This document is published in:

Neurocomputing, (2012), 75 (1), 78-87.
DOI: http://dx.doi.org/10.1016/j.neucom.2011.03.051

A probabilistic, discriminative and distributed system for the recognition of human actions from multiple views

Rodrigo Cilla , Miguel A. Patricio, Antonio Berlanga, Jose´ M. Molina

Computer Science Department, Universidad Carlos III de Madrid, 28270 Colmenarejo, Madrid, Spain

E-mail addresses: rcilla@inf.uc3m.es (R. Cilla), mpatrici@inf.uc3m.es (M.A. Patricio), aberlan@ia.uc3m.es (A. Berlanga), molina@ia.uc3m.es (J.M. Molina).

Abstract: This paper presents a distributed system for the recognition of human actions using views of the scene grabbed by different cameras. 2D frame descriptors are extracted for each available view to capture the variability in human motion. These descriptors are projected into a lower dimensional space and fed into a probabilistic classifier to output a posterior distribution of the action performed according to the descriptor computed at each camera. Classifier fusion algorithms are then used to merge the posterior distributions into a single distribution. The generated single posterior distribution is fed into a sequence classifier to make the final decision on the performed activity. The system can instantiate different algorithms for the different tasks, as the interfaces between modules are clearly defined. Results on the classification of the actions in the IXMAS dataset are reported. The accuracy of the proposed system is similar to state-of-the-art 3D methods, even though it uses only well-known 2D pattern recognition techniques and does not need to project the data into a 3D space or require camera calibration parameters.

Keywords: Human action recognition, Bayesian networks, Computer vision, Machine learning

1. Introduction

Attaching a semantic meaning to human actions occurring in video streams is useful in different situations. Detecting a crowd running away of a building can be a sign of something having gone wrong inside. An elderly person detected lying on the floor in a room suggests that he may have suffered an accident and needs attention. In both cases, an alarm can be automatically raised to the emergency services to require their presence. Beyond surveillance, human actions can be used to interact with automated systems. Playing video games using realistic body gestures or automatically adjusting the lighting of a room when somebody is detected reading are just a couple of such interactions.

The wide variety of applications of human action recognition has brought the field into the focus of computer vision researchers for the past two decades. Recently published surveys in the area give an idea of the progress made over this time [1–3]. There has been an evolution from the single view systems used in the early days [4–6] to the multiple view setups currently being deployed [7–10]. Multiple view systems have exploited multiple view geometry to overcome the main weaknesses of the single view systems: robustness to occlusions and viewpoint invariance.

Most existing multi-view human action recognition systems have followed a similar scheme. First, some low level image processing is done to the images grabbed at each view to extract 2D features such as a silhouette [7] or body limb segmentation [11] of the monitored human being. These features are then projected into a common scene model to generate a 3D human representation, such as visual hulls [7] or body limb configurations [11].

While these approaches have shown high accuracy in human action recognition tasks, they suffer from some drawbacks [9,12]:

1. *Need for camera calibration*: The usual human action recognition procedure is to project the perceived views into a common representation of the scene. Camera calibration parameters are necessary in order to project the recovered 2D features into a 3D scene model [13]. These requirements are an obstacle to system deployment, because calibration parameters have to be recovered every time a new camera is added or its position changes (maybe accidentally), which is taxing.
2. *Centralized processing*: It is common practice in existing approaches to send the features detected at the camera nodes to a central server, where the action recognition is performed in a common scene model. The server usually accounts for most of the computational requirements of the system, its performance degrades as the number of cameras increases. To make the system scalable, the amount of resources that have to be allocated to the central server when a new camera is added should be minimized or, at least, bounded. The projection and

matching of the 2D features in the 3D scene model is one of the most load-demanding tasks to be performed. A possible way to avoid this is to embed part of the action classification into the cameras, using 2D pattern recognition to make a classification of the action for the view. The different view classifications are then combined using lightweight algorithms to create a single representation of what is going on in the scene.

Another reason for avoiding centralized processing is fault tolerance. In a centralized processing scheme, the action recognition process will not work if the central node breaks. The use of a distributed processing scheme affords fault tolerance, as the different action recognition steps are performed by different nodes.

3. *Bandwidth requirements*: As noted earlier, the camera nodes need to send the computed features to a central server. The amount of information that has to be sent is correlated with how much processing is done at each camera. The bandwidth requirements for sending the acquired raw pixels are greater than for just sending the extracted foreground, which again requires more bandwidth than sending segmented body part locations or just trajectory information. In order to prevent the saturation of communication channels when new cameras are added to the system, it is important to reduce the amount of information exchanged by the nodes, while retaining the utmost informativeness about what is going on. Posterior probabilities may be a compact way of reducing the transmitted information without so much loss.

Hybrid Artificial Intelligence systems [14–16] combine different kinds of techniques to efficiently solve a wide variety of real world problems. In this paper we propose a system for the classification of human actions perceived from multiple viewpoints without performing any explicit 3D reconstruction, exploiting the synergies between probabilistic reasoning and image understanding. 2D features characterizing human motion are extracted for each view of the scene. The features are introduced into a probabilistic classifier to create a posterior distribution on the performed action. A central server gathers the posterior distributions for all the views and combines them into a single posterior for the action. Finally, a dynamic Bayesian network (DBN) is used to model the uncertainty of the temporal evolution of the single frame posteriors. This DBN is used to classify the test action sequences entered into the system. To test the performance of the proposed system, it will be trained using action sequences grabbed from different synchronized views of the scene. Then, new action sequences are presented to the system and the ratio of correctly classified sequences is taken as the quality measure. With the proposed approach, we are able to outdo some of the drawbacks of other distributed multi-camera human action recognition systems [12,17,18], that assume a constant number of cameras in the system or do not handle the uncertainty in the classification in a proper way.

This paper is an extended version of the work presented in [8]. The fusion algorithms presented there are now described in the context of an architecture for the distributed recognition of human actions, and a more extensive validation of them is provided.

1.1. Contributions

The main contributions of this paper is a hierarchical discriminative system for the recognition of human actions from multiple cameras. Different feature descriptors, classifiers, classifier fusion algorithms and sequence models can be instantiated for the different system levels, choosing the most appropriate one for the action recognition task to be performed. The proposed system achieves an accuracy similar to state-of-the-art 3D methods when applied to the IXMAS dataset classification, using only standard pattern recognition techniques applied at each of the available views and combining the results of the local classifications.

1.2. Paper organization

The paper is organized as follows. Section 2 illustrates the components of the proposed system. In Section 3, the methods used to process the images grabbed from each camera are introduced. In Section 4, the algorithms used to combine the results of the local processing are presented. Section 5 describes the sequence classification algorithm used in the system. In Section 6, the system is tested on the IXMAS dataset, and the results are shown and discussed. In Section 7, the state of the art on view-invariant action recognition is reviewed. Finally, Section 8 presents the conclusions of this research.

2. System overview

Fig. 1 illustrates the proposed multi-camera action recognition architecture. C different cameras observe a scene from different viewpoints. It is assumed that there is only a single individual in the scene. This way, we can ignore data for tracking association problems. Without loss of generality, it is also assumed that all C cameras always have a perception of the individual in the scene, although the number of cameras observing the individual may be different at every instant t. This should simplify ongoing formulations. The goal of the system is to select the action α performed by the individual from a set of N predefined actions $A=(a_1,\ldots,a_N)$ known a priori given a set of image sequences $\{I(x,y,t)^c\}$, $1\le t\le T$, $1\le c\le C$, of length T simultaneously acquired by the C cameras observing the scene.

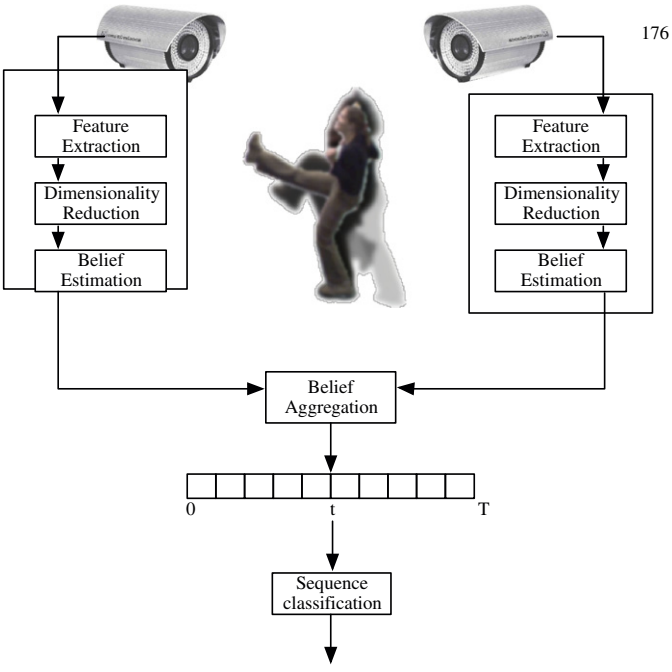

Fig. 1. Overview of the proposed system.

The first step in order to make this decision is to compute an action descriptor $f_t^c \in \mathcal{X}$ from the data grabbed from each view c. $\mathcal{X}$ is the inner product space where the descriptor is defined and typically $\mathcal{X} \equiv \mathcal{R}^D$, although other choices are also possible, for example when using histogram descriptors [19]. f_t^c must capture

enough variability in the data to be able to differentiate the actions in A. Another desirable property is that $\mathcal{X}$ should be compact in order to overcome the problems caused by the *curse of dimensionality*. Manifold learning techniques can be employed to project the original descriptors into a more compact subspace, if necessary.

Once an action descriptor f_t^c has been obtained, a probabilistic classifier is used to create a posterior probability distribution on the performed action given the observed descriptor, $p(a_i|f_t^c)$, $a_i \in A$, $\sum_{i=1}^{N} p(a_i|f_t^c) = 1$. This posterior probability distribution measures the uncertainty of the observed descriptors of being an instance of each one of the categories.

The posterior probability distributions computed for each one of the C views of the scene are combined using a classifier fusion algorithm, generating a posterior distribution $p(\alpha_t|f_t^1 \dots f_t^C)$ on the action performed given the descriptor computed by the different views.

Finally, the posterior probability distributions created at each instant t are entered into a sequence classifier to generate a single posterior distribution on the performed action given the observation sequence $p(\alpha|f_1^1 \dots f_T^C)$. This distribution will be finally used to predict the action of the observed individual.

This architecture distributes the decision making process across multiple nodes. Each node processes the image grabbed from each camera, and makes a partial decision on the action using the information contained just in that image. A central node then grabs the decisions taken by each node and combines them to make the final decision on the performed action. One advantage of this approach is that if a camera breaks the action recognition decision can still be made, as the central node would be still collecting the decisions made by the other nodes. Other advantage is that the computational resources needed to process the image sequences are allocated across different nodes, reducing the amount of resources needed at the central node.

A possible alternative way of structuring the system would be to first classify each sequence at each camera and then sending just one posterior distribution to the central node, as in [12]. However, we are interested in performing frame by frame action segmentation at the central node in the future, assuming different actions happen on the input sequences. If the system would be structured in such way it would be more difficult to make this extension.

3. Single view processing

3.1. Human action representation

The first step in the proposed architecture is to compute a descriptor to capture the variability of the input images. Two different action descriptors will be tested in our system.

3.1.1. Motion history image

The motion history image (MHI), introduced by Bobick and Davies [20], is an appearance descriptor that has been widely used for the recognition of human actions. It recursively accumu-lates the silhouettes of the moving person up to the current frame. It is used in the system as it is the best example of a descriptor incorporating temporal information while providing a framewise output. Let $D(x,y,t)$ be a binary image representing the silhouette of the observed person at time t. A MHI ζ is recursively defined as

$$\zeta(x,y,t) = \begin{cases} 255, & D(x,y,t) = 1, \\ \max(\zeta(x,y,t-1)-\rho,0), & D(x,y,t) = 0, \end{cases} \quad (1)$$

where ρ is a parameter that adjusts the amount of time the presence of the silhouette in a given pixel is remembered. A higher value of ρ implies a shorter memory. The bounding box of the observed person in the MHI is tracked across frames, and resized to a 35×20 box. The resulting pixels are concatenated to generate a descriptor with D_{MHI}=700 dimensions. An example MHI image is shown on Fig. 2.

Fig. 2. Motion history image.

3.1.2. Tran's descriptor

Tran et al. [21] proposed a frame descriptor combining optical flow and appearance. It is used in the system because it has shown a high experimental performance. The bounding box of a human being is normalized to a square box, from which human shape and optical flow are computed. Vertical and horizontal planes of the optical flow are split and blurred. A radial histogram is computed over each of the optical flow planes and the shape. The three histograms are concatenated into 216-d vector. Lastly, a principal component analysis (PCA) reduction of the surrounding past and future vectors is appended to finally generate a descriptor of D_{TRAN}=286 dimensions. Readers are referred to [21] for more details.

3.2. Dimensionality reduction

The action descriptors just introduced have a large dimensionality (D_{TRAN}=286, D_{MHI}=700) and need to be projected into a lower dimensional space in order to prevent the problems derived from "the curse of dimensionality". There is a large corpus of dimensionality reduction techniques suitable for solving this problem (see [22] for a recent survey).

Dimensionality reduction techniques can be divided in two major subgroups: (1) unsupervised dimensionality reduction, whose objective is to project the data to a lower dimension where their variance is maximized and (2) supervised dimensionality reduction, also called discriminant analysis, whose objective is to project the data to a lower dimension where the separation between the different categories of the data is maximized.

As the purpose of this paper is to introduce a general system for the recognition of actions, only the simplest technique of each group will be tested. Principal component analysis (PCA) is the standard unsupervised dimensionality reduction technique and projects the

data points into a lower dimensional subspace where the variance of the training data is maximized. Linear discriminant analysis (LDA) finds projective directions by maximizing the ratio of between-class scatter to within-class scatter. Both methods have been used many times in image processing tasks, the most notable being face classification [23,24]. Readers are referred to any pattern recognition book, such as [25], for more details.

3.3. Action classification

The action descriptor f_t^c computed at each frame will be introduced into a probabilistic classifier in order to generate the posterior probabilities of the performed action given the evidence grabbed at that instant. Examples of suitable classifiers are mixtures of Gaussians [26] or conditional random fields [27]. A support vector machine or a C4.5 tree would be invalid classifiers, as they do not provide a calibrated output suitable for conversion into a posterior probability.

We have chosen a parametric (k-means + naive Bayes) and a non-parametric (nearest neighbor conditional density estimator) density estimator to test our system. The parametric splits the feature space in different regions, estimating the conditional probabilities of each class at each region. The non-parametric estimates the conditional probabilities of each class according to the neighbourhood of a test point. This way the possibility of using a local or a global approach to classification is incorporated to the system.

3.3.1. Nearest neighbor conditional density estimator

The nearest neighbor conditional density estimator (kNN) [25] is a well-known non-parametric conditional density estimator. The estimator locally captures the conditional density around a given test point x. Let K be a fixed neighborhood size and K_i, $\sum_i K_i = K$ the number of neighbors of class a_i

$$p(x|a_i) = \frac{K_i}{K}. \tag{2}$$

3.3.2. K-means + naive Bayes

The space of feature descriptors f_t^c will be quantified using a codebook of size K. Each feature vector will be associated with its nearest center to obtain the word w_k. Codebook centers are computed using the k-means algorithm.

$$p(w_k|a_i) = \frac{p(a_i|w_k)p(w_k)}{p(a_i)}. \tag{3}$$

4. Action fusion

After extracting a set of posterior probability distributions $p(a_t^c|f_t^c)$ from the frame descriptor f_t^c computed for each view, they have to be combined to generate a joint posterior probability distribution $p(\alpha_t|f_t^1,\ldots,f_t^C)$ representing the uncertainty in the classification with respect to the evidence perceived by the different cameras at an instant t.

Two different algorithms will be tested for this task. The first is a voting scheme. The second is a Bayesian network modeling the errors in local classifications.

4.1. Voting

The first algorithm that we tested for the fusion of single view soft classifications is defined as the product of the posterior probabilities:

$$p(\alpha_t|f_t^1,\ldots,f_t^C) \propto \prod_{c=1}^{C} p(a_c|f_t^c). \tag{4}$$

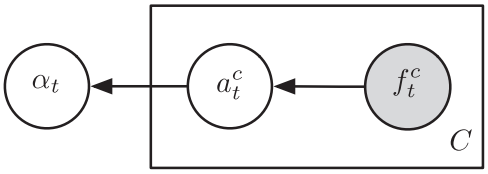

Fig. 3. Plate model of the Bayesian network used to combine the outputs from the classifiers at each camera.

This algorithm is tested as baseline to measure the efficiency of the bayesian network.

4.2. Bayesian network

The second algorithm that we tested for the fusion of single view soft classifications is based on the Bayesian network shown in Fig. 3. The network is composed of observation nodes f_t^c, representing the observation at instant t and camera c, a node α_t representing the activity at time t and a set of latent nodes a_t to model the single view classification.

Given a set of frame descriptors $\mathbf{f}_t = f_t^1,\ldots,f_t^C$, a set of latent variables $\mathbf{a}_t = a_t^1,\ldots,a_t^C$, and the activity label α_t, their joint probability is factorized as

$$P(\alpha_t,\mathbf{a}_t,\mathbf{f}_t) = P(\alpha_t|\mathbf{a_t})P(\mathbf{a_t}|\mathbf{f_t})P(\mathbf{f_t}). \tag{5}$$

The conditional probability given $\mathbf{f_t}$ is then:

$$P(\alpha_t,\mathbf{a}_t|\mathbf{f}_t) = \frac{P(\alpha_t,\mathbf{a}_t,\mathbf{f}_t)}{P(\mathbf{f}_t)} = P(\alpha_t|\mathbf{a_t})P(\mathbf{a_t}|\mathbf{f_t}). \tag{6}$$

The probability $P(\alpha_t,\mathbf{a}_t,\mathbf{f}_t)$ is defined as a product of independent factors, assuming hidden variables a_t^c to be independent:

$$P(\alpha_t|\mathbf{a_t}) \doteq \prod_{c=1}^{C} P(\alpha_t|a_t^c). \tag{7}$$

With this assumption we rule out modeling correlations between local classification errors. In this way, this assumption reduces to two the exponential number of probability distributions that would otherwise need to be estimated. Thus, Eq. (6) can be rewritten as

$$P(\alpha_t,\mathbf{a}_t|\mathbf{f}_t) = \prod_{c=1}^{C} p(\alpha_t|a_t^c)p(a_t^c|f_t^c). \tag{8}$$

Marginalizing over a_t^c:

$$P(\alpha_t|\mathbf{f}_t) = \prod_{c=1}^{C}\sum_{a^c} p(\alpha_t|a_t^c)p(a_t^c)p(f_t^c|a_t^c). \tag{9}$$

Bayesian network parameters are estimated using labeled training samples. $p(a_t^c|f_t^c)$ is known, being provided by the single view soft classifiers, so only $p(\alpha^t|a_c^t)$ needs to be estimated. Let $O^c = (o_1^c,\ldots,o_K^c\}$ be the set of K training frame descriptors computed at camera c with their respective activity labels $Y^c = \{y_1^c,\ldots,y_K^c\}$, $y_k^c \in A$. Model parameters are estimated as

$$p(\alpha^t = a_i|a_t^c = a_j) = \frac{\sum_{k=1}^{K}\gamma_k p(a_t^c = a_j|o_k^c)}{\sum_{l=1}^{N}\sum_{k=1}^{K}\gamma_k p(a_t^c = a_l|o_k^c)}, \tag{10}$$

where $\gamma_k = 1$ if $y_k = a_j$ and $\gamma_k = 0$ otherwise.

5. Sequence classification

Human actions are not isolated occurrences, they happen in sequence. By this time, the reader will probably have noted the t subscript in our formulation. The method proposed until now

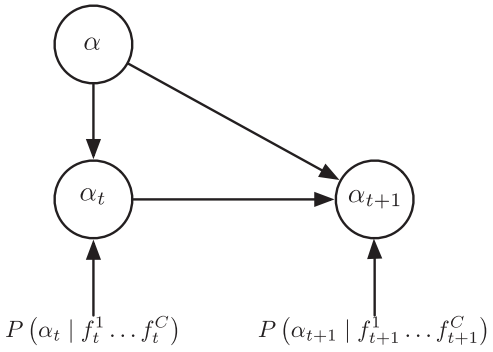

Fig. 4. Dynamic Bayesian network for sequence classification.

considers individual frame descriptors, but ignores sequence dynamics. So, given a sequence of frame descriptors computed at each camera $F = \{f_1^1, \dots, f_1^C, \dots, f_T^1, \dots, f_T^C\}$, we need to associate it with their respective activity α, assuming that there is only one activity performed in the sequence. The sequence length T is not needed to be the same for all sequences.

In this paper a discriminative Hidden Markov Model (HMM) [28] is employed for this task. The probability of a path of hidden node values $H = \alpha_1, \dots, \alpha_T$ given an action class α and an observed sequence F is defined as

$$p(H|F,\alpha) = p(\alpha_1|\alpha)p(\alpha_1|f_1^1 \dots f_1^C) \prod_{t=2}^{T} p(\alpha_t|\alpha_{t-1},\alpha)p(\alpha_t|f_t^1 \dots f_t^C), \tag{11}$$

where $p(\alpha_t|\alpha_{t-1},\alpha)$ is a transition model for each action. This factorization of the probability distribution is graphically shown on Fig. 4. The action α^* performed given a sequence of observed actions F is

$$\alpha^* = \arg\max_{\alpha} p(\alpha|F), \tag{12}$$

where $p(\alpha|F)$ is defined as

$$p(\alpha|F) \propto \sum_{\alpha_T} p(\alpha_T|F,\alpha)p(\alpha). \tag{13}$$

The above quantity can be recursively estimated using the standard forward-backward procedure [28].

The parameters of the model, $p(\alpha_1|\alpha)$ and $p(\alpha_t|\alpha_{t-1},\alpha)$, can be estimated from labeled training samples in a similar way as for the Bayesian network in Section 4.2. We assume an uniform prior on $p(\alpha)$.

6. Experiments

6.1. Experimental setup

Experiments with different instantiations of the proposed system will be conducted using IXMAS: INRIA Xmas Motion Acquisition Sequences [7]. IXMAS is composed of 36 clips recorded by five different cameras in which 12 different actors perform 14 different activities at least three times each. Sample frames are shown in Fig. 5. Only the 11 activities tested in [7] will be used. The frame descriptor proposed by Tran et al. [21] has been downloaded from their web page.[1] The MHI has been extracted from the dataset using a parameter of $\sigma = 10$. The code of our system is available online.[2]

[1] http://vision.cs.uiuc.edu/projects/activity/
[2] http://www.giaa.inf.uc3m.es/miembros/rodri/

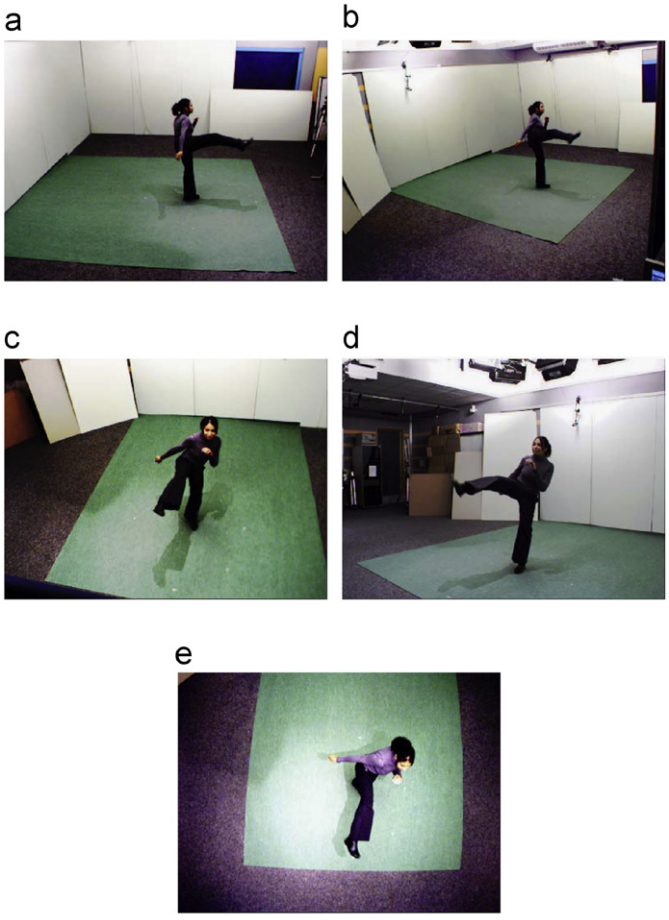

Fig. 5. The *kick* action in the IXMAS dataset from the five available views: (a) Camera 1, (b) Camera 2, (c) Camera 3, (d) Camera 4 and (e) Camera 5.

Two different evaluation protocols will be used to evaluate the system: Leave-One-Sequence-Out (LOSO-CV) Cross-Validation and Leave-One-Actor-Out Cross-Validation (LOAO-CV). LOSO-CV trains the system using all the clips in the dataset except one, that is used for validation. The process is repeated until all the clips have been used for validation once. LOAO-CV trains the system using the clips of all the actors except one, that is used for validation, and repeating the process until every actor in the system has been used for validation. LOAO-CV is a harder evaluation protocol than LOSO-CV, aiming to study how the system performance is expected to degrade when it observes an unknown actor.

PCA and LDA are used to project the frame descriptors into a lower dimensional subspace. In the case of PCA, it has been tested for $d = \{10, 15, 20, 25, 30, 35, 40, 45\}$.

The size of the codebook used in the BN classifier has been experimentally adjusted to $k = 300$ words. The k-NN density estimators will be tested using $k = 3$, $k = 5$ and $k = 7$ neighbors.

A different dimensionality reducer and classifier is trained for each camera in the system, with the images that they grabbed. Classifier fusion and sequence classifiers are then run on the results provided by these classifiers.

In order to compare the different system setups to be tested, we compute the accuracy in the classification as the performance criteria. The accuracy is defined as the ratio between the number of correctly classified samples with respect to the total number of samples presented to the system for validation.

6.2. Results

6.2.1. Single camera classification

Table 1 shows the accuracy of the single frame classifiers. Irrespective of the frame descriptors used, the results reported for cameras 1–4 are quite similar, whereas the accuracy drops by around 10% for camera 5.

Classifiers trained using Tran's descriptor generally provide better results than those trained with the MHI descriptor. Additionally, the accuracy of Tran's descriptors increases with the dimensionality of the PCA projection, whereas it appears to reach saturation point and even decrease in the case of MHI descriptors. In the case of MHI descriptors, the LDA projection always results in worse accuracy than PCA, whereas accuracy is better when combined with the BN classifier in the case of Tran's descriptor.

Regarding the classification algorithms employed, k-NN algorithms are more accurate than the BN algorithm for almost all the choices of descriptor and projection algorithm. As regards the choice of the number of neighbors to use, 7-NN was found to return better results than 3-NN and 5-NN, but the difference is not substantial.

6.2.2. Classifier fusion

Table 2 shows the accuracies achieved after applying the classifier fusion algorithms to the posterior distribution generated from each camera. We find that whereas the voting whereas the algorithm always improves the accuracy of the BN classifiers at least a little, this is not the case for the k-NN classifiers, where the final accuracy is always worse than for the best single view classifier. However, the accuracy provided by the BN algorithm is always better than the best single view classifier by about 10–20%.

6.2.3. Sequence classification

Finally, the results for sequence classification are shown in Table 3. The accuracy improvement is notable when compared with frame-by-frame classification.

Table 1
Results obtained after single camera classification of the IXMAS dataset.

Descriptor		Tran's				MHI			
		Classifier							
Camera	Reducer	3-NN	5-NN	7-NN	NB	3-NN	5-NN	7-NN	NB
1	PCA_{10}	00.4258	0.4283	0.4499	0.4575	0.4315	0.4337	0.4489	0.4574
	PCA_{15}	00.435	0.471	0.4882	0.4947	0.4511	0.4691	0.4781	0.4846
	PCA_{20}	00.4477	0.4867	0.5029	0.5121	0.4566	0.4798	0.488	0.4921
	PCA_{25}	00.4494	0.5013	0.5162	0.5169	0.4559	0.4839	0.4961	0.5063
	PCA_{30}	00.4549	0.5073	0.5239	0.527	0.4577	0.4892	0.5012	0.5045
	PCA_{35}	00.4574	0.5196	0.5303	0.5344	0.458	0.4989	0.5044	0.5084
	PCA_{40}	00.4595	0.5228	0.5324	0.5363	0.4595	0.4972	0.5045	0.51
	PCA_{45}	00.4615	0.5259	0.5408	0.5432	0.4465	0.4993	0.5042	0.5098
	LDA	00.4809	0.4654	0.4859	0.4953	0.4307	0.425	0.4382	0.4426
2	PCA_{10}	00.4142	0.4147	0.4341	0.443	0.4438	0.4593	0.4743	0.4835
	PCA_{15}	00.435	0.4631	0.4783	0.4815	0.4754	0.5168	0.528	0.534
	PCA_{20}	00.4415	0.4801	0.501	0.5042	0.4902	0.5213	0.5273	0.5359
	PCA_{25}	00.4564	0.4986	0.5154	0.5198	0.4896	0.5306	0.5399	0.5441
	PCA_{30}	00.4595	0.5141	0.53	0.5359	0.4817	0.5288	0.5394	0.5461
	PCA_{35}	00.4621	0.5216	0.5357	0.5393	0.4867	0.5341	0.5434	0.548
	PCA_{40}	00.4722	0.531	0.5421	0.545	0.4821	0.531	0.5423	0.5447
	PCA_{45}	00.4719	0.5304	0.5456	0.5491	0.4684	0.5388	0.5431	0.546
	LDA	00.4886	0.4772	0.499	0.5107	0.4528	0.4535	0.4656	0.4707
3	PCA_{10}	00.4236	0.4478	0.4652	0.4694	0.4431	0.4508	0.4646	0.4716
	PCA_{15}	00.4659	0.5066	0.5203	0.5255	0.4609	0.499	0.5121	0.5171
	PCA_{20}	0.4699	0.5231	0.5388	0.5411	0.4868	0.5141	0.5279	0.5327
	PCA_{25}	0.4726	0.5324	0.5466	0.5495	0.4731	0.5192	0.53	0.5378
	PCA_{30}	0.4713	0.5436	0.5505	0.5545	0.4626	0.5195	0.528	0.5326
	PCA_{35}	0.4767	0.5446	0.5568	0.5571	0.4627	0.5137	0.5258	0.5337
	PCA_{40}	0.4677	0.5481	0.5577	0.5608	0.4535	0.5238	0.5308	0.5316
	PCA_{45}	0.4692	0.5491	0.5608	0.5612	0.4548	0.5176	0.5237	0.5292
	LDA	0.4722	0.459	0.4828	0.4894	0.3944	0.401	0.4135	0.4223
4	PCA_{10}	0.4301	0.4274	0.449	0.4586	0.4468	0.4681	0.4843	0.4893
	PCA_{15}	0.4461	0.4692	0.4904	0.4957	0.4755	0.5013	0.5174	0.524
	PCA_{20}	0.4648	0.493	0.5107	0.5158	0.4906	0.522	0.5336	0.5432
	PCA_{25}	0.4772	0.5214	0.5338	0.5373	0.4958	0.5299	0.5418	0.5429
	PCA_{30}	0.4828	0.5307	0.5414	0.55	0.4995	0.5287	0.539	0.5439
	PCA_{35}	0.482	0.5346	0.5503	0.5499	0.4852	0.5307	0.5392	0.5444
	PCA_{40}	0.4824	0.5372	0.5531	0.5571	0.4961	0.5286	0.5438	0.5463
	PCA_{45}	0.4836	0.5398	0.5544	0.5589	0.4821	0.5306	0.5444	0.5472
	LDA	0.5033	0.4965	0.5195	0.5287	0.4817	0.4889	0.5012	0.5065
5	PCA_{10}	0.3708	0.3987	0.4202	0.4329	0.345	0.3628	0.3693	0.3757
	PCA_{15}	0.3614	0.4211	0.4398	0.4477	0.3458	0.3767	0.3862	0.3938
	PCA_{20}	0.3658	0.4337	0.4488	0.455	0.3641	0.3993	0.4093	0.4201
	PCA_{25}	0.3563	0.4387	0.4511	0.4543	0.3651	0.4039	0.4103	0.4157
	PCA_{30}	0.3696	0.4471	0.4579	0.4645	0.3634	0.4004	0.4103	0.4174
	PCA_{35}	0.3606	0.4464	0.457	0.4604	0.3505	0.4001	0.4088	0.4117
	PCA_{40}	0.3724	0.4506	0.4584	0.4615	0.36	0.3966	0.4071	0.4132
	PCA_{45}	0.3711	0.4583	0.4638	0.4649	0.3434	0.3995	0.4062	0.4092
	LDA	0.3354	0.3155	0.341	0.3535	0.2842	0.281	0.2927	0.2991

Table 2
Accuracy obtained after applying classifier fusion algorithms to the IXMAS dataset.

Descriptor		Tran's				MHI			
		Classifier							
Fusion method	Reducer	3-NN	5-NN	7-NN	NB	3-NN	5-NN	7-NN	NB
Bayesian network	PCA_{10}	0.5558	0.5818	0.5989	0.6063	0.6033	0.6052	0.6289	0.6376
	PCA_{15}	0.5833	0.6254	0.6371	0.6412	0.6272	0.6422	0.6604	0.6701
	PCA_{20}	0.5894	0.6475	0.6567	0.6575	0.6424	0.6551	0.6739	0.6821
	PCA_{25}	0.5994	0.6581	0.6637	0.6675	0.6482	0.6632	0.6767	0.686
	PCA_{30}	0.6057	0.6674	0.674	0.677	0.6379	0.6636	0.6768	0.6858
	PCA_{35}	0.6044	0.6705	0.6757	0.6778	0.6429	0.6633	0.6811	0.691
	PCA_{40}	0.6073	0.6722	0.6777	0.6826	0.6398	0.6622	0.6803	0.6865
	PCA_{45}	0.6035	0.6753	0.6841	0.6851	0.6363	0.6647	0.6773	0.6838
	LDA	0.6213	0.6246	0.6371	0.6467	0.591	0.5818	0.5945	0.604
Vote	PCA_{10}	0.515	0.4389	0.4889	0.5154	0.5106	0.4033	0.4575	0.4965
	PCA_{15}	0.5315	0.4751	0.5239	0.5498	0.5232	0.4323	0.4898	0.5271
	PCA_{20}	0.5412	0.4913	0.5359	0.5584	0.5338	0.4551	0.5166	0.552
	PCA_{25}	0.5469	0.4991	0.5433	0.5683	0.544	0.4639	0.5203	0.5544
	PCA_{30}	0.5478	0.5086	0.5557	0.5785	0.5323	0.4632	0.5125	0.5515
	PCA_{35}	0.55	0.5119	0.5534	0.5752	0.5393	0.4605	0.5133	0.5447
	PCA_{40}	0.5538	0.5161	0.557	0.5789	0.5336	0.4606	0.5163	0.5448
	PCA_{45}	0.5492	0.5211	0.5613	0.5849	0.5301	0.4587	0.5117	0.5439
	LDA	0.5433	0.4074	0.468	0.5036	0.4491	0.3451	0.3917	0.4214

Table 3
Accuracy obtained after applying sequence classification algorithm to the IXMAS dataset.

Descriptor		Tran's				MHI			
		Classifier							
Fusion method	Reducer	3-NN	5-NN	7-NN	NB	3-NN	5-NN	7-NN	NB
Bayesian network	PCA_{10}	0.7327	0.8762	0.8688	0.8515	0.8688	0.7995	0.8144	0.8366
	PCA_{15}	0.7723	0.901	0.9059	0.901	0.8812	0.7525	0.8193	0.8342
	PCA_{20}	0.7822	0.9158	0.9158	0.9134	0.8886	0.75	0.8243	0.8441
	PCA_{25}	0.7995	0.9233	0.9158	0.9233	0.896	0.7426	0.802	0.8218
	PCA_{30}	0.8144	0.9257	0.9356	0.9332	0.8936	0.7624	0.8094	0.8366
	PCA_{35}	0.8391	0.9307	0.948	0.9406	0.8812	0.745	0.8045	0.8243
	PCA_{40}	0.8589	0.9233	0.9356	0.9381	0.8911	0.7475	0.7847	0.8045
	PCA_{45}	0.8589	0.9158	0.9455	0.9406	0.8713	0.7351	0.802	0.8119
	LDA	0.8837	0.9035	0.9084	0.9059	0.8144	0.7748	0.7896	0.7921
Vote	PCA_{10}	0.8243	0.8342	0.8441	0.854	0.7772	0.7252	0.7921	0.7698
	PCA_{15}	0.8663	0.8663	0.8762	0.8837	0.7772	0.7054	0.7351	0.7475
	PCA_{20}	0.8614	0.8762	0.8762	0.8861	0.7277	0.7178	0.7178	0.7178
	PCA_{25}	0.8688	0.8861	0.896	0.8886	0.7797	0.7129	0.6708	0.7228
	PCA_{30}	0.8837	0.8812	0.901	0.8985	0.797	0.7005	0.698	0.6683
	PCA_{35}	0.8837	0.8713	0.8911	0.9059	0.7475	0.6856	0.703	0.6634
	PCA_{40}	0.8985	0.8787	0.8936	0.901	0.7599	0.6634	0.552	0.6188
	PCA_{45}	0.8936	0.8985	0.8985	0.8985	0.7574	0.6436	0.6386	0.6139
	LDA	0.8762	0.8267	0.8564	0.8787	0.7376	0.6733	0.7277	0.7376

The behavior of the sequence classification algorithm depends on the origin of the instant classification posteriors that it combines. When using the output from the BN classifier fusion algorithm, the result varies slightly with respect to the number of dimensions used in the frame descriptor for any given classifier. In the case of k-NN classifiers some overfitting can be observed, as the final accuracy starts to drop as the dimensionality grows. When using the voting algorithm, the variation of the results is greater. While the behavior is similar to BN's when applied to Tran's descriptor, the result quickly overfits when applied to the MHI descriptor and drops with the dimensionality.

6.3. Discussion

The results reported in Section 6.2 show that Tran's descriptor is better than the MHI descriptor at the task of recognizing the actions included in the IXMAS dataset. A possible explanation is that Tran's descriptor includes appearance and local motion cues, whereas MHI is based on appearance only. The use of different cues improves the variability of the descriptor, better capturing the variance of the action. Note that classifiers using MHI descriptor start to overfit earlier than classifiers using Tran's descriptor when the dimensionality increases. This again suggests that the content of the Tran's descriptor is richer than the content of MHI descriptor: the latter can be compressed to a smaller number of dimensions than the former.

Another remarkable result is that the use of label information for dimensionality reduction does not improve the results in most cases, or at least not significantly. This might be due to the fact that LDA assumes that each class is unimodal and can be approximated by a Gaussian distribution, whereas the data actually used probably do not fit that assumption.

Table 4
Comparison of the accuracy of the proposed method to other works evaluated with IXMAS dataset.

Method	Protocol	Accuracy	Type
Tran et al. [21]	LOSO	81	2D
Srivastava et al. [12]	LOAO	81.4	Multi-camera
Weinland et al. [7]	LOSO	93.33	3D
Peng et al. [29]	LOSO	94.59	3D
Our	LOSO	94.88	Multi-camera
	LOAO	91.3	

The BN classifier fusion algorithm has been proved to outperform the voting algorithm. The reason is that the BN attaches different weights to the posteriors produced by each camera, according to a model of the usual errors in the classification, whereas the voting algorithm does not use any prior information about classification accuracy.

The results for sequence classification, when compared to instant classification, show that actions are not isolated occurrences, but happen in sequence. It is not enough to consider just one instant in order to recognize actions, and, whenever already available, the past and the future frames have to be employed to make the decision about what is happening or happened.

When globally examining the results, there is one discouraging observation: the best algorithm configuration found for one tier of the system does not guarantee that the best accuracy will be achieved on the next tier up. We observed many times that the accuracy given after the classifier fusion by the classifiers with the best single frame performance is smaller than the reported for other classifiers with a worse performance at the single frame level. There are also similar examples of these phenomena involving the classifier fusion and sequence classification results. This implies that action recognition systems cannot be constructed incrementally in order to find the best configuration, as the configuration with the best result at the highest level is not the configuration with the best result at intermediate levels.

Finally, the accuracy of the proposed system is compared to other proposals reporting results on the IXMAS dataset. Table 4 compares the proposed system to other alternatives. All algorithms are deterministic for stored images. To the best of our knowledge, the proposed system achieves an accuracy similar to the best reported to date [29]. Let us stress that while the best result was based on the classifications of the 3D visual hull, this proposal relies on only well-known simple 2D pattern recognition techniques, without any need of recovering camera calibration parameters.

7. Related work

The research considering how to achieve view-invariant action recognition can be divided into two separate groups: (1) methods proposing action representations that are invariant to camera view and (2) methods combining the perceptions from multiple cameras to take a view-independent decision.

7.1. View-invariant features

The problem of viewpoint action recognition has been studied from the geometrical perspective. Rao et al. [30] introduce a 2D view- invariant descriptor for 3D point trajectories projected in the affine plane. They search the spatio-temporal trajectory curvature to find instants of change. Parameswaran and Chellappa [31] present 2D and 3D invariants for body pose configurations. Gritai et al. [32] propose a metric to compare the trajectories of body parts under anthropometric, temporal and viewpoint transforms. Sheikh et al. [33] approximate the variability in action data as a linear combination of different action bases in spatio-temporal space. The main drawback of these approaches is that they assume that an accurate 3D tracking of the body parts is available, and this is very difficult to achieve in a real scenario.

Other authors have relied on machine learning techniques to create view- invariant action models using only 2D features. Martinez-Contreras et al. [34] project motion history images (MHI) [20] into a subspace that groups viewpoint and movement in a principal manifold using Kohonen self-organizing feature maps. The winner neuron is used to classify the action being performed using HMM smoothing. Tran et al. [21] proposed another approach to achieve view invariance, where view-invariant models are learned using non-parametric classification from a frame descriptor extracted from multiple views including appearance and local motion information. The main weakness of these approaches is the use of only a single view to predict new actions, as different categories may appear similar if they are not observed from the appropriate viewpoint.

7.2. Multi-view systems

Traditionally, multi-view systems have projected the silhouettes obtained from the different views into 3D to create a visual hull [35] of the observed human being. Then, different action descriptors can be extracted from the visual hull. Weinland et al. [7] extended MHI to 3D, introducing motion history volume (MHV). Peng et al. [29] perform a multi-linear analysis of the visual hull to create a descriptor of reduced dimensionality that is introduced in a HMM. These approaches achieve good recognition performance, but visual hull computation requires camera calibration and a lot of centralized processing.

A number of ideas have been proposed to avoid visual hull computation. Srivastava et al. [12] compute a histogram over quantized spatio-temporal salient points [36] for each view. They are then concatenated, and a k-NN classifier is used to decide the performed action. Wu et al. [17] and Määttä et al. [18] propose different ways to combine 2D descriptors computed from different views, but their proposals either assume that there is a constant number of cameras observing the view or use the data coming from the best one only. Our approach improves their proposals as uncertainty is propagated to the upper levels every time that they are classified, taking into account the observations from all the cameras.

8. Conclusions

This paper has presented a distributed human action recognition system. 2D descriptors have been extracted for the frames captured at each one of the available views. They have been projected into a lower dimensional space and introduced into a probabilistic classifier to generate a posterior probability of the performed action. The posteriors for the different cameras have been merged using a classifier fusion algorithm, whose results have been fed into a sequence classifier to make the final decision on the performed action. The system has been tested with different algorithms, exploiting the flexibility provided by the well-defined interfaces between levels. As result, the system achieves an accuracy similar to the state-of-the-art of human action recognition algorithms for classifying the IXMAS dataset.

In the future, we intend to explore the possibility of exploiting the well-defined interfaces between the system levels to include other types of sensors, such as time-of-flight cameras [37] or motion capture devices [38], in order to improve the accuracy of the recognition process. Implementing these ideas with a

multi-agent system, such as the one proposed by Castanedo et al., would be another challenging task [39]. Other future line would be to test in the system more advanced methods. Authors are specially interested in exploring the dimensionality reduction literature to find appropriate methods to reduce the high dimensionality of the action descriptors.

Acknowledgments

Authors acknowledge the valuable feedback given by the anonymous reviewers.

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

[1] 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.
[2] G. Lavee, E. Rivlin, M. Rudzsky, Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video, IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews 39 (5) (2009) 489–504.
[3] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing 28 (6) (2010) 976–990. doi:10.1016/j.imavis.2009.11.014.
[4] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1997, 1997, pp. 994–999 〈doi:10.1109/CVPR.1997.609450〉.
[5] L. Davis, S. Fejes, D. Harwood, Y. Yacoob, I. Hariatoglu, M. Black, Visual surveillance of human activity, Computer Vision ACCV'98 (1997) 267–274.
[6] S. Hongeng, F. Bremond, R. Nevatia, Representation and optimal recognition of human activities, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2000, 2000, pp. 818–825.
[7] 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.
[8] R. Cilla, M. Patricio, A. Berlanga, J. Molina, Fusion of single view soft k-NN classifiers for multicamera human action recognition, Hybrid Artificial Intelligence Systems (2010) 436–443.
[9] H. Aghajan, C. Wu, R. Kleihorst, Distributed vision networks for human Pose analysis, Signal Processing Techniques for Knowledge Extraction and Information Fusion (2008) 181–200.
[10] M. Patricio, F. Castanedo, A. Berlanga, O. Perez, J. García, J. Molina, Computational intelligence in visual sensor networks: improving video processing systems, Computational Intelligence in Multimedia Processing: Recent Advances (2008) 351–377.
[11] D. Gavrila, L. Davis, 3-D model-based tracking of humans in action: a multi-view approach, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1996 (CVPR96), 1996, pp. 73–80.
[12] C. Srivastava, H. Iwaki, J. Park, A.C. Kak, Distributed and lightweight multi-camera human activity classification, in: Third ACM/IEEE Conference on Distributed Smart Cameras, 2009, pp. 1–8.
[13] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, Cambridge, 2000.
[14] A. Abraham, E. Corchado, J. Corchado, Hybrid learning machines, Neurocomputing 72 (13–15) (2009) 2729–2730.
[15] Á. Herrero, E. Corchado, M. Pellicer, A. Abraham, MOVIH-IDS: a mobile-visualization hybrid intrusion detection system, Neurocomputing 72 (13–15) (2009) 2775–2784.
[16] E. Corchado, A. Abraham, A. De Carvalho, Hybrid intelligent algorithms and applications, Information Sciences 180(14) (2010).
[17] C. Wu, A. Khalili, H. Aghajan, Multiview activity recognition in smart homes with spatio-temporal features, in: Fourth IEEE/ACM International Conference on Distributed Smart Cameras 2010, ICSDC 2010, 2010, pp. 142–149.
[18] T. Määttä, A. Härmä, H. Aghajan, On efficient use of multi-view data for activity recognition, in: Fourth IEEE/ACM International Conference on Distributed Smart Cameras 2010, ICSDC 2010, 2010, pp. 158–165.
[19] R. Chaudhry, A. Ravichandran, G. Hager, R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009) 1932–1939. doi:10.1109/CVPRW.2009.5206821.
[20] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (3) (2001) 257–267.
[21] D. Tran, A. Sorokin, D. Forsyth, Human activity recognition with metric learning, in: Proceedings of the 10th European Conference on Computer Vision, Springer-Verlag, 2008, p. 561.
[22] C. Burges, Dimension Reduction: A Guided Tour, Foundations and Trends in Machine Learning 2(4) (2009).
[23] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.
[24] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (2002) 711–720.
[25] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
[26] P. Ribeiro, J. Santos-Victor, Human activity recognition from video: modeling, feature selection and classification architecture, in: International Workshop on Human Activity Recognition and Modeling (HAREM), 2005.
[27] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, Hidden conditional random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (10) (2007) 1848–1852. doi:10.1109/TPAMI.2007.1124.
[28] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.
[29] B. Peng, G. Qian, S. Rajko, View-Invariant Full-Body Gesture Recognition via Multilinear Analysis of Voxel Data, in: Third ACM/IEEE Conference on Distributed Smart Cameras, 2009.
[30] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, International Journal of Computer Vision 50 (2) (2002) 203–226.
[31] V. Parameswaran, R. Chellappa, View invariants for human action recognition, Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 613–619.
[32] A. Gritai, Y. Sheikh, M. Shah, On the use of anthropometry in the invariant analysis of human actions, Proceedings of the 17th International Conference on Pattern Recognition 2004, ICPR 2004, vol. 2, 2004, pp. 923–926.
[33] Y. Sheikh, M. Sheikh, M. Shah, Exploring the space of a human action, 10th IEEE International Conference on Computer Vision 2005, ICCV 2005, vol. 1, 2005, pp. 144–149.
[34] F. Martínez-Contreras, C. Orrite-Uruñuela, E. Herrero-Jaraba, H. Ragheb, S. Velastin, Recognizing human actions using silhouette-based HMM, in: IEEE Conference on Advanced Video and Signal-based Surveillance, 2009, pp. 43–48.
[35] A. Laurentini, The visual hull concept for silhouette-based image understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence (1994) 150–162.
[36] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance 2005, 2005, pp. 65–72. doi:10.1109/VSPETS.2005.1570899.
[37] M. Holte, T. Moeslund, P. Fihl, View invariant gesture recognition using the CSEM SwissRanger SR-2 camera, International Journal of Intelligent Systems Technologies and Applications 5 (3) (2008) 295–303.
[38] D. Minnen, T. Starner, J. Ward, P. Lukowicz, G. Troster, Recognizing and discovering human actions from on-body sensor data, in: IEEE International Conference on Multimedia and Expo 2005, ICME 2005, IEEE, 2005, pp. 1545–1548.
[39] F. Castanedo, J. García, M. Patricio, J. Molina, Data fusion to improve trajectory tracking in a cooperative surveillance multi-agent architecture, Information Fusion 11 (3) (2010) 243–255.