Multi-view human movement recognition based on Fuzzy distances and Linear Discriminant Analysis

Alexandros Iosifidis[†*], Anastasios Tefas[†], Nikos Nikolaidis[†*] and Ioannis Pitas[†*]

[†] *Informatics and Telematics Institute*
Centre for Research and Technology Hellas, Greece
[*] *Department of Informatics, Aristotle University of Thessaloniki*
Thessaloniki 54124, Greece Tel,Fax: +30-2310996304

{*tefas,nikolaid,pitas*}*@aiia.csd.auth.gr*

Abstract

In this paper, a novel multi-view human movement recognition method is presented. A novel representation of multi-view human movement videos is proposed that is based on learning basic multi-view human movement primitives, called multi-view dynemes. The movement video is represented in a new feature space (called dyneme space) using these multi-view dynemes, thus producing a time invariant multi-view movement representation. Fuzzy distances from the multi-view dynemes are used to represent the human body postures in the dyneme space. Three variants of Linear Discriminant Analysis (LDA) are evaluated to achieve a discriminant movement representation in a low dimensionality space. The view identification problem is solved either by using a circular block shift procedure followed by the evaluation of the minimum Euclidean distance from any dyneme, or by exploiting the circular shift invariance property of the Discrete Fourier Transform (DFT). The discriminant movement representation combined with camera viewpoint identification and a nearest centroid classification step leads to a high human movement classification accuracy.

Keywords: Activity recognition; Multi-view Dynemes; Fuzzy Vector Quantization; Linear Discriminant Analysis

1. Introduction

Human movement recognition and analysis is an important task for various applications. It can be used as a pre-processing stage for human behavior analysis

in a wide variety of fields, such as surveillance [1], human–computer interaction and games [2], model-based compression [3], augmented reality [4] and semantic video annotation. The term human movement has been used with various meanings in the literature. Sometimes it is used interchangeably with the terms human motion and human action or activity. In this paper we adopt the taxonomy used in [5], where movement, activity and action correspond to low-level, middle-level and high-level motion patterns, respectively. Many approaches have been proposed in order to formally describe human movement patterns. Two approaches that exploit the global human body information in order to describe human body posture shape are described in [6] and [7]. In [6], Motion Energy Image (MEI) and Motion History Image (MHI) are introduced. MEI is a binary image, which depicts the moving regions in white and still regions in black. MHI is a grayscale image whose intensity is a function of motion recency. Alternatively, movements can be described by a sequence of movement primitives, the so-called *dynemes* [7]. This approach is inspired from speech recognition, where the term phoneme is used to denote the smallest constructive speech unit [8].

Most movement recognition algorithms involve a training phase. The main challenges that a movement recognition method should be able to face and we address in this paper include:

- Inter-class variations: Several movement types are quite similar, for example jog and run.
- Intra-class variations: Variations in motion speed, execution style, as well as anthropometric ratios can be observed between individuals.
- Capture conditions: Person localization might be difficult in cluttered or dynamic environments. Self occlusions or occlusions of human body parts from other objects may result to poor human body representation.
- Human body orientation: The orientation of the person with respect to the camera might be different from that in the training videos (e.g. side vs frontal view). Moreover, during a movement, the person may change motion direction. A suitable human body representation which will deal with these changes should be utilized and the movement recognition accuracy should not be affected.
- Distance between the camera(s) and the person: The person may move in an arbitrary distance from the camera(s) used. This will affect the size of his/her body projection in the camera(s) plane(s).

- Continuous operation: The method should allow continuous movement recognition over time.
- Camera setup: The camera(s) used in the training and test phases may differ in resolution and frame rate. In the case of multiple cameras, synchronization errors between the frames coming from different cameras might occur. Furthermore, multi-camera setups, usually require camera calibration.

The plethora of movement recognition algorithms that have been proposed can be divided in three categories, depending on the adopted camera setup and their ability to perform view-independent human movement recognition: single-view, single-view/view-invariant and multi-view ones [9, 10].

Up to now, the majority of human movement recognition algorithms that have been proposed use one fixed camera (single-view video) in both the training and recognition phases. In [11], a codebook of movelets for each body part is produced to represent body posture images. A movelet is defined as a collection of the image patches that correspond to shape, motion and occlusion of the main human body parts. Hidden Markov Models (HMMs) estimate the most likely sequence of movelets and the movement depicted in a sequence. In [12], the human contour is described by a feature vector generated by a shape descriptor. Shape context features are clustered in Dominant Sets in each posture image. Classification is achieved using a nearest neighbor algorithm. In [7, 13], tracking information is exploited to form motion vectors in every video frame. Then, HMMs are used to recognize the human movement. In [14], Locality Preserving Projections (LPP) are used to project a sequence of moving silhouettes associated to a movement video on a low-dimensional space. The median Hausdorff distance or the normalized spatiotemporal correlation is used to classify an unknown movement within a nearest-neighbor framework. In [15], movement prototypes are represented by dynemes produced by Fuzzy Vector Quantization (FVQ). Linear Discriminant Analysis (LDA) is used to project fuzzy vector distances of each posture vector within a movement sequence from the dynemes to a low dimensionality space. In this space, the minimum Mahalanobis distance or the maximum cosine similarity from movement class centers is used for human movement classification. In [6], MEI and MHI representing a movement are concatenating in order to produce vector containing shape and time information. Movement classification is achived by performing a nearest neighbor procedure. An improvement of this work, which does not require tracking is presented in [16]. Although these algorithms achieve good recognition results, they require the same camera view angle during both training and recognition phases. This angle must, ideally, be the one that captures

the most discriminant motion information and, usually, corresponds to the side view. This assumption leads to a constrained recognition environment, because such algorithms will fail, if the person under study is captured from a different view angle or its motion direction changes over time.

In order to overcome this limitation, researchers have come up with view invariant single-view movement representation and recognition approaches. In [17], a computational representation of human movement that captures abrupt changes in the motion speed and the direction represented by the spatio-temporal curvature of a $2D$ trajectory was introduced. In [18], the view-invariant movement recognition problem was tackled by utilizing the geometric invariant theory, based on point-light displays. A convenient $2D$ invariant representation was obtained by combining patches of a $3D$ scene. In [19], a novel movement representation was proposed using the so-called spatio-temporal movement volumes (STV). Given the object contours at each time instance, a movement volume was generated by computing point correspondences between consecutive contours, based on graph theory. Then, a movement representation in terms of the sign of mean and Gaussian curvatures was obtained by analyzing the differential geometry of the local volume surfaces. These movement descriptors were employed to define a movement sketch, which is invariant to the camera viewing angle. In [20], an example-based movement recognition approach was demonstrated, using dependencies between three dimensional movement exemplars and their $2D$ projections on the image plane. $3D$ movement exemplars were used to produce $2D$ image information in the training phase, while, in the recognition phase, HMMs were employed in order to identify the movement sequence that best explains the image observations. An HMM variant, the Conditional Random Fields (CRFs) are used in [21] for human movement recognition. CRFs overcome the observations independence assumption in human movement analysis. In [22], human movements were represented as three-dimensional shapes, induced by the silhouettes in a space-time volume. The solution of the Poisson equation is exploited to extract space-time features, such as local space-time saliency, movement dynamics, shape structure and orientation. These features are subsequently utilized for shape representation and classification. The methods described above are invariant within a viewing-angle range and, thus, their application is limited to some special cases.

Recently, researchers have proposed algorithms that use multi-camera setups. The use of multiple cameras has several advantages. The human body is captured from multiple views, and, thus, fully view-independent movement recognition can be achieved. Moreover, a person being occluded in one, or more, camera(s) may be visible from other cameras, and, thus, movement recognition can still be pos-

sible. Finally by exploiting the multi-view human body information better recognition accuracy can be obtained. However, the need to process multiple video streams leads to higher computational cost and a multi-camera setup is more difficult to setup and more costly than a single camera one. In [23, 24], multi-view information is exploited to achieve view-invariant movement recognition. A temporal segmentation method is introduced to split a continuous movement sequence into primitive actions. Visual hulls are computed and accumulated over a time period into the so-called Motion History Volumes (MHVs) which are extensions of the MHIs proposed in [6]. MHVs, transformed into cylindrical coordinates around their vertical axis, are used to produce view-invariant features in the Fourier domain. In [25], Combined Local-Global (CLG) optical flow is used to extract a motion flow feature. Invariant moments with flow deviations are used to extract a global shape flow from multi-view image sequences. Multidimensional HMMs (MDHMMs) are used to classify an unknown incoming movement.

In this paper, a novel view-invariant method that exploits information captured by a multi-camera setup is proposed. Binary human body masks are obtained from a background subtraction procedure [26, 27], or using a chroma keying technique, or any other moving object segmentation technique. In most applications, such as video surveillance, this is an efficient way to obtain the moving object silhouettes. In cases where this approach can not be applied, human body pose estimation techniques [28, 29] can be applied to the video frames of each camera to produce binary human body masks. In case of noisy binary masks, simple post processing techniques, such as morphological operations or more advanced filtering techniques, can be applied in order to improve their quality. The human body is tracked in consecutive single-view video frames [30, 31] and binary masks corresponding to the same person coming from all cameras are subsequently combined to represent multi-view posture patterns. These patterns are clustered determining a number of multi-view posture primitives, the so-called multi-view dynemes, are determined. The fuzzy distances between every multi-view posture pattern and every multi-view dyneme are obtained in order to create a new representation space for the multi-view body postures, the so-called multi-view dyneme space. This new movement representation is motion speed and duration invariant. Furthermore, it has proven capable to generalize over variations within one class, distinguish between actions of different classes and cope with usual synchronization errors. Linear Discriminant Analysis (LDA) is performed to reduce dyneme space dimensionality by discovering an optimal discriminant subspace. The mapping of movement representations on this subspace produces the so-called discriminant movement representation, which is used for movement classification by

employing either the Euclidean or the Mahalanobis distance from the discriminant movement class centers.

The proposed method is a non parametric one and exploits the rich information captured by multiple synchronized and uncalibrated cameras in order to achieve high human movement classification accuracy. It assumes that the person is at short or medium distance from the cameras. The binary masks of the human body are rescaled in low resolution posture frames. Thus, the method can operate in settings where the human body silhouettes are of low resolution, i.e., the height of the body is higher than 30 pixels. In settings where the body size is smaller, the body binary mask may be affected by the presence of noise. This will affect the recognition accuracy. The solution of the camera viewpoint identification problem, i.e., the identification of the camera position with respect to the human body, before proceeding to movement recognition in a new test video leads to a view-independent movement recognition technique. This is achieved by determining the camera re-arrangement which provides the same view angle ordering as in the training phase, or by exploiting a new, view-invariant human body representation based on the circular shift invariance property of the Discrete Fourier Transform. The usage of the multi-view dynemes, combined with the projection of the movement representations in a low dimensionality discriminant feature space results to fast and accurate movement recognition.

The main novel contributions of this paper are: 1) the proposal of a novel view-invariant movement representation (multi-view dynemes), 2) the solution of the camera viewpoint identification problem using a circular shift procedure on the multi-view posture representation, followed by the minimum Euclidean distance from any multi-view dyneme, or by exploiting the circular shift invariance property of DFT, 3) the use of LDA variants for dimensionality reduction in the multi-view dyneme space.

The remainder of this paper is structured as follows. Section 2 provides an overview of the recognition framework used in the proposed approach and a small discussion concerning the movement recognition task. Section 3 presents technical details that clarify the processing steps performed in the proposed method. Section 4 presents experiments conducted for assesing the performance of the proposed method. Finally, conclusions are drawn in Section 5.

2. Problem Statement

One of the most commonly used multi-camera setups is the converging one, where all N synchronized cameras involved point to the observation space center,

as shown in Figure 1a) for eight ($N = 8$) cameras. The capture volume is the space which can be seen from all N cameras. The distance between the cameras and the person is application related. In the case of movement recognition in short range settings, such as indoors movement recognition, this distance will probable be small, while in far field settings, such as movement recognition in outdoor environments, e.g., a parking lot, this distance will usually be higher. In the later case, the size of the human in the videos will be small and this might affect the recognition accuracy. Every camera captures one video frame at each time instance, that will be called single-view frame. A collection of frames from all cameras acquired at the same time instance is referred as an N-view frame. An example is shown in Figure 1b).

(a) (b)

Figure 1: *a) A converging eight-view camera setup and its capture volume, b) an eight-view video frame*

Let $\mathcal{M}$ be a set of M elementary movement classes, such as walk, run, bend, etc. Since most human movements are periodic, the term elementary movement corresponds to a single period of the movement, e.g., to one walking step. In the case of non-periodic movements, e.g. 'bend', the elementary movement includes the whole movement sequence. Let a person perform an elementary movement m, $1 \leq m \leq M$ inside the camera capture volume, being captured in a N-view video, called elementary movement video. Obviously different elementary movements have different durations N_{t_m}, $m = 1, ..., M$. For example, a run period consists on average of only 9 video frames in a 25 fps video, whereas a bend period consists of 40 video frames. The problem to be solved is to recognize the elementary movement at hand. In the following, the word movement will denote elementary movement, unless otherwise stated. The proposed method copes with the elementary movement recognition problem, but is also extended, (see section 3.7) in order to recognize continuous movement videos containing many consec-

utive movement periods. This is done by using a sliding window consisting of adequate number of video frames. By moving this window, elementary videos are obtained and recognition is performed at every sliding window position.

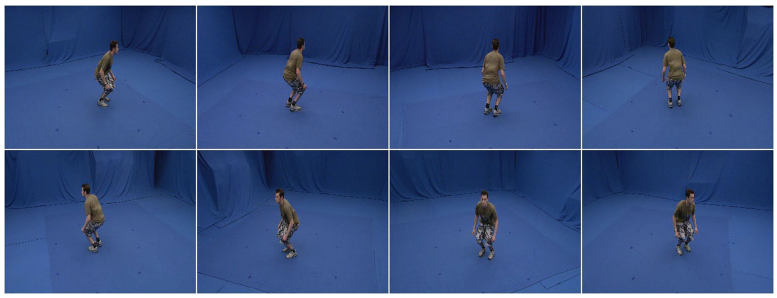
Elementary movement classes highly overlap in the video frame space, since the same body postures (N-view video frames) appear in different movements. This can be seen in Figure 2, where three eight-view video frames of a person performing the movements 'jump in place', 'jump forward' and 'sit' are shown. Even a human observer can be confused when he/she must decide which posture corresponds to each movement, when viewed in isolation and not within a sequence. However, there are certain postures that characterize uniquely certain movement classes. Furthermore, as already mentioned, different movements differ in duration. A movement representation scheme should take into account all these observations, in order to achieve good recognition results. Finally, a movement recognition technique should allow continuous movement recognition, be fast and take into account various factors related to camera setup, such as the camera-actor distances and the synchronization errors.

Movement recognition is a difficult task in such a setup. Indeed, the person may freely move inside the cameras capture volume. Thus, his/her view angle from a certain camera may change during movement. For example, whereas camera #1 can, at a certain time instance, depict a side view of a person, a change in his/her movement direction may result in a frontal view in this camera. Thus, the camera viewpoint identification problem should be solved in order to achieve a view-independent movement recognition. An alternative solution could be the use of view-invariant movement representation.

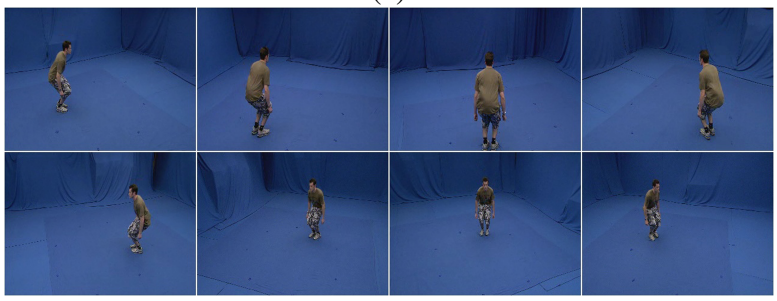
3. Proposed Method

3.1. Preprocessing

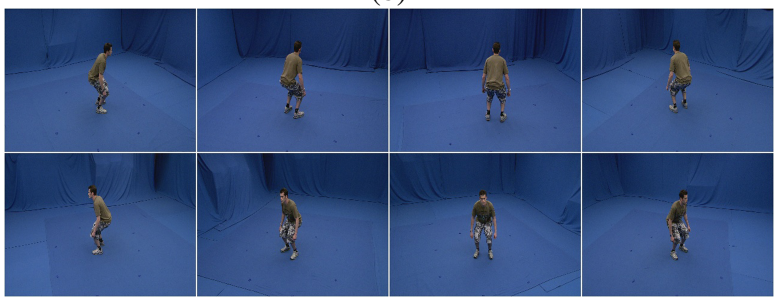
As previously described, a movement performed inside the camera capture volume is captured from all N cameras in a N-view movement video consisting of N_{t_m} N-view video frames that depict one movement period. The number of frames N_{t_m} in a N-view video frame may vary, according to the movement class m, $1 \leq m \leq M$. During training, a N-view movement video depicting a number of consecutive periods is manually split in elementary N-view videos that are subsequently used in the training procedure. During testing, in the case of continuous movement recognition, a sliding window of an appropriately chosen length that moves over a N-view video segment is used and recognition is performed for each temporal position of this window.



(a)



(b)



(c)

Figure 2: *Multi-view video frames of (a) 'jump in place', (b) 'jump forward' and (c) 'sit' sequences.*

Moving object segmentation techniques [26, 27] are applied to the frames of a N-view video to create binary single-view masks depicting the person's body in white over a black background. These masks are centered at the person center of mass. Binary single-view posture masks of size equal to that of the maximum bounding box (region of interest, ROI) that encloses the person body in each single-view video are created and rescaled accordingly to $H \times W = 64 \times 64$ pixels to produce binary single-view posture masks of fixed size. Apart from reducing the computational cost, the use of low resolution single-view posture

masks reduces the effect of local image segmentation errors which may occur in the cases of clutter background. Single-view binary posture masks of eight movements {'walk', 'run', 'jump in place', 'jump forward', 'bend', 'sit', 'fall' and 'wave one hand'} performed by the same person and being captured from various view-angles are shown in Figure 3.

Figure 3: *Binary single-view posture masks of eight movements captured from various view angles. From left to right: 'bend', 'jump forward', 'fall', 'run', 'sit', 'walk', 'wave one hand' and 'jump in place'.*

In the training phase, binary single-view posture masks of all the N views corresponding to the same time instance t are combined by placing the one that depicts the person's frontal view first and the remaining in a clockwise manner to create N-view binary posture masks. An eight-view binary posture mask from a 'bend' video is shown in Figure 4.

Figure 4: *An eight-view posture mask of a bend sequence.*

Binary N-view posture masks are scanned column-wise to produce posture vectors. That is, every multi-view movement video of $N \times H \times W$ pixels (per N-view frame) consisting of N_{t_m} video frames, is described N_{t_m} N-view posture vectors $\mathbf{p}_i = [\mathbf{p}_{i1}^T, \mathbf{p}_{i2}^T, ..., \mathbf{p}_{iN}^T]^T,\ i = 1, ..., N_{t_m},\ \mathbf{p}_{ij} \in \mathbb{R}^{H \times W}$.

3.2. Dynemes Calculation

In the training phase, all the N-view posture vectors $\mathbf{p}_i,\ i = 1, ..., L,\ L = \sum_{m=1}^{M} N_{t_m} \cdot N_T$ of all the N_T different training N-view elementary movement videos having $N_{t_m},\ m = 1, ..., M$ frames each, are clustered to K clusters without using the known movement labels. This approach is followed in order to produce movement independent multi-view movement primitives, the so-called *N-view dynemes*. Although this procedure can be performed by applying various clustering techniques, such as Spectral Clustering [32, 33], Self Organizing Maps [34], Fuzzy C-Means [35], etc., it was observed, through experimentation, that a simple K-Means algorithm [36] can provide satisfactory N-view dynemes. The

K-Means algorithm seeks to partition the N-view posture vectors of the training videos into K clusters having centers $\mathbf{v}_j,\ \ j = 1, ..., K$, so that the following expression is minimized:

$$\sum_{j=1}^{K} \sum_{i=1}^{L} a_{ij} \parallel \mathbf{p}_i - \mathbf{v}_j \parallel^2, \tag{1}$$

where $a_{ij} = 1$ if the N-view posture vector $\mathbf{p}_i$ is assigned to the cluster j (having cardinality $n_j = \sum a_{i,j}$) and zero otherwise.

The N-view dynemes $\mathbf{v}_j,\ j = 1, ..., K$ are obtained by calculating the arithmetic mean of the vectors assigned to each of these K clusters:

$$\mathbf{v}_j = \frac{1}{n_j} \sum_{j=1}^{L} a_{ij} \mathbf{p}_i. \tag{2}$$

The optimal dyneme number K is determined using the cross-validation procedure [37]. This procedure is used to determine how a learning algorithm will operate on data that it was not trained upon. During cross-validation, the learning algorithm is trained multiple times (folds), each time using all but some of the training samples that are subsequently used for testing. Depending on the samples used for testing, there are various cross-validation variations. In our case, the videos that are being excluded from the training set (test videos) are all the elementary movement videos depicting one person, i.e., the leave-one-person-out cross validation procedure was applied. The cross-validation procedure is applied for different number of dynemes and the optimal dyneme number K is the one that provides the best movement recognition rate.

A set of twenty eight-view dynemes describing eight movements ('walk', 'run', 'jump in place', 'jump forward', 'bend', 'fall', 'sit' and 'wave one hand') are shown in Figure 5. In this Figure, every row depicts an 8-view dyneme. As can be seen, some dynemes correspond to body postures that appear in more than one movements, e.g., dynemes 1 and 17 include postures of movements 'walk' and 'run', while dynemes 2, 8, 9, 10 and 12 include postures of movements 'jump in place', 'jump forward' and 'sit'. Some dynemes, however correspond to postures that appear only in one movement. Such a dyneme can uniquely determine this movement, e.g. dynemes 14 and 20 describe movement 'bend', dynemes 4, 11, 15, 16 and 19 describe movement 'wave one hand'. Two different movement classes should contain one or more different body postures. Using this approach, a movement can be described as a unique combination of dynemes, even if some

of the dynemes appear in more than one similar movements. Overall, it will subsequently proven that the dynemes have enough discriminant power to provide a good representation space for N-view movement posture vectors.

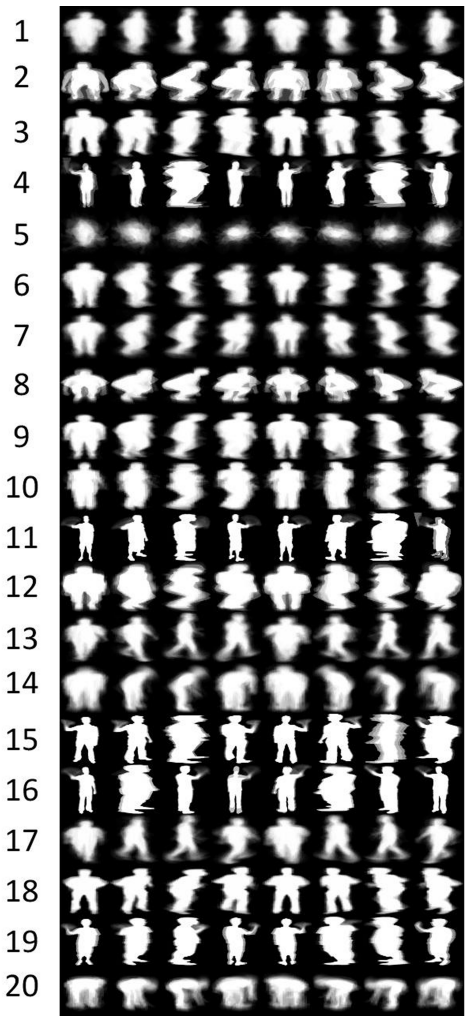


Figure 5: *A set of twenty dynemes derived from eight-view posture vectors of eight movements ('walk', 'run', 'jump in place', 'jump forward', 'bend', 'fall', 'sit' and 'wave one hand').*

3.3. Movement Representation

As already mentioned, every elementary movement video is described by a set of N_{t_m} N-view posture vectors $\mathbf{p}_i \in \mathbb{R}^{N_s}$, $N_s = N \times H \times W$, $i = 1, ..., N_{t_m}$,

where N_{t_m} varies, accross movement types. After the dyneme calculation, the fuzzy distances:

$$d_{ik} = (\parallel \mathbf{p}_i - \mathbf{v}_k \parallel_2))^{-\frac{2}{q-1}} \tag{3}$$

of every N-view posture vector $\mathbf{p}_i$ from all the K dynemes $\mathbf{v}_k$, $k = 1, ..., K$ are calculated. q is the fuzzification parameter ($q > 1$), which is set equal to 1.1 for all the experiments presented in this paper. Each N-view posture vector $\mathbf{p}_i$ is thus mapped to the following distance vector:

$$\mathbf{d}_i = [d_{i1}, d_{i2}, ..., d_{iK}]^T \in \mathbb{R}^K. \tag{4}$$

The distance vectors $\mathbf{d}_i$ are normalized to produce membership vectors $\mathbf{u}_i \in \mathbb{R}^K$, which are the posture vectors representations in the K-dimensional dyneme space:

$$\mathbf{u}_i = \frac{\mathbf{d}_i}{\parallel \mathbf{d}_i \parallel}. \tag{5}$$

If a movement video sequence consists of N_{t_m} video frames, the membership vectors $\mathbf{u}_i$ corresponding to each posture vector (one per video frame) are combined to produce the so-called *movement vector* $\mathbf{s} \in \mathbb{R}^K$, which represents the movement video in the dyneme space. To this end, simple arithmetic mean is chosen, in order to avoid taking into account any temporal information in the movement representation:

$$\mathbf{s} = \frac{1}{N_{t_m}} \sum_{i=1}^{N_{t_m}} \mathbf{u}_i. \tag{6}$$

Finally, the movement vectors $\mathbf{s}$ representing every movement video in the training dataset are normalized to have zero mean and unit standard deviation.

3.4. LDA Projection

In order to discriminate movement classes, the labeling information available in the training phase can be exploited. The dimensionality of training movement vectors $\mathbf{s}_{mj} \in \mathbb{R}^K$ can be reduced to $D < K$ dimensions, using a discriminant subspace method. Assuming that the movement classes are linearly separable, LDA [38] is used to project them to a low-dimensional discriminant subspace $\mathbb{R}^D$, $D < K$. The use of LDA in movement recognition, where the number of movement classes is equal to M, can be performed either in a multi-class setting, or by formulating M one-against-all problems, followed by $\frac{M(M-1)}{2}$ two-class problems. In this paper, three variants of the LDA algorithm are utilized. The first

is the traditional multi-class algorithm, that will be presented in Subsection 3.4.1. The second one is a multi-class algorithm similar to Weighted Piecewise Linear Discriminant Analysis (WPLDA) [39] to be described in Subsection 3.4.2. The third approach is a combination of one-versus-all and two-class LDA problems that will be described in Subsection 3.4.3. LDA algorithms specify the optimal discriminant subspace by minimizing:

$$\mathbf{\Psi}_{opt} = \arg\min_{\mathbf{\Psi}} \frac{trace\{\mathbf{\Psi}^T \mathbf{S}_w \mathbf{\Psi}\}}{trace\{\mathbf{\Psi}^T \mathbf{S}_b \mathbf{\Psi}\}}. \quad (7)$$

The matrix $\mathbf{\Psi}$ represents a linear transformation and $\mathbf{S}_b$, $\mathbf{S}_w$ are the between and within scatter matrices of the training movement vectors $\mathbf{s}_{cj},\ c = 1, ..., C,\ j = 1, ...N_c$ belonging to the C classes of the problem at hand [40], N_c being the number of training videos belonging to class c. In multi-class LDA problems, the number of classes C is equal to the number of movement classes, i.e., $C = M$. In each of the one-against-all LDA problems, the training movement vectors are divided in positive and negative training vector samples and the number of classes in each problem is $C = 2$. Finally, in the one-versus-one case, the training movement classes are divided in $\frac{M(M-1)}{2}$ movement class pairs and, thus, the number of classes in each of them is $C = 2$. The rank of $\mathbf{S}_w$ is at most $N_c - K$. $\mathbf{S}_w$ is invertible, if the number N_c of the training videos belonging to class c is adequately larger than the number of dynemes K. The optimal matrix $\mathbf{\Psi}_{opt}$ is formed by the $C - 1$ generalized eigenvectors that correspond to the largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$. In the case where $\mathbf{S}_w$ is not invertible, i.e., when the number of training samples belonging to every problem class is smaller than the dimensionality of the dyneme space ($N_c < K$), one of the LDA variants described in [41, 42] can be used instead.

3.4.1. Multi-Class LDA

The training movement vectors $\mathbf{s}_{mj},\ m = 1, ..., M,\ j = 1, ...N_m$ that represent each movement video are labeled. The application of multi-class LDA will result to the projection of every movement vector $\mathbf{s}_{mj} \in \mathbb{R}^K$ in a $(M-1)$-dimensional discriminant space $\mathbb{R}^{M-1}$, producing discriminant movement vectors $\mathbf{y}_{mj} \in \mathbb{R}^{M-1}$ $\mathbf{y}_{mj} = \mathbf{\Psi}_{opt}^T \mathbf{s}_{mj}$. In this space, discriminant movement vectors $\mathbf{y}_{mj}$ belonging to different movement classes are well separated. The optimal matrix $\mathbf{\Psi}_{opt}$ is formed by the $M-1$ generalized eigenvectors that correspond to the largest

eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$. In this case, the two scatter matrices mentioned above are:

$$\mathbf{S}_w = \sum_{m=1}^{M}\sum_{j=1}^{N_m} \frac{(\mathbf{s}_{mj} - \boldsymbol{\mu}_m)(\mathbf{s}_{mj} - \boldsymbol{\mu}_m)^T}{N_m} \tag{8}$$

$$\mathbf{S}_b = \sum_{m=1}^{M} \frac{(\boldsymbol{\mu}_m - \boldsymbol{\mu})(\boldsymbol{\mu}_m - \boldsymbol{\mu})^T}{N_m} \tag{9}$$

where $\boldsymbol{\mu}_m$ is the mean vector of class m, and $\boldsymbol{\mu}$ is the mean vector of all the training movement vectors and N_m is the number of movement vectors belonging to class m.

In the testing phase, the movement vector $\mathbf{s}$, representing the test multi-view movement video, is mapped to the discriminant movement subspace $\mathbb{R}^{M-1}$. In this space, the reduced dimensionality vector $\mathbf{y} = \boldsymbol{\Psi}_{opt}^T\mathbf{s}$ is classified to the nearest class centroid, using either Euclidean or Mahalanobis distance.

3.4.2. Weighted Piecewise Multi-Class Linear Discriminant Analysis (WPLDA)

In WPLDA the training movement vectors $\mathbf{s}_{mj} \in \mathbb{R}^K$, of dimensionality equal to the number of dynemes K, are broken down in lower dimensionality feature vectors $\mathbf{s}_{mjn} \in \mathbb{R}^{K_s}$, such that $\mathbf{s}_{mj} = [\mathbf{s}_{mj1}^T, ...\mathbf{s}_{mjn}^T...\mathbf{s}_{mjN_s}^T]^T$, thus creating N_s subsets of feature vectors, each having dimensionality $K_s = \frac{K}{N_s},\ n = 1, ..., N_s$. N_s is chosen in such a way, so that the number of training movement vectors in each class is sufficient for applying LDA, that is, the resulting scatter matrices are invertible, i.e., $N_m > K_s \geq M - 1$. By applying LDA, using the scatter matrices defined similarly as in (8) and (9), discriminant feature vectors $\mathbf{y}_{mjn} \in \mathbb{R}^{M-1}$ are produced. The resulting discriminant features $\mathbf{y}_{mjn}$ are concatenated forming $\mathbf{y}_{mj} = [\mathbf{y}_{mj1}^T, ...\mathbf{y}_{mjn}^T...\mathbf{y}_{mjN_s}^T]^T$ and a second LDA step is applied. This results to the generation of the discriminant movement vectors $\mathbf{z}_{mj} \in \mathbb{R}^{M-1}$ that represent each movement video in this discriminant subspace. The procedure described above is illustrated in Figure 6.

After the determination of the final discriminant subspace, all training movement vectors $\mathbf{s}_{mj},\ m = 1, ..., M,\ \ j = 1, ..., N_m$ are mapped to this subspace and the class centroids are found.

In the test phase, the movement vector $\mathbf{s}$, representing the test multi-view movement video, is broken down to N_s vectors $\mathbf{s}_n,\ \ n = 1, ..., N_s$, which are mapped in discriminant subspaces to produce N_s discriminant feature vectors. These feature vectors are concatenated and a second mapping to a discriminant subspace is applied. This results to the generation of the discriminant movement

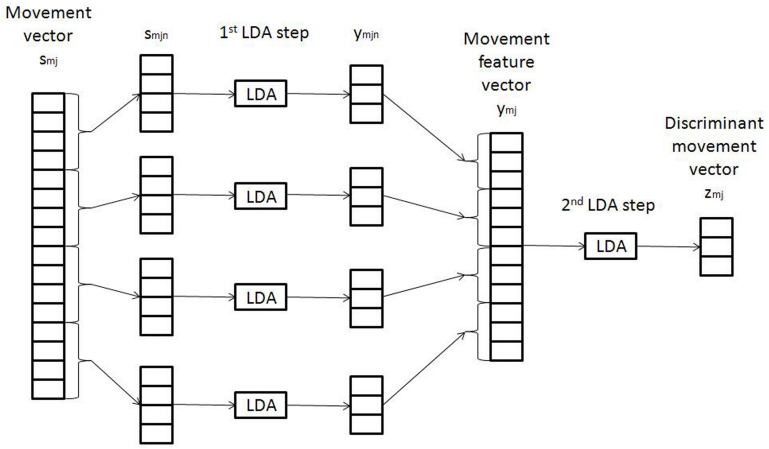


Figure 6: *Discriminant movement representation using two LDA steps.*

vector $\mathbf{z}$ that represents the movement video in this feature space. Then $\mathbf{z}$ is classified to the nearest centroid, using either Euclidean or Mahalanobis distance.

3.4.3. One-versus-all plus two-class LDAs

The M-class classification problem can be split in M one-against-all problems, followed by $\frac{M(M-1)}{2}$ two-class classification problems. For all these problems, the dimensionality of the training vectors is reduced to 1. In other words, after the projection, the sample $\mathbf{s}_{mj},\ m = 1, ..., M,\ j = 1, ..., N_m$ becomes a scalar: $y_{mj} = \mathbf{\Psi}_{opt}\mathbf{s}_{mj}$. For each of the one-versus-all problems, the movement vectors belonging to the specified movement class are used as positive samples, while the remaining movement vectors are used as negative samples. The scatter matrices are defined in the form:

$$\mathbf{S}_w = \sum_p \sum_{j=1}^{N_p} \frac{(\mathbf{s}_{pj} - \boldsymbol{\mu}_p)(\mathbf{s}_{pj} - \boldsymbol{\mu}_p)^T}{N_p} \tag{10}$$

where p indexes the positive and the negative training samples for each one-against all problems and

$$\mathbf{S}_b = (\boldsymbol{\mu}_p - \boldsymbol{\mu}_n)(\boldsymbol{\mu}_p - \boldsymbol{\mu}_n)^T \tag{11}$$

where $\boldsymbol{\mu}_p$, $\boldsymbol{\mu}_n$ are the mean vectors of the positive and the negative classes, respectively. These projections are combined with nearest centroid classification.

In the second step, if more than two classifiers decide positively for a given test sample, then the final classification decision is taken according to the projections that use information only from these classes. Thus $\frac{M(M-1)}{2}$ LDA projection vectors are calculated, one for each pair of classes. For each of the $\frac{M(M-1)}{2}$ two-class LDA problems for movement classes $p, q \in \{1, ..., M\}$, the scatter matrices are defined as follows:

$$\mathbf{S}_w = \sum_i \sum_{j=1}^{N_i} \frac{(\mathbf{s}_{ij} - \boldsymbol{\mu}_i)(\mathbf{s}_{ij} - \boldsymbol{\mu}_i)^T}{N_i} \tag{12}$$

$$\mathbf{S}_b = (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^T \tag{13}$$

where i takes values in $\{p, q\}$ and $\boldsymbol{\mu}_p$, $\boldsymbol{\mu}_q$ are mean movement vectors of the $p,\ q$ movement classes, respectively.

In the test phase, the movement vector $\mathbf{s}$, representing the test multi-view movement video, is mapped to each of the discriminant subspaces that were determined in the training phase for all the one-versus-all LDA problems and is classified to a movement class depending on its Euclidean or Mahalanobis distance from the class centroid of every positive movement class defined in the corresponding discriminant subspace. In the case of only one positive classification result, the movement vector $\mathbf{s}$ is classified to the corresponding movement class. In the case of more than one positive classification results, the movement vector $\mathbf{s}$ is mapped to the discriminant subspaces specified for the two-class problems that correspond to every couple of the previously recognized movements classes. In these spaces, the movement vector is classified in one of the two classes, according to the Euclidean or Mahalanobis distance from discriminant movement prototypes. This procedure is repeated until the final classification of the test movement video in one movement class. An example is illustrated in Figure 7 in order to better explain the procedure followed in the one-versus-all plus two-class classification problem. In this example, a multi-view video depicting a person performing the movement 'run' is mapped to the discriminant subspaces determined by the one-versus-all problems, which separate every movement from all others. Euclidean, or Mahalanobis distance from the positive discriminant movement prototype defines if this sequence is recognized to contain the examined movement or not. In this example, the one-versus-all classification problems related to the movements walk and run produced positive results, while the rest ones produced negative ones (not jump, not bend, etc). In this case, the two-class problem that separates 'walk' from 'run' is used to recognize the correct movement class. The movement vector

describing the video is mapped to the discriminant subspace that separates movements walk and run. In this space, the Euclidean or Mahalanobis distance from class centers is used to conclude that the test movement is a 'run'.

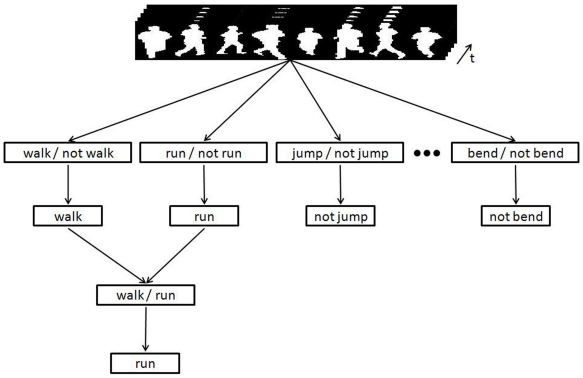


Figure 7: *An example of the one-versus-all plus two-class classification procedure.*

3.5. Camera Viewpoint Identification Problem

As previously described, the arrangement of the N single-view movement videos within a test N-view movement video should be consistent to the camera viewpoint arrangement used during the training phase to form N-view posture vectors. This means that the first video should correspond to the frontal view and all other ones should follow in a clock-wise manner (e.g. 45^o, right side view, 135^o, etc). Obviously, such a camera arrangement does not necessary hold for a newly acquired multi-view movement video. Thus, the camera viewpoint identification problem should be solved, before the recognition (test) process starts. To this end, two solutions are proposed. The first one discovers the arrangement of views corresponding to that used in the training procedure, while the second one exploits the circular invariance property of the DFT. Both methods achieve a view independent posture vector representation, which results to view independent human movement representation and recognition.

3.5.1. Multi-view posture vector rearrangement

As already mentioned, in the training phase, all available views of every N-view movement video are manually arranged. After this procedure, all training N-view posture vectors $\mathbf{p}_i$ depict the movement in a consistent way, i.e., by placing the frontal view first, followed by all other views in a clockwise manner. This results to the construction of consistent N-view dynemes.

In the recognition (test) phase, the arrangement of the posture vectors should be the same as in the training phase. To achieve this in the general case, all possible arrangements of the N available views should be considered. Fortunately, in most cases (including the camera setup used in this paper) the spatial relationship between cameras is a priori known, i.e., camera $\#1$ is followed by camera $\#2$, camera $\#2$ is followed by camera $\#3$ in a clockwise manner and so on. Thus, the number of arrangements that should be examined is equal to the number of the cameras N and are obtained by applying a block circular shift $\mathbf{p}'_{ij} = \mathbf{p}_{ij'}$, $j' = N - j \, mod \, N$ of the N-view posture vector elements (single-view binary posture vectors).

To obtain the correct arrangement of single-view posture vectors $\mathbf{p}'_{ij}$, $i = 1, ..., N_{t_m}$, $j = 1, ..., N$, of an incoming test N-view movement video, all N block shifted N-view posture vectors $\mathbf{p}'_i = [\mathbf{p}'^T_{i1}, \mathbf{p}'^T_{i2}, ..., \mathbf{p}'^T_{iN}]^T$, $i = 1, ..., N_{t_m}$ are compared with every N-view dyneme obtained in the training phase. This procedure is applied to all the N-view multi-view posture vectors comprising the test movement video and involves computing the Euclidean distance between every dyneme vector and each circularly shifted posture vector $\mathbf{p}'_{ij}$ of the incoming movement video. The shifted version $\mathbf{p}'_{ij}$ that provides the minimum distance indicates the correct view order and all posture vectors are rearranged with respect to this view order:

$$\arg \min_{j} \min_{ik} \| \, \mathbf{p}_{ij} - \mathbf{v}_k \, \|_2, \quad i = 1, ..., N_{t_m}, \quad j = 1, ..., N, \quad , k = 1, ..., K. \tag{14}$$

where $\mathbf{v}_k$ denotes the k-th dyneme.

3.5.2. Fourier view-invariant posture representation

A new, view-invariant posture representation is proposed to solve the camera viewpoint identification problem. This representation exploits the circular shift invariance of the magnitude of DFT coefficients:

$$P(k) = | \sum_{n=0}^{N_s - 1} p(n) e^{-i \frac{2\pi k}{N_s} n} |, \quad k = 1, ..., N_s - 1. \tag{15}$$

The view-invariant posture vector representation is obtained by concatenating all the N single-view posture vectors $\mathbf{p}_j$, $j = 1, ..., N$ in a single vector $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, ..., \mathbf{p}_N^T]^T$, in the order they are produced by the cameras and computing the magnitude of its Discrete Fourier transform coefficients $P(k)$. In the case of a multi-camera setting, only circular shifts by $H \times W$ elements (entire single-view frames) have a physical meaning. It will be proven experimentally in section

4 that the transformed posture vector representation is indeed robust to camera viewpoint circular shifts.

3.6. Movement Classification (test phase)

To classify an unknown N-view video containing N_{t_m} binary masks of a moving person from each of the N views, the ROI of every body posture ROI is centered at the person's center of mass and binary single-view posture videos of frame size equal to the maximum ROI that encloses the person's body are created for every view. These are rescaled to the size $H \times W$ pixels used in the training phase (64×64 in the experiments presented in this paper) and vectorized to produce N single-view posture vectors. These vectors are concatenated, by placing the posture vector that corresponds to the first camera in the first position, followed by the single-view posture vectors that come from all other cameras in a clockwise manner. The final N-view posture vectors $\mathbf{p}_i,\ i = 1, ..., N_{t_m}$ are produced either by calculating the magnitude of the Discrete Fourier Transform of the N-view posture vectors using (15) or by solving the camera viewpoint identification problem using the circular shift procedure described in Subsection 3.5.1. The fuzzy distances $d_{ij},\ i = 1, ..., N_{t_m},\ j = 1, ..., K$ between the resulting N-view posture vectors $\mathbf{p}_i$ and the N-view dynemes $\mathbf{v}_j$ are calculated using (3) and combined in (4) in order to map all the N-view posture vectors from the input space to the dyneme space and produce the membership vectors $\mathbf{u}_i,\ i = 1, ..., N_{t_m}$. Membership vectors $\mathbf{u}_i$ are normalized using (5) and combined through (6) to produce the movement vector $\mathbf{s}$ for the multi-view test video, which is subsequently normalized using the mean and standard deviation vectors of the training movement vectors. Finally, the movement vector $\mathbf{s}$ is projected to the discriminant subspace specified in the training phase for one of the LDA variants described in Subsection 3.4 and is classified to one of the movement classes $m,\ 1 \leq m \leq M$.

3.7. Continuous movement recognition

As noted in Section 2, a movement recognition technique should not be confined to elementary movement recognition (i.e., over one movement period) but should allow continuous movement recognition over time. In order to achieve continuous operation, a sliding window can be utilized. Thus, for movement recognition at time instance t using a sliding window consisting of N_W frames, the video frames $\mathbf{f}_i,\ i = t, t-1, ..., t-N_W+1$ are used. Since the average length N_{t_m} of the elementary movement of different classes varies, the sliding window should contain a sufficient number N_W of video frames to enable the method to

correctly recognize movement classes that their elementary periods consist of different video frame numbers. By performing recognition at every sliding window position, a continuous recognition operation is achieved over time. The procedure described above is illustrated in Figure 8.

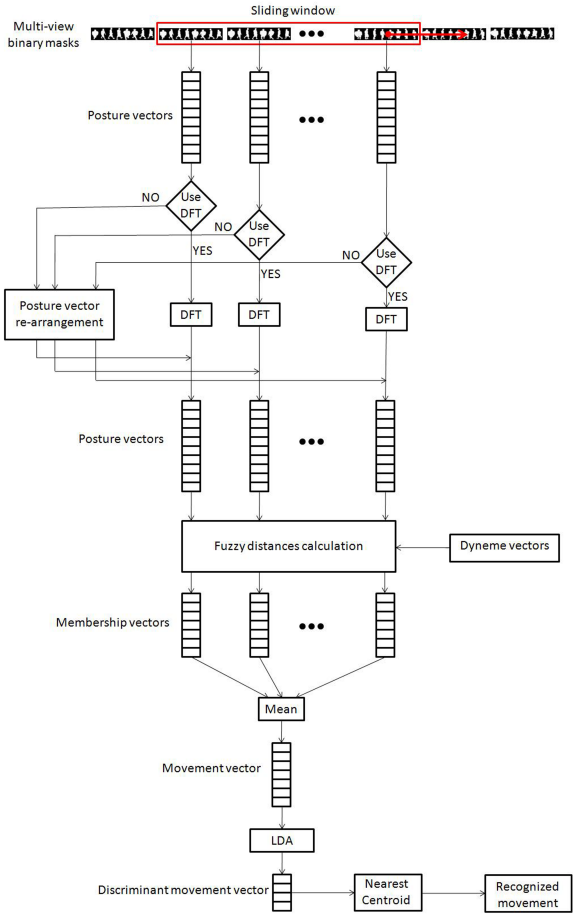


Figure 8: *Continuous movement recognition procedure.*

4. Experimental Results

In this Section, the experimental results produced on the i3DPost multi-view movement video database [43] are presented. Furthermore, the ability of the proposed method to perform continuous movement recognition and its robustness to synchronization errors that can occur in a multi-camera setup are demonstrated.

Finally, we compare our method with state of the art methods aiming to view-invariant movement recognition on the INRIA IXMAS multi-view movement recognition database [23].

4.1. The i3DPost multi-view database

The i3DPost multi-view movement video database contains 64 high-resolution 1920×1080 pixel image sequences of eight persons (six males and two females), each performing eight movements. Every movement is captured from eight views. The video capture took place in a studio at University of Surrey with blue background and capture volume dimensions of $4 \times 3 \times 2$ cubic meters. The cameras were positioned around the capture volume at a height of 2m above the studio floor and were equally spaced in a ring of 8 m diameter. In these 64 eight-view image sequences, the persons perform one or more periods of the following movements: 'walk' (wk), 'run' (rn), 'jump in place' (jp), 'jump forward' (jf), 'bend' (bd), 'fall' (fl), 'sit' (st) and 'wave one hand' (wo). Binary masks for every single-view image sequence were extracted by thresholding the blue color in the HSV color space, i.e., pixels with values $H > 200,\ \ S > 0.3,\ \ V > 0.1$ and $MAX(R, G, B) = B$ were set to zero (denoting background) and all remaining pixels were set to 1 (foreground).

4.2. The IXMAS multi-view database

The INRIA (Institut National de Recherche en Informatique et Automatique) IXMAS Motion Acquisition Sequences database contains 330 low resolution 291×390 pixel image sequences of ten persons (five males and five females), each performing eleven movements. Every movement is performed three times from each person and is captured from five views. The persons freely change position and orientation during movement execution. The movements performed are: 'check watch' (cw), 'cross arm' (ca), 'scratch head' (sh), 'sit down' (sd), 'get up' (gu), 'turn around' (tu), 'walk in a circle' (wk), 'wave hand' (wh), 'punch' (ph), 'kick' (kk), and 'pick up' (pu). Binary masks of the persons' body are provided by the database.

4.3. Cross-validation in i3DPost multi-view database

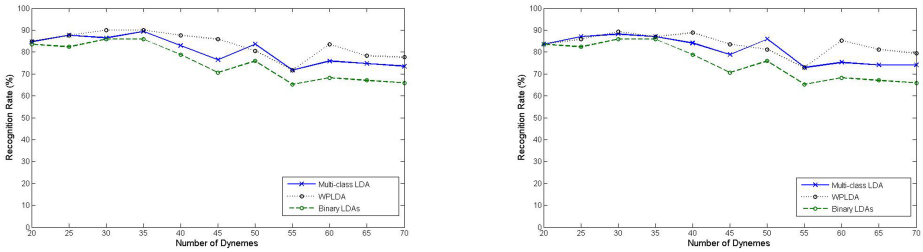
In an off-line preprocessing procedure, elementary videos containing one single movement period, e.g., one walk period, were manually created during both training and testing. These videos were further preprocessed, as discussed in Subsection 3.1, to produce videos containing single-view binary posture masks. In this preprocessing step, the dimensions required to contain the person's body in

all frames were determined in every video and bounding boxes of this size were extracted, centered at the person's center of mass and rescaled to $H \times W = 64 \times 64$ pixels for every video frame.

As previously described, the leave-one-person-out cross validation procedure has been used to identify the optimal number of multi-view dynemes. Thus, in every fold of this procedure preprocessed videos of seven persons were used for training and the videos of the eighth person were used for testing. The experiment included eight folds of the cross-validation procedure, one for each person left out. In the test phase, the order of the single-view movement videos followed the arrangement of the cameras, i.e., the first video was that from the camera $\#1$, followed by the videos of the rest of the cameras in a clockwise manner. In other words, the test videos were fed in a random order, in terms of the relative position of the views with respect to the person. The plots in Figures 9 and 10 illustrate the movement recognition rates obtained for every LDA variant described in Section 3.4 with respect to the number of eight-view dynemes for the posture vector representation in the spatio-temporal and the DFT domain, respectively. It can be seen that the multi-class approach outperforms the one-versus-all plus two-class classification approach. In the case of the one-versus-all plus two-class approach, the best recognition rate, 85.88%, was achieved using the DFT posture vector representation, 35 eight-view dynemes and the Euclidean distance. In the case of multi-class LDA, the best recognition rate, 89.41%, was achieved using the DFT posture vector representation, 25 eight-view dynemes and the Euclidean distance. The WPLDA approach proved to be the best, as it provides higher classification rates in most experiments. A 94.37% classification rate was achieved using the DFT posture vector representation and 60 eight-view dynemes by splitting the 60-dimensional movement vectors to six 10-dimensional vectors and using the Euclidean distance. To summarize: $1)$ the DFT posture vector representation followed by the Euclidean distance provide the best classification rates; $2)$ the multi-class approach outperforms the one-versus-all plus two-class approach; $3)$ the WPLDA approach outperforms the simple multi-class approach.

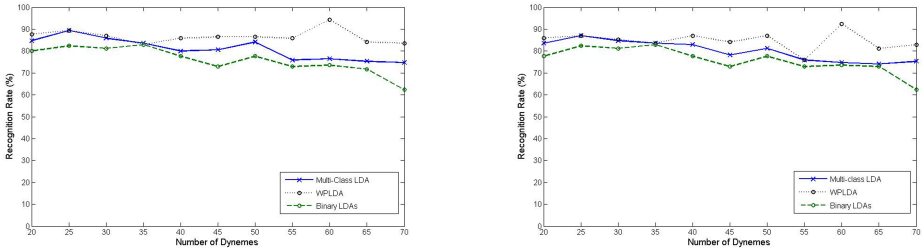
The confusion matrix corresponding to the optimal parameters and procedures (DFT posture vector representation, 60 eight-view dynemes, WPLDA movement vector, dimensionality reduction to 10 and Euclidean distance) is shown in Table 1. In this Table, a row represents the actual movement class and a column the movement recognized by the algorithm. As can be seen, movements which contain discriminant postures, e.g., 'walk', 'run', 'bend', 'fall' and 'wave one hand', are well separated and classified with 100% recognition accuracy. Movements that share a large number of similar postures, i.e. 'jump in place', 'jump forward'

and 'sit' are more difficult to recognize. Movement 'jump in place' is confused to 'jump forward' and to 'sit' in 12% and 8% of the cases respectively. Movement 'sit' is misclassified as 'jump in place' in 25% of the cases.



(a) (b)

Figure 9: *Movement recognition rate vs the number of dynemes using movement posture vector rearrangement combined with: a) Euclidean, b) Mahalanobis distance.*



(a) (b)

Figure 10: *Movement recognition rate vs the number of dynemes for Fourier representation combined with: a) Euclidean, b) Mahalanobis distance.*

4.4. Continuous movement recognition

This section illustrates the capability of the proposed method to perform continuous recognition. A multiple movement video depicting one of the persons performing ten iterations of every elementary movement was used for this study. This video was created by concatenating video segments that depict the person performing elementary movements. The movement recognition algorithm was trained using as training samples the binary videos of the remaining seven persons. The DFT posture vector representation, 60 eight-view dynemes, movement vectors broken down to 6 10-dimensional vectors and the Euclidean distance were

Table 1: Confusion matrix containing classification rates (%) in movement recognition on the i3DPost database.

actual \ recognized	wk	rn	jp	jf	bd	st	fl	wo
wk	100							
rn		100						
jp			80	12		8		
jf				100				
bd					100			
st			25			75		
fl							100	
wo								100

used. A sliding window was employed and recognition was performed at every sliding window position. Since the length N_{t_m} of the elementary movement periods of different classes varies in the range of 9 to 40, it was decided to use $N_W = 21$ video frames within the sliding window, so that the window contains a sufficient number of frames to accommodate lengthier movements. The larger the window length N_W used, the bigger the movement recognition accuracy is, when the person performs only one movement. However, a large N_W requires bigger computational effort, postpones the first classification decision till N_W frames are available and has bigger problems at the transition between two different movements. Therefore, $N_W = 21$ was experimentally found to be a good trade off. Figure 11a) illustrates the results of this experiment. In this Figure, the movement label ground truth is illustrated by a continuous line and the recognition results by a dashed line. It can be seen that no movement was recognized for the first 20 frames, as the algorithm needs at least 21 frames to perform recognition ($N_W = 21$). Furthermore, it can be seen that correct classification of the new movement is delayed up to N_W frames in the transition between two movement classes. Despite the fact that $N_W = 21$ was chosen, which is significantly lower than the duration of the elementary bend ($N_{t_m} = 40$), this movement was correctly classified, because the body postures that appear in a bend uniquely sequence characterize this movement. Therefore, a few video frames are sufficient for its recognition. As expected, the only recognition errors occurred during movement transitions. To eliminate these errors, a majority voting filter over the recognition results has been used inside a window of length $N_W = 21$ frames. Figure 11b) illustrates the results achieved after majority voting. As it can be

seen, the continuous movement recognition results are very satisfactory.

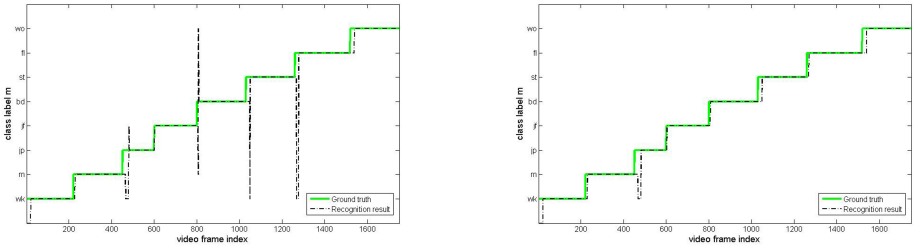


Figure 11: *Continuous movement recognition results: a) output of the algorithm, b) after applying a majority voting filter.*

4.5. Robustness against synchronization errors

It was noted in Section 3 that the cameras used to capture multi-view movement videos should be synchronized. However, in multi-camera setups, synchronization errors are rather frequent, leading to arbitrary time lags between the frames of different cameras. This section illustrates the robustness of the proposed method to these errors.

To do so, an experiment was performed, where desynchronized videos are tested in a system that has been trained using synchronized videos. The test video was desynchronized by shifting the video frames of different cameras in time by a random shift, with respect to the frames captured by camera $\#1$. Two eight-view posture frames that correspond to the same time instance can be seen in Figure 12. The top one is synchronized, while the bottom one is desynchronized, with synchronization errors equal to 0, 2, 7, 4, 5, 4, 6 and 6 frames, respectively (from left to right).

Figure 12: *Synchronized (a) and desynchronized (b) multi-view posture frame.*

The leave-one-person-out cross-validation procedure, using 60 eight-view dynemes, movement vectors broken down to six 10-dimensional vectors and the Euclidean distance, was applied for different desynchronization levels of the test

videos. The random frame shift followed a uniform distribution in the range $[0, n_D]$. The movement recognition accuracy achieved in these desinchronised videos versus the frame shift range n_D is illustrated in Figure 13. As can be seen, synchronization errors with range up to $n_D = 2$ frames shifts do not influence at all the performance of the proposed method. Moreover, desynchronization shifts of up to 7 frames shifts does not influence significantly the performance of the proposed approach. This can be explained by the fact that temporally adjacent body postures do not differ much. Furthermore, the proposed movement representation utilizes movement posture primitives and similar postures produce similar memberships on the dynemes. This results to similar movement vectors, which are finally classified to the correct movement class.

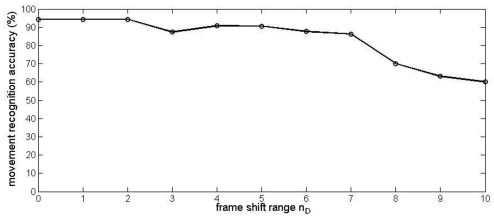


Figure 13: *Recognition rate vs synchronization error (in frames).*

4.6. Comparison with other methods

In order to compare our method with state of the art methods aiming to view-independent movement recognition we conducted experiments on the IXMAS multi-view movement recognition database using the same experimental setup. The leave-one-person-out cross validation procedure was performed. In every fold of this procedure preprocessed videos of nine persons were used for training and the videos of the tenth person were used for testing. The experiment included ten folds of the cross-validation procedure, one for each person left out. In the test phase, the test videos were fed in a random order, in terms of the relative position of the views with respect to the person. Because the camera setup used in the database does not provide a 360^o coverage of the scene, the DFT posture vector representation is not applicable. In order to obtain view invariant posture vector representation, the posture vector rearrangement procedure described in subsection 3.5.1 was applied, examining all possible single view posture vector rearrangements. A 83.47% classification rate was achieved using 80 five-view dynemes by splitting the 80-dimensional movement vectors to eight 10-

dimensional vectors and using the Euclidean distance. The corresponding confusion matrix is shown in Table 2. Table 3 contains comparison results with several other methods proposed in the literature. As can be seen, the proposed method achieves state of the art movement classification rates.

Table 2: Confusion matrix containing classification rates (%) in movement recognition on the IXMAS database.

	cw	ca	sh	sd	gu	tu	wk	wh	ph	kk	pu
cw	75.7	15.2	9.1								
ca	6.1	81.8	12.1								
sh	3.1	9.1	84.8								
sd				84.9	10.1	3.0					
gu				12.1	75.7	3.0		6.1			6.1
tu						93.9	6.1				
wk							100				
wh	9.1	6.1	9.1					72.7	3.0		
ph	15.1								63.7	21.2	
kk										100	
pu				3.0	9.1				3.0		84.9

Table 3: Comparison results in the IXMAS five-view action recognition database.

Method	**Accuracy**
Method in [44]	81.3%
Method in [45]	81%
Method in [46]	80.6%
Proposed method	83.47%

5. Discussion and Conclusion

In this paper, a novel view-invariant human movement representation and recognition method that exploits synchronized and uncalibrated multi-view video was presented. View-invariant representation is achieved either by circular shifts of the available views or by exploiting the circular shift invariance property of DFT. Three variants of the LDA projection method were evaluated using these movement representations. It has experimentally been found that the multi-class

classification approach outperforms the one-against-all and the two-class classification procedure. The use of a discriminant feature representation leads to well separated movement classes and a simple Nearest Centroid classification algorithm is sufficient to provide correct classification. The use of a low-computation 3D posture representation combined with the movement representation in a low dimensional discriminant space results to a fast movement recognition method, which achieves high recognition rates and is not affected by movement speed variations across persons. The proposed approach can be easily applied for continuous movement recognition, can tolerate moderate camera synchronization errors, and performs better than other state of the art methods operating on multi-view video sets.

Acknowledgment

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).

References

[1] L. Weilun, H. Jungong, P. With, Flexible Human Behavior Analysis Framework for Video Surveillance Applications, International Journal of Digital Multimedia Broadcasting 2010 (2010) 9, ISSN 1687-7578.

[2] P. Barr, J. Noble, R. Biddle, Video game values: Human-computer interaction and games, Interacting with Computers 19 (2) (2007) 180–195.

[3] B. Song, E. Tuncel, A. Chowdhury, Towards A Multi-Terminal Video Compression Algorithm By Integrating Distributed Source Coding With Geometrical Constraints, Journal of Multimedia 2 (3) (2007) 9–16.

[4] T. Hfllerer, S. Feiner, D. Hallaway, B. Bell, M. Lanzagorta, D. Brown, S. Julier, Y. Baillot, L. Rosenblum, User interface management techniques for collaborative mobile augmented reality, Computers and Graphics 25 (5) (2001) 799–810.

[5] A. Bobick, Movement, activity and action: the role of knowledge in the perception of motion, Philosophical Transactions of the Royal Society B: Biological Sciences 352 (1358) (1997) 1257–1265.

[6] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analalysis and Machine Intelligence 23 (3) (2001) 257–267.

[7] R. Green, L. Guan, Quantifying and recognizing human movement patterns from monocular video images-part I: A new framework for modeling human motions, IEEE Transactions on Circuits and Systems for Video Technology 14 (2) (2004) 179–190.

[8] H. Dudley, S. Balashek, Automatic recognition of phonetic patterns in speech, The Journal of the Acoustical Society of America 30 (1958) 721 pages.

[9] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1473–1488.

[10] X. Ji, H. Liu, Advances in View-Invariant Human Motion Analysis: A Review, IEEE Transactions on Systems, Man and Cybernetics Part–C 40 (1) (2010) 13–24.

[11] X. Feng, P. Perona, Human action recognition by sequence of movelet codewords, in: Proceedings of the 1st International Symposioum on 3D Data Processing Visualization and Transmission, 717–733, 2002.

[12] Q. Wei, W. Hu, X. Zhang, G. Luo, Dominant sets-based action recognition using image sequence matching, in: Proceedings of IEEE International Conference on Image Processing, vol. 6, 113–136, 2007.

[13] R. Green, L. Guan, Quantifying and recognizing human movement patterns from monocular video images-part II: Applications to biometrics, IEEE Transactions on Circuits and Systems for Video Technology 14 (2) (2004) 191–198.

[14] L. Wang, D. Suter, Learning and matching of dynamic shape manifolds for human action recognition, IEEE Transactions on Image Processing 16 (6) (2007) 1646–1661.

[15] N. Gkalelis, A. Tefas, I. Pitas, Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition,

IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1511–1521.

[16] T. Xiang, S. Gong, Beyond tracking: Modelling activity and understanding behaviour, International Journal of Computer Vision 67 (1) (2006) 21–51, ISSN 0920-5691.

[17] J. Niebles, L. Fei-Fei, View-Invariant representation and recognition of actions, International Journal of Computer Vision 50 (2) (2002) 203–226.

[18] V. Parameswaran, R. Chellappa, View invariants for human action recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 613–619, 2003.

[19] A. Yilmaz, M. Shah, Actions sketch: a novel action representation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 984–989, 2005.

[20] D. Weinland, F. Grenoble, E. Boyer, R. Ronfard, A. Inc, Action recognition from arbitary views using 3D exemplars, in: Proceedings of IEEE Conference on Computer Vision, 1–7, 2007.

[21] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of IEEE Conference on Machine Learning, 282–289, 2001.

[22] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: Proceedings of IEEE Conference on Computer Vision, vol. 2, 1395–1402, 2005.

[23] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, Computer Vision and Image Understanding 104 (2–3) (2006) 249–257.

[24] D. Weinland, R. Ronfard, E. Boyer, Automatic discovery of action taxonomies from multiple views, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 1639–1645, 2006.

[25] M. Ahmad, S. Lee, Human action recognition using shape and CLG-motion flow from multi-view image sequences, Pattern Recognition 41 (7) (2008) 2237–2252.

[26] M. Seki, T. Wada, H. Fujiwara, K. Sumi, Background subtraction based on the cooccurrence of image variations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 65–72, 2003.

[27] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 246–252, 1999.

[28] A. Agarwal, B. Triggs, Recovering 3D human pose from monocular images, IEEE transactions on pattern analysis and machine intelligence (2006) 44–58ISSN 0162-8828.

[29] M. Lee, R. Nevatia, Human pose tracking in monocular sequence using multilevel structured models, IEEE transactions on pattern analysis and machine intelligence (2008) 27–38ISSN 0162-8828.

[30] B. Lei, L. Xu, Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management, Pattern recognition letters 27 (15) (2006) 1816–1825, ISSN 0167-8655.

[31] J. Xue, N. Zheng, J. Geng, X. Zhong, Tracking multiple visual targets via particle-based belief propagation, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 38 (1) (2008) 196–209, ISSN 1083-4419.

[32] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference, 849–856, 2001.

[33] U. Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416, ISSN 0960-3174.

[34] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (9) (2002) 1464–1480, ISSN 0018-9219.

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

[36] A. Webb, Statistical Pattern Recognition, 2nd ed, Wiley, 2002.

[37] P. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, 1982.

[38] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd ed, Wiley-Interscience, 2000.

[39] M. Kyperountas, A. Tefas, I. Pitas, Weighted piecewise LDA for solving the small sample size problem in face verification, IEEE Transactions on Neural Networks 18 (2) (2007) 506–519.

[40] H. Oja, S. Sirkia, J. Eriksson, Scatter matrices and independent component analysis, Austrian Journal of Statistics 35 (2–3) (2006) 175–189.

[41] J. Yang, A. Frangi, J. Yang, D. Zhang, Z. Jin, KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2) (2005) 230–244.

[42] M. Zhu, A. Martinez, Subclass discriminant analyzis, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (8) (2006) 1274–1286.

[43] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, I. Pitas, The i3DPost multi-view and 3D human action/interaction database, in: 6th Conference on Visual Media Production, 159–168, 2009.

[44] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3d exemplars, in: in Proceedings International Conference Computer Vision, IEEE, 1–7, 2007.

[45] D. Tran, A. Sorokin, Human activity recognition with metric learning, Computer Vision–ECCV 2008 (2008) 548–561.

[46] F. Lv, R. Nevatia, Single view human action recognition using key pose matching and viterbi path searching, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 1–8, 2007.