Visual Identification by Signature Tracking

Mario E. Munich, *Member*, *IEEE*, and Pietro Perona, *Member*, *IEEE*

Abstract—We propose a new camera-based biometric: visual signature identification. We discuss the importance of the parameterization of the signatures in order to achieve good classification results, independently of variations in the position of the camera with respect to the writing surface. We show that affine arc-length parameterization performs better than conventional time and Euclidean arc-length ones. We find that the system verification performance is better than 4 percent error on skilled forgeries and 1 percent error on random forgeries, and that its recognition performance is better than 1 percent error rate, comparable to the best camera-based biometrics.

Index Terms—Systems and applications, active and real-time vision, signature verification, signature recognition, biometrics.

1 Introduction and Motivation

Identifying yourself to a machine is the first step of most automated transactions. The desire for ever increasing convenience and security motivates the development of biometric techniques in order to replace keys, passwords, and smart cards. The most popular biometric identification techniques are face recognition [4], [28], [32], fingerprint recognition [13], iris recognition [5], retina recognition [9], speaker identification [3], and signature verification [16], [18], [29].

Signature verification presents three likely advantages over other biometric techniques from the point of view of adoption in the market place. First, it is a socially accepted identification method already in use in bank and credit card transactions; second, most of the new generation of portable computers and personal digital assistants (PDAs) use handwriting as the main input channel; third, a signature may be changed by the user, similarly to a password, while it is not possible to change fingerprints, iris, or retina patterns. Therefore, automatic signature verification has the unique possibility of becoming the method of choice for identification in many types of electronic transactions.

Most signature verification systems require either the use of electronic tablets or digitizers for online capturing [16], or optical scanners for offline conversion [37]. These interfaces are bulky (they need to have a footprint at least as large as the largest signature) and require the presence of dedicated hardware. Cameras, on the other hand, may be made as small as a pen cap and are becoming ubiquitous in the current computer environment. We have demonstrated [20], [24], [25] the feasibility of building an interface for handwriting using off-the-shelf camera and frame grabber. This paper explores the use of the interface for signature verification and signature recognition, a vision-based personal identification system that could be integrated as a component of a complete visual pen-based computer environment.

- *M.E. Munich is with Evolution Robotics, 130 W. Union Street, Pasadena, CA 91103. E-Mail: mariomu@vision.caltech.edu.*
- *P. Perona is with the Department of Electrical Engineering—136-93, California Institute of Technology, Pasadena, CA 91125. E-mail: perona@caltech.edu.*

Manuscript received 24 Aug. 1 2001; revised 17 May 2002; accepted 14 July 2002.
Recommended for acceptance by S. Sclaroff.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 114842.

The literature on signature verification is quite extensive (see [16], [18], [29] for surveys) and is divided into two main areas of research: offline and online systems. Offline systems deal with a static image of the signature; online systems capture the position of the pen tip as a function of time. Our system is based on capturing the full motion sequence of the act of signing and, therefore, it falls in the second category (see Fig. 1 for a comparison between the data used by offline and online systems).

Automatic signature verification systems, like all other biometric verification systems, involve two processing modes: training and testing. In the *training* mode, the user provides signature samples that are used to construct a model or prototype representing some distinctive characteristics of his signature. In the *testing* mode, the user provides a new signature, along with the alleged identity, and the system judges the likely authenticity of the presented sample with respect to the alleged class model. A signature recognition system differs from a verification system in that it has to find the corresponding class of each new incoming signature from a pool of M possible classes. Fig. 2 compares the two systems.

A signature verification or recognition system has to be designed to meet a set of requirements in terms of performance, training set size, and robustness conditions (see [7], [16], [29] for more information on design requirements). These specifications are application dependent. The required performance could vary from 0.01 percent false acceptance for credit cards authentication systems to 2-3 percent false acceptance/rejection for an identification system for a personal desktop computer. The required system invariance would depend on the signature acquisition set up. For very restricted acquisition conditions such as fixed signing location and direction, only invariance with respect to small changes in signature size would be required. For more relaxed conditions in which the user could sign in any place, additional invariance with respect to translation and rotation of the signature are expected. Finally, for a system such as the one described in this paper, invariance with respect to affine deformations is also needed since we suppose that the location of the camera could change with regard to the signing surface.

One common characteristic of many online systems for signature verification is that the training set has a small number of signature examples. Few users would be willing to provide more than 5-10 examples of their signature, and even if they would, the boredom of the repetitive task of signing

Fig. 1. **Offline and online captured signatures**. The top three rows show signatures captured offline with a scanner. The bottom two rows show signatures captured online with the visual pen-tracking interface. These examples display the changes in the signature pattern from subject to subject and the variability in the signatures from the same subject. The two first rows show pictures of signatures attributed to Napoleon Bonaparte. Note that his signatures changed over time from "Bonaparte" to "Napoleon" and, finally, to a simplified script. The third row presents pictures of signatures from George Washington whose signatures are much more consistent than Napoleon's. The last two rows show example signatures collected with the visual pen-tracking interface; the dots represent the captured sample points. The first subject signs inconsistently: loops are added, deleted, and distorted in the middle of his signatures. The second subject is much more consistent: the overall shape of his signatures is maintained even though no two signatures are equal.

would introduce uncommon distortion in the data; hence, signatures captured in this condition would not be representative of the class of objects to be modeled. This data dependence on the state of mind of the subject is inherent to behavioral biometrics like signature verification and speaker identification; good signature samples may be collected if the subject is relaxed and motivated, otherwise, the samples could present a high degree of unusual variability.

A few additional issues need to be resolved in the design of an online signature verification system. The main ones are the choice of a parameterization for the signatures, the choice of a metric to be used for comparing signatures, and, more in general, the extraction of a model of a complicated and variable object, such as signatures, from very few training examples. All these issues are addressed in this paper.

The contributions of this paper are the following: It is the first study of signature acquisition using a camera, rather than a tablet or other device; the system is shown to achieve similar or better verification and recognition performance than other comparable camera-based identification systems presented in the literature, e.g., face recognition or hand shape recognition (iris and fingerprint recognition are more accurate; however, they require dedicated capture devices instead of a common multipurpose camera and, therefore, are not comparable to our system). Moreover, we show that our system achieves a verification performance that is comparable to the best signature verification performances cited in the literature. We also find that an affine-invariant parameterization of the signatures provides a better verification performance than conventional time parameterization or Euclidean arc-length parameterization.

This paper is organized as follows: Section 2 describes the signature acquisition system and the data acquisition protocol, Section 3 presents the method used to perform the comparison between signatures and discusses the different signature parameterizations, Section 4 describes the system implementation and the method used to evaluate the performance of the system, Section 5 provides the experimental results, and Section 6 discusses the results and describes possible extensions.

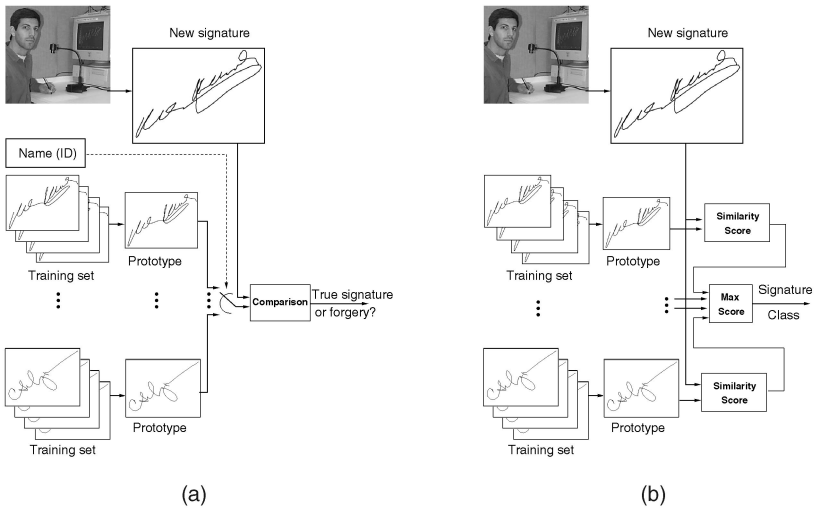


Fig. 2. **Verification versus recognition**. (a) **Signature verification system**. The training set is used to extract the subject's prototype signature. A new signature provided along with the claimed identity is compared with the corresponding prototype and classified as a true signature or a forgery. (b) **Signature recognition system**. The prototypes are extracted in the same way as in the verification system. A new signature is compared with the prototypes from different subjects and is assigned to the class with maximum similarity score.

2 Capturing Handwriting with Vision

Signatures are captured with the system described in [20], [21], [24], [25]. We briefly review the main characteristics of the system in this section. Fig. 3 shows a block diagram of the system and the experimental setup.

The system has been implemented in real time with a hardware that consists of a video camera, a frame grabber, and a Pentium II 230 PC. The camera is a commercial Flexcam ID, manufactured by Videolabs, equipped with manual gain control. It has a resolution of 480×640 pixels per interlaced image. The frame grabber is a PXC200 manufactured by Imagination. The input camera image is digitized by the grabber and transferred to memory at 60Hz through the PCI bus. All further computations are performed with the PC. We achieved a total processing time of 14ms per frame. The pen tip detection accuracy has been reported ([20], [24]) to be roughly one-tenth of a pixel. The resolution of signature acquisition is difficult to measure since the size of the signature in the camera image varies depending on the position and orientation of the camera. A typical signature from our experiments occupies about 20 image pixels per centimeter of signature; hence, the resolution of signature acquisition is roughly 200 samples per centimeter of signature.

2.1 Data Collection

We collected two sets of signatures[1] with the camera-based interface in order to evaluate the verification performance of the system. The first set was used throughout the design, development, and tuning of the system, so the performance on this set may be overly optimistic due to overfitting. The second set was captured after the system was fully developed and the corresponding error rates were computed *only once*, providing, in this way, a better estimation of the generalization error of the system. All data was collected using a *fully automatic* experimental set-up in which the subject *autonomously* interacted with the windows-based application shown in Fig. 3b.

The first data set consists of signatures from 56 subjects, 18 of them women and four left handed. Each subject was asked to provide 25 signatures, 10 to be used as the training set, and 15 to be used as the test set. The second data set consists of signatures from 50 subjects (no intersection with the first set), 14 of them women and six left handed. Each subject was asked to provide 30 signatures, 10 to be used as the training set and 20 to be used as the test set. Fig. 4 shows one signature from each of the subjects in the two databases (subject s048 of set 2 was left out from the figure for lack of space, but Figs. 8 and 19 show s048's signatures).

The data for each subject was collected in three sessions that took place on different days. This procedure provides a sample of the variability of the subject's signatures, while at the same time, avoids the distortion produced by the boredom of the repetitive task of signing. The camera was not placed at a fixed position and height; it was changed from subject to subject and from session to session. Five of the signers additionally provided 10 forgeries for each of the subjects in the database (see Figs. 11 and 19 for examples of forgeries). Each set of forgeries for a particular subject was collected in one session. The forger was shown the ink trace of a set of real signatures and given enough time to practice until he felt comfortable writing the forgeries. The visual tracker was set up such that the subject could not remain still in the same place for more than 500 milliseconds, not allowing the forger to copy the signatures at a very slow speed but rather forcing him to produce them at normal signing pace. The forger knew that the system was acquiring the full signing trajectory. The system performed verification on-the-fly to give the forger feedback on whether the forgery was accepted as a true signature.

3 Signature Comparison

Comparing two signatures is not easy. Signatures may differ in many ways, even if they are generated by the same subject. Possible differences include variations in length, additions and deletions of portions of them, and changes in

1. The signatures used in this paper are publicly available for academic use at http://www.vision.caltech.edu/mariomu/research/data/.

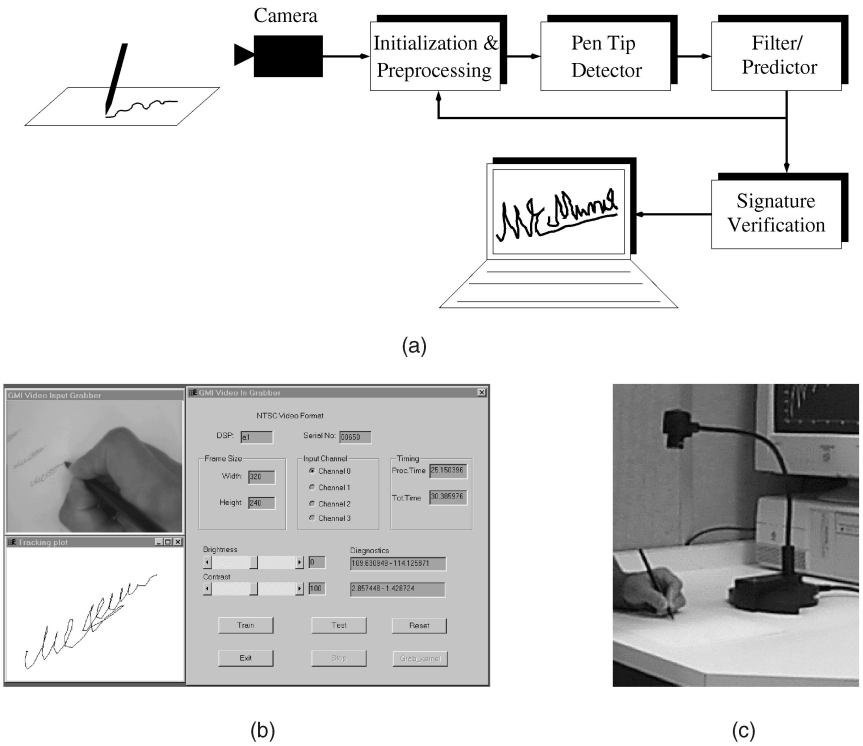


Fig. 3. **Vision-based signature acquisition system**. (a) **Block diagram of the system**. The camera provides a sequence of images to the initialization/preprocessing stage. This block initializes the system, i.e., it finds the initial position of the pen and selects the template (rectangular subregion of the image) corresponding to the pen tip. The initialization method is a semiautomatic one that requires a small amount of user cooperation. We display the image captured by the camera on the screen of the computer as shown on (b) and a rectangular box is overlaid at a particular location on the image. The user is required to place the pen tip inside the displayed box, ready to start writing. The system detects when the pen enters into the box and selects the template used to perform tracking. In subsequent frames, the preprocessing stage extracts the region of interest in which the tip detector obtains the current position of the pen tip. Assuming that the changes in size and orientation of the pen tip during the sequence of images are small, the most likely position of the pen tip in each frame is given by the location of the maximum of the correlation between the template and the region of interest. The filter is a recursive estimator that predicts the position of the tip in the next frame based on an estimate of the current position, velocity, and acceleration of the pen. The filter also estimates the most likely position of the pen tip for missing frames. Finally, the last block of our system performs signature verification. (b) **Graphical user interface**. GUI of the windows-based application that implements our system. The biggest window is a Dialog Box that allows the user to input parameters and run commands; a second window is used to show the image that the camera is providing to the system; the last window shows the captured trajectory. The GUI allows the subjects to *autonomously* use the system for data collection. (c) **Experimental setup**. The camera is looking at a piece of paper in which the user is signing with a common pen. The image captured by the camera is shown on the GUI to provide visual feedback to the user. The system does not require any calibration. The user has the flexibility of arranging the relative positions of the camera and the piece of paper in order to write comfortably provided that the system has a clear sight of the pen tip.

velocity due to pauses or hesitations of the writer. Fig. 1 shows examples of this variability.

Different methods have been proposed in the literature in order to provide a measure of the similarity between two signatures. Neural networks, hidden Markov models, elastic matching, regional correlation, and dynamic programming are some of the algorithms used to compare signatures. In particular, Dynamic Programming Matching (DPM) is a technique that finds correspondence between samples points of two signatures, using some predefined metric. Given this correspondence, it is possible to calculate a "distance" between the signatures. The use of DPM for comparison of time functions was initially proposed in the field of speech recognition by Sakoe and Chiba [33] and is described in detail in by Rabiner and Juang in [31] with the name of Dynamic Time Warping (DTW). DPM has been successfully used for signature verification by many researchers [7], [8], [11], [19], [22], [23], [26], [27], [34], [39]; we use this technique to establish correspondence between signatures acquired with our visual pen-tracking system.

Sato and Kogure [34] proposed to use DPM in order to align the shape of signatures consisting only of pen-down strokes, after having normalized the data with respect to translation, rotation, trend, and scale. They further used the result of DPM to compute the alignment of the pressure function and to calculate a measure of the difference in writing motion. They perform the classification based on three measures: the residual distance between shapes after time alignment, the residual distance between pressure functions, and the distance between writing motions.

Parizeau and Plamondon [27] evaluated the use of DPM for signature verification by aligning either horizontal or vertical position ($x(t)$, $y(t)$), horizontal or vertical velocity ($v_x(t)$, $v_y(t)$), or horizontal or vertical acceleration ($a_x(t)$, $a_y(t)$). In their work, they used complete signing trajectories, consisting of both pen-down and pen-up strokes.

Hastie et al. [8] obtained a statistical model of signatures that allows for variations in the speed of writing as well as affine transformations. DPM was used to find the correspondence between speed signals of pairs of signatures. The distance measure provided by DPM was used as the classification parameter. During training, the signature with the lowest distance to all others was chosen as the reference and its speed signal was used to perform letter segmentation. All other signatures were also segmented into letters by using

Fig. 4. **Signature databases**. Signatures collected with the camera-based interface.

the correspondence provided by DPM. Letter templates were extracted from the segmented signatures and were used for comparison and classification during testing.

Huang and Yan [11] presented the use of DPM for matching signature strokes by finding a warp path that, at the same time, minimizes the cost of aligning the shape, the

Fig. 5. **Consistency of pen-up strokes**. Examples of signatures acquired with the visual interface and corresponding images captured by the camera after finishing the acquisition. The images show the ink trace left on the paper, i.e., they display only the pen-down strokes of the signatures. The signature plots show the complete signing trajectories captured by the interface; these trajectories consists both of pen-down and pen-up strokes. The pen-up strokes of the signatures are as consistent as the pen-down ones; hence, we use the complete signing trajectory to perform signature verification and recognition.

velocities, and the accelerations of the individual strokes. Pen-up strokes are merged with pen-down strokes in the preprocessing phase of their algorithm.

Nalwa [26] parameterized the pen-down strokes of the signature using arc length instead of time; a number of characteristic functions such as coordinates of the center of mass, torque, and moments of inertia were computed using a sliding computational window and a moving coordinate frame. A simultaneous DPM over arc-length of all these characteristic functions for the two signatures under comparison provided a measure of similarity to be used for classification.

Summarizing, DPM has been used to match position, velocity, speed, acceleration, pressure, as well as other functions derived from the signatures. Some researchers used only the pen-down strokes for matching, while some others used the full signing trajectory. Some systems normalized the signatures with respect to translation, rotation, trend, or scale. Time parameterization of the signatures has been used by most researchers with the exception of Nalwa [26] that parameterized the signatures in arc-length.

Our implementation of DPM for signature verification [20], [22], [23] attempts to perform the best alignment of the 2D shape of the signatures using a translation-invariant measure of curve similarity, i.e., we find the time warping function that has the minimum cost of aligning the planar curves that represent signatures. The translation-invariant distance was proposed by Serra and Berthod [35], and it is described in Section 3.1. We note that the pen-up strokes drawn by each subject were as consistent as the pen-down strokes, as shown in Fig. 5. This observation agrees with the evidence [18] that signatures are produced as a ballistic or reflex action, with minimal visual feedback; therefore, we use the full signing trajectory in our experiments. We observed that, on average, users were very consistent in their style of signing: They wrote their signatures with a similar slant, in a similar amount of time, with similar dimensions, and with a similar motion.

At the beginning of each data collection session, the subject was allowed to adjust the camera until reaching a comfortable signing position. The movement of the camera introduced rotation and affine distortion to the signatures that needed to be compensated by the system. Section 3.2 describes the normalization of the signatures for rotation and Section 3.3 introduces a parameterization of the signatures that increases the robustness of the system with respect to affine deformations.

3.1 Curve Matching Using Dynamic Programming

Given two signatures, we regard them as two-dimensional discretized curves $C_1 = \{\mathbf{X}(i), i = 1, \cdots, N_x\}$ and $C_2 = \{\mathbf{Y}(j), j = 1, \cdots, N_y\}$ as in Fig. 6a. Let us assume that we have a warping or correspondence map $\phi = (t, s)$ between C_1 and C_2, such that a point $\mathbf{X}(t_k) \in C_1$ corresponds to a point $\mathbf{Y}(s_k) \in C_2$, for $k \in \{1, \cdots, N\}$, $t_k \in \{1, \cdots, N_x\}$, $s_k \in \{1, \cdots, N_y\}$. Let $\mathbf{X}(t_k)$ correspond to $\mathbf{Y}(s_k)$ and $\mathbf{X}(t_{k-1})$ correspond to $\mathbf{Y}(s_{k-1})$ as shown in Fig. 6a; let us define the elementary distance $d((t_{k-1}, s_{k-1}), (t_k, s_k))$ between two signature segments as in (1) and let us define the similarity measure between the two curves given ϕ, $\mathbf{D}_\phi(C_1, C_2)$, as the sum of elementary distances shown by (2).

$$d((t_{k-1}, s_{k-1}), (t_k, s_k)) = \|\overrightarrow{\mathbf{X}(t_k)\mathbf{Y}(s_k)} - \overrightarrow{\mathbf{X}(t_{k-1})\mathbf{Y}(s_{k-1})}\|^2 \quad (1)$$

$$\begin{aligned} \mathbf{D}_\phi(C_1, C_2) &\triangleq \sum_{k=2}^{N} d((t_{k-1}, s_{k-1}), (t_k, s_k)) \\ &= \sum_{k=2}^{N} \|\overrightarrow{\mathbf{X}(t_k)\mathbf{Y}(s_k)} - \overrightarrow{\mathbf{X}(t_{k-1})\mathbf{Y}(s_{k-1})}\|^2. \end{aligned} \quad (2)$$

Having defined the distance between curves given the warping function $\phi = (t, s)$, the actual problem to solve is:

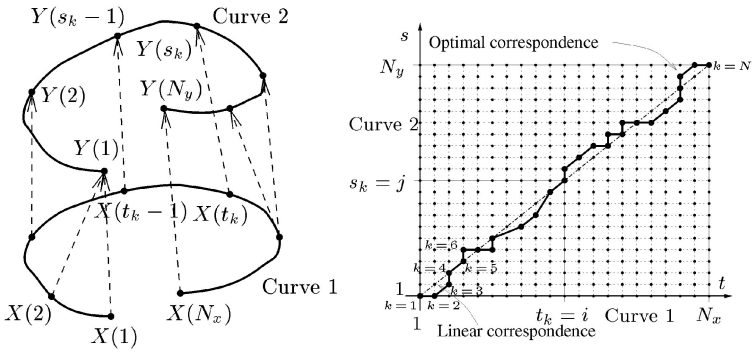


Fig. 6. **Curve correspondence**. (a) Correspondence between the two curves C_1 and C_2. (b) Matching process on the *warping plane*. The diagonal dot and dashed line shows the linear correspondence between the curves; the solid line shows the optimal matching obtained with Dynamic Programming Matching. The correspondence function is parameterized by $k \in \{1, \cdots, N\}$.

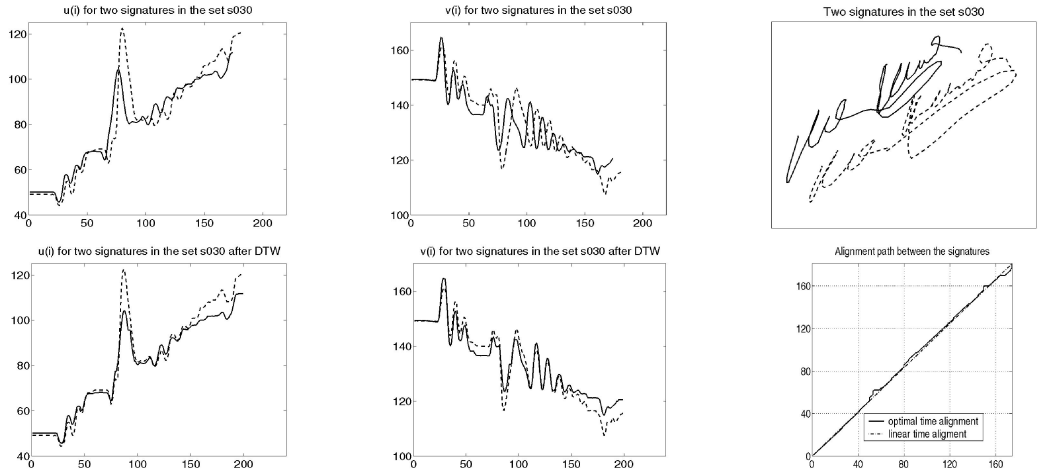


Fig. 7. **Dynamic Programming Matching (DPM) example**. Two signatures from subject s030 matched with DPM. The first column shows the horizontal coordinate $u(i)$ of both signatures before and after matching; the second column shows the vertical coordinate $v(i)$ of both signatures before and after matching. The upper plot of the third column shows the signatures. The lower plot of the third column shows the optimal time warping function compared with a linear matching function. We note that the matching is quite good regardless of the differences in $u(i)$ and $v(i)$. The remaining mismatch between these signals accounts for the differences in shape of the signatures.

Find the function ϕ that minimizes $\boldsymbol{D}_\phi(C_1, C_2)$ and obtain the resulting minimum distance between the curves. The matching process may be visualized on the "warping plane" of Fig. 6b where t_k is represented on the x-axis and s_k is represented on the y-axis. The set of sample points on C_1 and C_2 defines a grid on the warping plane; the correspondence function ϕ joins different nodes of the grid and defines a curve or a path on this plane that is parameterized by $k \in \{1, \cdots, N\}$; if the warping path crosses one vertex (i, j) of the grid, it means that point $\mathbf{X}(i) \in C_1$ corresponds to $\mathbf{Y}(j) \in C_2$. Fig. 7 shows an example of DPM applied to two signatures.

Let us summarize the Dynamic Programming Matching algorithm. In order to backtrack the warping path after obtaining the minimum distance, the algorithm needs to recall the parent node of each point (i, j) within the allowed region of the warping plane. Let us store the parent node of point (i, j) in $\zeta(i, j)$.

1. Initialization:

$$\mathcal{D}(1,1) = 0,$$
$$\zeta(1,1) = (1,1).$$

2. Recursion: for $1 \le i \le N_x$, $1 \le j \le N_y$, such that i and j stay within the allowed grid and follow a predefined set of constraints, compute:

$$\mathcal{D}(i,j) = min \begin{cases} \mathcal{D}(i-1,j) + d((i-1,j),(i,j)) \\ \mathcal{D}(i-1,j-1) + d((i-1,j-1),(i,j)) \\ \mathcal{D}(i,j-1) + d((i,j-1),(i,j)), \end{cases}$$
$$\zeta(i,j) = argmin \begin{cases} \mathcal{D}(i-1,j) + d((i-1,j),(i,j)) \\ \mathcal{D}(i-1,j-1) + d((i-1,j-1),(i,j)) \\ \mathcal{D}(i,j-1) + d((i,j-1),(i,j)). \end{cases}$$

3. Termination:

$$\mathbf{D}(C_1, C_2) = \mathcal{D}(N_x, N_y),$$
$$\phi_1 = (N_x, N_y).$$

4. Path Backtracking:

$$\text{do } \phi_{k+1} = \zeta(\phi_k) \text{ until } \phi_{k+1} = (1,1).$$

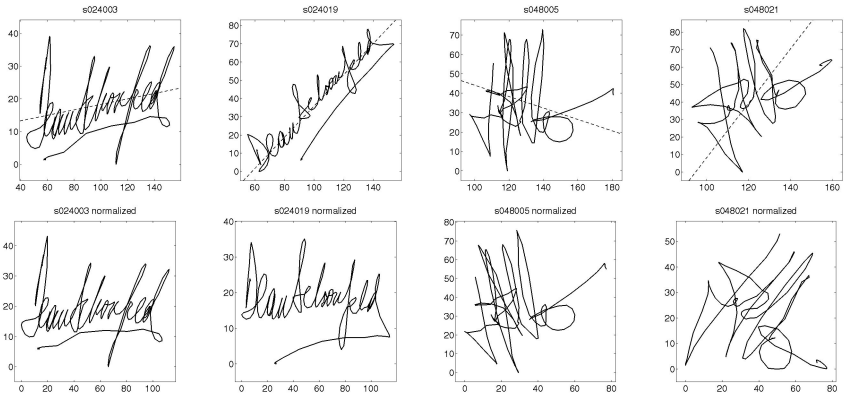


Fig. 8. **Rotation normalization**. The signatures in the first row are the original ones captured with the visual tracker and the signatures on the second row are the corresponding ones after rotation normalization. The dotted line shows the axis of least inertia of the original signatures. The normalization works quite well for subject s024's signatures and fails for subject s048's signatures.

3.2 Rotation Normalization

The distance defined by (2) is not invariant to rotation, so we need to normalize each signature with respect to rotation. The normalization is performed by extracting the axis of least inertia of the signature and rotating the curve until this axis coincides with the horizontal axis. Given the curve

$$C = \{\mathbf{X}(i) = \begin{bmatrix} u(i) \\ v(i) \end{bmatrix}, i = 1, \cdots, N\}.$$

Let $\bar{u} = \frac{1}{N}\sum_{i=1}^{N} u(i)$ and $\bar{v} = \frac{1}{N}\sum_{i=1}^{N} v(i)$ be the coordinates of the center of mass of the signature. It can be shown (see [10]) that the orientation of the axis of least inertia is given by the orientation of the least eigenvector of the matrix

$$I = \begin{pmatrix} \overline{u^2} & \overline{uv} \\ \overline{uv} & \overline{v^2} \end{pmatrix},$$

where $\overline{u^2} = \frac{1}{N}\sum_{i=1}^{N}(u(i) - \bar{u})^2$, $\overline{v^2} = \frac{1}{N}\sum_{i=1}^{N}(v(i) - \bar{v})^2$, and $\overline{uv} = \frac{1}{N}\sum_{i=1}^{N}(u(i) - \bar{u})(v(i) - \bar{v})$ are the second order moments of the signature.

Experiments 1 and 2 compare the performance of the verification system with and without normalization for rotation. This normalization is not always successful since it assumes that the signatures have a clearly defined axis of least inertia. Fig. 8 shows a subject (s024 $\in$ set 1) for which the rotation normalization works quite well and another subject (s048 $\in$ set 2) for which the rotation normalization fails. Subject s048 is the only one in our two sets of signatures for whom the rotation normalization fails. The individual equal error rates for subject s048 are 5.75 percent for skilled forgeries and 0.75 percent for random forgeries. For this subject, rotation normalization makes several authentic signatures appear quite dissimilar from the prototype signature extracted from the training set; therefore, the system classifies these signatures as forgeries, making the error rate be high. Subject s048 was not excluded from any of the experiments presented in Section 5.

3.3 Signature Parameterization

Having normalized the signatures for rotation, the repositioning of the camera could still introduce affine deformation to the signatures as shown in Fig. 9. This section describes a parameterization of the signatures that incorporates a certain degree of invariance with respect to affine transformations into the system.

In most of the previous signature verification and recognition work, a time-based parameterization of the signatures under comparison has been used. There is no clear reason for using this parameterization other than the convenience of being automatically provided by the capture device, the assumption that each time sample contains an identical amount of information, or both. To our knowledge, only Nalwa [26] used an arc-length parameterization of the signatures for computing the distinctive functions proposed in his paper. Arc-length parameterization of the signature is only loosely dependent on time and on the dynamics of signing, even though it maintains the causality of the signature's generation. The weak dependence on the

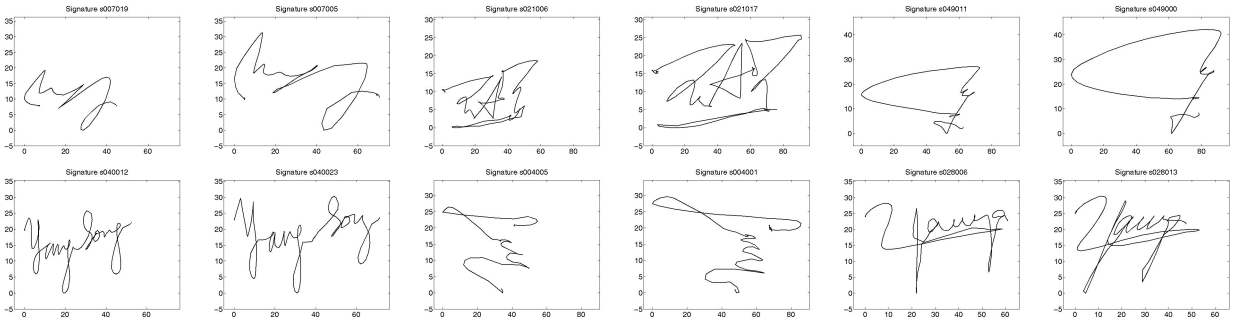


Fig. 9. **Affine distortion of the signatures**. Example signatures captured varying the position of the camera. The signatures are shown after rotation normalization. The first row shows the effect of zooming into the writing surface. The first two subjects of the second row present the effect of horizontal scaling. The last subject of the second row displays the effect of shear.

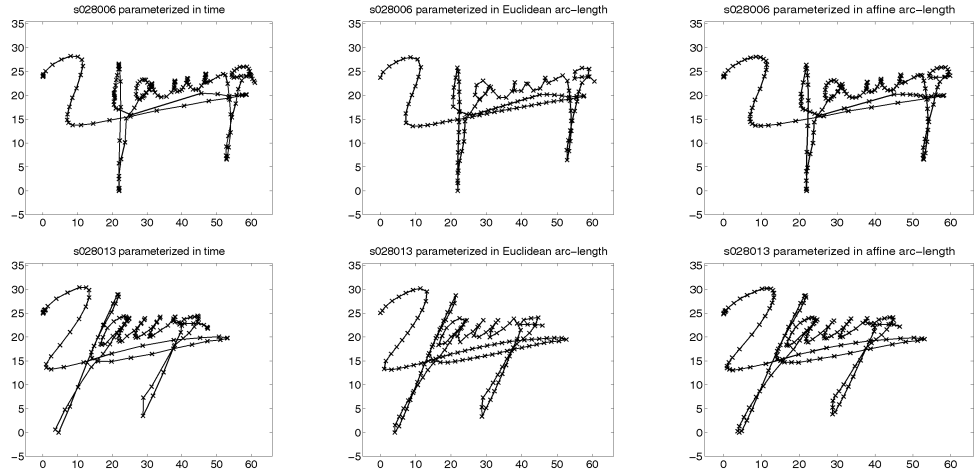


Fig. 10. **Signature parameterizations**. The first column shows two original signatures parameterized in time (the signatures are presented after rotation normalization). The second and third columns display the signatures parameterized in Euclidean and affine arc-length, respectively. The affine arc-length parameterization assigns fewer points to straight segments than to curved ones, i.e., more samples are assigned to fine details of the signatures.

dynamics of signing seems contrary to the traditional idea that pen dynamics is a key element in detecting forgeries. However, the use of the arc-length parameterization is a first step toward achieving invariance with respect to Euclidean transformations of the signatures. Going one step further, we could use a parameterization that provides a certain degree of invariance with respect to affine transformations of the signatures. This parameterization has been described in the literature [2] and has been called *affine arc-length* by Pollick and Sapiro [30].

Several studies (see [15], [30], [38] and references therein) show that the generation and perception of planar movements by humans involve a direct relationship between the tangential velocity of the hand and the radius of curvature of the planar curve. Experimental results show that the tangential velocity decreases as the curvature increases. A mathematical fitting of these results gives rise to a power law in which the tangential velocity is proportional to the $\frac{1}{3}$ power of the radius of curvature. While the relationship between these two quantities is very intuitive, there is no clear explanation for the exact factor $\frac{1}{3}$ in the power law. Pollick and Sapiro [30] showed that this power law precisely implied motion at a constant affine velocity, i.e., that curves with equal affine length would be drawn in equal time. The main question is why affine parameters seem to be embedded in the representation of planar motion. One possible explanation presented in [30] notes that affine transformations are obtained when a planar object is rotated and translated in space, and then projected into the eye via parallel projection. This approximated model for the human visual system is valid when the object is flat enough and away from the eye, as in the case of drawing and planar point motions. These observations are the main motivation for using affine arc-length in our experiments.

Arc-length parameterization of curves is a standard practice in differential geometry [14], [17]. Let us briefly describe the relations used to reparameterize the signatures on Euclidean[2] and affine arc-lengths (adapted from [30]). A planar curve may be defined as the locus of points

$$C(p) = \begin{bmatrix} u(p) \\ v(p) \end{bmatrix} \in \mathbb{R}^2,$$

with $p \in [0, 1]$. Different parameterizations p define the same curve but give rise to different velocities along the curve $\frac{\partial C}{\partial p}$. Given an increasing function $q(p) : [0, 1] \rightarrow [0, 1]$, the curve defined by $C(q) = C(q(p))$ will be the same as the one defined $C(p)$, even though the velocities along the curve will be different $\frac{\partial C}{\partial p} \neq \frac{\partial C}{\partial q}$. One of the best-known curve parameterizations is the *Euclidean arc-length* ν defined such that the curve is traveled with constant velocity, i.e., $\| \frac{\partial C}{\partial \nu} \| = 1$. The Euclidean arc-length parameterization defines a curve length that is invariant with respect to rotations and translations (Euclidean transformations). Given a curve parameterized with an arbitrary parameterization p, we use the relation $\nu(p) = \int_0^p \| \frac{\partial C(t)}{\partial t} \| \, dt$ in order to reparameterize it in Euclidean arc-length.

If we allow for affine transformations rather than Euclidean ones, the Euclidean length ν is not invariant any more. A new parameterization s on *affine arc-length* is defined such that the resultant affine curve length is invariant with respect to affine transformations. Given the curve with an arbitrary parameterization p, the reparameterization in affine arc-length s is defined by the condition $\left| \frac{\partial C}{\partial s} \times \frac{\partial^2 C}{\partial s^2} \right| = 1$, which means that the area of the parallelogram defined by the vectors $\frac{\partial C}{\partial s}$ and $\frac{\partial^2 C}{\partial s^2}$ is constant. The reparameterization of the curve is obtained with the relation $s(p) = \int_0^p \left| \frac{\partial C}{\partial t} \times \frac{\partial^2 C}{\partial t^2} \right|^{\frac{1}{3}} dt$, this equation is singular for the case of straight lines since the vectors $\frac{\partial C}{\partial s}$ and $\frac{\partial^2 C}{\partial s^2}$ are collinear; hence, we thresholded the minimum value of the vector product in order to avoid the singularity. In our experiments, we found only one signature for which the thresholding was necessary.

Fig. 10 shows two signatures acquired with the visual tracking system and the corresponding reparameterizations on Euclidean and affine arc-length. The average Euclidean and affine displacement between sample points were extracted from each subject's training set and were used to reparameterize the corresponding subject's signatures and forgeries. Signature matching using DPM is performed after reparameterization of the signatures.

2. The standard arc-length parameterization of curves, that was used by Nalwa in his system, is called Euclidean arc-length in this paper in order to differentiate it from the affine arc-length.

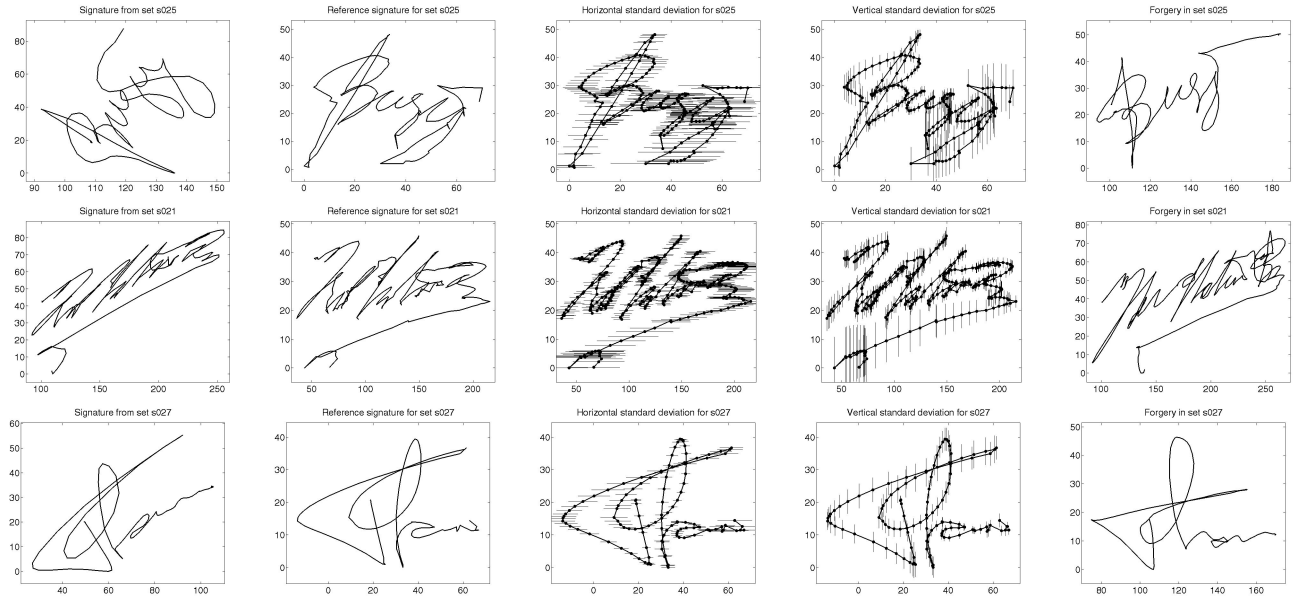


Fig. 11. **Prototype signatures**. The first column displays signatures captured with the visual tracker and the last column presents corresponding skilled forgeries. The central columns show the prototype signature obtained with the method described in Section 4.1. The local horizontal and vertical standard deviation are overimposed on the prototype in order to show the stability of the training algorithm. The prototypes of s025 and s021 appear "noisy," presenting a great variation of the local standard deviation. In contrast, the prototype of s027 is more stable, presenting a more constant value of local standard deviation. The "noiseness" of the prototypes is in part due to the method used to compute the prototype, in part due to the variability of the signatures in the training set, and in part due to the quantization effect produced by the discrete temporal and spatial sampling of the signatures.

4 System Implementation and Performance Evaluation

4.1 Training

During training, the system must *learn* a representation of the training set that yields minimum generalization error. DPM provides correspondence between two signatures; hence, acquisition noise could be reduced by averaging the signatures along the warping path; the mean signature would be a more robust representation of the class. In the case in which there are more than two examples in the training set, the mean signature could be computed if we have correspondence among all examples; however, there is no clear way of establishing this type of correspondence. In principle, one could think of performing the simultaneous matching of all the examples working on an N-dimensional tensor instead of a matrix. The disadvantage of this approach is that it is difficult to define the elementary distance associated to the arc joining two nodes of this tensor.

We propose a suboptimal training procedure: We perform only pairwise matching in order to find all pairwise mean signatures from the training set; the mean signature that yields minimum alignment cost with all the remaining signatures in the training set is chosen as a reference. Correspondence across all signatures in the training set is obtained by placing the signatures in correspondence with the reference signature; the prototype that represents the training set is computed as the mean of the aligned signatures. The warping path provided by DPM may be noninvertible; in other words, we could have many samples of one signature that are in correspondence with only one sample of the reference signature and vice versa. In order to compute the prototype, we take all points from all the signatures in the training set that are in correspondence with each particular sample of the reference; the mean of all these points provides the corresponding sample of the prototype; the standard deviation of all these samples is used later to compute the weighted correlation measure between signatures. Fig. 11 shows several examples of signatures, training prototypes, and corresponding skilled forgeries.

The prototype and the local standard deviation summarize the local statistics of the matching process among signatures in the training set. The individual residual distances between signatures in the training set and the prototype are collected in order to estimate global statistics of the alignment process. We extract the median and the median absolute deviation (MAD) of these distances in order to use them as normalizing factors for classification (see Section 4.5). We also use these distances to define the rejection threshold for the field test presented in the experiments. The threshold is user-dependent and is set equal to the mean of the distances plus five times the standard deviation of the distances.

4.2 Testing

Depending on the availability of data, a signature verification system can be validated with different types of forgeries (see [29]). The two most common types of forgeries are **Random Forgeries**, where the forger uses his own signature as the signature to be verified, and **Skilled Forgeries**, where the forger tries and practices imitating as closely as possible the static and dynamic information of the true signature. We used both types of forgeries in the experiments described in Section 5.

The verification performance of the system was evaluated both for each subject individually and for the whole data sets. Error trade-off curves were obtained with both random and skilled forgeries. The test set and the forgeries set for each subject were used to obtain the curves for skilled forgeries and to gather the field test error rates; the

test set and all signatures from other subjects were used to obtain the curves for random forgeries. The test sets were used to evaluate the recognition performance of the system.

Time duration of the signatures was used in order to screen gross forgeries and to speed up the experiments. The mean time duration of each subject's signature was extracted from the training set. Each signature under test was checked to have its time duration within three standard deviations of the mean duration for the corresponding subject. Signatures that fell outside this bound were rejected as forgeries while signatures within bounds were matched to the prototype using DPM. This gross screening reduced the number of DPM comparisons to be performed by approximately 40 percent and did not generate any false rejection.

4.3 Error Rates

Signature verification is a two-class pattern recognition problem, one class consisting of genuine signatures and the other consisting of forgeries. The performance of a verification system is generally evaluated with Type I and Type II error rates. The Type I error rate, or False Rejection Rate (FRR), measures the number of genuine signatures classified as forgeries as a function of the classification threshold. The Type II error rate, or False Acceptance Rate (FAR), evaluates the number of false signatures classified as real ones as a function of the classification threshold.

The curve of FAR as a function of FRR, using the classification threshold as a parameter, is called the *error trade-off curve*. In practice, this curve is often simplified into a number, the *equal error rate* (EER), that is the error rate at which the percentage of false accepts equals the percentage of false rejects. The equal error rate provides an estimate of the statistical performance of the algorithm, i.e., it provides an estimate of its generalization error. Fig. 12 shows typical curves of FRR and FAR as a function of the classification threshold and the corresponding error trade-off curve.

A different characterization of the verification performance of the system is provided by an actual field test, i.e., a test that resembles the actual deployment and implementation of the system in a real environment. A single rejection threshold is extracted from the training set and is kept fixed during the test; thus, the field test performance corresponds to a point in the error trade-off curve as it is shown in Experiment 5.

Signature recognition is an M-class pattern recognition problem in which the system has to select the class to which a given signature belongs. Signature recognition is a harder problem than signature verification since the system does not know in advance the alleged class of a given example. The recognition error rate, that measures the number of misclassified signatures, is the parameter used to evaluate the recognition performance of the system. Signature recognition results are presented on Experiment 6.

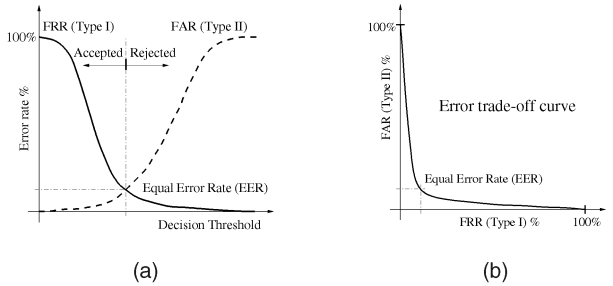


Fig. 12. **Graphical representation of the verification error rates**. (a) **FRR and FAR versus classification threshold**. Clearly, we can tradeoff one type of error for the other type of error: Accepting every signature implies a 0 percent FRR and a 100 percent FAR; rejecting every signature implies a 100 percent FRR and a 0 percent FAR. (b) **Error trade-off curve**. It provides the behavior of the algorithm for any operating regime and it is the best descriptor of the performance of the system.

4.4 Duplicated Examples

One common problem of many online systems for signature verification is the large number of examples required to both obtain a reliable model for a signature and assess the performance of the algorithm. We have to build a signature model that performs well in practice and we have to infer the generalization error of the system, all with very few examples. If we knew that the model that we are building should be invariant with respect to some transformation of the data, we could increase the number of examples in both the training and test sets by using *Duplicate Examples* (see [1]). The prototype signature is extracted from the training set as described on Section 4.1 using only the captured signatures that are assigned to the training set. Duplicated examples are use in training to estimate the in-sample statistics of the matching process; duplicated examples are used in testing to increase the size of the true signatures and skilled forgeries test sets.

In our experiments, we used two transformations to obtain duplicated examples. One of the transformations is a time origin translation since our system should be insensitive to the particular instant of time in which we started acquiring the signature. New signatures were obtained by resampling captured ones using a spline interpolation. The other transformation is an affine deformation of the signatures

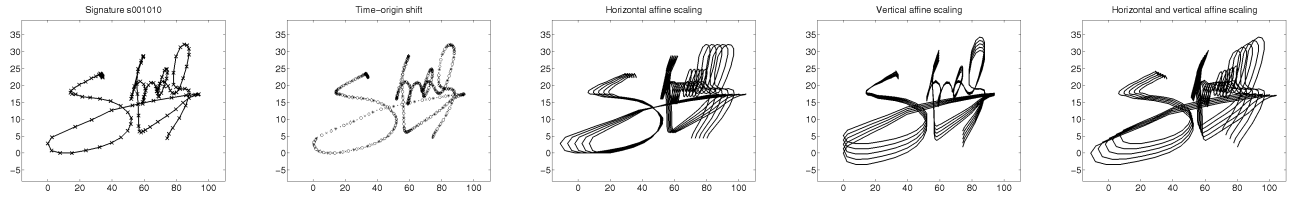


Fig. 13. **Duplicated examples**. The first plot shows the original signature captured with the visual tracker. The second plot displays the position of the new samples when performing time origin shifting. The third and fourth plots show the result of applying a horizontal and a vertical affine scaling to the original signature. The fifth plot presents the result of applying both scalings at the same time. The maximum and minimum values of scaling were estimated from the training set.

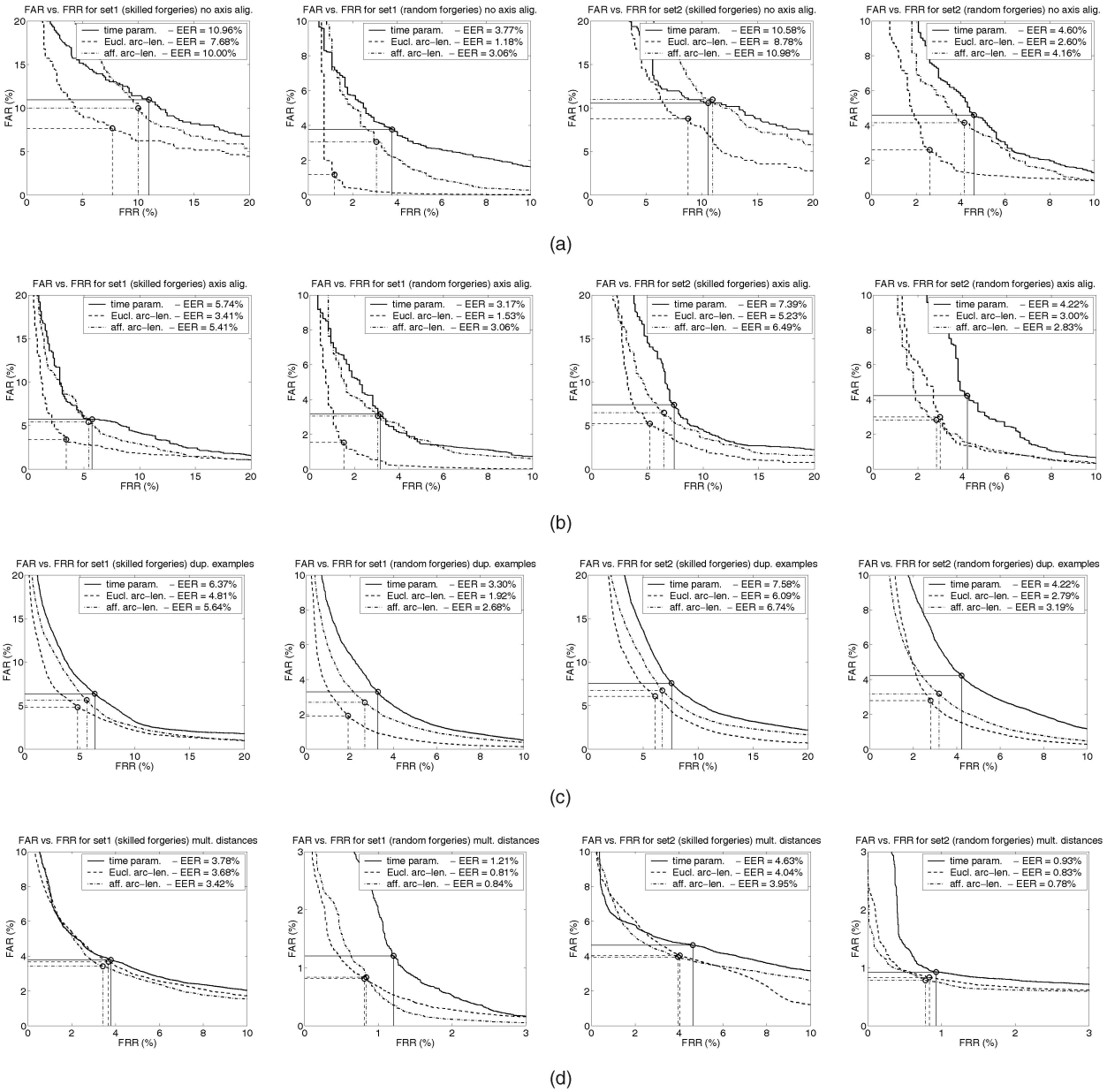


Fig. 14. (a) **Performance of different parameterizations**. Error trade-off curves for the described parameterizations. The error rates are high since no rotation normalization was used in the experiment. The curves present a staircase pattern since the number of test examples is only 10-15. Euclidean arc-length is the parameterization that performs best in all cases. (b) **Performance with rotation normalization**. Error trade-off curves produced with rotation normalization. The error rates are generally smaller than the ones of Experiment 1, with the only exception of the random forgeries test of Euclidean arc-length on set 2. Overall, rotation normalization improves the performance of the system, especially in the skilled forgeries case. Euclidean arc-length is the parameterization that presents the best performance on both sets. (c) **Performance with duplicated examples**. Error trade-off curves obtained using duplicated examples to generate extra signatures for the true signatures and skilled forgeries test sets. Rotation normalization is applied on the signatures in this experiment. The change in the error rates is expected since the experiments presented in (a) and (b) have a much coarser representation of the curves. The addition of duplicate examples adds a much finer representation of the FRR curves and of the FAR curve for skilled forgeries and, therefore, there is a better characterization of the error rates. Note that the error rates are, in most cases, bigger than the error rates of Experiment 2, but smaller than the error rates of Experiment 1. Euclidean arc-length is the parameterization that performs best for all cases. (d) **Performance with multiple distances**. Error trade-off curves obtained using multiple distances for classification. The error rates are smaller than the ones of Experiment 3, presenting a relative decrease of 35 percent on average for skilled forgeries and a relative decrease of 65 percent on average for random forgeries. Affine arc-length is the parameterization that achieves the best overall performance. The error rates obtained with random forgeries show that the system has good discrimination capabilities and could be used for signature recognition.

since the modification of the position of the camera from acquisition to acquisition introduces affine deformation in the data. Since all signatures are normalized for rotation, only affine scaling is used to generate duplicate examples. The horizontal and vertical scale ranges of each subject's signatures were extracted from the training set. Affine-scaling duplicated examples were generated by uniformly sampling the scale ranges. Fig. 13 shows a signature captured with the tracking system and a set of duplicated examples obtained by time origin shifting and affine scaling.

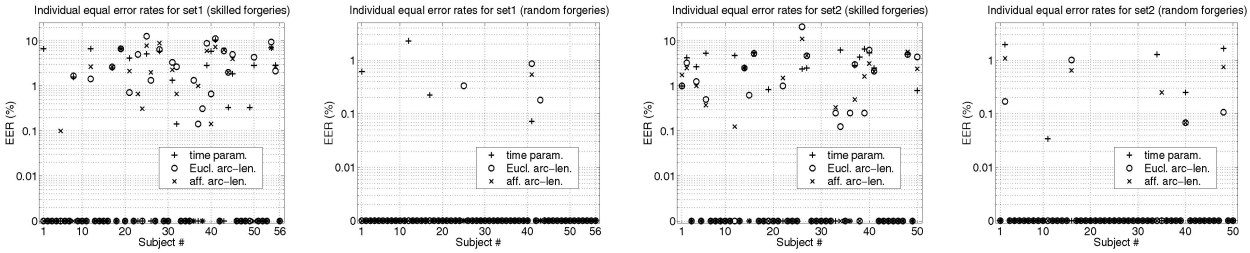


Fig. 15. **Individual performances**. Equal error rates for each subject obtained using multiple distances. The error rates are shown in a semilog scale for better visualization (The 0 percent error rate points have been plotted at 0.001 percent and the vertical axis has been relabeled accordingly). The subject with worst performance for skilled forgeries and affine arc-length (subject s026 from set 2) has an error rate of 11 percent. The skilled forgeries error rates for affine arc-length are below 1 percent for 75 percent of the subjects (only 15 subjects in set 1 and 12 subjects in set 2 have skilled forgeries error rates bigger than 1 percent). Only one subject in set 1 and five subjects in set 2 have random forgeries error rates different from 0 percent (only one subject in set 2 has random forgeries error rate bigger than 1percent).

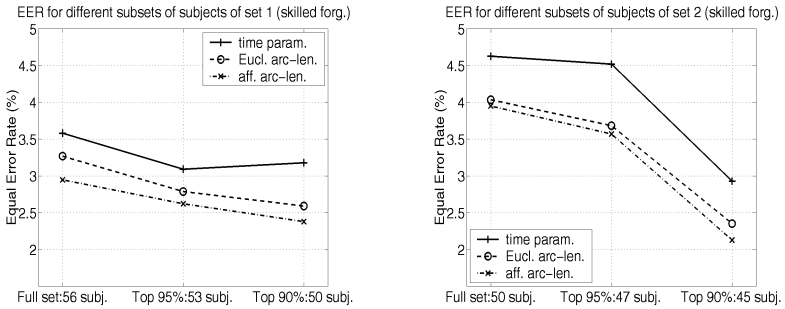


Fig. 16. **Variability of the verification EER for skilled forgeries**. This figure shows the variation in the EER of the system when the worst-performing subjects are excluded from the experiments. The plots show the EER obtained with the complete data sets, with the best 95 percent of the subjects, and with the best 90 percent of the subjects. An approximately 10 percent relative improvement in the performance of the system is achieved when excluding the three worst subjects from each data set; such a variation in performance is due to the relatively small size of the data sets and underlines the impact of the individual subject's performances on the overall error rate; bigger sets of signatures should be used in order to obtain performance measures that are less sensitive to individual error rates. Affine arc-length parameterization achieves the lowest error rates in all cases. Euclidean arc-length parameterization is always better than time parameterization.

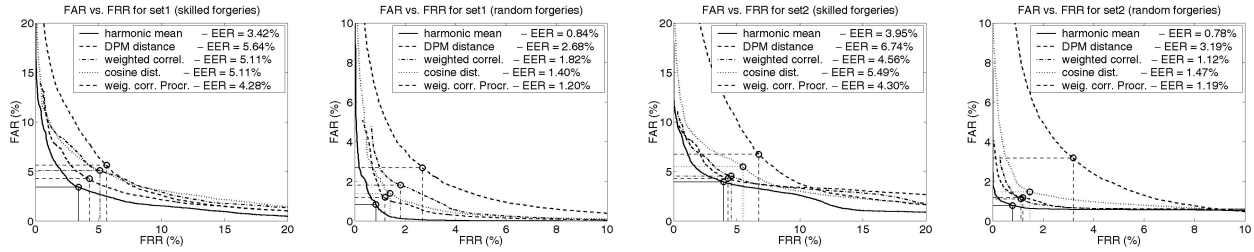


Fig. 17. **Performance of the various distances**. Error trade-off curves for the distances used Experiment 4 obtained with affine arc-length parameterization. The resulting distance after DPM of the prototype and the test signature is the individual distance that performs worst; the harmonic mean takes the best of each distance and merges it into a single parameter that achieves a better performance than any of the individual distances.

4.5 Distance Measures

The residual distance after DPM is the classification parameter used in the first three experiments presented on Section 5. However, many other similarity measures could be used once correspondence between samples has been established. The fourth experiment presents the performance of the system using four different measures: the resulting distance after DPM between the prototype and the test signature; the weighted correlation between the prototype and the test signature; the cosine distance [36] between the prototype and the test signature; and the weighted correlation between the prototype and the test signature after having performed a Procrustes transformation [6] on them. The weighting function represents the stability of each point of the prototype; it is computed as the reciprocal of the local standard deviation. The Procrustes transformation provides the optimal, in the least-squares sense, Euclidean transformation (translation, rotation, and scaling) between the prototype and the test signature, given the correspondence between their samples.

These similarity measures are boiled down to a single parameter by the use of the harmonic mean [26]. Given two distances d_1 and d_2, the *weighted harmonic mean* d of d_1 and d_2 is defined as $\frac{1}{d} = \frac{1}{\alpha_1 d_1} + \frac{1}{\alpha_2 d_2}$, where α_1 and α_2 are the weighting factors. The definition of weighted harmonic mean can be generalized to more than two distances as $\frac{1}{d} = \sum_i \frac{1}{\alpha_i d_i}$. The weighting factors normalize the distances to comparable values. In our system, we compute the median absolute deviation (MAD) of the distances in the training set and we use the reciprocals of the MAD as the corresponding weighting factors α_i.

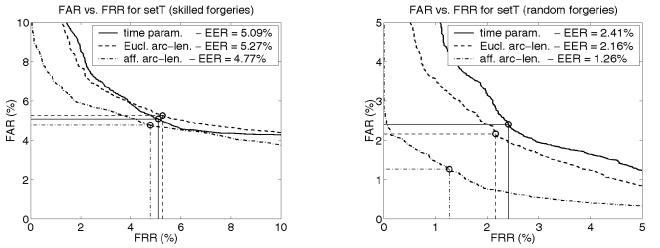


Fig. 18. **Performance for tablet-captured data**. Error trade-off curves obtained using multiple distances. Affine arc-length is the parameterization that provides the best performance, in agreement with the experimental results of Experiment 4. The size of the tablet-captured data set is 25 percent smaller than each of the visually-captured data sets; therefore, a bigger number of signatures is needed to perform a more accurate comparison of performance.

5 Experiments

The signatures from each subject were divided into a training and a test set. The first 10 captured signatures were allocated to the training set and the remaining ones were assigned to the test set. This division is similar to the actual operation of a real verification system in which the first signature examples would be used to train the system and the following ones would be used to test the system.

Several experiments are conducted in order to evaluate the verification performance of the parameterizations, the verification performance of rotation normalization, and the verification performance of the proposed distance measures. We present the verification results in Fig. 14 by using the error trade-off curves, showing only the portion of the curves that is most informative; we display the curves for skilled and random forgeries. The first two columns of the figure corresponds to set 1 and the last two columns corresponds to set 2; the equal error rate (EER) condition is indicated with a circle. A separate experiment shows the field test verification performance of the system. Another experiment presents the recognition performance of the system.

We show performance on set 1 and set 2 separately because set 1 was used throughout the design and tuning of the system, while set 2 was used only once during the final evaluation phase.

Experiment 1. Performance as a function of parameterization. Fig. 14a shows the error trade-off curves obtained with the parameterizations described above. No normalization for rotation was used in the experiment. The curves present a staircase pattern since no duplicated examples were used and, therefore, the number of test examples was small.

TABLE 1
Field Test Verification Performance

	time		Euclidean arc-length		affine arc-length	
	FAR	FRR	FAR	FRR	FAR	FRR
set 1	3.06%	5.33%	3.83%	3.42%	2.98%	4.34%
set 2	4.13%	6.15%	4.16%	3.85%	3.86%	4.16%
set T	4.35%	8.21%	5.07%	5.84%	4.51%	6.11%

The rejection threshold used for this experiment is automatically determined and user-dependent. The mean and the standard deviation of the distances between the prototype and the signatures in the training set define the threshold, that is set equal to the mean plus five times the standard deviation. The error rates shown in the table correspond to a single point of the error trade-off curves. The trade-off between false rejection and false acceptance rates is clearly noticeable when one compares the error rates presented in the table with the EER values shown on Figs. 14 and 18. Note that the error rates presented in the table were obtained using skilled forgeries.

TABLE 2
Recognition Error Rates

	time		Euclidean arc-length		affine arc-length	
	train	test	train	test	train	test
set 1	0.71%	4.00%	0.0%	1.41%	0.0%	1.06%
set 2	0.40%	4.99%	0.20%	0.799%	0.20%	0.299%
set T	0.27%	2.43%	0.27%	1.35%	0.27%	0.0%

Following previous experiments, signatures were normalized for rotation before reparameterization and multiple distances were used for classification. Only true signatures were used in the experiment. Affine arc-length is the parameterization that achieves best recognition performance.

Experiment 2. Performance using rotation normalization. Fig. 14b shows the error trade-off curves obtained with rotation normalization. The error rates are smaller than the ones presented in Experiment 1; thus, rotation normalization improves the performance of the system since most signatures on our data sets have a clearly defined axis of least inertia. Note that rotation normalization is performed before reparameterizing the signatures on either Euclidean or affine arc-length.

Experiment 3. Performance using duplicate examples. Fig. 14c shows the error trade-off curves obtained with duplicated examples. We generated duplicate examples both for training and testing, using the two transformations described in Section 4.4. We produced 19 duplicate examples (four examples of time-origin shift, five examples of horizontal affine scaling, five examples of vertical affine scaling, and five examples of horizontal and vertical affine scaling) for each captured signature or forgery. We did not use duplicate examples to generate random forgeries since we have enough data to reliably estimate the FAR. The error rates are generally slightly bigger than the error rates of Experiment 2 due to the increase in size of the test sets.

Experiment 4. Performance using different distance measures. Fig. 14d presents the error trade-off curves obtained using the similarity measures described in Section 4.5. The harmonic mean of the distances is used as the classification parameter. The best performance is achieved by affine arc-length parameterization of the signatures. The individual equal error rates for each subject are presented in Fig. 15. The variation of the equal error rates obtained by excluding the worst 5 percent and 10 percent of the subjects is shown in Fig. 16. The error trade-off curves for each of the distances obtained with affine arc-length are presented in Fig. 17 where we observe that the harmonic mean of the distances provides quite an improvement in performance.

Experiment 5. Performance using tablet-captured signatures. This experiment is designed to investigate whether the signatures captured with the camera-based interface provide as much information as the ones captured with a conventional tablet digitizer, from the signature verification point of view. A third data set of signatures was collected from 38 subjects using a Wacom digitizer (active area: 153.6×204.8 mm; resolution: 50 lpmm; accuracy: $\pm$ 0.25 mm; and maximum report rate: 205 points/second). Each subject provided 20 signatures, 10 to be used as the training set and 10 to be used as the test set. The signatures were collected by students of a class offered at Caltech over a period of three years, following a procedure similar to the one described on Section 2.1. Three signers provided 10 forgeries for each of the subjects in the data set.

TABLE 3
Comparison of Our Novel Camera-Based Biometrics with Other Camera-Based Biometrics Presented in the Literature

		Error Rates	Experimental Conditions
Recognition	signature	1.06%	106 subjects, 10 training examples per subject, 10-15 test examples per subject
	face	6.5% [28]	Best average results on the FERET database using a part of the FB probe, 200 training examples and 200 testing examples, the testing and training examples were taken under the same conditions but have different facial expression
Verification	signature	0.84%	106 subjects, 10 training examples per subject (200 with dup. ex.), 10-15 test examples per subject (200-300 with dup. ex.)
	face	1.0% [32]	FERET database using the FB probe, 1196 training examples and 1195 testing examples, the testing and training examples were taken under the same conditions but have different facial expression
	hand shape	2.75% [12]	353 images from 53 people (2-15 images per subject), all images used to get ROC curves

The experiment was conducted following the same experimental protocol used on Experiment 4, i.e., the full signing trajectory (both pen-down and pen-up strokes) was used, rotation normalization was applied prior to reparameterization and matching, duplicated examples were added to the true signatures and skilled forgeries test sets, and multiple distances were used for classification. Only the position signal provided by the tablet was used in the experiment; the pressure signal was discarded in order to make a fair comparison of the performances obtained with the two types of capture systems. Fig. 18 presents the error trade-off curves that show that affine arc-length is the parameterization that achieves the best performance.

Experiment 6. Signature verification field test. Table 1 shows the performance of the system for a field test experiment performed with the signature databases. The decision threshold used to classify the signatures was extracted from the training set in a user-dependent fashion as described in Section 4.1. The error rates obtained in this experiment account for just a single point of the error trade-off curves shown in Figs. 14d and 18.

Experiment 7. Signature recognition performance. Table 2 presents the recognition performance of the system for the three described data sets. We compute the recognition error rates on both the training and the test sets. In agreement with the results of previous experiments, affine arc-length is the parameterization that achieves best recognition performance.

5.1 Discussion

A comparison with other camera-based biometrics systems that are similar to our system in terms of experimental set-up,[3] such as face recognition and hand shape recognition, is presented on Table 3. From these data, it would appear that visual signature recognition is as reliable or more reliable than other camera-based biometrics.

The performance obtained with both Euclidean and affine arc-length parameterizations is shown to be always superior than the performance obtained with time parameterization. Fig. 19 shows examples of signatures from the visually-captured data sets for which the algorithm has a verification error rate greater than 5 percent.

The verification error rates obtained with tablet-captured signatures are bigger than the error rates of visually-captured ones. The tablet-captured database has fewer examples than the visually-captured ones; hence, data sets of similar size should be used to provide a fairer comparison. In any case, the error rates achieved by our system are comparable[4] to the best signature verification performances presented in the literature [7], [8], [11], [16], [18], [19], [26], [27], [29], [34], [39].

The error rates are quite encouraging, providing evidence that the system is a good candidate for camera-based user identification. A more reliable estimate of the generalization error should be computed using bigger data sets, e.g., a data set of more than 2,000 subjects is needed to achieve a sensitivity lower than 1 percent using 20 test signatures per subject. Such scale of testing is justified in the context of an industrial R&D project.

6 Conclusions and Further Work

We have presented the performance of a novel vision-based biometric technique for personal identification. The system does not require any dedicated hardware, unlike fingerprint verification, iris, or retina scanning systems, but rather a conventional camera which may be already in use for security, video conferencing, or other applications. We have shown recognition error rates of 1 percent, a result that indicates that the algorithm is quite able to discriminate whether a signature belongs to a certain class (or, in other words, to a certain subject). We have also presented verification error rates that are better than 3.95 percent for skilled forgeries and better than 0.84 percent for random forgeries. Such performance is comparable if not better than

3. Note that comparison of performances across systems that use different data (faces, hand shapes, signatures, etc.) acquired with the same sensor (a camera) is quite difficult and unreliable. We present the performance of various camera-based biometrics systems in order to have a qualitative comparison.

4. A comparison of various signature verification and recognition systems should use only one data set of signatures for performance evaluation. Unfortunately, there is no publicly available database of signatures for that purpose and, therefore, the comparison of the performance of our system with the results presented in the literature could only be made at a qualitative level. The data sets of signatures that have been used in our experiments are available for academic use in our Web site http://www.vision.caltech.edu/mariomu/research/data/.

Fig. 19. **Worst verification performance cases.** Subjects with verification error rate greater than 5 percent. The three first subjects are from set 1 and the three last are from set 2. The plots show a visually-acquired signature, the corresponding prototype, a falsely rejected signature, and a falsely accepted skilled forgery. In the first four cases, high error rates correspond to signatures that are very simple and therefore easy to forge. The last two cases correspond to a subject (s026) whose signatures present a high degree of variability and to a subject (s048) for whom rotation normalization fails.

the performances of other similar camera-based recognition systems presented in the literature.

The experimental results show that the parameterization of the signatures is quite critical for achieving low error rates. The best performance is obtained by parameterizing the signatures with affine arc-length, using the harmonic mean of several similarity measures as the classification parameter. From the results of the experiments, it can be inferred that shape similarity and causality in the generation of the signature (arc-length parameterization) are more important than matching the dynamics of signing (time parameterization). The causality of the signature, i.e., the order in which parts are produced, is still valuable and is used in the DPM paradigm since the correspondence between signatures is established sequentially. In fact, causality is the added information that our online system is using to outperform systems that do comparison from still pictures of signatures since causality provides the sequence in which matching between signatures would be performed. This information is vital in establishing correspondence between strokes that cross each other and is difficult to extract from a picture of a signature.

We have also shown that the use of duplicate examples provides a better estimation of the generalization error, given that we know that our algorithm has to be invariant to a certain transformation. In our experiments, we used only time origin shifting and horizontal and vertical affine scaling as the transformation class. A full affine transformation could be used to generate duplicate examples provided that a reasonable range of the parameters of this transformation could be estimated from the training data.

The signature verification and recognition algorithms could be made more robust by adding more global

descriptors of the signatures that would allow the system to discard coarse forgeries. One problem that was not addressed in the present scheme is dealing with dramatic changes in scale and it is one of the areas of further research, as well as the development of better similarity measures.

All biometric techniques have false accepts generated by the imperfections of the classification method or by errors in the acquisition device. However, behavioral biometric techniques, such as signature verification, compared with physiological biometric techniques, such as fingerprint verification or face recognition, have the additional disadvantage that a forger with enough information about the true signature and with enough training could deceive the algorithm. This weakness is inherent to all behavioral biometrics and may or may not be important depending on the particular application. In our system, the pen-up portions of the signatures decrease the risk of accepting skilled forgeries since they are not available to the forger.

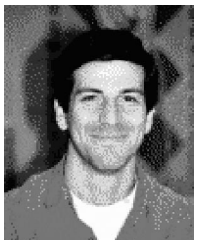
Signature verification and recognition could be employed to replace the use of computer passwords; in this case, the daily use of the system could make people sign more consistently and could provide them with a number that quantifies the variability of the signature. It would also be possible to have a system in which the user signs with an ink-less pen, leaving no trace of the signature in order to prevent possible forgers from knowing it.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support from the US National Science Foundation Engineering Research Center on Neuromorphic Systems Engineering at Caltech (NSF) Cooperative Agreement No. EEC-9402726). They would also like to express their gratitude to all the subjects that collaborated in the experiments by providing their time and their signatures to build the example databases.

REFERENCES

[1] Y.S. Abu-Mostafa, "Hints," *Neural Computation,* vol. 7, pp. 639-671, 1995.
[2] A.M. Bruckstein, R. Holt, A. Netravali, T. Richardson, "Invariant Signatures for Planar Shape Recognition under Partial Occlusion," *CVGIP: Image Understanding,* vol. 58, no. 1, pp. 49-65, 1993.
[3] J.P. Campbell, "Speaker Recognition: A Tutorial," *Proc. IEEE,* vol. 85, pp. 1437-1462, 1997.
[4] R. Chellappa, C. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey," *Proc. IEEE,* vol. 83, no. 5, pp. 705-740, 1995.
[5] J.G. Daugman, "High Confidence Visual Recognition of Persons by a Test of a Statistical Independence," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1148-1161, Nov. 1993.
[6] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis.* John Wiley & Sons, 1998.
[7] M.C. Fairhurst, "Signature Verification Revisited: Promoting Practical Exploitation of Biometric Technology," *Electronics and Comm. Eng. J.,* pp. 273-280, 1997.
[8] T. Hastie, E. Kishon, M. Clark, and J. Fan, "A Model for Signature Verification," *Proc. IEEE Conf. Systems, Man, and Cybernetics,* pp. 191-196, 1991.
[9] R. Hill, "Retina Identification," *Biometrics: Personal Identification in Networked Society,* A. Jain, R. Bolle, and S. Pankanti, eds., Kluwer Academic, 1999.
[10] B.K.P. Horn, *Robot Vision.* The MIT Press, 1986.
[11] K. Huang and H. Yan, "On-Line Signature Verification Based on Dynamic Segmentation and Global and Local Matching," *Optical Eng.,* vol. 34, no. 12, pp. 3480-3487, 1995.
[12] A.K. Jain and N. Duta, "Deformable Matching of Hand Shapes for Verification," *Proc. IEEE Int'l Conf. Image Processing,* 1999.
[13] A.K. Jain, L. Hong, S. Pankanti, and E. Bolle, "An Identity-Authentication System Using Fingerprints," *Proc. IEEE,* vol. 85, no. 9, pp. 1365-1388, 1997.
[14] E. Kreyszig, *Differential Geometry.* Dover, 1991.
[15] F. Lacquaniti, C. Terzuolo, and P. Viviani, "The Law Relating the Kinematic and Figural Aspects of Drawing Movements," *Acta Psychologica,* vol. 54, pp. 115-130, 1983.
[16] F. Leclerc and R. Plamondon, "Automatic Signature Verification," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 8 no. 3, pp. 643-660, 1994.
[17] M.M. Lipschutz, *Differential Geometry.* Mc Graw-Hill, 1969.
[18] G. Lorette and R. Plamondon, "Dynamic Approaches to Handwritten Signature Verification," *Computer Processing of Handwriting,* pp. 21-47, 1990.
[19] R. Martens and L. Claesen, "On-Line Signature Verification by Dynamic Time-Warping," *Proc. 13th Int'l Conf. Pattern Recognition,* pp. 38-42, 1996.
[20] M.E. Munich, "Visual Input for Pen-Based Computers," PhD thesis, Calif. Inst. of Technology, Pasadena, Jan. 2000. http://www.vision.caltech.edu/mariomu/thesis/thesis.ps.gz.
[21] M.E. Munich and P. Perona, "Visual Input for Pen-Based Computers," *Proc. 13th Int'l Conf. Pattern Recognition,* 1996.
[22] M.E. Munich and P. Perona, "Visual-Based ID Verification by Signature Tracking," *Proc. Second Int'l Conf. Audio- and Video-Based Person Authentication,* 1999.
[23] M.E. Munich and P. Perona, "Visual Signature Verification Using Affine Arc-Length," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 180-186, 1999.
[24] M.E. Munich and P. Perona, "Visual Input for Pen-Based Computers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 313-328, Mar. 2001.
[25] M.E. Munich and P. Perona, "Apparatus and Method for Tracking Handwriting from Visual Input," US Patent 6,044,165 filed 6/15/1995, granted Mar. 2000.
[26] V.S. Nalwa, "Automatic On-Line Signature Verification," *Proc. IEEE,* vol. 85, no. 2, pp. 215-239, 1997.
[27] M. Parizeau and R. Plamondon, "A Comparative Analysis of Regional Correlation, Dynamical Time Warping and Skeletal Tree Matching for Signature Verification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 710-717, July 1990.
[28] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The Feret Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 10, pp. 1090-1104, Oct. 2000.
[29] R. Plamondon and G. Lorette, "Automatic Signature Verification and Writer Identification, the State of the Art," *Pattern Recognition,* vol. 22, no. 2, pp. 107-131, 1989.
[30] F.E. Pollick and G. Sapiro, "Constant Affine Velocity Predicts the 1/3 Power Law of Planar Motion Perception and Generation," *Vision Research,* vol. 37, no. 3, pp. 347-353, 1997.
[31] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, Inc., 1993.
[32] S.A. Rizvi, P.J. Phillips, and H. Moon, "The Feret Verification Testing Protocol for Face Recognition Algorithms," *Proc. Int'l Conf. Automatic Face- and Gesture-Recognition,* pp. 48-53, 1998.
[33] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. Acoustics, Speech, Signal Processing,* vol. 26, no. 1, pp. 43-49, 1978.
[34] Y. Sato and K. Kogure, "On-Line Signature Verification Based on Shape, Motion and Writing Pressure," *Proc. Sixth Int'l Conf. Pattern Recognition,* pp. 823-826, 1982.
[35] B. Serra and M. Berthod, "Subpixel Contour Matching Using Continuous Dynamic Programming," *Proc. IEEE Conf. Computer SC Computer Vision and Pattern Recognition,* pp. 202-207, 1994.
[36] C.G. Small, *The Statistical Theory of Shape.* Springer-Verlag, 1996.
[37] C.C. Tappert, C.Y. Suen, and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, pp. 787-808, 1990.
[38] P. Viviani and G. McCollum, "The Relation between Linear Extent and Velocity in Drawings Movements," *Neuroscience,* vol. 10, no. 1, pp. 211-218, 1983.
[39] B. Wirtz, "Stroke-Based Time Warping for Signature Verification," *Proc. Int'l Conf. Document Analysis and Recognition,* pp. 179-182, 1995.



Mario E. Munich received the degree of Electronic Engineer (with honors) from the National University of Rosario, Argentina, in 1990, and the MS and PhD degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1994 and 2000, respectively. After graduation, he joined Vocalpoint Technologies, San Francisco, California, where he developed noise-robust speech recognition engines. Currently, he is working at Evolution Robotics, Pasadena, where he leads the Human-Robot Interface group. His research interests include machine vision, handwriting and speech recognition, human-computer interfaces, pattern recognition, machine learning, and affective computing. He is member of the IEEE, the IEEE Computer Society, and the ACM.

Pietro Perona received the degree of Dottore in Ingegneria Elettronica from the University of Padova, Italy in 1985, and the PhD degree in electrical engineering and computer science from the University of California, Berkeley, in 1990. He is currently a professor of electrical engineering and computation and neural systems, as well as the Director of the National Science Foundation Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at the California Institute of Technology, Pasadena. His research interests include both human and machine vision. His current activity is focused on visual recognition and the perception of 3D shape. He has also worked on the use of diffusive PDE's for image processing (anisotropic diffusion) and filtering techniques for early vision and modeling of human vision. He is member of the IEEE and the IEEE Computer Society.