Measuring the Performance of Face Localization Systems

Yann Rodriguez, Fabien Cardinaux, Samy Bengio, Johnny Mariéthoz

IDIAP Research Institute, Rue du Simplon 4, CP 592, CH-1920 Martigny, Switzerland.

Abstract

The purpose of Face localization is to determine the coordinates of a face in a given image. It is a fundamental research area in computer vision because it serves, as a necessary first step in any face processing system, such as automatic face recognition, face tracking or expression analysis. Most of these techniques assume, in general, that the face region has been perfectly localized. Therefore, their performances depend widely on the accuracy of the face localization process. The purpose of this paper is to mainly show that the error made during the localization process may have different impacts on the final application. We first show the influence of localization errors on the face verification task and then empirically demonstrate the problems of current localization performance measures when applied to this task. In order to properly evaluate the performance of a face localization algorithm, we then propose to *embed* the final application (here face verification) into the performance measuring process. Using two benchmark databases, BANCA and XM2VTS, we proceed by showing empirically that our proposed method to evaluate localization algorithms better matches the final verification performance.

Key words: Face Detection and Localization, Face Verification, BANCA and XM2VTS Databases

1 Introduction

Face localization (FL) is the process of finding the exact position of a face in a given image [1,2]. It is generally used as an important step in several applications such as face tracking [3–5] or person authentication [6,7]. Unfortunately, it is difficult to measure the performance of a face localization algorithm, as no universal criterion has been acknowledged in the literature for this purpose. In fact, we argue in this paper that such a criterion does not exist and we propose instead the use of a criterion that would be specifically tailored for each application for which the localization algorithm would be designed.

In order to illustrate this argument, this paper concentrates on a face verification (FV) task [8,9]. In that context, the best localization algorithm should be the one that minimizes the number of errors made by a specific verification algorithm.

The paper thus starts by analyzing how various kinds of localization errors affect the performance of two different face verification algorithms, on two different benchmark databases. This empirical analysis, presented in Section 4, clearly demonstrates that not all localization errors induce the same verification error, even if the current localization performance measures, such as those presented in Section 2, would have rated them similarly.

In the second part of this paper, we go one step further: knowing that verification in itself is not error-free, we propose a new localization measure adapted to the task of verification. This measure estimates directly the verification errors as a function of the errors made by the localization algorithm. In this paper, we estimate this measure using a simple K nearest neighbor (KNN) algorithm. We then show empirically that the localization measure estimated by this simple procedure better reflects the performance of a face localization algorithm when used for a face verification task.

The paper is organized as follows. Section 2 presents an overview of classical measures currently used in the literature in order to evaluate the performance of a face localization algorithm. Section 3 then presents the empirical framework (databases, face verification and face localization systems) used in this paper to analyze face localization algorithms and evaluate our proposed method. Section 4 presents two different empirical analyses that both show that the performance of a localization algorithm can only make sense in the context of the application for which the localization algorithm was built for. This is then followed by Section 5, which presents the idea consisting in estimating the error made by the verification process given the error made by the localization process. Section 6 evaluates empirically how this new performance measure behaves on a real benchmark database, and finally Section 7 concludes the paper.

Note that this paper builds on the initial ideas presented in [10], which are extended in several respects, including a thorough empirical analysis of the relation between localization and verification errors for face verification systems.

2 Performance Measures for Face Localization

2.1 Lack of Uniformity

Direct comparison of face localization systems is a very difficult task, mainly because there is no clear definition of what a good face localization is. While most concerned papers found in the literature provide localization and error rates, almost none mention the way they count a correct/incorrect hit that leads to computation of these rates. Furthermore, when reported, the underlying criterion is usually not clearly described. For instance, in [11] and [12], a detected window is counted as a true or false detection based on the visual observation that the box includes both eyes, the nose and the mouth. According to Yang's survey [13], Rowley *et al.* [14] *adjust the criterion until the experimental results match their intuition of what a correct detection is (i.e. the square window should contain the eyes and also the mouth).* In some rare works, the face localization criterion is more precisely presented. In [15] for instance, Lienhart *et al.* count a correct hit if the Euclidean distance between the centers of the detected and the true face is less than 30% of the width of the true face, and the width of the detected face is within ±50% of the true face. In [16], the authors consider a true detection if the measured face position (through the position of the eyes) and size (through the distance between the eyes) do not differ more than 30% from the true values. Unfortunately, the lack of uniformity between reported results makes them particularly difficult to compare and reproduce.

2.2 A Relative Error Measure

Recently, Jesorsky *et al.* [17] introduced a relative error measure based on the distance between the detected and the expected (ground-truth) eye center positions. Let C_l (respectively C_r) be the true left (resp. right) eye coordinate position and let $\tilde{C}_l$ (resp. $\tilde{C}_r$) be the left (resp. right) eye position estimated by the localization algorithm. This measure can be written as

$$d_{eye} = \frac{\max(d(C_l, \tilde{C}_l), d(C_r, \tilde{C}_r))}{d(C_l, C_r)} \tag{1}$$

where $d(a, b)$ is the Euclidean distance between positions a and b. A successful localization is accounted if $d_{eye} < 0.25$ (which corresponds approximately to half the width of an eye).

This is, to the best of our knowledge, the first attempt to provide a unified face localization measure. We can only encourage the scientific community to use it and mention it when reporting detection/error rates when the task is localization only. Researchers seem to only start to be aware of this problem of uniformity in the reporting of localization errors and now sometimes report cumulative histograms of d_{eye} [18,19] (detection rate vs. d_{eye}), but this still concerns only a minority of papers. Furthermore, a drawback of this measure is that it is not possible to differentiate errors in translation, rotation and scale.

2.3 A More Parametric Measure

More recently, Popovici *et al.* [20] proposed a new parametric scoring function whose parameters can be tuned to more precisely penalize each type of errors. Since face localization is often only a first step of a more complex face processing system (such as a face recognition module), analyzing individually each type of errors may provide useful hints to improve the performance of the upper level system.

In the same spirit as in [20], let us now define four basic measures to represent the difference in horizontal translation (Δ_x), vertical translation (Δ_y), scale (Δ_s) and rotation (Δ_α):

$$\Delta_x = \frac{\overline{dx}}{d(C_l, C_r)} \ , \tag{2}$$

$$\Delta_y = \frac{\overline{dy}}{d(C_l, C_r)} \ , \tag{3}$$

$$\Delta_s = \frac{d(\tilde{C}_l, \tilde{C}_r)}{d(C_l, C_r)} \ , \tag{4}$$

$$\Delta_\alpha = \widehat{\overrightarrow{C_l C_r}, \overrightarrow{\tilde{C}_l \tilde{C}_r}} \ , \tag{5}$$

where $\overline{dx}$ is the algebraic measure of vector $\overrightarrow{dx}$. All these measures are sum-

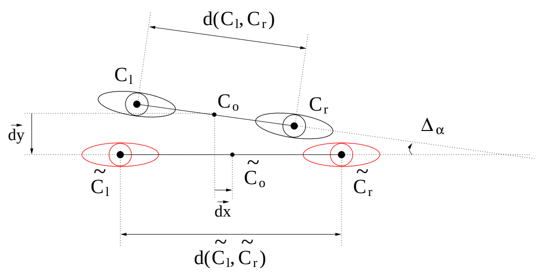


Fig. 1. Summary of some basic measurements made in face localization. C_l and C_r (resp. $\tilde{C}_l$ and $\tilde{C}_r$) represent the true (resp. the detected) eye positions. C_0 (resp. $\tilde{C}_0$) is the middle of the segment $[C_l C_r]$ (resp. $[\tilde{C}_l \tilde{C}_r]$).

marized in Figure 1. The four delta measures are easily computed given the ground-truth eye positions (C_l and C_r) and the detected ones ($\tilde{C}_l$ and $\tilde{C}_r$). Furthermore, as it will appear useful later in the paper, one can artificially create detected positions given these four delta measures. Note finally that both the choices of Jesorsky's threshold (0.25) and Popovici's weights on each of these delta measures (in order to obtain a single measure) still remain subjective.

2.4 Application-Dependent Measure

In this paper, we argue that a universal objective measure for evaluating face localization algorithms *does not exist*. A given localized face may be correct for the task of initializing a face tracking system [3], but may not be accurate enough for a face verification system [6]. We therefore think that there can be no absolute definition of what a *good face localization* is. We rather suggest to look for an application-dependent measure representing the final task. Moreover, in the context of face verification, there has been several empirical evidence [6] showing that the verification score obtained with a perfect (manual) localization is significantly better than the verification score obtained with a not-so-perfect (automatic) localization, which shows the importance of measuring accurately the quality of a face localization algorithm for verification.

Hence, in the remainder of the paper, we will empirically show, using some real datasets, how face localization errors affect face verification results, and how it can be more accurately measured than using currently proposed measures.

3 Baseline System

In this Section, we describe the environment used to perform all the experiments of this paper. We first describe the databases, then the localization system, and finally the verification systems.

3.1 The Face Databases

In all the experiments described in the paper, we used two different databases. The XM2VTS database is used mainly for preliminary analysis and training purposes while the BANCA database is used to evaluate the quality of face localization performance measures (see Figure 2 for example images of each database). The XM2VTS database contains synchronized video and speech

(a) XM2VTS (controlled conditions): uniform background and lighting

(b) BANCA English (uncontrolled conditions): complex background and lighting variability

Fig. 2. Comparison of XM2VTS (a) and BANCA (b) face image conditions.

data from 295 subjects, recorded during four sessions taken at one month intervals. The subjects were divided into a set of 200 training clients, 25 evaluation impostors and 70 test impostors. We performed the experiments following the *Lausanne Protocol Configuration I* described in [21].

The BANCA database [22] was designed to test multi-modal identity verification with various acquisition devices under several scenarios (controlled, degraded and adverse). In the experiments described here we used the face images from the French and English corpora, each containing 52 subjects. Each subject participated in 12 recording sessions in different conditions and with different cameras. Each of these sessions contains two video recordings: one true client access and one impostor attack. Five "frontal" face images were extracted from each video recording. Following the *BANCA Experimental Protocol* [22], these five images should be considered as a single access; however, in order to estimate and test our proposed measure (see Section 5), we used each image as an independent access. Out of the 7 protocols, we decided to use protocol P, which appears to be the most realistic one.

3.2 The Face Localization System

In this paper, we used the real-time frontal face detector presented by Fröba and Ernst [23] for face localization. We used a *Modified* version of the *Census Transform* (MCT) to compute local 3x3 kernel features which capture the local spatial image structure. At each pixel location in an image, the feature is defined as an ordered set of pixel intensity comparisons. Due to their local structure, MCT features are invariant to gray scale transformation which makes them robust against illumination changes. The classification is performed by a cascade classifier framework, inspired by the work of Viola and Jones [24]. The classifier structure is however much simpler than previous approaches, consisting of only four stages (instead of more than 20 in the original approach). As in [24], we used the *AdaBoost* [25] algorithm for both feature selection and training. An on-line demo program of our face localization system can be found on the internet *http://www.idiap.ch/∼marcel/en/detector.php.*

Like many popular recent systems, this detector is an *image-based* approach, using the principle of a *scanning window*. A test image is exhaustively scanned at multiple positions and scales; each window is then classified as either containing a face or not. The main scanning parameters are the horizontal and vertical steps between two consecutive scanning windows and the scale factor (see Figure 3). Localization precision is closely related to these parameters, as is the computational cost (number of windows to scan).

3.3 The Face Verification Systems

A face verification system (FV) usually consists in image normalization and feature extraction followed by classification [26–28]. In this study we used two kinds of FV, namely *DCT/GMM* and *PCA/Gaussian* systems, which we briefly describe here.

In both systems, a 80×64 (rows $\times$ columns) face window is first cropped out, based on the result of the face localization process. Each face window should contain the face area from the eyebrows to the chin. Moreover, the location of the eyes should be the same on each face window (via geometric normalization). Histogram equalization is then used afterward in order to normalize

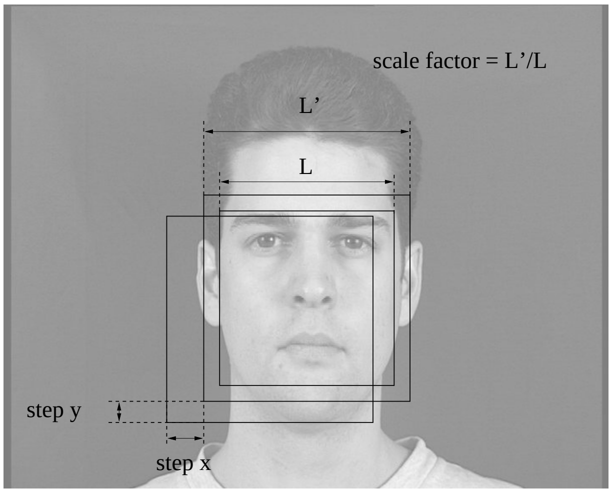
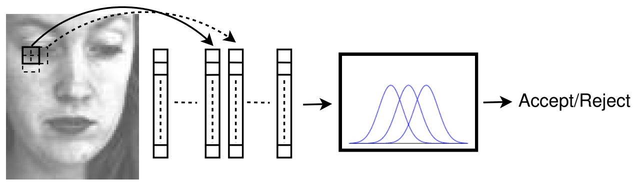
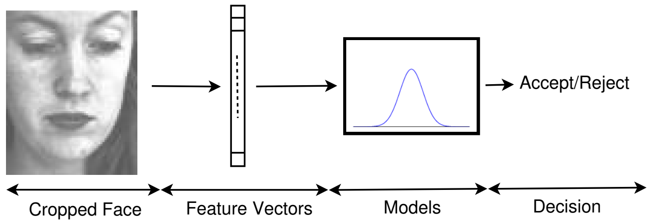


Fig. 3. Face localization scanning parameters: step x, step y and scale factor. The choice of these parameters both affects the speed of the system as well as accuracy.



(a) DCT/GMM



(b) PCA/Gaussian

Fig. 4. Conceptual representations of the two face verification systems

the face images photometrically.

Using the DCT/GMM system [6,29], we then extract a set of *DCTmod2* feature vectors $\mathbf{X}$ from each face image [30]. The DCT/GMM system was implemented using a Gaussian Mixture Model (GMM) technique similar to those used in text-independent speaker verification systems [31]. A generic GMM is trained with the features computed on several faces (non-client specific), in order to maximize $p(\mathbf{X}|\Omega)$, the likelihood of a face $\mathbf{X}$ given the generic GMM parameters Ω, for all $\mathbf{X}$ of the training database. This GMM is then adapted for each client i in order to produce a new GMM model of $p(\mathbf{X}|C_i)$, the likelihood of a face $\mathbf{X}$ given the parameters of a client C_i. The ratio between these likelihoods represents the score of the verification model, which is then compared to a threshold θ in order to take a final decision. A conceptual example of the DCT/GMM system is represented in Figure 4(a).

In comparison, the PCA/Gaussian model is based on Principal Component Analysis (PCA) feature extraction [32]. The classifier used for the PCA system is somewhat similar to the DCT/GMM system; the main difference is that only two Gaussians are used: one for the client and one to represent the generic model [1]. Due to the small size of the client specific training dataset, and since PCA feature extraction results in one feature vector per face, each client model inherits the covariance matrix from the generic model and the mean of each client model is the mean of the training vectors for that client. A similar

[1] The number of Gaussians of the DCT/GMM model is in general much higher and is normally tuned on some validation set.

system has been used in [33,34]. A conceptual example of the PCA/Gaussian system is represented in Figure 4(b).

The FV performance is generally measured in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR), defined as:

$$\mathrm{FAR}(\theta) = \frac{\text{number of FAs}}{\text{number of impostor accesses}}\,, \tag{6}$$

$$\mathrm{FRR}(\theta) = \frac{\text{number of FRs}}{\text{number of true claimant accesses}}\,, \tag{7}$$

where θ is the chosen decision threshold. In order to help the interpretation of performance, the two error measures are often combined using the Half Total Error Rate (HTER), defined as:

$$\mathrm{HTER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2}\,.$$

Furthermore, since in real life applications the decision threshold θ has to be chosen *a priori*, we selected it in order to obtain Equal Error Rate (EER) performance, where $\mathrm{FAR}(\theta)$=$\mathrm{FRR}(\theta)$ on the validation set [2] . The same threshold is then used on the test set to obtain the final HTER.

4 Robustness of Current Measures

The purpose of this Section is to analyze the relation between the tasks of face localization and face verification, by observing how errors reported by the FL system affect the FV system. We start by observing, in Section 4.1, the performance of a of FV system when we artificially introduce some localization errors in the tested face images. Then, in Section 4.2, we empirically demonstrate for a particular case that a generic face localization measure is not accurate. These preliminary experiments are performed on the XM2VTS database using the associated protocol. The experiments were carried out with the two different FV approaches briefly described in Section 3.3, namely DCT/GMM and PCA/Gaussian. The models are trained with manually located images and the decision threshold is chosen *a priori* at EER on the validation set (also using manually located images). The FV systems are thus independent of the FL system used. The FAR, FRR and HTER performance measures are then computed with perturbed face images from the test set.

[2] Since the terminology is not consistent in the evaluation protocols associated with the XM2VTS [21] and BANCA [22] databases, we chose in this paper to name "*validation set*" the image set used to tune the system hyper-parameters (including the decision threshold) and "*test set*" the set of images used to evaluate the performance.

4.1 Effect of FL Errors

In Section 2.2, four types of localization errors were defined: horizontal and vertical translations (respectively Δ_x and Δ_y), scale (Δ_s) and rotation (Δ_α). As a preliminary analysis, we studied how each type of localization error affects the FV performance. Specifically, the eye positions were artificially perturbed in order to generate a configurable amount of translation (horizontal and vertical), scale and rotation errors. Then experiments were performed for each type of errors independently; i.e. when we generated one type of perturbation, the others were kept null.

Figure 5 shows the FV performance as a function of the generated perturbations for the two FV systems. Several conclusions can be drawn from these curves:

(1) Regarding HTER curves, as expected, the FV performance is affected by localization errors. The minimum of the HTER curves are always obtained at the ground-truth positions.
(2) In the tested range, FRR is more sensitive to localization errors, the FAR is not significantly affected. In other words, localization errors in a reasonable range do not induce additional false acceptances. This was expected since, after all, a non face rarely becomes a face by simple geometric transpositions.
(3) HTER curves demonstrate that the two FL approaches are not affected in the same way. Generally, the DCT/GMM system is more robust to perturbed images than the PCA/Gaussian system; justification of this result is discussed further in [29]. Moreover, we remark that the two systems are not sensitive to the same type of errors; while DCT/GMM is affected by scale and rotation errors and very robust to translation errors, the PCA/Gaussian system is very sensitive to all types of errors, including translation.

4.2 Indetermination of d_{eye}

In Section 2, we discussed the important problem of a universal measure to evaluate face localization performance, in order to get fair and clean system comparisons. We also introduced the currently unique existing measure, proposed by Jesorsky *et al.* [17], based on the true and the detected eye positions (1). We also underlined that this measure does not differentiate errors in translation, scale or rotation.

For the specific task of FV, prior empirical evidence showed that the performance is closely related to the accuracy of the face localization system. In

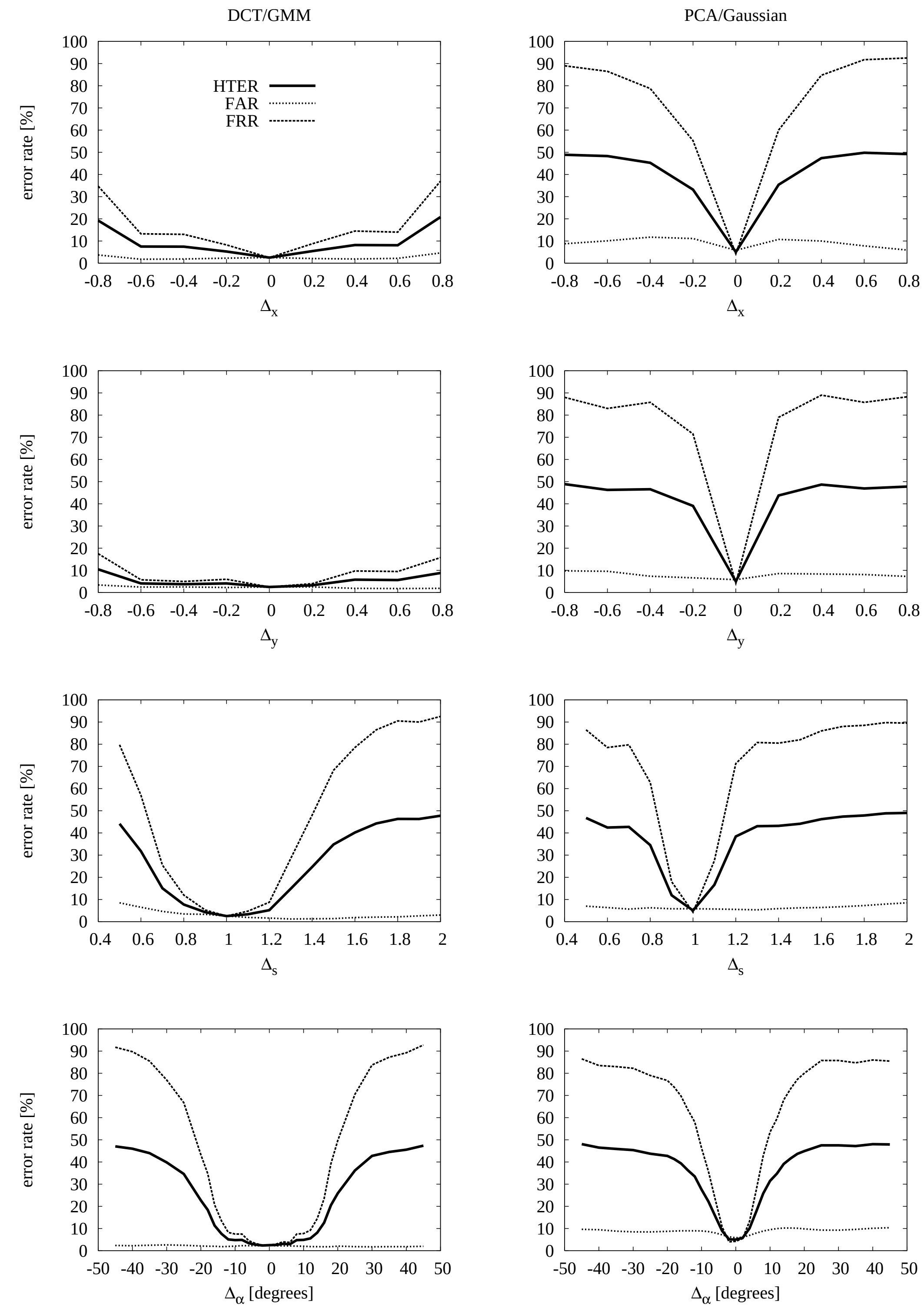


Fig. 5. Face verification performance (in terms of FAR, FRR and HTER error rates) as a function of face localization errors. The error rates are shown for the DCT/GMM (left column) and for the PCA/Gaussian (right column) face verification systems.

Section 4.1, we went a little bit further by explaining that this performance is closely related to the type of error introduced by the FL system and that this dependency varies from one FV system to another (eg. DCT/GMM vs PCA/Gaussian). We then argued that a universal criterion like d_{eye} is not adapted to the final task of FV and that we thus need to search for an application-dependent measure.

To illustrate this opinion more clearly, let us look again at the d_{eye} measure and show why it is not adapted to the FV task. In order to understand the limitations of this measure, we analyze here each type of localization error independently, as done in Section 4.1.

Table 1
For the specific case of $d_{eye} = 0.2$, the first column contains the corresponding Δ values and the third column contains the resulting HTER

delta error	d_{eye}	HTER
$\Delta_x = -0.2$	0.2	5.27
$\Delta_x = 0.2$	0.2	5.43
$\Delta_y = -0.2$	0.2	4.14
$\Delta_y = 0.2$	0.2	3.27
$\Delta_s = 0.6$	0.2	31.75
$\Delta_s = 1.4$	0.2	24.65
$\Delta_\alpha = 23°$	0.2	32.35
$\Delta_\alpha = -23°$	0.2	31.24

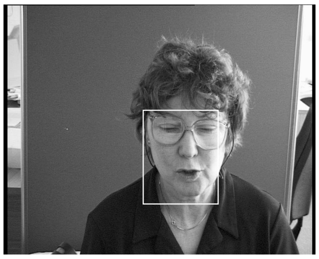
We first arbitrarily selected a value of $d_{eye} = 0.2$, which commonly means that the detected pattern is a face (since it is lower than 0.25). We then selected all kinds of delta errors which would yield $d_{eye} = 0.2$. Details of how to obtain these corresponding delta errors are given in Appendix. Figure 6 shows examples of localizations obtained for each of these delta errors. The corresponding Δ values are reported in the first column of Table 1. The last column shows the resulting face verification performance, in terms of HTER, using the DCT/GMM face verification system. This experiment basically shows the following:

(1) There is a significant variation in HTER for the same value of d_{eye}.
(2) The DCT/GMM system is more robust to errors in translation than to errors in scale or rotation (for the same $d_{eye} = 0.2$).

Note that in practice, a face detector does not fail only on one type of error. However, this experiment clearly shows that a face localization performance measure such as d_{eye} is not adapted if we want to take into account the performance of the whole system.

(a) ground-truth ($d_{eye} = 0.0$)

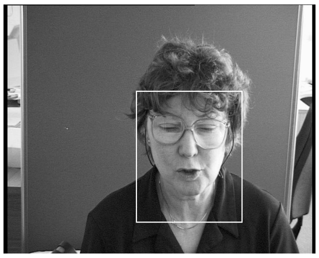
(b) $\Delta_x = 0.2$ ($d_{eye} = 0.2$)



(c) $\Delta_x = -0.2$ ($d_{eye} = 0.2$)

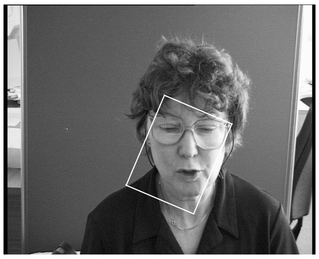
(d) $\Delta_y = 0.2$ ($d_{eye} = 0.2$)

(e) $\Delta_y = -0.2$ ($d_{eye} = 0.2$)

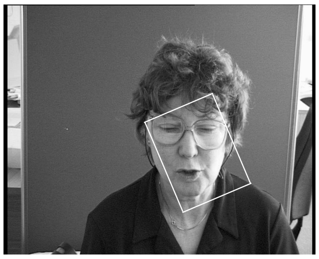


(f) $\Delta_s = 1.4$ ($d_{eye} = 0.2$)

(g) $\Delta_s = 0.6$ ($d_{eye} = 0.2$)



(h) $\Delta_\alpha = 23°$ ($d_{eye} = 0.2$)



(i) $\Delta_\alpha = -23°$ ($d_{eye} = 0.2$)

Fig. 6. Figure (a) shows the face bounding box for the ground-truth annotation. For the given value of $d_{eye} = 0.2$, Figures (b) to (i) illustrate the bounding box resulting from perturbations in horizontal translation (b,c), vertical translation (d,e), scale (f,g) and rotation (h,i).

5 Approximate Face Verification Performance

The preliminary experiments conducted in Section 4 should have convinced that current FL measures are not adapted to the FV task, and we also argued that it is probably not adapted to any other particular task. Hence, as explained in Section 2, instead of searching for a universal measure assessing the quality of a face localization algorithm, we propose here to estimate a specific performance measure adapted to the target task. We here concentrate on the task of face verification, hence a good face localization algorithm in that context is a module which produces a localization such that the expected error of the face verification module is minimized. More formally, let $\mathbf{x}_i$ be the input vector describing the face of an access i, as defined more precisely in Section 3.2, $\mathbf{y}_i = \mathrm{FL}(\mathbf{x}_i)$ be the output of a face localization algorithm applied to $\mathbf{x}_i$ (generally in terms of eye positions), $z_i = \mathrm{FV}(\mathbf{y}_i)$ be the decision taken by a face verification algorithm (generally accept or reject the access) and $\mathrm{Error}(z_i)$ be the error generated by this decision. The ultimate goal of a face localization algorithm in the context of a face verification task is thus to minimize the following criterion:

$$\mathrm{Cost} = \sum_i \mathrm{Error}(\mathrm{FV}(\mathrm{FL}(\mathbf{x}_i))) \ . \tag{8}$$

Our proposed solution for a meaningful FL measure adapted to a given task is thus to embed all subsequent functions (FV and Error) into a single box and to estimate this box using some universal approximator:

$$\mathrm{Cost} = \sum_i f(\mathrm{FL}(\mathbf{x}_i); \theta) \tag{9}$$

where $f(\cdot;\theta)$ is a parametric function that would replace the rest of the process following localization using parameters θ. In this paper, we consider as function $f(\cdot)$ a simple K nearest neighbor (KNN) algorithm [35]. In order to be independent of the precise localization of the eyes, we modified in fact slightly this approach by changing the input of function $f(\cdot)$ in order to contain instead the error made by the localization algorithm in terms of very basic measures: Δ_x, Δ_y, Δ_s and Δ_α, as described in Section 2. Let $\mathrm{GT}(\mathbf{x}_i)$ be the ground-truth eyes position of $\mathbf{x}_i$ and $\mathrm{Err}(\mathbf{y}_i, \mathrm{GT}(\mathbf{x}_i))$ be the function that produces the face localization error vector; we thus have

$$\mathrm{Cost} = \sum_i f(\mathrm{Err}(\mathrm{FL}(\mathbf{x}_i), \mathrm{GT}(\mathbf{x}_i)); \theta) \ . \tag{10}$$

In order to train such a function $f(\cdot)$, we used the following methodology. First, in order to cover the space of localization errors, we create artificial examples based on all available training accesses. The training examples of $f(\cdot)$ are thus uniformly generated by adding small perturbations (localization errors) bounded by a reasonable range. For each generated example, a verification is performed and a corresponding target value of 1 (respectively 0) is assigned when a verification error appears (respectively does not appear).

6 Experiments and Results

This Section is devoted to verifying experimentally if our proposed method to measure the performance of localization algorithms in the context of a face verification task improves with respect to other known measures.

6.1 Training Data

The XM2VTS database was used to generate examples to estimate our function $f(\cdot)$, which should yield the expected verification error given a localization error. For each of the 1000 available client images [3], 50 localization errors were randomly generated following a uniform distribution in a predefined interval $[-1, 1]$ for Δ_x and Δ_y, $[0.5, 1.5]$ for Δ_s and $[-20°, 20°]$ for Δ_α. The training set thus contains 50000 examples. A verification is performed for each example, which will be assigned a target value of 1 (respectively 0) when the verification algorithm accepts the client (respectively rejects him). Furthermore, a separate validation set of 50000 examples was created using the same procedure (with the same set of clients, but a different random seed). The hyper-parameter K of the KNN model, which controls the capacity [36] of $f(\cdot)$, was then chosen as the one which minimized the out-of-sample error on the validation set.

6.2 Face Localization Performance Measure

Given the set of errors $\Delta = \{\Delta_x, \Delta_y, \Delta_s, \Delta_\alpha\}$ generated by the FL algorithm on an image n we define the error of the KNN localization algorithm as:

$$\varepsilon_{\text{KNN}}(\Delta^n) = \frac{1}{K} \sum_{k \in \text{KNN}(\Delta^n)} C_k \tag{11}$$

[3] The preliminary analysis of Section 4.1 showed that FAR is not significantly affected by localization errors, so we did not use any impostor access for this step.

where $\text{KNN}(\Delta^n)$ is the set of the K nearest training examples of Δ^n and C_k is the error made on example k defined as:

$$C_k = \begin{cases} 0 & \text{if} \quad \text{Accepted Client} \\ 1 & \text{if} \quad \text{Rejected Client} \ . \end{cases} \tag{12}$$

We then estimate the performance of the FL system on a set of N images using:

$$E_{\text{KNN}} = \frac{1}{N} \sum_{n=1}^{N} \varepsilon_{\text{KNN}}(\Delta^n) \ . \tag{13}$$

Similarly, we measure the error made by the d_{eye} measure as follows:

$$\varepsilon_{eye}(n) = \begin{cases} 0 & \text{if} \quad \text{Accepted Client and } d_{eye}(n) < 0.25 \\ 1 & \text{if} \quad \text{otherwise} \end{cases} \tag{14}$$

and

$$E_{eye} = \frac{1}{N} \sum_{n=1}^{N} \varepsilon_{eye}(n) \ . \tag{15}$$

6.3 KNN Function Evaluation

In order to verify that the obtained KNN function is robust to the choice of the training dataset, we chose to evaluate it on another dataset, namely the English BANCA corpus. In order to extract the faces from the access images, we used the face localization algorithm described in Section 3.2. This system involves some scanning parameters typically chosen empirically, such as horizontal and vertical steps and scale factor. When minimizing these parameters, the localization is expected to be more accurate, however the computational cost then becomes intractable. These two parameters should thus be selected in order to have a good *performance/computational cost* trade-off. In order to obtain a good trade-off we can either favor translation accuracy by reducing horizontal and vertical steps or scale accuracy by reducing the scale factor.

Note that the localization system only deals with upright frontal faces. It can not be used to test the effect of rotational errors, which is actually independent of the scanning parameters.

We decided to test two different versions of the localization system, as follows:

(1) The first system, FL_{shift}, uses larger values for horizontal and vertical step factors. This system is expected to introduce more errors in translation.

(2) The second system, FL_{scale}, uses finer translational step factors, but a larger scale factor, expected to introduce errors in scale.

We thus have two scenarios. We want to verify that our KNN function is able to measure which is the best FL system, or in other words the one which minimizes the FV error. Table 2 compares the localization errors obtained with the d_{eye} criterion (second column) computed using equation (15), our proposed function (third column) computed using equation (13), and the actual verification score decomposed into its FAR, FRR and HTER components (last 3 columns), on all the accesses of the BANCA database using protocol P and the DCT/GMM FV system. Basically, several conclusions can be drawn from this table:

Table 2
Comparison of two FL performance measures for two face localization systems as well as for a perfect localization (ground-truth). The last 3 columns contains the face verification score in terms of FAR, FRR and HTER for the DCT/GMM system.

FL Systems	Measures		Verification		
	E_{eye}	E_{KNN}	FAR [%]	FRR [%]	HTER [%]
ground-truth	0.00	0.05	15.1	23.9	19.5
FL_{shift}	0.10	0.12	11.7	30.3	21.0
FL_{scale}	0.04	0.15	14.7	33.8	24.3

(1) As expected, the best verification score (HTER = 19.5) is obtained with perfect localization (first conclusion of Section 4.1). Then follows the FL_{shift} system, which yields an HTER of 21.0 and finally the FL_{scale} system with an HTER of 24.3. This ordering was also expected, following the third conclusion of Section 4.1.
(2) Our proposed function correctly identifies the best localization system (FL_{shift}, the system which minimizes the FV error), while the d_{eye}-based measure fails to order the two modules. This can be mainly explained because the d_{eye} measure does not differentiate errors in translation, shift or rotation, while the DCT/GMM FV system is more affected by a certain type of error (third conclusion of Section 4.1).
(3) The KNN almost perfectly predicts the FRR delta between the FL systems and the ground-truth ($0.12 - 0.05 \simeq (30.3 - 23.9)/100$ and $0.15 - 0.12 \simeq (33.8 - 30.3)/100$). Remember that only client accesses were used to train the KNN function (Section 6.1).
(4) We remark that the FAR corresponding to the FL_{shift} system (11.7) and the FL_{scale} system (14.7) are lower than the FAR with perfect localization (15.1). This is because of impostor accesses, a bad face localization only pushes the system to reject more accesses (including impostors accesses), yielding a lower FAR.

Furthermore, the proposed KNN measure only takes 20 ms on a PIV 2.8 Ghz to evaluate an image access, while it would take 350 ms for the DCT/GMM system (preprocessing, feature extraction and classification).

7 Conclusion

In this paper, we have proposed a novel methodology to compare face localization algorithms in the context of a particular application, namely face verification. Note that the same methodology could have been applied to any other task that builds on localization, such as face tracking. We have first shown that current measures used in face localization are not accurate for localization. We have thus proposed a method to estimate the verification errors induced specifically by the use of a particular face localization algorithm. This measure can then be used to compare more precisely several localization algorithms. We tested our proposed measure using the BANCA database on a face verification task, comparing two different face localization algorithms. Results show that our measure does indeed capture more precisely the differences between localization algorithms (when applied to verification tasks), which can be useful to select an appropriate localization algorithm. Furthermore, our function is robust to the training dataset (training on XM2VTS and test on BANCA) and compared to the DCT/GMM face verification system, the KNN performs more than 15 times faster. Finally, in order to compare FL modules, we do not need to run face verification on the entire database, but we only use our function on a subset of face images.

In this paper we used a KNN to estimate the target function. Given Figure 5, the KNN could probably be replaced by a simpler parametric function. For example, under the reasonable assumption that the final error is a simple combination of scale, shift, and rotation errors, the resulting function could be a simple combination of four polynomial functions.

In fact, one can view the process of training a localization system as a selection procedure where one simply selects the best localization algorithm according to a given criterion. In that respect, an interesting future work could concentrate on the use of such a measure to effectively *train* a face localization system for the specific task of face verification.

A From d_{eye} to Δ Measures

In this appendix, we explain how to compute the Δ values (first column of Table 1) corresponding to $d_{eye} = 0.2$.

Each type of localization error is examined independently. For the four cases (Δ_x, Δ_y, Δ_s and Δ_α), we have $d(C_l, \tilde{C}_l) = d(C_r, \tilde{C}_r))$. If we set $D = d(C_l, C_r)$ (distance between the true eye positions), equation (1) can be rewritten as:

$$d_{eye} = \frac{d(C_l, \tilde{C}_l)}{D} \; . \tag{A.1}$$

We now examine each type of error:

- **x translation**
  We have: $d(C_l, \tilde{C}_l) = |\vec{dx}|$
  From (A.1) and (2), we obtain:

$$\Delta_x = \pm d_{eye}. \tag{A.2}$$

- **y translation**
  We have: $d(C_l, \tilde{C}_l) = |\vec{dy}|$
  In the same way, from (A.1) and (3), we obtain:

$$\Delta_y = \pm d_{eye}. \tag{A.3}$$

- **scale**
  An error in scale only induces a perturbation along the x axis. We set $D' = d(\tilde{C}_l, \tilde{C}_r)$ (distance between the detected eye positions) and we distinguish two cases:

(1) $D' > D : d(C_l, \tilde{C}_l) = \frac{D'-D}{2}$
From (A.1) and (4): $d_{eye} = \frac{\Delta_s - 1}{2}$,
(2) $D' < D : d(C_l, \tilde{C}_l) = \frac{D-D'}{2}$
From (A.1) and (4): $d_{eye} = \frac{1-\Delta_s}{2}$, then:

$$\Delta_s = 1 \pm 2d_{eye}. \tag{A.4}$$

- **rotation**
  An error in rotation induces a perturbation both along the x and y axis. For clarity, we define $\vec{v} = \overrightarrow{C_l\tilde{C}_l}$. The distance between the true and the detected left eye position $d(C_l, \tilde{C}_l)$ can then be written as:

$$d(C_l, \tilde{C}_l) = \sqrt{v_x^2 + v_y^2} \tag{A.5}$$

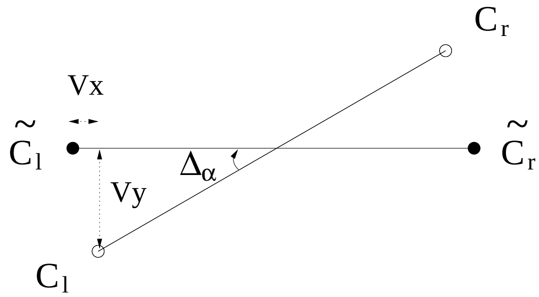


Fig. A.1. v_x and v_y translation error induced by an error in rotation.

where $v_x = ||\vec{v_x}||$ and $v_y = ||\vec{v_y}||$, x and y components of vector $\vec{v}$. By combining (A.1) and (A.5), we get:

$$d_{eye} = \frac{\sqrt{v_x^2 + v_y^2}}{D} \ . \tag{A.6}$$

According to Figure A.1, we have:

$$v_x = \frac{D}{2} \sin \Delta_\alpha \quad , \tag{A.7}$$

$$v_y = \frac{D}{2}(1 - \cos \Delta_\alpha) \quad . \tag{A.8}$$

Using (A.7) and (A.8) in (A.5) we get:

$$d_{eye} = \frac{\sqrt{(\frac{D}{2})^2 \sin^2 \Delta_\alpha + (\frac{D}{2})^2 (1 - \cos \Delta_\alpha)^2}}{D}$$

$$d_{eye} = \sqrt{\frac{\sin^2 \Delta_\alpha + \cos^2 \Delta_\alpha + 1 + 2\cos \Delta_\alpha}{4}}$$

$$d_{eye} = \sqrt{\frac{1 - \cos \Delta_\alpha}{2}}$$

which finally leads to:

$$\Delta_\alpha = \pm \arccos{(1 - 2d_{eye}^2)}. \tag{A.9}$$

From our choice of $d_{eye} = 0.2$ and equations (A.2),(A.3), (A.4) and (A.9), we get the following Δ values:

$$\begin{aligned}
\Delta_{x_{1,2}} &= \pm 0.2 \\
\Delta_{y_{1,2}} &= \pm 0.2 \\
\Delta_{s_1} &= 0.6 \\
\Delta_{s_2} &= 1.4 \\
\Delta_{\alpha_1} &= 23^\circ \\
\Delta_{\alpha_2} &= -23^\circ
\end{aligned}$$

Acknowledgment

This research has been partially carried out in the framework of the Swiss NCCR project (IM)2. It was also supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, funded in part by the Swiss Federal Office for Education and Science (OFES). This publication only reflects the authors' views. All experiments were done using the *Torch3* library [37] and the *Torch3vision* package [4]

References

[1] K. Yow, Automatic human face detection and localization, Ph.D. thesis, University of Cambridge, Department of Engineering (1998).

[2] V. Vezhnevets, Method for localization of human faces in color-based face detectors and trackers, in: Proceedings of The Third International Conference on Digital Information Processing And Control In Extreme Situations, Minsk, Belarus, 2002, pp. 51–56.

[3] K. Huang, M. Trivedi, Robust real-time detection, tracking, and pose estimation of face in video streams, in: Proceedings of the International Conference on Pattern Recognition (ICPR), Cambridge, UK, 2004, pp. 965–968.

[4] D. Comaniciu, V. Ramesh, Robust detection and tracking of human faces with an active camera, in: Proceedings of the IEEE International Workshop on Visual Surveillance, Dublin, Ireland, 2000, pp. 11–18.

[5] S. Spors, R. Rabenstein, A real-time facetracker for color video, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, USA, 2001.

[4] see: http://www.idiap.ch/~marcel/en/torch3/introduct ion.php.

[6] F. Cardinaux, C. Sanderson, S. Marcel, Comparison of MLP and GMM classifiers for face verification on XM2VTS, in: Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), Guilford, UK, 2003, pp. 911–920.

[7] F. Tsalakanidou, S. Malasiotis, M. Strintzis, Face localization and authentication using color and depth images, IEEE Transactions on Image Processing 14 (2) (2005) 152–168.

[8] R. Chellappa, C. Wilson, S. Sirohey, Human and machine recognition of faces: A survey, Proceedings of the IEEE 83 (5) (1995) 705–741.

[9] M. Sadeghi, J. Kittler, A. Kostin, K. Messer, A comparative study of automatic face verification algorithms on the banca database, in: Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), Halmstad, Sweden, 2003, pp. 35–43.

[10] Y. Rodriguez, F. Cardinaux, S. Bengio, J. Mariéthoz, Estimating the quality of face localization for face verification, in: Proceedings of the IEEE International Conference on Image Processing, (ICIP), Singapore, 2004, pp. 581–584.

[11] F. Tek, Face detection using learning networks, Master's thesis, The Middle East Technical University, Departement of Electrical and Electronics Engineering, Ankara, Turkey (2002).

[12] R.-J. Huang, Detection strategies for face recognition using learning and evolution, Ph.D. thesis, George Mason University, Fairfax, Virginia (1998).

[13] M.-H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 24 (1) (2002) 34–58.

[14] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 20 (1) (1998) 23–38.

[15] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, in: Proceedings of the 25th DAGM-Symposium, Magdeburg, Germany, 2003, pp. 297–304.

[16] B. Fröba, C. Küblbeck, Robust face detection at video frame rate based on edge orientation features, in: Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), Washington, D.C., USA, 2002, pp. 342–347.

[17] O. Jesorsky, K. Kirchberg, R. Frischholz, Robust face detection using the hausdorff distance, in: Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), Halmstad, Sweden, 2001, pp. 90–95.

[18] S. Behnke, Face localization in the neural abstraction pyramid, in: Proceedings of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES), Oxford, UK, 2003, pp. 139–145.

[19] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. K. Kälviäinen, Affine-invariant face detection and localization using gmm-based feature detector and enhanced appearance model, in: Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), Seoul, Korea, 2004, pp. 67–72.

[20] V. Popovici, Y. Rodriguez, J.-P. Thiran, S. Marcel, On performance evaluation of face detection and localization algorithms, in: Proceedings of the International Conference on Pattern Recognition (ICPR), Cambridge, UK, 2004, pp. 313–317.

[21] J. Lüttin, G. Maître, Evaluation protocol for the extended m2vts database (xm2vtsdb), IDIAP Communication 98-05, IDIAP Research Institute, Martigny, Switzerland (1998).

[22] E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruiz, J.-P. Thiran, The BANCA database and evaluation protocol, in: Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), Guilford, UK, 2003, pp. 625–638.

[23] B. Fröba, A. Ernst, Face detection with the modified census transform, in: Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), Seoul, Korea, 2004, pp. 91–96.

[24] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Porceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 2001, pp. 511–518.

[25] Y. Freund, R. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the IEEE International Conference on Machine Learning (ICML), Bari, Italy, 1996, pp. 148–156.

[26] F. Cardinaux, C. Sanderson, S. Bengio, Face verification using adapted generative models, in: Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), Seoul, Korea, 2004, pp. 825–830.

[27] S. Li, A. Jain, Handbook of Face Recognition, Springer, 2004.

[28] G. Medioni, S. Kang, Emerging Topics in Computer Vision, Prentice-Hall, 2004, Ch. Face Detection and Recognition.

[29] F. Cardinaux, C. Sanderson, S. Bengio, User authentication via adapted statistical models of face images, To appear in IEEE Transaction on Signal Processing.

[30] C. Sanderson, K. Paliwal, Fast features for face authentication under illumination direction changes, Pattern Recognition Letters 24 (14) (2003) 2409–2419.

[31] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted gaussian mixture models, Digital Signal Processing 10 (1-3) (2000) 19–41.

[32] M. Turk, A. Pentland, Eigenface for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 70–86.

[33] C. Sanderson, S. Bengio, Extrapolating single view face models for multi-view recognition, in: Proceedings of the International Conference of Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Melbourne, Australia, 2004, pp. 581–586.

[34] C. Sanderson, K. Paliwal, Identity verification using speech and face information, Digital Signal Processing 14 (5) (2004) 449–480.

[35] C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[36] V. Vapnik, Statistical Learning Theory, Wiley, Lecture Notes in Economics and Mathematical Systems, volume 454, 1998.

[37] R. Collobert, S. Bengio, J. Mariéthoz, Torch: a modular machine learning software library, Technical Report 02-46, IDIAP Research Institute, Martigny, Switzerland (2002).