# A prototype system for unconstrained face verification based on statistical learning

Augusto Destrero, Alberto Lovato, Francesca Odone

# A prototype system for unconstrained face verification based on statistical learning

Augusto Destrero[1,2], Alberto Lovato[1], and Francesca Odone[2]

[1] Imavis srl, Via Greto di Cornigliano 6R, 16152 Genova, ITALY
{ augusto.destrero, alberto.lovato } @imavis.com
[2] DISI, Via Dodecaneso 35,16146 Genova, ITALY
odone@disi.unige.it

**Abstract.** This paper discusses the video processing modules of a prototype face verification system. The main modules (face detection, registration, and verification) are all based on a feature selection plus classification pipeline that implements recently proposed statistical learning algorithms. All these modules are running on the prototype since January 2008, performing face verification in real time. Here we report quantitative results obtained both off-line, on a previously recorded dataset, and on-line, on the verification system.

## 1 Introduction

After many years of theoretical and algorithmic investigations, machine learning findings are nowadays mature for exploitation and technology transfer. One application domain where such a transition is already taking place is image and video processing. In particular, the literature is rich with many interesting approaches related to face detection and face recognition that may lead to effective solutions for the biometrics market.

In this paper we propose a prototype verification system based on a simple hardware infrastructure and on software modules that combine state-of-the-art video processing and statistical learning algorithms. The adopted solution allows us to achieve real time face verification that may be applied to monitor entrances in unconstrained environments. Here we mainly focus on the prototype video-processing functionalities, in particular on face detection, registration and verification in videos. All these phases are addressed according to a common pipeline: during training we first look for a data-driven image representation by means of a regularized feature selection algorithm recently proposed [1,2]. Then, we use this representation to train an appropriate classifier — here we simply adopt an SVM classifier[3]. At run time we evaluate the video stream on a frame-by-frame basis: for what concerns face detection we look for faces in each frame, as for face verification we analyse each frame of a video and then compute a cumulative feedback (genuine user or impostor) based on the evidence gathered from all frames.

The choices adopted are based on the following *system requirements*:

1. do not put constraints on users posture and behaviours
2. allow for fast computation (overall processing in real time)
3. set the basis for a simple enrollment phase
4. produce compact models for future engineering (e.g., storage of visual identifiers on small memory buffers)

The limited amount of constraints allowed calls for the use of models that are robust to appearance variations. At the same time, robust model-based approaches (e.g., 3D models) are not suitable for real time processing. *Learning from examples* guarantees a trade-off between the two requirements. We address both detection and verification as binary classification problems, relying on appropriate datasets acquired in the calibration and enrollment phase respectively. The adopted learning tools allow us to obtain satisfactory performances with relatively small datasets, with a clear advantage for the design of the enrollment phase. Requirement 3 is also addressed by *exploiting video information*, since videos are easier to capture than independent sets of images, and by *designing an appropriate operative environment*: users walk down a corridor in front of the camera and therefore datasets of reasonable size are easily collected. As for requirement 4 we rely on a *data-driven feature selection* mechanism to find the most appropriate description for each enrolled individual. Starting off from an over-complete description, based on features successfully used for face recognition (LBP features), we select meaningful features for each enrolled individual via data-driven feature selection.

Let us briefly sketch pertinent state-of-the-art in order to set the adopted algorithms in the appropriate frame. In recent years many face-related problems have been addressed by learning from examples. In particular this paradigm dominated the face detection scene since the early 90s and contributed to obtaining promising solutions that deal with view-point changes, rotations, scale and illumination variations — see [4]. Among such a vast literature, component-based approaches highlighted the fact that local areas are often more descriptive and more appropriate for dealing with occlusions and deformations [5,6]. Very popular approaches are derived from Adaboost (see, for instance, [7,8]). As for face recognition [9] our approach relates to feature-based methods, specifically on template-based descriptions. Possibly the first local approach to face recognition is due to Pentland and his co-workers [10]. Among local descriptors for face recognition, Local Binary Patterns (LBP) [11] are becoming very popular for their ability to capture descriptive features of a given texture and in particular of faces (see, for instance, [11,12,13]).

The paper is organized as follows. Section 2 is an overview of the prototype system architecture, Section 3 reports the various video analysis phases, from preprocessing to verification. Sec. 4 reports an analysis of the performances of the system. The results reported are mainly releted to face verification, in particular on the effectiveness of the feature selection strategy for this specific problem, and on the way information coming from various frames is condensed in a unique feedback. Sec. 5 is left to a final discussion and to an account of future works.

## 2 System architecture

The face verification prototype is composed by (1) a PC-based verification unit, (2) a RFID card reader, and (3) a monitoring and feedback unit — see Figure 1, left. The *verification unit* is based on the ImAnalysis library [3]. Together with video acquisition and processing functionalities it contains the users database and the events database, and may be accessed remotely in particular to perform administration tasks. The *RFID card reader* is a MIFARE (ISO/IEC 14443) proximity reader with an activation range of [5-7]cm; it is configured to read RFID tags that store the ID of the enrolled users. The *monitoring and feedback unit* is composed of a color analog video camera and a monitor used to attract the users attention and to send them feedbacks on the verification status (Fig. 1, right). The prototype is installed in a corridor with the card reader installed on one side and the monitoring unit on the other side, by the side of the monitored entrance. The user has to walk in front of the camera for a few seconds. In the case of a genuine user, as soon as the system has gathered enough positive evidence, it returns a positive feedback via the monitor.

We now analyse the system operative phases, first considering the case of a user known to the system (i.e., previously enrolled). Verification starts when the user brings the RFID tag in his possession near the RFID reader. Upon a positive feedback from the reader, obtained in a few seconds, the user may walk down the corridor at a normal speed. Considering the corridor extension (about 4 meters) it takes about 10 seconds to reach the other end of the corridor. In the meanwhile, the card reader associates to the current user an identity to be verified and sends the verification system, via a TCP/IP connection, the RFID-id. The verification system, that hosts the identities database, retrieves the gallery associated to the RFID-id, then starts building the current appearance model of the user walking down the corridor. The verification phase is based on comparing the gallery with current model, as described in Section 3.

To enroll a new user, the system associates to him the enrollment state, where user personal data have been inserted into the database, but no gallery is available. If the user is in the enrollment state, when he slides the RFID tag, the system acquires data to the purpose of building a dataset for training and validating the module.

Notice that the gallery of a user is based on all the data collected when he is in the enrollment mode, and could consist of information obtained from various videos. Instead, when the user is in verification mode a new test model (probe) is built each time he requires his identity to be verified. In this case the probe is derived from current video only.

## 3 Video processing

The verification unit described in Sec. 2 consists of various software and system modules including the video processing functionalities. In this section we

[3] http://www.imavis.com

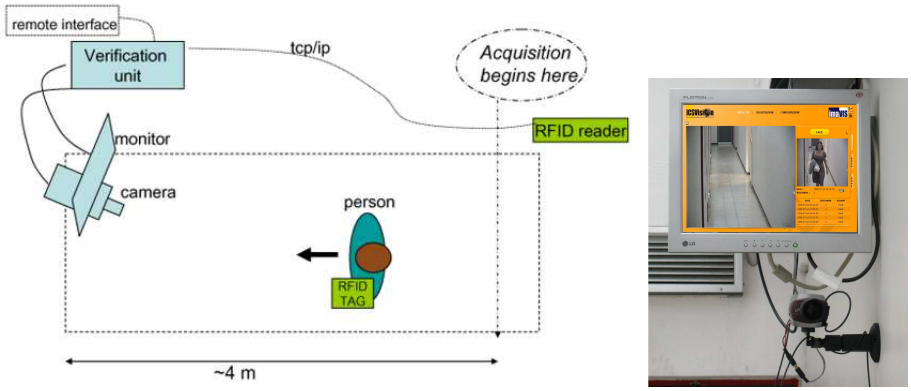


**Fig. 1.** Left: a sketch of the system architecture and of the operative environment. Right: a picture of the monitoring and feedback unit.

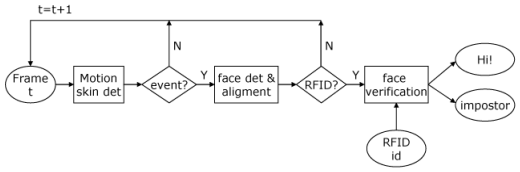


**Fig. 2.** The video processing workflow implemented on the verification unit.

refer in particular to such functionalities. Figure 2 shows the video processing workflow. The pre-processing phase, based on motion detection and skin detection, was reported in [14]; we do not detail it any further.

### 3.1 Data-driven feature selection

In this section we briefly report the feature selection method upon which we base all our image representations. The method was originally proposed in [1] and was adopted in [2] for face detection. In [2] a comparison with other dimensionality reduction methods is reported, in particular with Adaboost feature selection. Here we simply recall that the main advantages of regularized feature selection are, on one side, a well ground theory to rely on, and on the other size, an easy to implement algorithm that leads to satisfactory results with a relatively small number of examples. We consider a binary classification setting, therefore we start from a training set of positive and negative examples described with a *dictionary of features* appropriate for the problem. If this dictionary is descriptive of the content carried by the data we may assume a linear dependence between input and output:

$$Af = g \tag{1}$$

where $A$ is the matrix of processed image data, and each $\{A_{ij}\}$ is obtained from $i = 1 \dots n$ images each of which is represented by $j = 1, \dots, p$ features. In a binary classification setting we may set $g_i \in \{-1, 1\}$; $f$ is the unknown vector that weighs the importance of the features for each image of the training set. In this setting, feature selection means looking for a sparse solution $f = (f_1, \dots, f_p)^\top$: features corresponding to non-zero $f_i$ are relevant to model the diversity of the two classes.

In general one could solve Problem (1) by matrix inversion, but, as we will see later in this section, the systems we consider are largely under-determined and may be severely ill-conditioned because of the features redundancy. One possible way to address these problems is to resort to some regularization strategy. In particular, since we aim at selecting a subset of features, a sparsity-enforcing penalty is ideal for our case. We consider the $L_1$ norm that leads to a feasible problem of the form

$$f_L = \arg\min_f \{|g - Af|_2^2 + 2\tau|f|_1 \tag{2}$$

( $|f|_1$ is the $L_1$-norm), usually referred to as *lasso* problem [15]. We solve it by an iterative method called *thresholded Landweber* that has been shown to converge to the minimizer of (2) under appropriate conditions on the matrix $A$ [1]. The iterative step is ($f_L^0 = \mathbf{0}^\top$):

$$f_L^{t+1} = S_\tau[f_L^t + A^\top(g - Af_L^t)] \quad t = 0, 1, \dots; \tag{3}$$

where $S_\tau$ is a *soft-thresholder*:

$$(S_\tau h)_n = \begin{cases} h_n - \tau sign(h_n) & \text{if} |h_n| \geq \tau \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

As we will see later, in our case, problem (3) involves the manipulation of matrices that may be very big. We therefore consider a *re-sampling procedure* based on solving $S$ smaller sub-problems obtained each time extracting, with replacement, a subset of $m$ features from the original set, $m \ll p$. The $S$ intermediate solutions are combined and used as input of a second regularized feature selection. Details and motivations of this 2-stages feature selection pipeline, as well as a discussion on parameter tuning, can be found in [2].

The solution obtained at the end of the two stages is sparse and consistent, but usually it is not small enough for our processing and storage requirements. We then apply a third selection stage on the features that survived the previous two stages, that further reduces the amount of redundancy in the set of selected features. We first restrict the set starting from a random feature and adding only features that are distant or appear not correlated according to the Spearman's correlation test. The complete feature selection module will then consist of 3 stages. We call the set of features obtained from the 3-stages method $S^{out}$.

Once features have been selected they are used to represent the data and then to address the classification problem of interest. In all the phases of our prototype we adopt a linear SVM as a classification algorithm, as it gives us the best compromise between performance and computational speed. In the next section we discuss classification issues specific of real-time object detection.

### 3.2 Coarse-to-fine classification

In general object detection must face the problem of a strong unbalance between positives and negatives in a given image. It is well known how, to perform real-time object detection, one has to consider a number of methods and heuristics to allow for a fast elimination of the millions of negative data. This issue has been debated in [7] where a coarse-to-fine procedure — a *cascade of classifiers* — is adopted. Here we take inspiration from that approach to build a cascade of SVM classifiers. Each classifier of the cascade is built extracting from $S^{out}$ small subsets of *at least* 3 distant features that are able to reach a fixed target performance on a validation set. We start by 3 mutually distant features and add further features until a target performance is obtained, testing a validation set with a linear SVM.

Target performance is chosen so to build weak SVM classifier that will not be likely to miss positive occurrences: we set the minimum hit rate to 99.5% and the maximun false positive rate to 50%. These modest targets allow us to achieve good performances with the global classifier, since global performance of the cascade will be computed according to the following [7]: $H = \prod_{i=1}^{K} h_i$ and $F = \prod_{i=1}^{K} f_i$ where $h_i$ is the hit rate and $f_i$ is the false positive rate for each weak classifier $i$. In our case, assuming a cascading of 10 weak classifiers, we will get $H = 0.995^{10} \sim 0.9$ and $F = 0.5^{10} \sim 3 \times 10^{-5}$.

### 3.3 Face detection and registration

Face detection refers to the problem of localizing human faces in digital images. We are mainly interested in detecting frontal faces, and to this purpose, we start from a dataset of $19 \times 19$ frontal face and non-face images acquired by our system. The dataset we consider to build the face detection module contains 4000 training examples, 2000 validation, and 3500 test examples, evenly distributed between positives and negatives [2]. Alternatively one may use a benchmark dataset, such as the CBCL-MIT — see [2] for results with this dataset.

For face detection we consider a dictionary of *rectangle features* [7], proved effective to describe the intrinsic structure of faces and their symmetries. We then proceed as follows:

- The linear system $Af = g$ is built from the available training set. Each row of matrix $A$ corresponds to a training image, each column to a specific rectangle feature of a given type, size, aspect ratio, position. As for vector $g$, we associate $+1$ to each positive (face) image, $-1$ to negative (non face) images.
- Considering $19 \times 19$ images we end up with about 64000 rectangle features; thus the system we consider is hugely under-determined and, because of images auto-correlation properties very much ill-conditioned. This justifies the use of the regularized method described earlier in the section. Also, because of the size of $A$ the re-sampling strategy discusses is recommended.

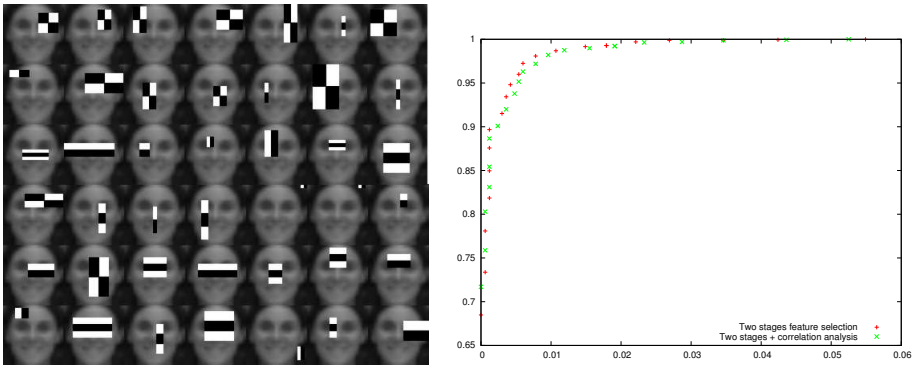


**Fig. 3.** Left: the 42 rectangle face features selected by the 3-stages feature selection method. Right: generalization performance of the features selected with a 2-stages and a 3-stages method.

Parameter tuning $\tau$ is performed on the validation set, with a cross-validation on the second selection stage [2]. The full 3-stages system leaves us with a compact set of 42 features to be used as *face descriptions*, shown in Fig. 3 (left). To evaluate the appropriateness of the selected features we test their generalization ability w.r.t. the test set. A classifier is trained, tuned, and tested over a dataset represented by the selected features. The ROC curves obtained from this analysis are shown in Figure 3 (right): here we report a comparison between the results obtained with the features selected by the first 2 stages and the compact set of features obtained adding the 3rd stage. A negligible performance loss is countered by an increase in description compactness.

In order to keep up with real time processing the face detection module implements the coarse-to-fine procedure described above.

We conclude this section on face detection by noticing that since our system does not put any constraints on the individual posture, some images will be inappropriate for recognition. Having at disposal a whole video for each session we discard poor quality images by means of a *eye detector*. Since eyes are the simplest feature to detect in a human face we discard faces where eyes are not clearly visible. The eye detection module replicates *exactly* the face detection module. Only the dataset has changed (extracted in this case from the FERET dataset), according to a data-driven philosophy. The eye-detection module is also used to automatically register the detected faces. Registration is very rough and simply consists of cropping an appropriate area of the face relative to the detected eyes positions (see Fig. 4, left). Faces below $40 \times 40$ pixels are discarded, and the remaining ones are scaled down to $40 \times 40$. (Fig. 4, right) shows samples extracted from various videos, and illustrates how the registration quality is quite reasonable.

**Fig. 4.** Face registration: on the left, the areas considered to form the new registered face. On the right, examples of automatically registered faces.

### 3.4 Face verification

Let us assume that for each individual enrolled in the system we dispose of a dataset of faces gathered and registered automatically by the face detection module. The dataset is divided, as usual, in training, validation and test. The feature dictionary we consider is based on the so-called *local binary patterns (LBP)* [11]. We now describe how we represent each individual $\mathcal{I}$:

- We start off from a set of $40 \times 40$ positive images $I_p$, $p = 1, \ldots, P$ of individual $\mathcal{I}$ and a set of negative images $I_n$ $n = 1, \ldots, N$ randomly sampled from other individuals.
- For each image $I_i$, each pixel neighbourhood is represented as a LBP. The neighbourhood size we consider is obtain choosing 8 sampling points on a circle of radius 2 around the pixel. Then, for each image, we compute $L$ LBP histograms on rectangular regions of at least $3 \times 3$ pixels, on all locations and aspect ratios; thus the description of image $I_i$ is a list of histograms $H_i^1, \ldots, H_i^L$.
- Notice that, here again, a feature selection procedure is important, since we obtained $L = 23409$ LBP histograms. We resort to the data-driven feature selection. Instead of exploiting directly positive and negative examples the linear system is built so to express the intra-personal and extra-personal variation of each histogram. With a similar approach to [12] we compute each row of the matrix $A$ as follows: for each pair of images $I_A$ and $I_B$ we compare corresponding LBP histograms (after normalization) using the $\chi^2$ distance. That is, for each pair of input images we obtain a feature vector $x_i$ whose elements are $\chi^2$ distances: $x_i = (\chi^2(I_A^1, I_B^1), \ldots, \chi^2(I_A^L, I_B^L))$. We associate a label $g_i \in \{+1, -1\}$, where $g_i = 1$ if both $I_A$ and $I_B$ are positive images, $g_i = -1$ if either $I_A$ or $I_B$ is negative.

For a given set of examples, the number of feature vectors we compute is very high — for a set of positive images of size $P$ we have $\binom{P}{2}$ positive feature vectors and many more negative feature vectors. In the current implementation, in order to obtain balanced systems of reasonable size we simply randomly sample at most 2000 positive and 2000 negative feature vectors and build matrix $A$. The vector $g$ is composed of the corresponding labels $g_i$ and the vector $f$ of unknowns will weight the importance of each LBP histogram for intra-person and extra-person

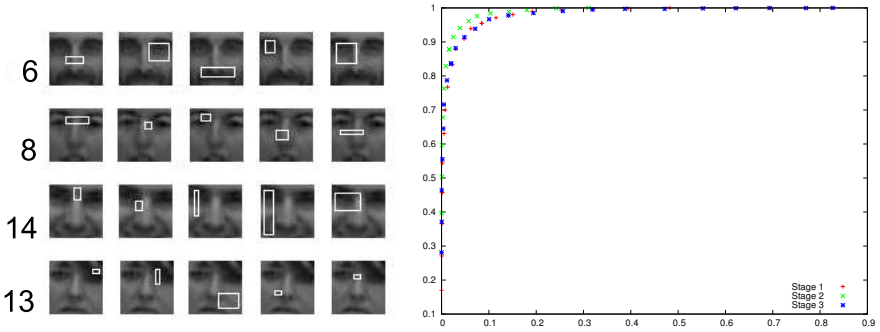


**Fig. 5.** Left: Top 5 features for some individuals. Right: Generalization performance of the features obtained from the three stages, for an individual arbitrarily chosen.

comparison. Once we have built a system for a given individual described, we select features following the 3-stages protocol.

Notice that for each individual we obtain a different set of distinctive features, the ones that best represent his/her peculiarity. Figure 5 (left) shows the top 5 features extracted for some individuals (n. 6, 8, 13, 14) of our training set (dataset $VAL01$, see Sect. 4) . Although the detected features do not necessarily reflect elements that a human observer would find meaningful, they are localized in the most distinctive areas of the face for each individual. For person 6, for instance, the beard is spotted. The most distinctive area of person 8 seems to be the top part, the eyes region. Person 14 is mainly characterized by vertical features. The top 2 features of person 13 are localized where she wears a fringe.

In conclusion, we briefly comment on the appropriateness of the 3-stages feature selection strategy for the problem of face authentication. We report similar experiments to the ones shown for the face detection case for a randomly chosen individual $\mathcal{I}_A$ (from dataset $VAL01$, Sect. 4). After feature selection we train and tune a classifier that should discriminate between images of $\mathcal{I}_A$ and images of impostors. Note that, unlike the experiments reported in Sect. 4, here we consider images instead than videos. Figure 5 (right) shows a comparison of the results obtained for the 3 stages. Similarly to face detection we notice that the three results are comparable, with a slight performance loss when adding the third stage, which is although responsible of a consistent decrease in the number of features and therefore allows for a very compact representation of each person (the models computed from dataset $VAL01$ range from 10 to 30 LBP features per individual).

### 3.5 Video-based face verification

At the end of the enrollment procedure of a given individual $\mathcal{I}$, once $s$ features have been selected as a "good" description of $\mathcal{I}$, we choose $g$ training images of $\mathcal{I}$, $\{G_1^{\mathcal{I}}, \ldots, G_g^{\mathcal{I}}\}$ as a gallery. This will lead us, in the verification stage, to evaluate

$g$ test data per each new probe $P_t$: $< G_i^{\mathcal{I}}, P_t >$ $\quad i = 1, \ldots, g, \quad t = 1, \ldots, T_{curr}$. We classify each of these test data, with a linear SVM classifier associated to identity $\mathcal{I}$, to evaluate how likely is $P_t$ to depict individual $\mathcal{I}$. We then associate a final confidence value to $P_t$, in the range $[0, 1]$, related to how many images of the gallery are similar to $P_t$.

When processing a video each frame is evaluated according to the above procedure. Different strategies may be adopted to associate to the video a unique positive or a negative feedback. So far we adopted two families of methods, compared in the experiments section: the first one, which we call *multiple peak*, returns a positive feedback (i.e., a positive verification) as soon as $N$ frames not necessarily adjacent of the video are associated a confidence value greater than a threshold $T1$. The second one, based on computing a *running average* of confidence values, returns a positive feedback if the average of the latest $M$ confidence values is greater than a threshold $T2$. Sect. 4 will report comparative experiments, that will highlight how the strategy choice does not seem to be crucial. The system is currently implementing a feedback based on a multiple peaks, with N=3.

## 4 Experiments on face verification

### 4.1 The datasets

We start describing in some details how we gathered a reference dataset $VAL01$ that we use for training, tuning, and testing the system.

The dataset $VAL01$ was built from several weeks of acquisition: at the end of two weeks we manually labeled the stored videos (a video is stored if faces are detected in it) and build models for all the individuals that have a rich dataset ($\geq 20$ frames). From the third week on data were gathered and labeled for testing: individuals that did not previously appear were stored to be used as negative examples (impostors) for all models. In total, 15 individuals were included in the training phase, while a total of 64 individuals were gathered for testing. $VAL01$ contains a set of training and tuning video frames per each of the 15 individuals, in the range of $[30-200]$ depending on the individual. It then contains about 1700 test frames of the 64 test individuals.

For what concerns real time experiments, the same 15 users received a RFID card each, they can use at their discretion. We do not have prior knowledge on the amount of impostors and not even on how collaborative the users are. In order to analyse the results obtained data (video, system output) are stored for 60 seconds after the RFID reader is activated. One video is captured for each person walking in the scene within that time frame. Data storage followed by manual labeling is important for a quantitative evaluation. A simple interface allows us to manually label the previously verified identities and thus to estimate the percentage of true positives (correctly verified users) and true negatives (correctly identified impostors). As a side effect we are also gathering a wider dataset to be available for future experiments. This dataset $VAL02$, currently

**Fig. 6.** Sample frames from videos in $VAL02$. Being the system entirely unconstrained the data include non collaborative people and clutter due to the presence of multiple people.

under construction, may be naturally used as a test set extension to $VAL01$. The experiments reported here refer to about 20 days recording. We labeled the data in two different ways: *(i)* all walks evaluated by the system are labeled in genuine users or impostors depending on the person tracked, *(ii)* only clean walks, containing 1 person only, are labeled. In the first case we gather 192 genuine users and 37 impostors; in the second case 144 genuine users and 22 impostors.

The data stored in $VAL02$ are quite challenging for various reasons. First, there is a 6 months temporal gap between the acquisition of $VAL01$ that we use to model the individuals, and the run time experiments that generate $VAL02$. This translates in various scene changes (in particular in a strong illumination change from winter to summer sun exposure) and the obvious appearance variation of the individuals enrolled in the system. Fig. 7 shows some illumination and appearance changes produced by time. Second, we do not put any restriction in people's behaviour, and we do not forbid multiple passage (that in a real access control system would inhibit entrance). Fig.6 depicts some common behaviours, including people walking in groups, people not looking at the camera, occlusions caused by people crossing the individual to verify.

### 4.2 Results

A first (offline) evaluation on $VAL01$ test set allows us to discuss the appropriateness of the face representation based on automatic LBP selection with respect to (1) a traditional global description based on PCA (2) the manually selected LBP features adopted in [12]. Table 1 reports the results obtained with the three methods in terms of acceptance rate and rejection rate. In these experiments we have assumed an equal probability of having a genuine user or an impostor, therefore for each individual we extract randomly from other individuals datasets as many negative test data as the number of available positives. The

**Fig. 7.** The same person in different day times and different days.

**Table 1.** Performance of different representation methods on the dataset VAL01.

| **Method** | **Acceptance Rate (AR)** | **Rejection Rate (RR)** |
|---|---|---|
| PCA (global) [16] | 0.68 | 0.70 |
| manual LBP select. [12] | 0.87 | 0.94 |
| data-driven LBP select. [our method] | 0.93 | 0.97 |

table shows how a local approach outperforms a global one. In particular PCA is not appropriate in our case, since images are only approximately registered. Manual selection of the most discriminative LBP features leads to good results, although inferior to our automatic feature selection, confirming that LBP are a good way to capture face peculiarities. Besides the very good results achieved, an important effect of our automatic feature selection procedure, is that an *ad hoc* description for each enrolled individual is achieved, allowing to capture different appearance features.

As for run time experiments (forming current $VAL02$) Fig. 8 shows very promising verification performances, while comparing various voting strategies to associate a single feedback to a video sequence. There is no real gain in adopting one strategy or another. In the multiple peaks case $N = 4$ produces slightly better results if one wants to keep false acceptance rates low. In the case of running average a temporal window of size M=3 is a good choice. Fig. 9, instead, shows that there is not a great advantage in keeping clean walks only. In absolute values the multiple peaks strategy, N=4, leads to 4 false acceptance (FA) and 40 false rejection (FR) over the full dataset and 2 (FA) and 27 (FR) over the clean dataset. Similarly, adopting running average over temporal windows with $M = 3$, we obtain 4 FA and 43 FR on the full dataset, and 2 FA and 25 FR on the clean one. The ROC of the multiple peaks case with $N = 3$ shows current on-line performance of the prototype.

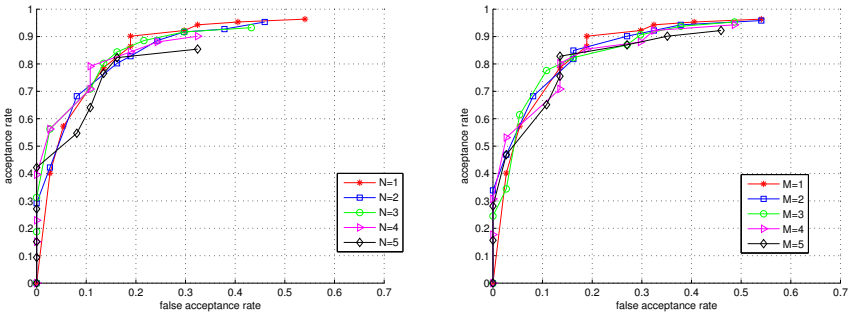


**Fig. 8.** Strategies to associate a single feedback to a video. Left: multiple peak (acceptance if $N$ confidence values are above $T1$ — see text). Right: running average (acceptance if the average of $M$ consecutive values is above $T2$ — see text). ROC curves are build varying $T1, T2$ between 0.1 and 1.

## 5 Discussion and future work

This paper describes a prototype face verification system based on videos. The system allows a user to declare his identity via a RFID card and verifies the identity in real-time by means of fast video-processing capabilities. The verification algorithms implement various recent findings from the statistical learning theory, in particular in obtaining a data-driven image representation. The very same feature selection + classification pipeline is implemented in various modules of the system, in particular to detect faces, to register them via eye-detection, and to verify identities. The video-processing workflow has been evaluated off-line, on a previously gathered dataset on videos, and is currently being evaluated on-line on the prototype system performing real-time. We report very promising quantitative results obtained by a manual labeling of videos acquired on a 20 days time frame. The system is currently installed in a very much unconstrained environment, where illumination may change drastically due to changes of natural light, and scene clutter may be significant, since we do not require that the scene is empty during verification. We highlight the fact that no real system would work under such challenging circumstances. We are currently implementing an update procedure that will allow us to keep up with appearance variations in time. Also we are evaluating different ways to exploit data redundancy derived from the temporal component, in particular adopting Extended Volume LBP [12].

## References

1. Daubechies, I., Defrise, M., Mol, C.D.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Comm. on Pure Appl. Math. **57** (2004)

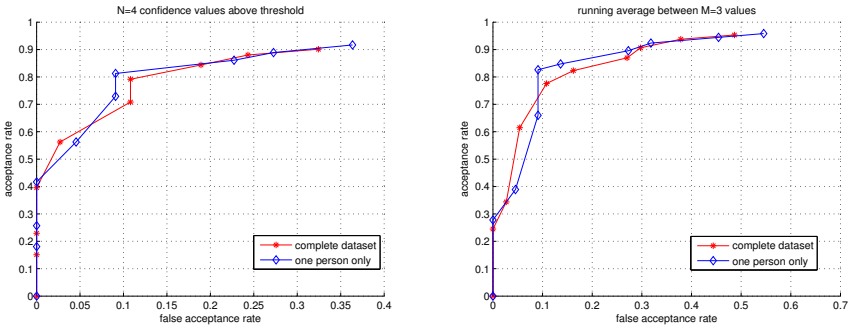


**Fig. 9.** Comparison between two different data labeling (the complete test set $VAL02$ and a set including only videos with one person at a time). Left: multiple peak, $N = 4$. Right: running average, $M = 3$.


2. Destrero, A., Mol, C.D., Odone, F., Verri, A.: A sparsity-enforcing method for learning face features. IEEE Trans. on Image processing in press.
3. Vapnik, V.N.: Statistical Learning Theory. Wiley (1998)
4. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: a survey. IEEE Trans. on Pattern Analysis and Machine Intelligenge **24**(1) (2002) 34–58
5. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Trans. on PAMI **23**(4) (2001) 349–361
6. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermediate complexity and their use in classification. Nature Neuroscience **5**(7) (2002)
7. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal on Computer Vision **57**(2) (2004) 137–154
8. Zhang, D., Li, S.Z., Gatica-Perez, D.: Real-time face detection using boosting in hierarchical feature spaces. In: Proc. of ICPR. (2004)
9. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.: Face recognition: A literature survey. ACM Computing Surveys (2003) 399–458
10. A.Pentland, Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: IEEE Int. Conf. on CVPR. (1994)
11. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. on PAMI **28**(12) (2006) 2037–2041
12. Hadid, A., Pietikäinen, M., Li, S.Z.: Learning personal specific facial dynamics for face recognition from videos. In: AMFG07. (2007) 1–15
13. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: AMFG. (2007) 168–182
14. Destrero, A., Odone, F., Verri, A.: A system for face detection and tracking in unconstrained environments. In: AVSS. (2007)
15. Tibshirani, R.: Regression shrinkage and selection via the lasso. J Royal. Statist. Soc. B **58**(1) (1996) 267–288
16. Liu, X., Chen, T., Kumar, B.V.: On modeling variations for face authentication. Pattern Recognition **36**(2) (2003) 313–328