 2002, pp. 717– 721.

[7] A. Efros, A. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2, Nice, France, Oct. 2003, pp. 726–733.

[8] R. Polana and R. Nelson, “Low level recognition of human motion (or how to get your man without finding his body parts),” in *Proc. 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA, Nov. 1994, pp. 77–82.

[9] M. J. Black, “Explaining optical flow events with parameterized spatio-temporal models,” in *Proc. IEEE Comput. Soc. Conf. on Comput. Vis. and Pattern Recognit.*, vol. 1, Ft. Collins, CO, USA, Jun. 1999, pp. 326–332.

[10] L. Wang and D. Suter, “Learning and matching of dynamic shape manifolds for human action recognition,” *IEEE Trans. on Image Proc.*, vol. 16, no. 6, pp. 1646–1661, Jun. 2007.

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[12] V. Kellokumpu, M. Pietikainen, and J. Heikkila, “Human activity recognition using sequences of postures,” in *Proc. IAPR Conf. on Mach. Vis. Applications*, Tsukuba Science City, Japan, May 2005, pp. 570–573.

[13] D. Cao, O. Masoud, D. Boley, and N. Papanikolopoulos, “Online motion classification using support vector machines,” in *Proc. 2004 IEEE Int. Conf. Robotics and Automation*, vol. 3, New Orleans, LA, USA, May 2004, pp. 2291– 2296.

[14] R. Cutler and L. Davis, “Robust real-time periodic motion detection, analysis, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 781–796, Aug. 2000.

[15] J. A. Beintema and M. Lappe, “Perception of biological motion without local image motion,” *Proceedings National Academy of Science*, vol. 99, no. 8, pp. 5661–5663, Apr. 2002.

[16] J. Lange, K. Georg, and M. Lappe, “Visual perception of biological motion by form: A template-matching analysis,” *Journal of Vision*, vol. 6, no. 8, pp. 836–849, Jul. 2006.

[17] L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis,” *Pattern Recognit.*, vol. 36, no. 3, pp. 585–601, Mar. 2003.

[18] T. B. Moeslund, A. Hilton, and V. Kruger, “A survey of advances in vision-based human motion capture and analysis,” *Comput. Vis. Image Understand.*, vol. 104, no. 2, pp. 90–126, Nov. 2006.

[19] A. K. Jain, R. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[20] Q. Wei, W. Hu, X. Zhang, and G. Luo, “Dominant sets-based action recognition using image sequence matching,” in *Proc. IEEE Int. Conf. Im. Proc. 2007*, vol. 6, San Antonio, TX, USA, Sept. 2007, pp. VI–133 – VI–136.

[21] Y. Ke, R. Sukthankar, and M. Hebert, “Event detection in croweded videos,” in *Proc. 11th IEEE Int. Conf. Comput. Vis.*, Pittsburgh, USA, Oct. 2007, pp. 1–8.

[22] R. Green and L. Guan, “Quantifying and recognizing human movement patterns from monocular video images-part II: Applications to biometrics,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 191–198, Feb. 2004.

[23] H. Yu, G. Sun, W. Song, and X. Li, “Human motion recognition based on neural network,” in *Proc. Int. Conf. on Communications, Circuits Syst.*, vol. 2, Hong Kong, China, May 2005, pp. 979–982.

[24] C. F. Juang and C. M. Chang, “Human body posture classification by a neural fuzzy network and home care system application,” *IEEE Trans. Syst. Man Cybernetics, A*, vol. 37, no. 6, pp. 984–994, Nov. 2007.

[25] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, “An introduction to kernel-based learning algorithms,” *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[26] J. Yang, A. Frangi, J. Y. Yang, D. Zhang, and Z. Jin, “KPCA plus LDA: a complete kernel fisher discriminant framework for feature extraction and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 230–244, Feb. 2005.

[27] M. Zhu and A. Martinez, “Subclass discriminant analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.

[28] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[29] N. B. Karayiannis, “Fuzzy vector quantization algorithms and their application in image compression,” *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1193–1201, Sep. 1995.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, 2nd ed.* John Wiley and Sons, 2001.

[31] K. Fukunaga, *Introduction to Statistical Pattern Recognition, 2nd ed.* New York: Academic Press, 1990.

[32] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[33] G. Goudelis, S. Zafeiriou, A. Tefas, and I. Pitas, “Class-specific kernel-discriminant analysis for face verification,” *IEEE Trans. Inf. Foren. and Secur.*, vol. 2, no. 3, pp. 570–587, Sep. 2007.

[34] J. Lu, K. Plataniotis, and A. Venetsanopoulos, “Face recognition using kernel direct discriminant analysis algorithms,” *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 117– 126, Jan. 2003.

[35] http://mocap.cs.cmu.edu, 2008.

[36] V. Parameswaran and R. Chellappa, “Human action-recognition using mutual invariants,” *Comput. Vis. and Image Understanding*, vol. 98, no. 2, pp. 295–325, May 2005.

[37] G. Sukthankar and K. Sycara, “A cost minimization approach to human behavior recognition,” in *Proc. 2005 Int. Conf. on Autonomous Agents, Proc. of 4th Int. joint Conf. on Autonomous Agents and multiagent systems*, Utrecht, Netherlands, Jul. 2005, pp. 1067–1074.

[38] http://www.curiouslabs.com, 2006.

[39] J. Niebles and L. Fei-Fei, “A hierarchical model of shape and appearance for human action classification,” in *IEEE Conf. on Comput. Vis. and Pattern Recognit.*, Minnesota, USA, Jun. 2007, pp. 1–8.

[40] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the 15th Int. Conf. on Multimedia*, Bombay, India, Sep. 2007, pp. 357–360.

[41] D. Batra, T. Chen, and R. Sukthankar, “Space-time shapelets for action recognition,” in *IEEE Workshop on Motion and video Computing*, Colorado, USA, Jan. 2008, pp. 1–6.