Behavior monitoring through automatic analysis of video sequences

Carlo Colombo
Dipart. Sistemi e Informatica
via S. Marta 3, Florence, Italy
colombo@dsi.unifi.it

Dario Comanducci
Dipart. Sistemi e Informatica
via S. Marta 3, Florence, Italy
comandu@dsi.unifi.it

Alberto Del Bimbo
Dipart. Sistemi e Informatica
via S. Marta 3, Florence, Italy
delbimbo@dsi.unifi.it

ABSTRACT

This paper addresses the problem of classifying actions performed by a human subject in a video sequence. A representation eigenspace approach based on the visual appearance is used to train the classifier. Before dimensionality reduction exploiting the PCA/LLE algorithms, a high dimensional description of each frame of the video sequence is constructed, based on foreground blob analysis. The classification task is performed by matching incrementally the reduced representation of the test image sequence against each of the learned ones, and accumulating matching scores until a decision is obtained; to this aim, two different metrics are introduced and evaluated. Experimental results demonstrate that the approach is accurate enough and feasible for behavior classification. Furthermore, we argue that the choice of both the feature descriptor and the metric for the matching score can dramatically influence the performance of the results.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: vision and scene understanding—*video analysis*

Keywords

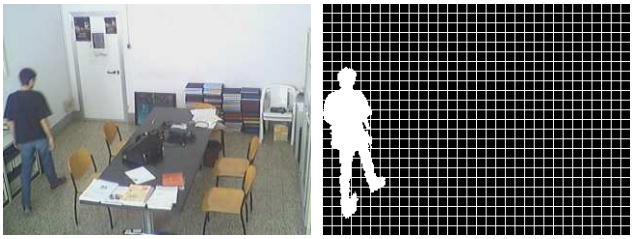
Visual surveillance, Behavior classification

1. INTRODUCTION

Human motion analysis based on compute vision techniques is becoming a core building block of many system for analysis of video information, including human-computer interaction, event based video retrieval and automated surveillance. The problem is to classify the action performed by a human in a video sequence. Several surveys have attempted to summarize the various approaches (e.g. [18], [10], [1]). Following Moeslund and Granum's taxonomy of system functionality for computer vision based human motion capture, the problem encompass either all or some of the following sub-problems: initialization, tracking, pose-estimation, action recognition. Our paper will mainly be focused on the last sub-problem. Various approaches to action recognition have been proposed in the recent past. In [4], two actions are matched by computing the normalized correlation between a set of features extracted from the optical flow. Polana and Nelson in [14] use template matching in the spatio-temporal space; the templates are generated from the statistics of the normal flow. Another common approach is based on spatial and temporal image derivatives. In [9] the temporal image gradient at various temporal scales is used; Laptev and Lindeberg in [8] match representative points from spatial and temporal gradients. The approach proposed by Bobick and Davis in [3] evaluates the Mahalanobis distance between the Hu moments based on the motion history templates from a stack of silhouettes. A stack of silhouettes is also used in [2] to extract spatio-temporal features through Poisson equations; the matching score between two actions is evaluated with the Euclidean distance between these features. Edge images are also used by Rahaman and Ishikawa [15], who use eigenspace analysis to represent actions as manifolds in a multidimensional space. Eigenspace approaches have been proposed for several application scenarios, such as face recognition [17], object representation and recognition [11] and gait analysis [12].

The objective of this work is to learn a representation eigenspace for modelling and classifying the actions performed by moving people. Behaviors are classified with respect to a predefined set of learned actions. The spatio-temporal visual appearance of an action is used to train the classifier. For this purpose, a high dimensional representation of each frame of the video sequence based on blob analysis is constructed, which is aimed at reducing both the amount of visual data being processed, and the effects of noise. Such representation is then made more compact through a process of dimensionality reduction, whose role is both to further decrease computations at matching time, and to retain only the essential characteristics of each behavior. Two different dimensionality reduction approaches are investigated: Local Linear Embedding (LLE) [16] and Principal Component Analysis (PCA) [6]. Each action defines a curve in the reduced space, obtained by interpolating the samples acquired from the video sequence. Once the training set of actions has been acquired, the classification task is performed by matching incrementally the reduced representation of the test image sequence against each of the learned curves, and accumulating matching scores until a decision is obtained. Two different metrics (in the reduced space and in the original space) to compute the matching score are introduced and

CIVR'07, July 9-11, 2007, Amsterdam, Netherlands.



(a) (b)

Figure 1: An original frame (left) and the foreground frame with superimposed representation grid (right).

evaluated. Experiments show that the approach is accurate enough and feasible for real time classification of human actions.

2. EXTRACTING THE REPRESENTATION

At both learning and matching times, video sequences undergo a representation process consisting into two phases: (1) feature selection (always on a frame-by-frame basis) and (2) feature reduction. At learning time, a set of video sequences is captured, one for each action to be recognized later by the classifier; the feature reduction phase is carried out on the whole sequence. At matching time instead, the feature reduction phase takes place for each individual frame of the query sequence.

2.1 Feature Selection

In the feature selection phase, the sequence is processed with a background subtraction algorithm [7] in order to extract the moving blobs corresponding to the subjects performing the action. For each frame, the blob mask is sub-sampled to retain only the lower spatial frequencies, that encode the most relevant features of the behavior at hand. This is achieved by dividing each frame into N rectangular cells and computing the percentage of foreground pixels in each cell. With a frame of size 320×240 pixels, a typical sub-sampling rate is $N = 32 \times 24 = 768$ cells. The resulting frame description effectively captures both the shape of the blob and its position in the image. Fig. 1 shows the sub-sampling grid and the extracted blobs.

2.2 Feature Reduction

In the case of video sequences, consecutive data points (each related to a frame) in the feature space are strongly correlated, and are likely to be close to each other, lying on lower dimensional manifolds (see Fig. 2). Feature reduction attempts to eliminate any redundancies in the original data set. Two feature reduction schemes have been considered and tested: PCA (Principal Component Analysis) and LLE (Local Linear Embedding). The former is simpler and faster, but works well only if data are distributed on linear subspaces. The latter is specifically addressed to problems where data are nonlinearly distributed in the feature space.

At learning time, feature reduction is performed for each training sequence separately: this is done with the idea of obtaining a better representation of individual actions than

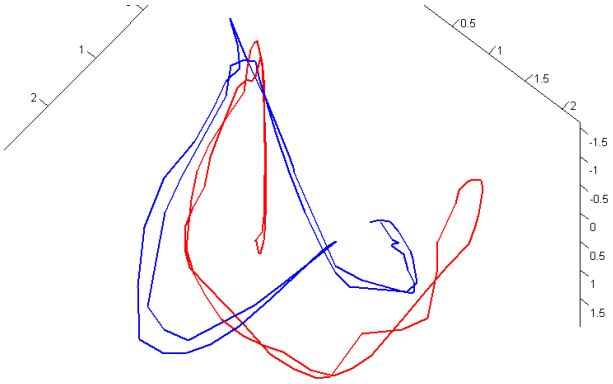


Figure 2: Two examples of actions in the reduced descriptor space ($\mathbb{R}^3$, in this case).

performing the data reduction on all the sequences of the training set together. Another motivation for this "one sequence, one manifold" approach is that it supports an incremental learning scheme, allowing the addition of any new behavior to the classifier block without needing to process again the whole training set.

2.2.1 PCA

Principal Component Analysis is a framework for reducing the dimensionality of a data set consisting of many interrelated variables, yet retaining the variation of the whole data set. This is done by transforming the original data into a new set of variables (the "principal components"), uncorrelated and ordered so that the first few of them hold most of the variation of the original set.

Let $\{\mathbf{X}_i, i = 1 \dots m\}$ be the high-dimensional data set with zero mean (if not, first subtract the average value $\overline{\mathbf{X}}$ of the data set) and compute the covariance matrix $\mathtt{Q} = \mathtt{P}\mathtt{P}^\top$, with $\mathtt{P} = [\mathbf{X}_1 \dots \mathbf{X}_m]$. The result of the eigenstructure decomposition of $\mathtt{Q}$ is a set of decreasing eigenvalues $\{\lambda_1 \dots \lambda_m\}$ and a corresponding set of orthonormal eigenvectors $\{\mathbf{e}_1 \dots \mathbf{e}_m\}$. Though all the eigenvectors are required to reconstruct perfectly any element $\mathbf{X}_i$ of the set, just few of them are enough to give a good approximation of it. Thus, the first n eigenvectors $\{\mathbf{e}_1 \dots \mathbf{e}_n\}$ constitute the principal components of the data set: they define a subspace where to project the original data. The projection matrix is given by

$$\mathtt{P}_{pca} = [\mathbf{e}_1 \dots \mathbf{e}_n]^\top \quad . \tag{1}$$

2.2.2 LLE

Local Linear Embedding is a recent framework proposed for non linear dimensionality reduction; it attempts to discover nonlinear structure in high dimensional manifolds assuming that each data point and its neighbors lie on or close to a locally linear patch of the manifold. Given enough samples $\mathbf{X}_i$ of the manifold in $\mathbb{R}^N$, the local geometry of each patch can be characterized the linear coefficients w_{ij} obtained by expressing each single point as a combination of its K neighbors. This is achieved by minimizing the global error

$$\sum_i \left| \mathbf{X}_i - \sum_j w_{ij} \mathbf{X}_j \right|^2 \tag{2}$$

with the constraints $w_{ij} = 0$ if $\mathbf{X}_j$ is not a neighbor of $\mathbf{X}_i$, and $\sum_j w_{ij} = 1$. A least squares solution for the optimization problem above can be found. The weights w_{ij} thus extracted are invariant to rotation, rescaling and translation of the patch, and characterize its intrinsic geometry. Suppose that the data lie on a smooth manifold of dimensionality $n \ll N$: since it is expected that the local geometry of the original data space is maintained after the reduction process, it is possible to find a set of points $\mathbf{x}_i \in \mathbb{R}^n$ that minimize the embedding cost function

$$\sum\nolimits_i \left| \mathbf{x}_i - \sum\nolimits_j w_{ij} \mathbf{x}_j \right|^2 \tag{3}$$

with the set of weights $\{w_{ij}\}$ kept fixed. The points $\mathbf{x}_i$ represent the samples of the manifold at lower dimension n and are obtained by solving an eigenvector problem.

LLE provides just a better representation of the manifold at low dimension, but cannot provide directly a mapping between the high dimensional original space and the low dimensional embedding space. A possible approach to obtain such a mapping is using the pairs $(\mathbf{X}_i, \mathbf{x}_i)$ as labelled examples for some learning algorithm. In [5] the Radial Basis Function interpolation framework [13] was used to learn such nonlinear mapping. The paper shows how to learn a mapping from the embedded manifold to the original space and – exploiting the particular form of the radial basis function employed – how to obtain the inverse mapping. The multiple radial basis function interpolants $f_k : \mathbb{R}^n \to \mathbb{R}$ are found, where k is the k-th dimension of the original space. The function chosen has the form

$$f_k(\mathbf{x}) = \begin{bmatrix} 1 & \mathbf{x}^\top \end{bmatrix} \mathbf{c}_k + \sum\nolimits_i u_{ki} \phi(|\mathbf{x} - \mathbf{x}_i|) \quad , \tag{4}$$

where $\mathbf{c}_k \in \mathbb{R}^{n+1}$, u_{ki} are real coefficients and $\phi(\cdot)$ is a real-valued basis function. Typical choices for $\phi(v)$ are thin-plate splines ($\phi(v) = v^2 \log v$), Gaussians ($\phi(v) = e^{-\alpha v^2}$) and bi-harmonic splines ($\phi(v) = v$). The whole mapping can be written in matrix form as

$$\mathbf{X} = f(\mathbf{x}) = \mathtt{B}\psi(\mathbf{x}) \ ,$$

with $\mathtt{B} \in \mathbb{R}_{N \times (m+n+1)}$ (m is the number of samples) and $\psi(\mathbf{x}) = \left[\phi(|\mathbf{x} - \mathbf{x}_1|), \ldots, \phi(|\mathbf{x} - \mathbf{x}_m|) \ 1 \ \mathbf{x}^\top\right]^\top$. Given a new input data $\mathbf{Y} \in \mathbb{R}^N$, the corresponding point $\mathbf{y}$ in the embedding space is found as the solution of

$$\mathbf{y} = \arg\min_{\mathbf{x}} |\mathbf{Y} - \mathtt{B}\psi(\mathbf{x})|^2 \ . \tag{5}$$

Given the particular form of $\psi(\mathbf{x})$, a linear approximation for the solution of (5) can be obtained by solving for $\psi(\mathbf{x})$ (using the pseudo-inverse of $\mathtt{B}$) and taking the last n rows of $\psi(\mathbf{x})$:

$$\mathbf{y} = \mathtt{S} \ \mathtt{V}\tilde{\mathtt{D}}\mathtt{U}^\top \mathbf{Y} = \mathtt{P}_{lle}\mathbf{Y} \ . \tag{6}$$

Matrices $\mathtt{V}$, $\tilde{\mathtt{D}}$, $\mathtt{U}$ are obtained by the SVD decomposition of $\mathtt{B} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$ and $\tilde{\mathtt{D}}$ is the diagonal matrix defined by taking the inverse of the nonzero singular values of $\mathtt{D}$ and setting the rest to zero. Matrix $\mathtt{S}$ has the form $\mathtt{S} = [\mathtt{0}_{n,N-n} | \mathtt{I}_n]$.

3. MATCHING AND CLASSIFICATION

At matching time, each frame of an action to be classified is projected onto the embedding space of each learned behavior, and the closest point on the low-dimensional manifold $\mathcal{M}$ is found as

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} \{ |\tilde{\mathbf{x}} - \mathbf{x}| : \tilde{\mathbf{x}} \in \mathcal{M} \} \ . \tag{7}$$

Such a point is used to compute the matching score between the query frame and each learned behavior.

Two different metrics for the matching score between a frame and a behavior have been defined. As a consequence of the "one sequence, one manifold" learning scheme, the embedding space for each learned behavior can have its own dimension. The first metric is computed in the embedding space exploiting the distance between $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$; such a distance is divided by the length $L_{\mathcal{M}}$ of the diagonal of the hypercube containing the manifold $\mathcal{M}$ in the embedding space. Normalization is introduced with the intention of eliminating the different size and dimension effect of a manifold with respect to another. The matching score for the distance computed in the embedding space is evaluated as:

$$\varepsilon^e = \exp\left(-\frac{|\mathbf{x} - \hat{\mathbf{x}}|}{L_{\mathcal{M}}} \right) \ ; \tag{8}$$

i.e. it has a value between 0 and 1 and decreases as the embedding error $|\mathbf{x} - \hat{\mathbf{x}}|$ increases. The second way to obtain a measure of the matching score is to use a "mean" reconstruction error instead of the distance in the embedding space. The mean reconstruction error can be obtained by back-projecting the closest point in the embedding space onto the original high-dimensional data space, then computing the Euclidean distance between the current and the reconstructed frame descriptors, and dividing it by dimension N of the feature space. Since the reconstruction error is computed in the original frame descriptor space, it can detect easily if the form of the reconstructed blob is similar to the input one. Furthermore, it is an independent measure with respect to each manifold. In particular, in the PCA case the embedded vector is evaluated as $\mathbf{x} = \mathtt{P}_{pca}\mathbf{X}$, and the reconstructed frame descriptor in the high-dimensional space is computed as $\hat{\mathbf{X}} = \mathtt{P}_{pca}^\top \hat{\mathbf{x}}$. On the other hand, in the LLE case the matrix obtained as the linear approximation $\mathtt{P}_{lle}$ in eq. (6) is used to compute the embedded vector $\mathbf{x} = \mathtt{P}_{lle}\mathbf{X}$, and the reconstructed frame descriptor can be obtained as $\hat{\mathbf{X}} = \mathtt{B}\psi(\hat{\mathbf{x}})$. In both cases, the matching score is then evaluated as

$$\varepsilon^r = \exp\left(-\frac{|\mathbf{X} - \hat{\mathbf{X}}|}{N} \right) \ . \tag{9}$$

Given M learned actions, the matching scores ε_i, $i = 1 \ldots M$ (what follows can be applied both to ε^e and ε^r) between the current frame and the i-th behaviour are collected frame by frame. Then all the scores are normalized such that $\sum_i \varepsilon_i = 1$, so as to represent probability values. Specifically, each normalized matching score is interpreted as the probability that the i-th action is being performed. Since each action a_i is an event incompatible with all the others, we can write

$$P\left(\sum_i^M a_i(t) \right) = \sum_i^M P(a_i(t)) \ , \tag{10}$$

where $P(a_i(t))$ represents the probability that at time t the action a_i is being performed.

To collect matching score observations in a compact way as time goes by, the probability $P(\hat{a}(t), \hat{a}(t-1), \ldots, \hat{a}(t_0))$ is evaluated; this is the probability that the same action $\hat{a}$ has been performed from time t_0 to time t. For simplicity, the event $\hat{a}(t)$ ("the action $\hat{a}$ is performed at time t") is considered independent from the previous events $\hat{a}(t-k)$ ("the action $\hat{a}$ is performed at time $t-k$"). Hence,

$$P\left(\prod_{k=0}^{t-t_0} \hat{a}(t-k)\right) = \prod_{k=0}^{t-t_0} P(\hat{a}(t-k)) \ . \qquad (11)$$

The system reaches a classification decision when there is an action with joint probability much higher than all the others:

$$\frac{\max_{a_i} P\left(\prod_{k=0}^{t-t_0} a_i(t-k)\right)}{\sum_i P\left(\prod_{k=0}^{t-t_0} a_i(t-k)\right)} > \gamma_t \ , \qquad (12)$$

where the threshold γ_t is a percentage value that gradually decreases as the decision time increases.

3.1 Detection of unknown behaviors

The classification approach described above cannot handle actions that do not belong to the action data set. Therefore, a strategy is needed to avoid that, if an unknown behavior is input to the system, the classification response be in any case one among the learned behaviors. To achieve that, a temporal threshold τ is used to classify as belonging to the extra class "unknown behavior" every input for which a classification decision has not been reached within τ steps. Indeed, if the query behavior is quite different from the learned ones, the matching score values with respect to the actions in the learning set are likely to be all approximately equal. As a consequence, the probabilistic approach of eq. (12) is likely not to reach a decision in the usual time required to classify known query inputs. The temporal threshold can be empirically chosen as

$$\tau = \mu_f + 2\sigma_f \ , \qquad (13)$$

where μ_f and σ_f are the estimated mean and standard deviation of the time needed for classifying known query inputs (see Fig. 3). Actually, τ should be considered as a "soft" threshold, the "hard" threshold being $\mu_f + 3\sigma_f$. In fact, before definitely labelling as "unknown" a query input, when the classification time τ is reached a check is performed to be sure that the classification probabilities are approximately equal. If it happens that a behavior has a dominant probability instead, an extra classification time $\Delta\tau = \sigma_f$ is allowed.

4. EXPERIMENTS AND RESULTS

In order to test system performance, a training set of actions taking place around a table was defined. Fig. 4 shows the experimental environment from the camera viewpoint. The training actions included walking along the four sides of the table and sitting on the four chairs, for a total of eight actions—see Fig. 5. The system learned a video sequence for each action at setup time; after that, similar actions were performed in random order by two different subjects, one of them also involved in the training sequences. Video sequences were temporally sub-sampled to obtain a frame rate of 10 frames/s.

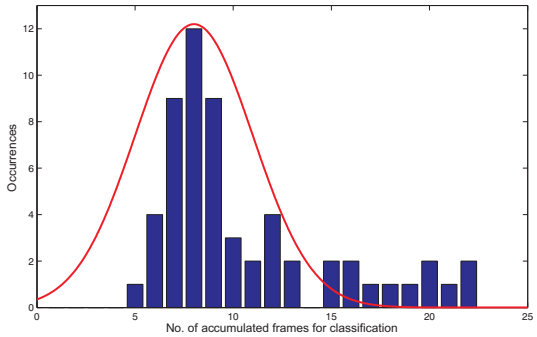


Figure 3: Statistical distribution of classification times.

Figure 4: The environment used for testing.

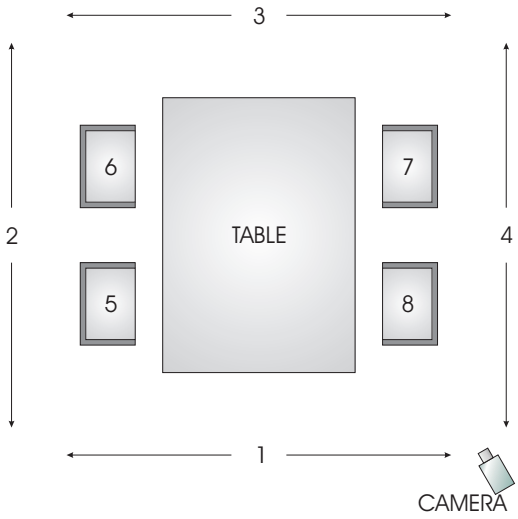


Figure 5: Actions performed around the table. From 1 to 4 they are walking along a side of the table; form 5 to 8 they are sitting in the relative chair.

	1	2	3	4	5	6	7	8
1	86.7	.	.	6.7	.	.	.	6.6
2	.	94.4	5.6	.	.	.	.	.
3	.	.	100	.	.	.	.	.
4	8.3	.	25.0	66.7	.	.	.	.
5	12.5	.	.	.	87.5	.	.	.
6	.	.	.	.	.	100	.	.
7	.	.	10.0	.	.	.	90.0	.
8	.	.	.	10.0	.	.	.	90.0

Table 1: Classification percentage using PCA and the same subject involved in the training phase.

	1	2	3	4	5	6	7	8
1	72.2	5.6	.	5.6	.	.	5.6	11.0
2	.	97.7	2.3	.	.	.	.	.
3	.	8.7	88.4	.	.	2.9	.	.
4	11.1	.	22.2	66.7	.	.	.	.
5	.	.	7.7	.	84.6	7.7	.	.
6	.	.	5.0	.	.	95.0	.	.
7	.	.	11.8	.	.	.	88.2	.
8	.	.	.	.	.	.	.	100.0

Table 2: Classification percentage using PCA and a subject different from the one involved in the training phase.

As first experiment the normalized metric in the embedding space of eq. 8 was tested, but such a metric did not give good results. A better performance was noted for LLE with respect to PCA: for PCA almost none of the action perfomed was correctly labelled, while for LLE the recognition rates of four action ranged between 62.61% and 100%. We explain the fact with the particular form of the feature descriptor: since many elements of the input vector are zero (no foreground in the cell), when projecting on the manifolds the corresponding column of the projection matrix are eliminated in the product computation. As a consequence, when a blob is not near the region where a given behavior must happen, it is projected in the embedding space in a unpredictable way: with this descriptor the distance in the embedding space is not a good measure for the matching score. The better performance of LLE is probably due to the fact that the projection matrix is computed in an independent way, involving radial basis functions along the manifold.

The second part of experiments deals with the use of the second metric (mean reconstruction error) in eq. 9 as matching score. Tabs. 1 and 2 show the confusion matrix using PCA, while Tabs. 3 and 4 deal with LLE. Performance is expressed in terms of percentage of right/wrong classifications. When an action is incorrectly confused with another action taking place in the same image region, the resulting misclassification is considered to be more acceptable than in the case when the two actions take place in completely different image regions. For instance, confusing actions 4 (i.e., walking along one right side of the table) and 2 (i.e., walking along the left side of the table) is interpreted as a much more serious misclassification than confusing actions 8 (i.e., sitting on a chair along the right side of the table)

	1	2	3	4	5	6	7	8
1	75.0	5.0	5.0	5.0	.	.	10.0	.
2	19.2	65.4	15.4	.	.	.	.	.
3	.	.	100.0	.	.	.	.	.
4	10.0	5.0	5.0	60.0	.	.	10.0	10.0
5	7.7	.	.	.	92.3	.	.	.
6	.	.	20.0	10.0	.	70.0	.	.
7	.	.	.	2.1	.	.	97.9	.
8	8.0	14.3	.	.	.	.	20.6	57.1

Table 3: Classification percentage with LLE and the same subject involved in the training phase.

	1	2	3	4	5	6	7	8
1	36.5	8.8	17.6	10.6	.	.	14.7	11.8
2	21.9	56.0	12.5	6.5	3.1	.	.	.
3	6.7	8.3	73.4	5.0	1.6	5.0	.	.
4	17.3	34.5	.	39.1	.	.	.	9.1
5	.	.	3.0	9.1	87.9	.	.	.
6	3.9	7.7	11.5	39.2	.	37.7	.	.
7	2.9	.	5.7	22.8	.	2.9	65.7	.
8	31.7	29.3	2.4	2.4	.	.	7.3	26.9

Table 4: Classification percentage with LLE and a subject different from that one involved in the training phase.

and 4. From the tables is apparent that PCA performs better than LLE. In fact, the PCA confusion matrix is much more similar to a diagonal matrix than LLE's for both the subjects under test. Moreover, PCA does not give rise to serious misclassifications, while LLE presents several cases of serious misclassifications, especially in the case when a different subject from the training one is involved (see e.g. the 34.5% confusion percentage for actions 2 and 4). The worse behavior of LLE as compared to PCA is due to the appearance of "ghosts" during the vector back-projection phase (see Fig. 6). "Ghosts" are foreground artifacts that arise due to spatial interferences among radial basis functions: their effect is to alter the matching scores of the various actions, some of which are increased, and some decreased in the wrong way. The emergence of ghosting effects calls for a revision of the remapping policy for LLE.

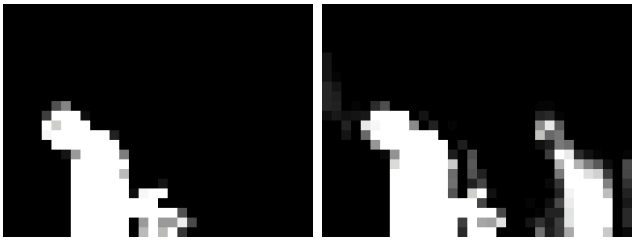


Figure 6: An original frame descriptor (a) and the back-projected one obtained with LLE (b). "Ghosts" are visible in the right side of the image.

	1	2	3	4	5	6	7	8	9
1	87.5	.	.	.	.	.	12.5	.	.
2	10.0	80.0	10.0	.	.	.	.	.	.
3	.	.	90.9	.	.	4.6	.	.	4.5
4	6.3	.	25.0	25.0	.	.	25.0	.	18.7
5	.	.	.	.	.	.	.	.	.
6	.	5.9	.	.	.	88.2	.	.	5.9
7	.	.	25.0	.	.	.	65.0	.	10.0
8	.	.	.	.	.	.	.	94.4	5.6
9	13.3	.	8.3	16.7	.	.	.	.	61.7

Table 5: Behavior classification percentage with the criterion of section 3.1 to detect unknown actions. PCA is used. The unknown behavior is labelled as number 9. The same person involved in the training phase performs the video sequence.

	1	2	3	4	5	6	7	8	9
1	100	.	.	.	.	.	.	.	.
2	7.1	85.8	.	.	.	.	.	.	7.1
3	.	.	100	.	.	.	.	.	.
4	10.0	.	40.0	30.0	.	.	10.0	.	10.0
5	.	.	.	.	.	.	.	.	.
6	.	.	2.8	.	.	97.2	.	.	.
7	.	.	.	.	.	.	.	.	.
8	23.5	.	.	.	.	.	.	70.6	5.9
9	.	20.0	.	.	.	.	.	.	80.0

Table 6: Experiment analogous to that related to Tab. 5, but with a different person employed for the test.

Tabs. 5 and 6 deal with thecriterion of section 3.1 employed to detect unknown behaviors. The data reduction algorithm used was PCA. The test sequences for these experiments were particularly challenging, as they featured not only actions different from the ones learned, but also interactions between the subjects and the environment, i.e. displacements of the chairs around the table, and of some objects on the table. The tables show that the correct classification percentage decreases with respect to the previous experiment—notice in particular that the confusion matrix is more sparse than before. Nevertheless, serious misclassifications are absent also in this case, and performance appears to be still acceptable.

5. CONCLUSIONS

An approach for the classification of actions performed by a human subject in a video sequence was presented. A high dimensional description of each frame of the video sequence is introduced, based on foreground blob analysis, and the intrinsic features of each behavior sequence are extracted by computing LLE/PCA feature reduction. Experimental results shows some problems with the choice of the feature descriptor and the projection in the embedding space, specially for PCA. On the other hand, back-projection provides good results for PCA and a worst performance for LLE. Thus, future works will deal with finding a better mapping between feature space and embedding space for LLE, together with the study of other feature descriptors. To end, more challenging action dataset will be investigated and the extension to monitoring more than one person will be addressed.

ACKNOWLEDGMENT

We thank Luigi Nuti for his help and contribution to this work.

6. REFERENCES

[1] J. Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU*, 73(3):428–440, 1999.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. of ICCV*, pages 1395–1402, October 2005.

[3] A. Bobick and J. Davis. The representation and recognition of action using temporal templates. *PAMI*, 23(3):257–267, 2001.

[4] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. of ICCV*, pages 726–733, October 2003.

[5] A. Elgammal. Nonlinear generative models for dynamic shape and dynamic appearance. In *Proc. of 2nd International Workshop on Generative-Model based vision*, pages 182–189, 2004.

[6] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[7] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *2nd European Workshop on AVSS.*, 2001.

[8] I. Laptev and T. Lindeberg. Space time interest points. In *Proc. of ICCV*, pages 432–439, October 2003.

[9] L. Z. Manor and M. Irani. Event-based analysis of video. In *Proc. of CVPR*, volume 2, pages 123–130, 2001.

[10] T. Moeslund and E. Granum. A survey of computer vision based human motion capture. *CVIU*, 81(3):231–268, 2001.

[11] H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appereance. *IJCV*, 14(5):39–50, 1995.

[12] H. Murase and R. Sakai. Moving object recognition in eigenspace representation: gait analysis and lip reading. *Pattern Recognition Letters*, 17(2):155–162, 1996.

[13] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of the IEEE*, 78(9), 1990.

[14] R. Polana and R. Nelson. Recognizing activities. In *Proc. of CVPR*, pages 815–818, 1994.

[15] M. Rahaman and S. Ishikawa. Human motion recognition using an eigenspace. *Pattern Recognition Letters*, 26(6):687–697, 2005.

[16] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[17] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. of CVPR*, pages 586–591, 1991.

[18] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.