View-invariant action recognition based on Artificial Neural Networks

Alexandros Iosifidis, Anastasios Tefas, Member, IEEE, and Ioannis Pitas, Fellow, IEEE

Abstract—**In this paper, a novel view invariant action recognition method based on neural network representation and recognition is proposed. The novel representation of action videos is based on learning spatially related human body posture prototypes using Self Organizing Maps (SOM). Fuzzy distances from human body posture prototypes are used to produce a time invariant action representation. Multilayer perceptrons are used for action classification. The algorithm is trained using data from a multi-camera setup. An arbitrary number of cameras can be used in order to recognize actions using a Bayesian framework. The proposed method can also be applied to videos depicting interactions between humans, without any modification. The use of information captured from different viewing angles leads to high classification performance. The proposed method is the first one that has been tested in challenging experimental setups, a fact that denotes its effectiveness to deal with most of the open issues in action recognition.**

Index Terms—**Human action recognition, Fuzzy Vector Quantization, Multi-layer Perceptrons, Bayesian Frameworks.**

I. Introduction

Human action recognition is an active research field, due to its importance in a wide range of applications, such as intelligent surveillance [1], human-computer interaction [2], content-based video compression and retrieval [3], augmented reality [4], etc. The term action is often confused with the terms activity and movement. An action (sometimes also called as movement) is referred to as a single period of a human motion pattern, such as a walking step. Activities consist of a number of actions/movements, i.e., dancing consists of successive repetitions of several actions, e.g. walk, jump, wave hand, etc. Actions are usually described by using either features based on motion information and optical flow [5], [6], or features devised mainly for action representation [7], [8]. Although the use of such features leads to satisfactory action recognition results, their computation is expensive. Thus, in order to perform action recognition at high frame rates, the use of simpler action representations is required. Neurobiological studies [9] have concluded that the human brain can perceive actions by observing only the human body poses (postures) during action execution. Thus, actions can be described as sequences of consecutive human body poses, in terms of human body silhouettes [10], [11], [12].

After describing actions, action classes are, usually, learned by training pattern recognition algorithms, such as Artificial Neural Networks (ANNs) [13], [14], [15], Support Vector Machines (SVMs) [16], [17] and Discriminant dimensionality reduction techniques [18]. In most applications, the camera viewing angle is not fixed and human actions are observed from arbitrary camera viewpoints. Several researchers have highlighted the significant impact of the camera viewing angle variations on the action recognition performance [19], [20]. This is the so-called viewing angle effect. To provide view-independent methods, the use of multi-camera setups has been adopted [21], [22], [23]. By observing the human body from different viewing angles, a view-invariant action representation is obtained. This representation is subsequently used to describe and recognize actions.

Although multi-view methods address the viewing angle effect properly, they set a restrictive recognition setup, which is difficult to be met in real systems [24]. Specifically, they assume the same camera setup in both training and recognition phases. Furthermore, the human under consideration must be visible from all synchronized cameras. However, an action recognition method should not be based on such assumptions, as several issues may arise in the recognition phase. Humans inside a scene may be visible from an arbitrary number of cameras and may be captured from an arbitrary viewing angle. Inspired from this setting, a novel approach in view-independent action recognition is proposed. Trying to solve the generic action recognition problem, a novel view-invariant action recognition method based on ANNs is proposed in this paper. The proposed approach does not require the use of the same number of cameras in the training and recognition phases. An action captured by an arbitrary number N of cameras, is described by a number of successive human body postures. The similarity of every human body posture to body posture prototypes, determined in the training phase by a self-organizing neural network, is used to provide a time invariant action representation. Action recognition is performed for each of the N cameras by using a Multi-Layer Perceptron (MLP), i.e., a feed-forward neural network. Action recognition results are subsequently combined to recognize the unknown action. The proposed method performs view-independent action recognition, using an uncalibrated multi-camera setup. The combination of the recognition outcomes that correspond to different viewing angles leads to action recognition with high recognition accuracy.

The main contributions of this paper are: a) the use of Self Organizing Maps (SOM) for identifying the basic posture prototypes of all the actions, b) the use of cumulative fuzzy distances from the SOM in order to achieve time-invariant action representations, c) the use of a Bayesian framework to combine the recognition results produced for each camera, d) the solution of the camera viewing angle identification problem using combined neural networks.

The remainder of this paper is structured as follows. An

A. Iosifidis, A. Tefas and I. Pitas are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. e-mail: {aiosif,tefas,pitas}@aiia.csd.auth.gr.

overview of the recognition framework used in the proposed approach and a small discussion concerning the action recognition task is given in Section I-A. Section II presents details of the processing steps performed in the proposed method. Experiments for assessing the performance of the proposed method are described in Section III. Finally, conclusions are drawn in Section IV.

A. Problem Statement

Let an arbitrary number of N_C cameras capturing a scene at a given time instance. These cameras form a N_C-view camera setup. This camera setup can be a converging one or not. In the first case, the space which can be captured by all the N_C cameras is referred as capture volume. In the later case, the cameras forming the camera setup are placed in such positions that there is not a space which can be simultaneously captured by all the cameras. A converging and a non-converging camera setup is illustrated in Figure 1. N_t video frames from a specific camera $\mathbf{f}_i,\ i = 1, ..., N_t$, form a single-view video $\mathbf{f} = [\mathbf{f}_1^T, \mathbf{f}_2^T, ..., \mathbf{f}_{N_t}^T]^T$.

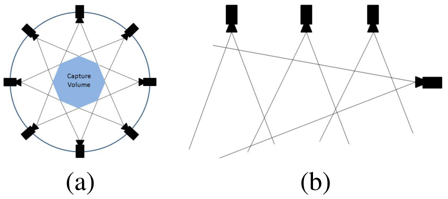


Fig. 1. *a) A converging and b) a non-converging camera setup.*

Actions can be periodic (e.g., walk, run) or not (e.g., bend, sit). The term elementary action refers to a single human action pattern. In the case of periodic actions, the term elementary action refers to a single period of the motion pattern, such as a walk step. In the case of non-periodic actions, the term elementary action refers to the whole motion pattern, i.e., a bend sequence. Let $\mathcal{A}$ be a set of N_A elementary action classes $\{a_1, ..., a_{N_A}\}$, such as walk, jump, run, etc. Let a person perform an elementary action $a_j,\ 1 \leq j \leq N_A$, captured by $N < N_C$ cameras. This results to the creation of N single-view action videos $\mathbf{f}_i = [\mathbf{f}_{i1}^T, \mathbf{f}_{i2}^T, ..., \mathbf{f}_{iN_{t_j}}^T]^T,\ i = 1, ..., N$ each depicting the action from a different viewing angle. Since different elementary actions have different durations, the number of video frames $N_{t_j},\ j = 1, ..., N_A$ that consist elementary action videos of different action classes varies. For example, a 'run' period consists of only 10 video frames, whereas a 'sit' sequence consists of 40 video frames in average at 25 fps. The action recognition task is the classification of one or more action videos, depicting an unknown person performing an elementary action in one of the known action classes specified by the action class set $\mathcal{A}$.

In the following, we present the main challenges for an action recognition method:

- The person can be seen from $N \leq N_C$ cameras. The case $N < N_C$ can occur either when the used camera setup is not a converging one, or when using a converging camera setup, the person performs the action outside the capture volume, or in the case of occlusions.
- During the action execution, the person may change movement direction. This affects the viewing angle he/she is captured from each camera. The identification of the camera position with respect to the person body is referred as the camera viewing angle identification problem and needs to be solved in order to perform view-independent action recognition.
- Each camera may capture the person from an arbitrary distance. This affects the size of the human body projection in each camera plane.
- Elementary actions differ in duration. This is observed in different realizations of the same action performed by different persons or even by the same person at different times or under different circumstances.
- The method should allow continuous action recognition over time.
- Elementary action classes highly overlap in the video frame space, i.e., the same body postures may appear in different actions. For example many postures of classes 'jump in place' and 'jump forward' are identical. Furthermore, variations in style can be observed between two different realizations of the same action performed either by the same person or by different persons. Considering these observations, the body postures of a person performing one action may be the same to the body postures of another person performing a different action. Moreover, there are certain body postures that characterize uniquely certain action classes. An action representation should take into account all these observations in order to lead to a good action recognition method.
- Cameras forming the camera setup may differ in resolution and frame rate and synchronization errors may occur in real camera setups, that is, there might be a delay between the video frames produced by different cameras.
- The use of multi-camera setups involves the need of camera calibration for a specific camera setting. The action recognition algorithms need to be retrained even for small variations of the camera positions.

The objective is the combination of all available information coming from all the available cameras depicting the person under consideration to recognize the unknown action. The proposed method copes with all the above mentioned issues and constraints. According to the best of the authors knowledge this is the first time where an action recognition method is tested against all these scenarios with very good performance.

II. Proposed Method

In this section, each step of the proposed method is described in detail. The extraction of posture data, used as input data in the remaining steps, is presented in subsection II-A. The use of a Self Organizing Map (SOM) to determine human body posture prototypes is described in subsection II-B. Action representation and classification are presented in subsections II-C and II-D, respectively. A variation of the original algorithm, that exploits the observation's viewing angle information is presented in subsections II-E and II-F. Finally, subsection II-G presents the procedure followed in the recognition phase.

A. Preprocessing Phase

As previously described, an elementary action is captured by N cameras in elementary action videos consisting of N_{t_j}, $1 \leq j \leq N_A$, video frames that depict one action period. The number N_{t_j} may vary over action classes, as well as over elementary action videos coming from the same action class. Multi-period action videos are manually split in elementary action videos, which are subsequently used for training and testing in the elementary action recognition case. In the case of videos showing many action periods (continuous action recognition), a sliding window of possibly overlapping video segments having suitably chosen length N_{tw} is used and recognition is performed at each window position, as will be described in Section III-D. In the following, the term action video will refer to an elementary action video.

Moving object segmentation techniques [25], [26] are applied to each action video frame to create binary images depicting person's body in white and the background in black. These images are centered at the person's center of mass. Bounding boxes of size equal to the maximum bounding box enclosing person's body are extracted and rescaled to $N_H \times N_W$ pixels to produce binary posture images of fixed size. Eight binary posture images of eight actions ('walk', 'run', 'jump in place', 'jump forward', 'bend', 'sit', 'fall' and 'wave one hand') taken from various viewing angles are shown in Figure 2.

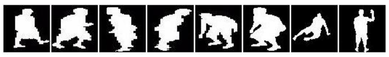


Fig. 2. *Posture images of eight actions taken from various viewing angles.*

Binary posture images are represented as matrices and these matrices are vectorized to produce posture vectors $\mathbf{p} \in \mathbb{R}^D$, $D = N_H \times N_W$. That is, each posture image is finally represented by a posture vector $\mathbf{p}$. Thus, every action video consisting of N_{t_j} video frames is represented by a set of posture vectors $\mathbf{p}_i \in \mathbb{R}^D$, $i = 1, ..., N_{t_j}$. In the experiments presented in this paper the values $N_H = 64$, $N_W = 64$ have been used and binary posture images were scanned columnwise.

B. Posture prototypes Identification

In the training phase, posture vectors $\mathbf{p}_i$, $i = 1, ..., N_p$, N_p being the total number of posture vectors consisting all the N_T training action videos, having N_{t_j} video frames each, are used to produce action independent posture prototypes without exploiting the known action labels. To produce spatially related posture prototypes, a SOM is used [27]. The use of SOM leads to a topographic map (lattice) of the input data, in which the spatial locations of the resulting prototypes in the lattice are indicative of intrinsic statistical features of the input postures. The training procedure for constructing the SOM is based on three procedures:

1) Competition: For each of the training posture vectors $\mathbf{p}_i$, its Euclidean distance from every SOM weight, $\mathbf{w}_{Sj} \in \mathbb{R}^D$, $j = 1, ..., N_S$ is calculated. The winning neuron is the one that gives the smallest distance:

$$j^* = arg \min_j \| \mathbf{p}_i - \mathbf{w}_{Sj} \|_2. \quad (1)$$

2) Cooperation: The winning neuron j^* indicates the center of a topological neighborhood h_{j^*}. Neurons are excited depending on their lateral distance, r_{j^*}, from this neuron. A typical choice of h_{j^*} is the Gaussian function:

$$h_{j^*k}(n) = \exp(-\frac{r_{j^*k}^2}{2\sigma^2(n)}), \quad (2)$$

where k corresponds to the neuron at hand, n is the iteration of the algorithm, r_{j^*k} is the Euclidean distance between neurons j^* and k in the lattice space and σ is the "effective width" of the topological neighborhoood. σ is a function of n: $\sigma(n) = \sigma_0 \exp(-\frac{n}{N_0})$, where N_0 is the total number of training iterations. $\sigma(0) = \frac{l_w + l_h}{4}$ in our experiments. l_w and l_h are the lattice width and height, respectively.

3) Adaptation: At this step, each neuron is adapted with respect to its lateral distance from the wining neuron as follows:

$$\mathbf{w}_{Sk}(n+1) = \mathbf{w}_{Sk}(n) + \eta(n) h_{j^*k}(n)(\mathbf{p}_i - \mathbf{w}_{Sk}(n)), \quad (3)$$

where $\eta(n)$ is the learning-rate parameter: $\eta(n) = \eta(0) \exp(-\frac{n}{N_0})$. $\eta(0) = 0.1$ in our experiments.

The optimal number of update iterations is determined by performing a comparative study on the produced lattices. In a preliminary study, we have conducted experiments by using a variety of iteration numbers for the update procedure. Specifically, we trained the algorithm by using 20, 50, 60, 80, 100 and 150 update iterations. Comparing the produced lattices, we found that the quality of the produced posture prototypes does not change for update iterations number greater than 60. The optimal lattice topology is determined using the cross-validation procedure, which is a procedure that determines the ability of a learning algorithm to generalize over data that was not trained on. That is, the learning algorithm is trained using all but some training data, which are subsequently used for testing. This procedure is applied multiple times (folds). The test action videos used to determine the optimal lattice topology were all the action videos of a specific person not included in the training set. A 12×12 lattice of posture prototypes produced using action videos of action classes 'walk', 'run', 'jump in place', 'jump forward', 'bend', 'sit', 'fall' and 'wave one hand' captured from eight viewing angles '0^o', '45^o', '90^o', '135^o', '180^o', '225^o', '270^o' and '315^o' (with respect to the person's body) is depicted in Figure 3.

As can be seen, some posture prototypes correspond to body postures that appear in more than one actions. For example, posture prototypes in lattice locations $(1, f)$, $(1, g)$, $(7, d)$ describe postures of actions 'jump in place', 'jump forward' and 'sit', while posture prototypes in lattice locations $(3, k)$, $(6, l)$, $(8, l)$ describe postures of actions 'walk' and 'run'. Moreover, some posture prototypes correspond to postures that appear to one only action class. For example, posture prototypes in lattice locations $(1, i)$, $(10, i)$, $(12, e)$ describe postures of action 'bend', while posture prototypes in lattice location $(4, f)$, $(3, i)$, $(5, j)$ describe postures of

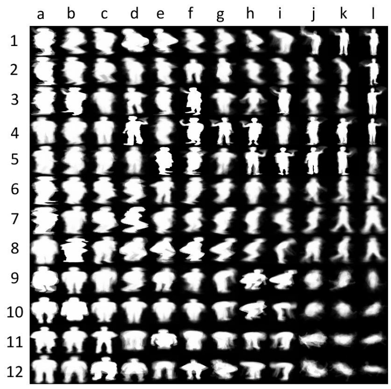


Fig. 3. *A* 12×12 *SOM produced by posture frames of eight actions captured from eight viewing angles.*

action 'wave one hand'. Furthermore, one can notice that similar posture prototypes lie in adjacent lattice positions. This results to a better posture prototype organization. To illustrate the advantage given by the SOM posture prototype representation in the action recognition task, Figure 4 presents the winning neurons in the training set used to produce the 12×12 lattice presented in Figure 3 for each of the action classes. In this Figure, only the winning neurons are shown, while the grayscale value of the enclosing square is a function of their wins number. That is, after determining the SOM, the similarity between all the posture vectors belonging to the training action videos and the SOM neurons was computed and the winning neurons corresponding to each posture vector was determined. The grayscale value of the enclosing squares is high for neurons having large number of wins and small for those having a small one. Thus, neurons enclosed in squares with high grayscale value correspond to human body poses appearing more often in each action type. As can be seen, posture prototypes representing each action are quite different and concentrate in neighboring parts of the lattice. Thus, one can expect that the more non-overlapping these maps are the more discriminant representation they offer for action recognition.

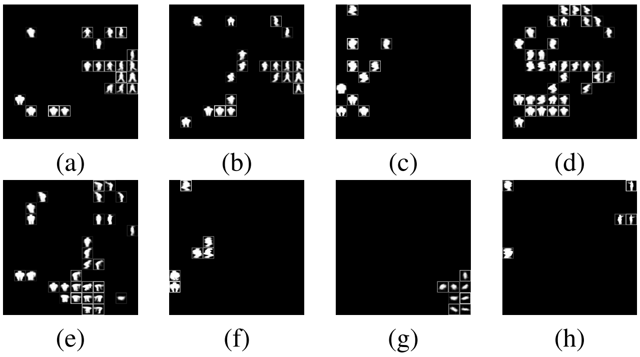

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 4. *Winning neurons for eight actions: a) walk, b) run, c) jump in place, d) jump forward, e) bend, f) sit, g) fall and h) wave one hand.*

By observing the lattice shown in Figure 3, it can be seen that the spatial organization of the posture prototypes defines areas where posture prototypes correspond to different viewing angles. To illustrate this, Figure 5 presents the wining neurons for all action videos that correspond to a specific viewing angle. It can be seen that the wining neurons corresponding to different views are quite distinguished. Thus, the same representation can also be used for viewing angle identification, since the maps that correspond to different views are quite non-overlapping. Overall, the SOM posture prototype representation has enough discriminant power to provide a good representation space for the action posture prototype vectors.

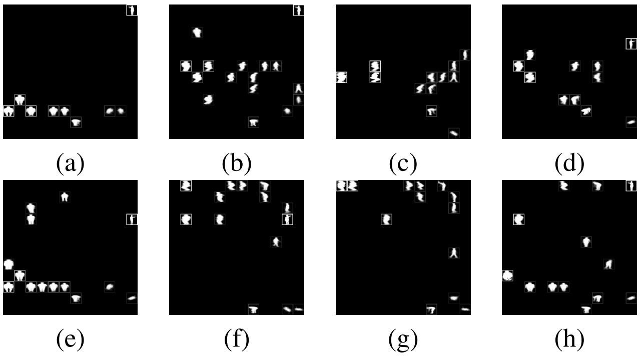

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 5. *Wining neurons for eight views: a)* 0^o, *b)* 45^o, *c)* 90^o, *d)* 135^o, *e)* 180^o, *f)* 225^o, *g)* 270^o *and h)* 315^o.

C. Action Representation

Let posture vectors $\mathbf{p}_i$, $i = 1, ..., N_{t_j}$, $j = 1, ..., N_A$ consist an action video. Fuzzy distances of every $\mathbf{p}_i$ to all the SOM weights $\mathbf{w}_{Sk}$, $k = 1, ..., N_S$ are calculated to determine the similarity of every posture vector with every posture prototype:

$$d_{ik} = (\parallel \mathbf{p}_i - \mathbf{w}_{Sk} \parallel_2)^{-\frac{2}{m-1}}, \qquad (4)$$

where m is the fuzzification parameter ($m > 1$). Its optimal value is determined by applying the cross-validation procedure. We have experimentally found that a value of $m = 1.1$ provides satisfactory action representation and, thus, this value is used in all the experiments presented in this paper. Fuzzy distances allow for a smooth distance representation between posture vectors and posture prototypes.

After the calculation of fuzzy distances, each posture vector is mapped to the following distance vector $\mathbf{d}_i = [d_{i1}, d_{i2}, ..., d_{iN_S}]^T$. Distance vectors $\mathbf{d}_i$, $i = 1, ..., N_{t_j}$ are normalized to produce membership vectors $\mathbf{u}_i = \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}$, $\mathbf{u}_i \in \mathbb{R}^{N_S}$, that correspond to the final representations of the posture vectors in the SOM posture space. The mean vector $\mathbf{s} = \frac{1}{N_{t_j}} \sum_{i=1}^{N_{t_j}} \mathbf{u}_i$, $\mathbf{s} \in \mathbb{R}^{N_S}$ of all the N_{t_j} membership vectors comprising the action video is called action vector and represents the action video.

The use of the mean vector leads to a duration invariant action representation. That is, we expect the normalized cumulative membership of a specific action to be invariant to the duration of the action. This expectation is enhanced by the observation discussed in Subsection II-B and illustrated in Figure 4. Given that the winning SOM neurons corresponding to different actions are quite distinguished, we expect that the distribution of fuzzy memberships to the SOM neurons will characterize actions. Finally, the action vectors representing all

N_T training action videos $\mathbf{s}_j,\ j = 1, ..., N_T$ are normalized to have zero mean and unit standard deviation. In the test phase, all the N action vectors $\mathbf{s}_k,\ k = 1, ..., N$ that correspond to N test action videos depicting the person from different viewing angles are normalized accordingly.

D. Single-view Action Classification

As previously described, action recognition performs the classification of an unknown incoming action captured by $N \leq N_C$ action videos, to one of the N_A known action classes $a_j,\ j = 1, ..., N_A$ contained in an action class set $\mathcal{A}$. Using the SOM posture prototype representation, which leads to spatially related posture prototypes, and expecting that action videos of every action class will be described by spatially related posture vectors, a MLP is proposed for the action classification task consisting of N_S inputs (equal to the dimensionality of action vectors $\mathbf{s}$), N_A outputs (each corresponding to an action class $a_j,\ j = 1, ..., N_A$) and using the hyperbolic tangent function $f_{sigmoid}(x) = \alpha\ tanh(bx)$, where the values $\alpha = 1.7159$ and $b = \frac{2}{3}$ were chosen [28], [29].

In the training phase, all N_T training action vectors $\mathbf{s}_i,\ i = 1, ..., N_T$ accompanied by their action labels are used to define MLP weights $\mathbf{W}_A$ using the Backpropagation algorithm [30]. Outputs corresponding to each action vector, $\mathbf{o}_i = [o_{i1}, ..., o_{iN_A}]^T$, are set to $o_{ik} = 0.95$ for action vectors belonging to action class k and $o_{ik} = -0.95$ otherwise, $k = 1, ..., N_A$. For each of the action vectors $\mathbf{s}_i$, MLP response $\hat{\mathbf{o}}_i = [\hat{o}_{i1}, ..., \hat{o}_{iN_A}]^T$ is calculated by:

$$\hat{o}_{ik} = f_{sigmoid}(\mathbf{s}_i^T \mathbf{w}_{Ak}) \tag{5}$$

where $\mathbf{w}_{Ak}$ is a vector that contains the MLP weights corresponding to output k.

The training procedure is performed in an on-line form, i.e., adjustments of the MLP weights are performed for each training action vector. After the feed of a training action vector $\mathbf{s}_i$ and the calculation of the MLP response $\hat{\mathbf{o}}_i$, the modification of weight that connects neurons i and j follows the update rule:

$$\Delta W_{Aji}(n+1) = c\Delta W_{Aji}(n) + \eta\delta_j(n)y_i(n), \tag{6}$$

where $\delta_j(n)$ is the local gradient for the j-th neuron, y_i is the output of the i-th neuron, η is the learning-rate, c is a positive number, called momentum constant ($\eta = 0.05$ and $c = 0.1$ in the experiments presented in this paper) and n is the iteration number. Action vectors are introduced to the MLP in a random sequence. This procedure is applied until the Mean Square Error (MSE) falls under an acceptable error rate ε:

$$E[(\frac{1}{N_T}(\hat{\mathbf{o}}_i - \mathbf{o}_i)^2)^{\frac{1}{2}}] < \varepsilon. \tag{7}$$

The optimal MSE parameter value is determined performing the cross-validation procedure using different threshold values for the mean square error (MSE) ε and the number of iterations parameters of the algorithm N_{bp}. We used values of ε equal to 0.1, 0.01 and 0.001 and values of N_{bp} equal to 100, 500 and 1000. We found the best combination to be $\varepsilon = 0.01$ and $N_{max} = 1000$ and used them in all our experiments.

In the test phase, a set $\mathcal{S}$ of action vectors $\mathbf{s}_i,\ i = 1, ..., N$ corresponding to N action videos captured from all the N available cameras depicting the person is obtained. To classify $\mathcal{S}$ to one of the N_A action classes a_j specified by the action set $\mathcal{A}$, each of the N action vectors $\mathbf{s}_i$ is fed to the MLP and N responses are obtained:

$$\hat{\mathbf{o}}_i = [\hat{o}_{i1}, ..., \hat{o}_{iN_A}],\ \ i = 1, ..., N. \tag{8}$$

Each action video is classified to the action class $a_j,\ j = 1, ..., N_A$ that corresponds to the MLP maximum output:

$$\hat{a}_i = \underset{j}{\operatorname{argmax}}\ \hat{o}_{ij}\ \ ,\ i = 1, ..., N,\ \ j = 1, ..., N_A. \tag{9}$$

Thus, a vector $\hat{\mathbf{a}} = [\hat{a}_1, ..., \hat{a}_N]^T \in \mathcal{R}^N$ containing all the recognized action classes is obtained. Finally, expecting that most of the recognized actions $\hat{a}_i$ will correspond to the actual action class of $\mathcal{S}$, $\mathcal{S}$ is classified to an action class by performing majority voting over the action classes indicated in $\hat{\mathbf{a}}$.

Using this approach, view-independent action recognition is achieved. Furthermore, as the number N of action vectors forming $\mathcal{S}$ may vary, a generic multi-view action recognition method is obtained. In the above described procedure, no viewing angle information is used in the combination of classification results $\hat{a}_i,\ i = 1, ..., N$, that correspond to each of the N cameras, to produce the final recognition result. As noted before, actions are quite different when they are captured from different viewing angles. Thus, some views may be more discriminant for certain actions. For example, actions 'walk' and 'run' are well distinguished when they are captured from a side view but they seem similar when they are captured from the front view. In addition, actions 'wave one hand' and 'jump in place' are well distinguished when they are captured from the front or back view, but not from the side views. Therefore, instead of majority vote, a more sophisticated method can be used to combine all the available information and produce the final action recognition result by exploiting the viewing angle information. In the following, a procedure based on a Bayesian framework is proposed in order to combine the action recognition results from all N cameras.

E. Combination of single-view action classification results using a Bayesian Framework

The classification of an action vector set $\mathcal{S}$ consisting of $N \leq N_C$ action vectors $\mathbf{s}_i,\ i = 1, ..., N$, each corresponding to an action video coming from a specific camera used for recognition, in one of the action classes $a_j,\ j = 1, ..., N_A$ of the action class set $\mathcal{A}$, can be performed using a probabilistic framework. Each of the N action vectors $\mathbf{s}_i$ of $\mathcal{S}$ is fed to the MLP and N vectors containing the responses $\hat{\mathbf{o}}_i = [\hat{o}_{i1}, \hat{o}_{i2}, ..., \hat{o}_{iN_A}]^T$ are obtained. The problem to be solved is to classify $\mathcal{S}$ in one of the action classes a_j given these observations, i.e., to estimate the probability $P(a_j|\hat{\mathbf{o}}_1^T, \hat{\mathbf{o}}_2^T, ..., \hat{\mathbf{o}}_{N_C}^T)$ of every action class given MLP responses. In the case where $N < N_C$, $N - N_C$ MLP outputs $\hat{\mathbf{o}}_i$ will be set to zero, as no recognition result is provided for these cameras. Since MLP responses are real valued, $P(a_j|\hat{\mathbf{o}}_1^T, \hat{\mathbf{o}}_2^T, ..., \hat{\mathbf{o}}_{N_C}^T)$ estimation is very difficult. Let $\hat{a}_i$ denote the action recognition result corresponding to the test action vector $\mathbf{s}_i$ representing the action video captured by the i-th camera, taking values in the action class set $\mathcal{A}$. Without loss of generality, $\mathbf{s}_i,\ i = 1, ..., N$ is assumed to be classified to the action class that provides the highest MLP response, i.e., $\hat{a}_i = \underset{j}{\operatorname{argmax}}\ \hat{o}_{ij}$. That is, the problem to be solved is to estimate the probabilities $P(a_j|\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C})$ of every action class a_j, given the discrete variables $\hat{a}_i,\ i = 1, ..., N_C$. Let $P(a_j)$ denote the a priori probability of action class a_j and $P(\hat{a}_i)$ the a priori probability of recognizing $\hat{a}_i$ from camera i. Let $P(\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C})$ be the joint probabilities of all the N_C cameras observing one of the N_A action classes a_j. Furthermore, the conditional probabilities $P(\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C}|a_j)$ that camera 1 recognizes action class $\hat{a}_1$, camera 2 recognizes action class $\hat{a}_2$, etc., given that the actual action class of $\mathcal{S}$ is a_j, can be calculated. Using these probabilities, the probability $P(a_j|\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C})$ of action class $a_j, j = 1, ..., N_A$, given the classification results $\hat{a}_i$ can be estimated using the Bayes formula:

$$P(a_j|\hat{a}_1, \hat{a}_2, ..., \hat{a}_N) = \frac{P(\hat{a}_1, \hat{a}_2, ..., \hat{a}_N|a_j) \cdot P(a_j)}{\sum_{l=1}^{N_A} P(\hat{a}_1, \hat{a}_2, ..., \hat{a}_N|a_l) \cdot P(a_l)}. \quad (10)$$

In the case of equiprobable action classes, $P(a_j) = \frac{1}{N_A}$. If this is not the case, $P(a_j)$ should be set to their real values and the training data should be chosen accordingly. Expecting that training and evaluation data come from the same distributions, $P(\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C}|a_j)$ can be estimated during the training procedure. In the evaluation phase, $\mathcal{S}$ can be classified to the action class providing the maximum conditional probability, i.e., $\hat{a}_{\mathcal{S}} = \underset{j}{\operatorname{argmax}}\ P(a_j|\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C})$. However, such a system cannot be applied in a straightforward manner, since the combinations of all the N_C cameras providing $N_A + 1$ action recognition results is enormous. The case $N_A + 1$ refers to the situation where a camera does not provide an action recognition result, because the person is not visible from this camera. For example, in the case of $N_C = 8$ and $N_A = 8$, the number of all possible combinations is equal to $(N_A + 1)^{N_C} = 43046721$. Thus, in order to estimate the probabilities $P(\hat{a}_1, \hat{a}_2, ..., \hat{a}_{N_C}|a_j)$ an enormous training data set should be used.

To overcome this difficulty, the action classification task could be applied to each of the N available cameras independently and the N classification results could subsequently be combined to classify the action vector set $\mathcal{S}$ to one of the action classes a_j. That is, for camera i, the probability $P(a_j|\hat{a}_i)$ of action class a_j given the recognized action class $\hat{a}_i$ can be estimated using the Bayes formula:

$$P(a_j|\hat{a}_i) = \frac{P(\hat{a}_i|a_j) \cdot P(a_j)}{\sum_{l=1}^{N_A} P(\hat{a}_i|a_l) \cdot P(a_l)}. \quad (11)$$

As previously described, since the person can freely move, the viewing angle can vary for each camera, i.e., if a camera captures the person from the front viewing angle at a given time instance, a change in his/her motion direction may result that this camera captures him/her from a side viewing angle at a subsequent time instance. Since the viewing angle has proven to be very important in the action recognition task, the viewing angle information should be exploited to improve the action recognition accuracy. Let $P(\hat{a}_i, \hat{v}_i)$ denote the joint probability denoting that camera i recognizes action class $\hat{a}_i$ captured from viewing angle $\hat{v}_i$. Using $P(\hat{a}_i, \hat{v}_i)$, the probability $P(a_j|\hat{a}_i, \hat{v}_i)$ of action class a_j, given $\hat{a}_i$ and $\hat{v}_i$ can be estimated by:

$$P(a_j|\hat{a}_i, \hat{v}_i) = \frac{P(\hat{a}_i, \hat{v}_i|a_j) \cdot P(a_j)}{\sum_{l=1}^{N_A} P(\hat{a}_i, \hat{v}_i|a_l) \cdot P(a_l)}. \quad (12)$$

The conditional probabilities $P(a_j|\hat{a}_i, \hat{v}_i)$ are estimated in the training phase. After training the MLP, action vectors corresponding to the training action videos are fed to the MLP in order to obtain its responses. Each action vector is classified to the action class providing the highest MLP response. Exploiting the action and viewing angle labels accompanying the training action videos, the conditional probabilities $P(\hat{a}_i, \hat{v}_i|a_j)$ corresponding to the training set are calculated. Finally, $P(a_j|\hat{a}_i, \hat{v}_i)$ are obtained using Equation (12). In the recognition phase, each camera provides an action recognition result. The viewing angle corresponding to each camera should also be estimated automatically to obtain $\hat{v}_i$. A procedure to this end, which exploits the vicinity property of the SOM posture prototype representation is presented in the next subsection. After obtaining the $\hat{a}_i$ and $\hat{v}_i$, $i = 1, ..., N$, the action vector set $\mathcal{S}$ is classified to the action class providing the maximum probability sum [31], i.e., $\hat{a}_{\mathcal{S}} = \underset{j}{\operatorname{argmax}}\ \sum_{i=1}^{N} P(a_j|\hat{a}_i, \hat{v}_i)$.

In (12), the denominator $\sum_{j=1}^{N_A} P(\hat{a}_i, \hat{v}_i|a_j) \cdot P(a_j) = P(\hat{a}_i, \hat{v}_i)$ refers to the probability that action vectors corresponding to action videos captured from the recognized viewing angle $\hat{v}_i$ belong to the recognized action class $\hat{a}_i$. This probability is indicative of the ability of each viewing angle to correctly recognize actions. Some views may offer more information to correctly classify some of the action vectors to an action class. In the case where $\hat{v}_i$ is capable to distinguish $\hat{a}_i$ from all the other action classes, $P(\hat{a}_i, \hat{v}_i)$ will be equal to its actual value, thus having a small impact to the final decision $P(a_j|\hat{a}_i, \hat{v}_i)$. In the case where $\hat{v}_i$ confuses some action classes, $P(\hat{a}_i, \hat{v}_i)$ will have a value either higher, or smaller than its actual one and will influence the decision $P(a_j|\hat{a}_i, \hat{v}_i)$. For example, consider the case of recognizing action class 'bend' from the front viewing angle. Because this action is well distinguished from all the other actions, when it is captured from the front viewing angle, $P(\hat{a}_i, \hat{v}_i)$ will not influence the final decision. The case of recognizing action classes 'jump in place' and 'sit' from a 270^o side viewing angle is different. Action videos belonging to these action classes captured from this viewing angle are confused. Specifically, all the action videos recognized to belong to action class 'jump in place' actually belong to this class, while some action videos recognized to belong to action class 'sit' belong to action class 'jump in place'. In this case $P(\hat{a}_i, \hat{v}_i)$ will be of high value for the action class 'sit', thus providing a low value of $P(a_j|\hat{a}_i, \hat{v}_i)$, while $P(\hat{a}_i, \hat{v}_i)$ will be of low value for the action class 'jump in place', thus providing a high value of $P(a_j|\hat{a}_i, \hat{v}_i)$. That is, the recognition of action classes 'sit' from the 270^o side viewing angle is ambiguous, as it is probable that the action video belongs to action class 'jump in place', while the recognition of the action class 'jump in place' from the same viewing angle is of high confidence, as action videos recognized to belong to action class 'jump in place' actually belong to this class.

The term $P(\hat{a}_i, \hat{v}_i|a_j)$ in (12) refers to the probability of the i-th action vector to be detected as action video captured from viewing angle $\hat{v}_i$ and classified to action class $\hat{a}_i$, given that the actual action class of this action vector is a_j. This probability is indicative of the similarity between actions when they are captured from different viewing angles angles and provides an estimate of action discrimination for each viewing angle. Figure 6 illustrates the probabilities $P(\hat{a}_i, \hat{v}_i|a_j),\ j = 1, ..., N_A,\ i = 1, ..., N_C,\ \hat{a}_i = a_j$, i.e., the probabilities to correctly classify an action video belonging to action class a_j from each of the viewing angles $\hat{v}_i$ for an action class set $\mathcal{A}$ = {'walk', 'run', 'jump in place', 'jump forward', 'bend', 'sit', 'fall', 'wave one hand'} produced using a 12×12 SOM and an 8-camera setup, in which each camera captures the person from one of the eight viewing angles $V = \{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$. In this Figure, it can be seen that action vectors belonging to action classes 'jump in place', 'bend', and 'fall' are almost correctly classified by every viewing angle. On the other hand, action vectors belonging to the remaining actions are more difficult to be correctly classified for some viewing angles. As was expected, the side views are the most capable in terms of classification for action classes 'walk', 'run', 'jump forward' and 'sit', while in the case of action class 'wave one hand' the best views are the frontal and the back ones.

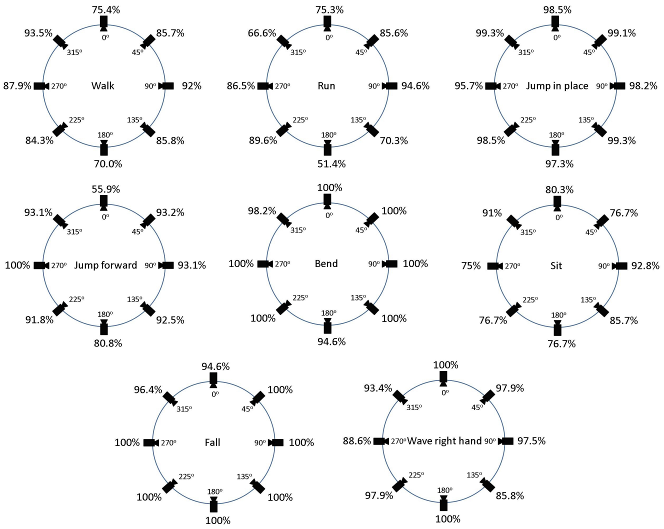


Fig. 6. *Single-view action classification results presented as input to the Bayesian framework for eight actions captured from eight viewing angles.*

F. Camera Viewing Angle Identification

The proposed method utilizes a multi-camera setup. The person that performs an action can freely move and this affects the viewing angles he/she is captured by each camera. Exploiting the SOM posture prototype representation in Figure 3a and the observation that posture prototypes corresponding to each viewing angle lie in different lattice locations as presented in Figure 5, a second MLP is proposed to identify the viewing angle $\hat{v}_i$ the person is captured from each camera. Similarly to the MLP used in action classification, it consists of N_S input nodes (equal to the dimensionality of action vectors $\mathbf{s}$), N_C outputs (each corresponding to a viewing angle) and uses the hyperbolic tangent function as activation function. Its training procedure is similar to the one presented in Subsection II-G. However, this time, the training outputs are set to $o_{ik} = 0.95,\ k = 1, ..., N_C$, for action vectors belonging to action videos captured from the k-th viewing angle and $o_{ik} = -0.95$ otherwise.

In the test phase, each of the N action vectors $\mathbf{s}_i$ consisting an action video set $\mathcal{S}$ that corresponds to the same action captured from different viewing angles, is introduced to the MLP and the corresponding to each action vector viewing angle $\hat{v}_i$ is recognized based on the maximum MLP response:

$$\hat{v}_i = \underset{j}{\text{argmax}}\ \hat{o}_{ij},\quad i = 1, ..., N,\ j = 1, ..., N_C. \tag{13}$$

G. Action Recognition (test phase)

Let a person performing an action captured from $N \leq N_C$ cameras. In the case of elementary action recognition, this action is captured in N action videos, while in the case of continuous action recognition, a sliding window consisted of N_{tw} video frames is used to create the N action videos used to perform action recognition at every window location. These videos are preprocessed as discussed in Section II-A to produce $N \times N_t$ posture vectors $\mathbf{p}_{ij},\ i = 1, ..., N,\ j = 1, ..., N_t$, where $N_t = N_{t_j}$ or $N_t = N_{tw}$ in the elementary and the continuous action recognition tasks, respectively. Fuzzy distances $\mathbf{d}_{ijk}$ from all the test posture vectors $\mathbf{p}_{ij}$ to every SOM posture prototype $\mathbf{w}_{Sk},\ k = 1, ..., N_S$, are calculated and a set $\mathcal{S}$ of N test action vectors, $\mathbf{s}_i$, is obtained. These test action vectors are fed to the action recognition MLP and N action recognition results $\hat{a}_i$ are obtained. In the case of majority voting, the action vector set $\mathcal{S}$ is classified to the action class a_j that has the most votes. In the Bayesian framework case, the N action vectors $\mathbf{s}_i$ are fed to the viewing angle identification MLP to recognize the corresponding viewing angle $\hat{v}_i$ and the action vector set $\mathcal{S}$ is classified to the action class a_j that provides the highest cumulative probability according to the Bayesian decision. Figure 7 illustrates the procedure followed in the recognition phase for the majority voting and the Bayesian framework cases.

III. Experiments

The experiments conducted in order to evaluate the performance of the proposed method are presented in this section. To demonstrate the ability of the proposed method to correctly classify actions performed by different persons, as variations in action execution speed and style may be observed, the leave-one-person-out cross-validation procedure was applied in the i3DPost multi-view action recognition database [32] and the experiments are discussed in subsection III-C. Subsection III-D discusses the operation of the proposed method in

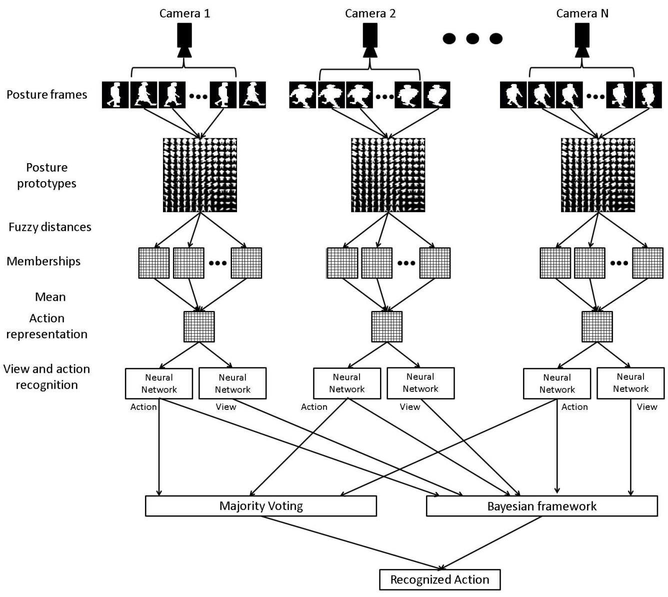


Fig. 7. *Action recognition system overview (test phase).*

case of multi-period action videos. Subsection III-E presents its robustness in the case of action recognition at different frame rates between training and test phases. In III-F a comparative study that deals with the ability of every viewing angle to correctly classify actions is presented. The case of different camera setups in the training and recognition phases is discussed in Subsection III-G, while the case of action recognition using an arbitrary number of cameras at the test phase is presented in subsection III-H. The ability of the proposed approach to perform action recognition in the case of human interactions is discussed in Subsection III-I.

A. The i3DPost multi-view database

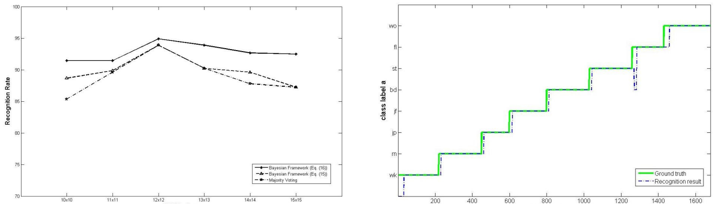
The i3DPost multi-view database [32] contains 80 high resolution image sequences depicting eight persons performing eight actions and two person interactions. Eight cameras having a wide 45^o viewing angle difference to provide 360^o coverage of the capture volume were placed on a ring of 8m diameter at a height of 2m above the studio floor. The studio was covered by blue background. The actions performed in 64 video sequences are: ’walk’ (wk), ’run’ (rn), ’jump in place’ (jp), ’jump forward’ (jf), ’bend’ (bd), ’fall’ (fl), ’sit on a chair’ (st) and ’wave one hand’ (wo). The remaining 16 sequences depict two persons that interact. These interactions are: ’shake hand’ (sh) and ’pull down’ (pl).

B. The IXMAS multi-view database

The INRIA (Institut National de Recherche en Informatique et Automatique) Xmas Motion Acquisition Sequences database [22] contains 330 low resolution (291×390 pixels) image sequences depicting 10 persons performing 11 actions. Each sequence has been captured by five cameras. The persons freely change position and orientation. The actions performed are: ’check watch’, ’cross arm’, ’scratch head’, ’sit down’, ’get up’, ’turn around’, ’walk in a circle’, ’wave hand’, ’punch’, ’kick’, and ’pick up’. Binary images denoting the person’s body are provided by the database.

C. Cross-validation in i3DPost multi-view database

The cross-validation procedure described in Subsection II-B was applied to the i3DPost eight-view database, using the action video sequences of the eight persons. Action videos were manually extracted and binary action videos were obtained by thresholding the blue color in the HSV color space. Figure 8a illustrates the recognition rates obtained for various SOM lattice topologies for the majority voting and the Bayesian framework cases. It can be seen that high recognition rates were observed. The optimal topology was found to be a 12×12 lattice. A recognition rate equal to 93.9% was obtained for the majority voting case. The Bayesian approach outperforms the majority voting one, providing a recognition rate equal to 94.04% for the view-independent approach. As can be seen, the use of viewing angle information results to an increase of the recognition ability. The best recognition rate was found to be equal to 94.87%, for the Bayesian approach incorporating the viewing angle recognition results. The confusion matrix corresponding to the best recognition result is presented in Table I. In this matrix, rows represent the actual action classes and columns the recognition results. As can be seen, actions which contain discriminant body postures, such as ’bend’, ’fall’ and ’wave right hand’ are perfectly perfectly classified. Actions having large number of similar body postures, such as ’walk’-’run’, or ’jump in place’-’jump forward’-’sit’, are more difficult to be correctly classified. However, even for these cases, the classification accuracy is very high.



(a) (b)

Fig. 8. *a) Action recognition rates vs various lattice dimensions of the SOM. b) Recognition results in the case of continuous action recognition.*

D. Continuous action recognition

This section presents the functionality of the proposed method in the case of continuous (multiple period) action recognition. Eight multiple period videos, each corresponding to one viewing angle, depicting one of the persons of the i3DPost eight-view action database were manually created by concatenating single period action videos. The algorithm was trained using the action videos depicting the remaining seven persons using a 12×12 lattice and combining the classification results corresponding to each camera with the Bayesian framework. In the test phase, a sliding window of $N_{tw} = 21$ video frames was used and recognition was performed at every sliding window position. A majority vote filter, of size equal to 11 video frames, was applied at every classification result.

Figure 8b illustrates the results of this experiment. In this Figure, ground truth is illustrated by a continuous line and recognition results by a dashed one. In the first 20 frames no action recognition result was produced, as the algorithm needs 21 frames (equal to the frames of the sliding window N_{tW}) to perform action recognition. Moreover, a delay is observed in the classification results, as the algorithm uses observations that refer to past video frames $(t, t-1, ..., t-N_t+1)$. This delay was found to be between 12 and 21 video frames. Only one recognition error occurred at the transition between actions 'sit' and 'fall'.

TABLE I

CONFUSION MATRIX FOR EIGHT ACTIONS.

	wk	rn	jp	jf	bd	st	fl	wo
wk	0.95	0.05						
rn	0.05	0.95						
jp			0.92	0.02		0.06		
jf			0.05	0.9		0.05		
bd					1			
st			0.13			0.87		
fl							1	
wo								1

E. Action recognition in different video frame rates

To simulate the situation of recognizing actions using cameras of different frame rates, between training and test phases, an experiment was set as follows. The cross-validation procedure using a 12×12 lattice and the Bayesian framework was applied for different camera frame rates in the test phase. That is, in the training phase the action videos depicting the training persons were used to train the algorithm using their actual number of frames. In the test phase, the number of frames consisting the action videos were fewer, in order to achieve recognition at lower frame rate. That is, for action recognition at the half frame rate, the test action videos consisted from the even-numbered frames, i.e., $n_t \, mod \, 2 = 0$, where n_t is the frame number of each video frame and mod refers to the modulo operator. In the case of a test frame rate equal to $1/3$ of the training frame rate, only frames with frame number $n_t \ mod \ 3 = 0$ were used, etc. In the general case, where the test to training video frame rate ratio was equal to $\frac{1}{K}$, the test action videos consisted of the video frames satisfying $n_t \ mod \ K = 0$. Figure 9a shows the results for various values of K. It can be seen that the frame rate variation between the training and test phases does not influence the performance of the proposed method. In fact, it was observed that, for certain actions, a single posture frame that depicts a well distinguished posture of the action is enough to produce a correct classification result. This verifies the observation made in Subsection II-D that human body postures of different actions are placed at different positions on the lattice. Thus, the corresponding neurons are responsible to recognize the correct action class. To verify this, the algorithm was tested using single body posture masks that depict a person from various viewing angles. Results of this experiment are illustrated in Figure 10. In this Figure, it can be seen that the MLP can correctly classify action vectors that correspond to single human body postures. Even for difficult cases, such as 'Walk 0^o', or 'Run 315^o, the MLP can correctly recognize the action at hand.

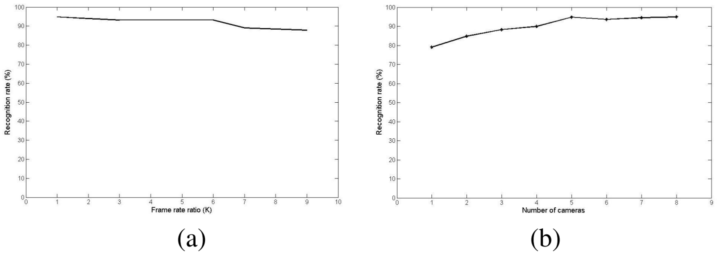
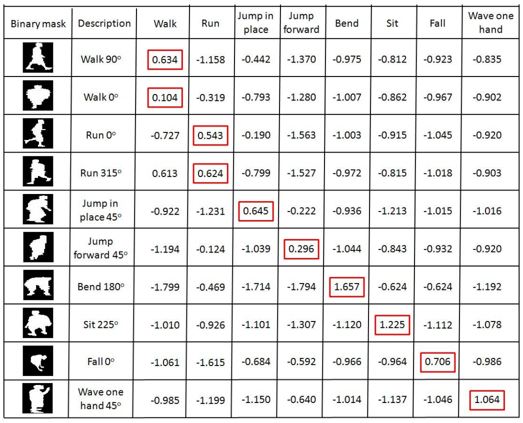


Fig. 9. *a) Recognition results for different video frame rates between training and test phase. b) Action recognition rates vs various occlusion levels.*



Binary mask	Description	Walk	Run	Jump in place	Jump forward	Bend	Sit	Fall	Wave one hand
	Walk 90°	0.634	-1.158	-0.442	-1.370	-0.975	-0.812	-0.923	-0.835
	Walk 0°	0.104	-0.319	-0.793	-1.280	-1.007	-0.862	-0.967	-0.902
	Run 0°	-0.727	0.543	-0.190	-1.563	-1.003	-0.915	-1.045	-0.920
	Run 315°	0.613	0.624	-0.799	-1.527	-0.972	-0.815	-1.018	-0.903
	Jump in place 45°	-0.922	-1.231	0.645	-0.222	-0.936	-1.213	-1.015	-1.016
	Jump forward 45°	-1.194	-0.124	-1.039	0.296	-1.044	-0.843	-0.932	-0.920
	Bend 180°	-1.799	-0.469	-1.714	-1.794	1.657	-0.624	-0.624	-1.192
	Sit 225°	-1.010	-0.926	-1.101	-1.307	-1.120	1.225	-1.112	-1.078
	Fall 0°	-1.061	-1.615	-0.684	-0.592	-0.966	-0.964	0.706	-0.986
	Wave one hand 45°	-0.985	-1.199	-1.150	-0.640	-1.014	-1.137	-1.046	1.064

Fig. 10. *MLP responses for single human body posture images as input.*

F. Actions versus viewing angle

A comparative study that specifies the action discrimination from different viewing angles is presented in this subsection. Using a 12×12 lattice and the Bayesian framework, the cross-validation procedure was applied for the action videos depicting the actions in each of the eight viewing angles $\{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$. That is, eight single-view elementary action recognition procedures were performed. Figure 11 presents the recognition rates achieved for each of the actions. In this Figure, the probability to correctly recognize an incoming action from every viewing angle is presented, e.g., the probability to correctly recognize a walking sequence captured from the frontal view is equal to 77.7%. As was expected, for most action classes, the side views are the most discriminant ones and result in the best recognition rates. In the case of action 'wave one hand', the frontal and the back views are the most discriminant ones and result in the best recognition rates. Finally, well distinguished actions, such as 'bend' and 'fall', are well recognized from any viewing angle. This can be explained by the fact that the body postures that describe them are quite distinctive at any viewing angle.

Table II presents the overall action recognition accuracy achieved in every single-view action recognition experiment. In this Figure, the probability to correctly recognize an incoming action from any viewing angle is presented. For example, the probability to correctly recognize one of the eight actions captured from the frontal view is equal to 77.5%. As can be seen, side views result in better recognition rates, because most of the actions are well discriminated when observed by side views. The best recognition accuracy is equal to $86,1\%$ and comes from a side view (135^o). It should be noted that the use of the Bayesian network improves the recognition accuracy as the combination of these recognition outputs leads to a recognition rate equal to 94.87%, as discussed in Subsection III-C.

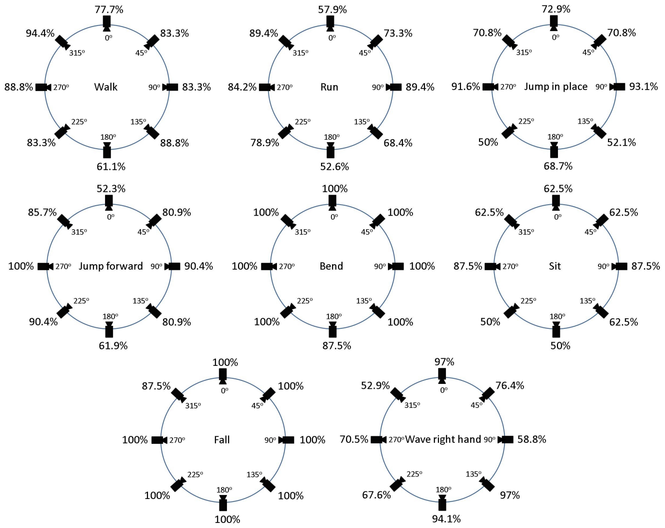


Fig. 11. *Recognition rates of different actions when observed from different viewing angles.*

TABLE II

RECOGNITION RATES OBTAINED FOR EACH VIEWING ANGLE.

$\mathbf{0^o}$	$\mathbf{45^o}$	$\mathbf{90^o}$	$\mathbf{135^o}$
77.5%	80.9%	82.4%	86.1%
$\mathbf{180^o}$	$\mathbf{215^o}$	$\mathbf{260^o}$	$\mathbf{315^o}$
74.1%	80.5%	79.2%	80.2%

G. *Action recognition using reduced camera setups*

In this subsection, a comparative study between different reduced camera setups is presented. Using a 12×12 lattice, the cross-validation procedure was applied for different reduced test camera setups. In the training phase, all action videos depicting the training persons from all the eight cameras were used to train the proposed algorithm. In the test phase, only the action videos depicting the test person from the cameras specified by the reduced camera setup were used. Because the movement direction of the eight persons varies, cameras in these experiments do not correspond to a specific viewing angle, i.e., camera #1 may or may not depict to the person's front view. Figure 12 presents the recognition rates achieved by applying this procedure for eight different camera setups. As can be seen, a recognition rate equal to 92.3% was achieved using only 4 cameras having a 90^o viewing angle difference to provide 360^o coverage of the capture volume. It can be seen that, even for 2 cameras placed at arbitrary viewing angles a recognition rate greater than 83% is achieved.

H. *Recognition in occlusion*

To simulate the situation of recognizing actions using an arbitrary number of cameras an experiment was set as follows. The cross-validation procedure using a 12×12 lattice and the Bayesian framework was applied for a varying number of cameras in the test phase. That is, in the training phase the action videos depicting the training persons from all the eight cameras were used to train the algorithm. In the test phase, the number and the capturing view of the testing person's action videos were randomly chosen. This experiment was applied for a varying number of cameras depicting the testing person. The recognition rates achieved in these experiments can be seen in Figure 9b. Intuitively, we would expect the action recognition accuracy to be low when using a small number of cameras in the test phase. This is due to the viewing angle effect. By using a large number of cameras in the test phase, the viewing angle effect should be addressed properly, resulting to an increased action recognition accuracy. Using one arbitrarily chosen camera, a recognition rate equal to 79% was obtained, while using four arbitrarily chosen cameras, the recognition rate was increased to 90%. Recognition rates equal to 94.85% and 94.87% were observed using five and eight cameras, respectively. As it was expected, the use of multiple cameras resulted to an increase of action recognition accuracy. This experiment illustrates the ability of the proposed approach to recognize actions at high accuracy, in the case of recognition using an arbitrary number of cameras that depict the person from arbitrary view angles.

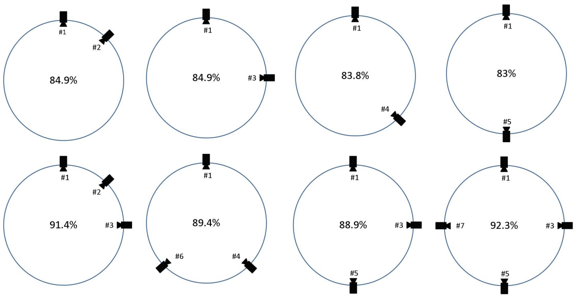


Fig. 12. *Recognition rates for the eight different camera setups.*

I. *Recognition of human interactions*

As previously described, the action recognition task refers to the classification of actions performed by one person. To demonstrate the ability of the proposed approach to correctly classify actions performed by more than one persons, i.e., human interactions, the cross-validation procedure was applied to the i3DPost eight-view database including the action videos that depict human interactions, e.g.: 'shake hands' and 'pull down'. A recognition rate equal to 94.5% was observed for a lattice of 13×13 neurons and the Bayesian framework. An example of 13×13 lattice is shown in Figure 13. The confusion matrix of this experiment can be seen in Table III.

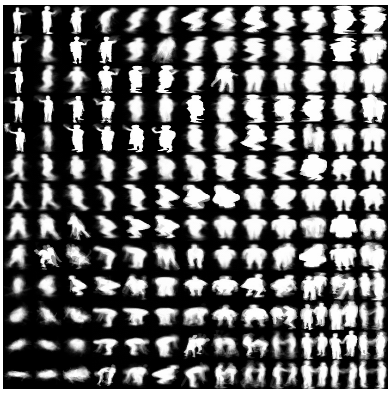


Fig. 13. *A 13 × 13 SOM produced by posture frames of actions and interactions.*

To illustrate the continuous action recognition functionality of a system that can recognize interactions, an experiment was set up as follows: the algorithm was trained using the action videos depicting the seven persons of the I3DPost action dataset including the two interactions ('shake hand' and 'pull dawn') using a 13 × 13 lattice topology and the Bayesian framework. The original action videos depicting the side views of the eight person performing these interactions was tested using a sliding window of $N_{tW} = 21$ video frames. Figure 14 illustrates qualitative results of this procedure. When the two persons were separated, each person was tracked at subsequent frames using a closest area blob tracking algorithm and the binary images depicting each person were fed to the algorithm for action/interaction recognition. When the two persons interacted, the whole binary image was introduced to the algorithm for recognition. In order to use the remaining cameras, more sophisticated blob tracking methods could be used [33], [34]. As can be seen, the proposed method can be extended to recognize interactions in a continuous recognition setup.

TABLE III

Confusion matrix for eight actions and two interactions

	wk	rn	jp	jf	bd	hs	pl	st	fl	wo
wk	0.95	0.05								
rn	0.05	0.95								
jp			0.81	0.1		0.02		0.07		
jf			0.05	0.95						
bd					1					
hs						1				
pl							1			
st			0.13					0.87		
fl									1	
wo			0.03	0.05						0.92

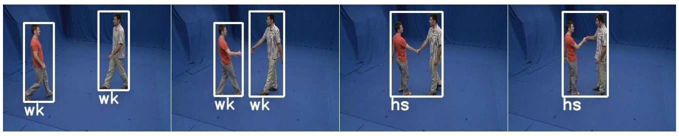


Fig. 14. *Continuous recognition of human interactions.*

J. Comparison against other methods

In this section we compare the proposed method with state of the art methods, recently proposed in the literature, aiming to view-independent action recognition using multi-camera setups. Table IV illustrates comparison results with three methods by evaluating their performance in the i3DPost multi-view action recognition database using all the cameras consisting the database camera setup. In [35], the authors performed the LOOCV procedure in an action class set consisting of the actions "walk", "run", "jump in place", "jump forward" and "bend". The authors in [36] included action "wave one hand" in their experimental setup and performed the LOOCV procedure using six actions and removed action "run", in order to perform the LOOCV procedure by using five actions. Finally, the authors in [37] applied the LOOCV procedure by using all the eight actions appearing in the databse. As can be seen in Table IV, the proposed method outperforms all the aforementioned methods.

TABLE IV

Comparison results in the i3DPost multi-view action recognition database.

	5 actions	8 actions	5 actions	6 actions
Method [35]	90%	-	-	-
Method [37]	-	90.88%	-	-
Method [36]	-	-	97.5%	89.58%
Proposed method	94.4%	94.87%	97.8%	95.33%

In order to compare our method with other methods using the IXMAS action recognition datase we performed the LOOCV procedure by using the binary images provided in the database. In an off-line procedure, each image sequence was split in smaller segments, in order to produce action videos. Subsequently, the LOOCV procedure has been performed by using different SOM topologies and the Bayessian framework approach. By using a 13 × 13 SOM an action recognition rate equal to 89.8% has been obtained. Table V illustrates comparison results with three methods evaluating their performance in the IXMAS multi-view action recognition database. As can be seen, the proposed method outperforms these methods providing up to 8.5% improvement on the action classification accuracy.

TABLE V

Comparison results in the IXMAS multi-view action recognition database.

Method [38]	Method [39]	Method [40]	Proposed method
81.3%	81%	80.6%	89.8%

IV. Conclusion

A very powerful framework based on Artificial Neural Networks has been proposed for action recognition. The proposed method highlights the strength of ANN in representing and classifying visual information. SOM human body posture representations is combined with Multilayer Perceptrons. Action and viewing angle classification is achieved independently for all cameras. A Bayesian framework is exploited in order to provide the optimal combination of the action classification results, coming from all available cameras. The effectiveness of the proposed method in challenging problem setups has been demonstrated by experimentation. According to authors' knowledge, there is no other method in the literature that can deal with all the presented challenges in action recognition. Furthermore, it has been shown that the same framework can be applied for human interaction recognition between persons, without any modification.

Acknowledgment

The research leading to these results has received funding from the Collaborative European Project MOBISERV FP7-248434 (http://www.mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

References

[1] L. Weilun, H. Jungong, and P. With, "Flexible human behavior analysis framework for video surveillance applications," *International Journal of Digital Multimedia Broadcasting*, vol. 2010, Article ID 920121, 9 pages, 2010.

[2] P. Barr, J. Noble, and R. Biddle, "Video game values: Human-computer interaction and games," *Interacting with Computers*, vol. 19, no. 2, pp. 180–195, Mar. 2007.

[3] B. Song, E. Tuncel, and A. Chowdhury, "Towards a multi-terminal video compression algorithm by integrating distributed source coding with geometrical constraints," *Journal of Multimedia*, vol. 2, no. 3, pp. 9–16, 2007.

[4] T. Hfllerer, S. Feiner, D. Hallaway, B. Bell, M. Lanzagorta, D. Brown, S. Julier, and Y. B. nad L. Rosenblum, "User interface management techniques for collaborative mobile augmented reality," *Computers and Graphics*, vol. 25, no. 5, pp. 799–810, Oct. 2001.

[5] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 288–303, feb. 2010.

[6] J. Hoey and J. Little, "Representation and recognition of complex human motion," in *Proceedings of IEEE Conference on Computer Vision*, vol. 1. IEEE, 2000, pp. 752–759.

[7] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 7, pp. 1310–1323, july 2011.

[8] H. J. Seo and P. Milanfar, "Action recognition from one example," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 867–882, May 2011.

[9] M. Giese and T. Poggio, "Neural mechanisms for the recognition of biological movements," *Nature Reviews Neuroscience*, vol. 4, no. 3, pp. 179–192, Mar. 2003.

[10] N. Gkalelis, A. Tefas, and I. Pitas, "Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition," *IEEE Transactions on Circuits Systems Video Technology*, vol. 18, no. 11, pp. 1511–1521, Nov. 2008.

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2247–2253, 2007.

[12] A. Iosifidis, A. Tefas, and I. Pitas, "Activity based person identification using fuzzy representation and discriminant learning," *IEEE Transactions on Information Forensics and Security*, no. 99.

[13] M. Ursino, E. Magosso, and C. Cuppini, "Recognition of abstract objects via neural oscillators: interaction among topological organization, associative memory and gamma band synchronization," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 316–335, 2009.

[14] M. Schmitt, "On the sample complexity of learning for networks of spiking neurons with nonlinear synaptic interactions," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 995–1001, 2004.

[15] B. Ruf and M. Schmitt, "Self-organization of spiking neurons using action potential timing," *IEEE Transactions on Neural Networks*, vol. 9, no. 3, pp. 575–578, 1998.

[16] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Proceedings of International Conference on Pattern Recognition*, vol. 3. IEEE, pp. 32–36.

[17] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," *Proceedings of IEEE International Conference on Computer Vision*, vol. 1, 2007.

[18] A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis," *Computer Vision and Image Understanding*, 2011.

[19] S. Yu, D. Tan, and T. Tan, "Modeling the effect of view angle variation on appearance-based gait recognition," in *Proceedings Asian Conf. Computer Vision*, vol. 1, Jan. 2006, pp. 807–816.

[20] O. Masoud and N. Papanikolopoulos, "A method for human action recognition," *Image and Vision Computing*, vol. 21, no. 8, pp. 729–743, Aug. 2003.

[21] M. Ahmad and S. Lee, "Human action recognition using shape and clg-motion flow from multi-view image sequences," *Pattern Recognition*, vol. 41, no. 7, pp. 2237–2252, July 2008.

[22] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2–3, pp. 249–257, Nov./Dec. 2006.

[23] M. Ahmad and S. Lee, "Hmm-based human action recognition using multiview image sequences," *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 1, pp. 263–266, 2006.

[24] F. Qureshi and D. Terzopoulos, "Surveillance camera scheduling: A virtual vision approach," in *Proceedings Third ACM International Workshop on Video Surveillance and Sensor Networks*, vol. 12, Nov. 2005, pp. 269–283.

[25] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *19th International Conference on Pattern Recognition, 2008. ICPR 2008*. IEEE, 2009, pp. 1–4.

[26] M. Piccardi, "Background subtraction techniques: a review," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4. Ieee, 2005, pp. 3099–3104.

[27] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 2002.

[28] S. Haykin, "Neural networks and learning machines," *Upper Saddle River, New Jersey*, 2008.

[29] Y. Le Cun, "Efficient learning and second order methods," in *Tutorial presented at Neural Information Processing Systems*, vol. 5, 1993.

[30] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," 1974.

[31] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.

[32] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3dpost multi-view and 3d human action/interaction database," in *6th Conference on Visual Media Production*, Nov. 2009, pp. 159–168.

[33] N. Papadakis and A. Bugeau, "Tracking with occlusions via graph cuts," *Transactions on Pattern Analysis and Machine Intelligence*, pp. 144–157, 2010.

[34] O. Lanz, "Approximate bayesian multibody tracking," *Transactions on Pattern Analysis and Machine Intelligence*, pp. 1436–1449, 2006.

[35] N. Gkalelis, N. Nikolaidis, and I. Pitas, "View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2009, pp. 394–397.

[36] M. B. Holte, T. B. Moeslund, N. Nikolaidis, and I. Pitas, "3D Human Action Recognition for Multi-View Camera Systems," in *First Joint 3D Imaging Modeling Processing Visualization Transmission (3DIM/3DPVT) Conference*. IEEE, 2011.

[37] A. Iosifidis, N. Nikolaidis, and I. Pitas, "Movement recognition exploiting multi-view information," in *International Workshop on Multimedia Signal Processing*. IEEE, 2010, pp. 427–431.

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

[39] D. Tran and A. Sorokin, "Human activity recognition with metric learning," *Computer Vision–ECCV 2008*, pp. 548–561, 2008.

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