# Camera-Based ID Verification by Signature Tracking

Mario E. Munich[1] and Pietro Perona[1,2]

[1] California Institute of Technology 136-93
Pasadena, CA 91125
{mariomu,perona}@vision.caltech.edu
[2] Università di Padova, Italia

**Abstract.** A number of vision-based biometric techniques have been proposed in the past for personal identification. We present a novel one based on visual capturing of signatures. This paper describes a system based on correlation and recursive prediction methods that can track the tip of the pen in real time, with sufficient spatio-temporal resolution and accuracy to enable signature verification. Several examples and the performance of the system are shown.

## 1 Introduction and Motivation

A number of biometric techniques have been proposed for personal identification in the past. Among the vision-based ones, we can mention face recognition [21], [22], [23], fingerprint recognition [6], iris scanning [4] and retina scanning. Voice recognition or signature verification are the most widely known among the non-vision based ones. Signature verification requires the use of electronic tablets or digitizers for on-line capturing and optical scanners for off-line conversion [20]. These interfaces have the drawback that they are bulky and complicated to use, increasing the complexity of the whole identification system. Cameras, on the other hand, are much smaller and simple to handle, and are becoming ubiquitous in the current computer environment. This paper presents a visual interface that can be built using video technology and computer vision techniques in order to capture signatures to be used for personal identification. This vision-based personal identification system could be integrated as a component of a complete visual pen-based computer environment. Some related work can be found in [3], [11].

Handwriting recognition is still an open problem, even though it has been extensively studied for many years. Signature verification is a reduced problem that still poses a real challenge for researchers. The literature on signature verification is quite extensive (see [1], [9], [16] for very comprehensive surveys) and shows two main areas of research, off-line and on-line systems. Off-line systems deal with a static image of the signature, i.e. the result of the action of signing while on-line systems work on the dynamic process of generating the signature, i.e. the action of signing itself. The system proposed in this paper falls within

the category of on-line systems since the visual tracker of handwriting captures the timing information in the generation of the signature.

Section 2 describes the system. Section 3 presents the experimental setup and the results of experiments. The final section summarizes the results and discusses future work.

## 2 Overview of the System

Figure 1 shows a basic block diagram of the system and the experimental setup. The preprocessing stage performs the initialization of the algorithm, i.e. it finds the position of the pen on the first frame of the sequence and selects the template corresponding to the pen tip to be tracked. In subsequent frames, the preprocessing stage performs one function only: it cuts a piece of image around the predicted position of the pen tip and feeds it into the next block. The pen tip tracker has the task of finding the position of the pen tip on each frame of the sequence. A Kalman filter predicts the position of the tip in the next frame based on an estimate of the current position, velocity and acceleration of the pen. Finally, the last block of our system performs signature verification. Section 2.1 describes in more detail the handwriting acquisition component of our system and section 2.2 describes the algorithm used for signature verification.

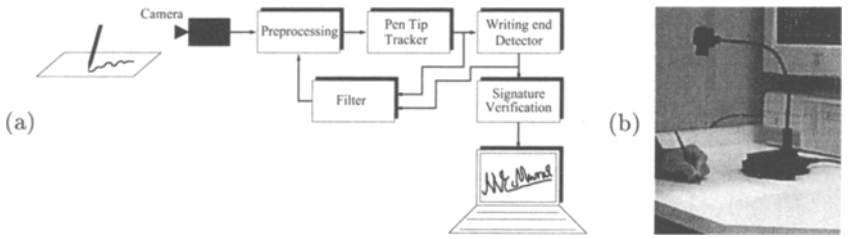


**Fig. 1.** (a) Block Diagram of the system. The camera feeds a sequence of images to the preprocessing stage. This block initializes the algorithm and selects the template to perform the tracking of the pen tip. The tip tracker obtains the position of the pen tip in each image of the sequence. The filter predicts the position of the pen tip in the next image. Finally, the last block of our system performs signature verification. (b) Experimental setup. The camera is looking at a person signing on a piece of paper.

### 2.1 Handwriting Acquisition

**Initialization.** The first problem to solve is locating the position of the pen tip in the first image of the sequence and selecting the kernel to be tracked. There are three possible scenarios: 1) In a batch analysis the tracker is initialized manually by mouse-clicking on the pen tip in the first frame. 2) The user writes with a pen that is familiar to the system. 3) An unknown pen is used.

The familiar-pen case is easy to handle: the system may use a previously stored template representing the pen tip and detect its position in the image by

correlation. There are a number of methods to initialize the system when the pen is unknown [11].

**Tracking the Pen.** The second block of the system has the task of finding the position of the pen tip in the current frame of the sequence. The solution of this task is well known in the optimal signal detection literature. The optimal detector is a filter matched to the signal (in our case a segment of the image) and the most likely position of the pen is given by the best match between the signal and the optimal detector.

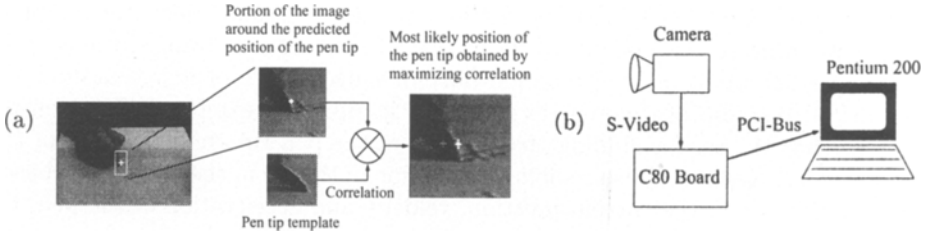


**Fig. 2.** (a) Given the predicted location of the pen tip in the current frame, the most likely position of the pen is obtained by finding the place that has maximum correlation with the previously stored template of the pen tip. (b) System configuration: The hardware architecture comprises a commercial camera, a video board, and a Pentium 200.

Assuming that the changes in size and orientation of the pen tip during the sequence 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 kernel and the image neighborhood, as shown in figure 2(a).

**Filtering.** Using the output of the correlation-based tracker, the filter predicts the position of the pen tip in the next frame based on an estimate of the position, velocity and acceleration of the pen tip in the current frame. This filter improves the performance of the system since it allows us to reduce the size of the neighborhood used to calculate correlation. The measurements are acquired faster and the measured trajectory is smoother due to the noise rejection of the filter. A Kalman Filter [2], [7], [8] is a suitable recursive estimation scheme for this problem. We assumed a simple random walk model for the acceleration of the pen tip on the image plane. The model is given by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{v}(t) \\ \dot{\mathbf{v}}(t) = \mathbf{a}(t) \\ \dot{\mathbf{a}}(t) = \mathbf{n}_a(t) \\ \mathbf{y}(t) = \mathbf{x}(t) + \mathbf{n}_y(t) \end{cases} \tag{1}$$

where $\mathbf{x}(t)$, $\mathbf{v}(t)$ and $\mathbf{a}(t)$ are the two dimensional-components of the position, velocity and acceleration of the tracked point, and $\mathbf{n}_a(t)$ is additive zero-mean,

Gaussian, white noise. The state of the filter $\mathbf{X}(t)$ includes three 2-dimensional variables, $\mathbf{x}(t)$, $\mathbf{v}(t)$ and $\mathbf{a}(t)$. This second order model is appropriate to describe the dynamics of a point object moving on a plane. The output of the model $\mathbf{y}(t)$ is the estimated position of the pen tip.

**Real-time Implementation.** Figure 2(b) shows a block diagram of the implementation hardware, which consists of a video camera, a video processing board, and a Pentium 200 PC. The camera is a commercial Flexcam ID, manufactured by Videolabs, equipped with manual gain control. It has a resolution of 480x640 pixels per interlaced image. A TI TMS3020C80 based signal and image processing board was used to perform the image capturing. The input camera image is digitized by the board and the even and odd fields of the image are separated for future processing. The even field of the image is transferred via DMA to the memory of the host PC. All further computations are performed with the Pentium 200. We achieved a total processing time of 31ms per frame.

### 2.2 Signature Verification

**Preliminaries.** We consider the algorithms presented in references [1], [9], [16] as well as in the work of V. Nalwa [12] in order to choose one of them to be implemented and included in our system. We decided to explore the performance of Dynamic Time Warping (DTW) applied to this problem. This algorithm was initially proposed in the field of Speech Recognition by Sakoe and Chiba [18] and it is described in full extent in the book of Rabiner and Juang [17]. In the area of signature verification, Sato and Kogure [19] used DTW to align signature shapes, Parizeau and Plamondon [13] compared the performance of DTW with regional correlation and skeletal tree matching for signature verification, Huang and Yan [5] applied DTW to align signature strokes and Nalwa [12] employed a similar dynamic programming technique to align different characteristic functions of the signature parameterized along its arc length.

Sato and Kogure [19] proposed to use DTW 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 DTW in order to compute the alignment of the pressure function and a measure of the difference in writing motion. Finally, they perform the classification based on the residual distance between shapes after time alignment, the residual distance between pressure functions and the distance between writing motions.

Parizeau and Plamondon [13] evaluated the use of DTW for signature verification by aligning either $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$, $a_x(t)$ or $a_y(t)$. In their work, they used the complete signing trajectories, i.e., pen down and pen up strokes.

Huang and Yan [5] presented the use of DTW for matching signature strokes by finding a warp path that minimizes the cost of aligning the shape, velocities and accelerations of the individual strokes at the same time. Pen up strokes are considered in the preprocessing phase of their algorithm, in order to be merged with the pen down strokes.

Nalwa [12] parameterized the pen down strokes of the signature along its arc length and then compute a number of characteristic functions such as coordinates of the center of mass, torque, and moments of inertia using a sliding computational window and a moving coordinate frame. He performed a simultaneously dynamic warping over arc length of all these characteristic functions for the two signatures under comparison. A measure of the similarity of the signatures is used for classification.

Our implementation of DTW for signature verification attempts to perform the best time alignment of the 2D shape of the signatures, i.e., we find the time warping function that has the minimum cost of aligning the planar curves that represent signatures. The visual tracker does not have the capability of detecting the positions in which the pen is up and not writing, so we used the full signing trajectory in our experiments. We note that the pen up strokes drawn by each subject were as consistent as the pen down strokes. This observation agrees with the belief [1] that signatures are produced as a ballistic or reflex action, without any visual feedback involved. We do not perform any type of normalization on the signatures since we consider that users are very consistent on their style of signing, they write their signatures with a similar slant, in a similar amount of time, with similar dimensions and with a similar motion. As presented, our DTW algorithm for signature verification is different from all the mentioned previous work although it shares with them some characteristics.

Figure 3 shows an example of dynamic time warping applied to align the 2D shape of two signatures. The first column shows $x(t)$ before and after time warping and the second column shows $y(t)$ before and after alignment. The upper plot of the third column shows the two signatures under comparison and the lower plot of the third column shows the alignment path. We note that the alignment is quite good regardless of the differences in the shapes of $x(t)$ and $y(t)$. The remaining mismatch between these signals accounts for the differences in shape of the signatures.

**Description of the Algorithm.** Dynamic time warping is a dynamic programming technique that performs alignment of two different examples of a signal. Given the two examples $X = (\mathbf{x}(1), \mathbf{x}(2), \cdots \mathbf{x}(T_x))$ and $Y = (\mathbf{y}(1), \mathbf{y}(2), \cdots \mathbf{y}(T_y))$, and a distance function $d(\mathbf{x}(t_x), \mathbf{y}(t_y))$, we can define the total dissimilarity between $X$ and $Y$ as follows

$$\mathbf{D}(X, Y) = \sum_{t=1}^{T} d(\mathbf{x}(t), \mathbf{y}(t)) \tag{2}$$

where the above summation is performed over a common time axis. In the simplest case of time alignment, i.e., linear time alignment, the above summation would be computed over one of the individual time axis $t_x$ or $t_y$ that would be related to each other by $t_x = \frac{T_x}{T_y} t_y$.

Linear time alignment assumes that the rates of production of the signals are constant and proportional to the duration of the signals. It corresponds to

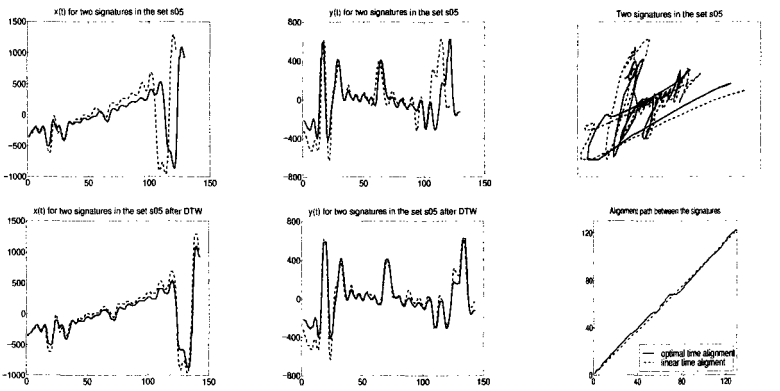


**Fig. 3.** Example of dynamic time warping applied to align the 2D shape of two realizations of the same signature. The first column shows $x(t)$ before and after time warping, the second column shows $y(t)$ before and after alignment. The upper plot of the third column shows the two examples of the signature. The lower plot of the third column shows the optimal time alignment path compared with a linear time alignment path.

the diagonal straight line showed in the $(t_x, t_y)$ plane of figure 4(a). In the general setting, signatures are generated at a variable rate showing local motion changes. Therefore, we need to obtain two warping functions $\phi_x$ and $\phi_y$ that relate the time indices of the examples with a common time axis. The accumulated distortion between $X$ and $Y$ in this case is

$$\mathbf{D}_\phi(X,Y) = \sum_{t=1}^{T} d(\mathbf{x}(\phi_x(t)), \mathbf{y}(\phi_y(t))) \tag{3}$$

with $\phi = (\phi_x, \phi_y)$. These warping functions define a warping path in the $(t_x, t_y)$ plane of figure 4(a) (solid line). Of all the possible warping functions that could be chosen, we would look for the one that minimize the distortion between $X$ and $Y$:

$$\mathbf{D}(X,Y) = \min_{\phi} \sum_{t=1}^{T} d(\mathbf{x}(\phi_x(t)), \mathbf{y}(\phi_y(t))) \tag{4}$$

The solution to this problem is obtained with a dynamic programming algorithm that relies on the following recursion:

$$\mathbf{D}(t_x, t_y) = \min_{(t'_x, t'_y)} \{\mathbf{D}(t'_x, t'_y) + c((t'_x, t'_y), (t_x, t_y))\} \tag{5}$$

where $\mathbf{D}(t'_x, t'_y)$ is the cumulated cost of the best path ending at node $(t'_x, t'_y)$, and $c((t'_x, t'_y), (t_x, t_y))$ is the elementary cost of the arc joining nodes $(t'_x, t'_y)$ and $(t_x, t_y)$ in the warping plane.

From the above recursion equation, we can see that a simple algorithm yields the optimal path: for each node on the discrete plane $(t_x, t_y)$, the minimum cumulated cost is computed with equation (5). The node that provides the minimum cost is stored in memory. The cumulative cost is computed serially starting from

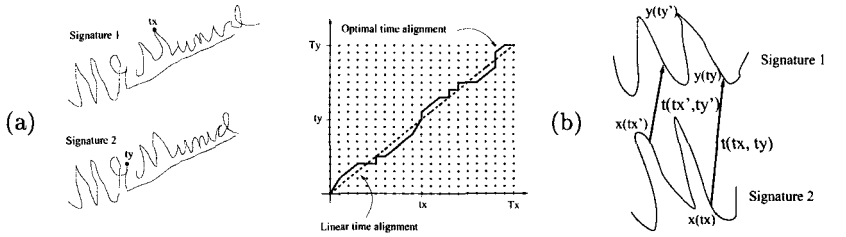


**Fig. 4.** (a) Two different realizations of the same signature and the discrete $(t_x, t_y)$ plane in which the time alignment is performed. The warping path in dashed line corresponds to linear time alignment while the warping path in solid line corresponds to a possible solution for the optimal time alignment. (b) Translation vectors between two pairs of points in each signature.

the first column until the last column in the discrete plane $(t_x, t_y)$. Finally, the last column is searched for the node with minimum cost and then the optimum warping path is found by backtracking the stored nodes.

For the time alignment process to be meaningful in terms of time normalization for different realizations of a signature, some constraints on the warping functions are necessary. Unconstrained minimization in equation (5) may conceivably result in a near-perfect match between two different signatures, thus making the comparison meaningless for recognition purposes. Typical time warping constraints that are considered reasonable for time alignment include endpoint constraints, monotonicity conditions, local continuity constraints, global path constraints and slope weighting (see [17] for a more extensive treatment of the subject).

We enforce only a few simple constraints. First, we only allow monotonic paths to be explored, so if point $t_x$ is matched with point $t_y$, then point $t_x + 1$ can only be matched with a point after point $t_y$. Second, we only allow point $(t_x, t_y)$ to be reached from points $(t_x - 1, t_y)$, $(t_x - 1, t_y - 1)$ and $(t_x, t_y - 1)$. Third, we require that the warping path start at point $(0, 0)$ and end at point $(T_x, T_y)$. Finally, we constrain the number of points $t_y$ that can be explored for each point $t_x$ in minimizing equation (5). With all these constraints, the algorithm can be summarized as follows:

1. Initialization: $\mathbf{D}(0,0) = 0$
2. Recursion: for $1 \leq t_x \leq T_x$, $1 \leq t_y \leq T_y$, such that $t_x$ and $t_y$ stay within the allowed grid, compute:

$$\mathbf{D}(t_x, t_y) = min \begin{cases} \mathbf{D}(t_x - 1, t_y) + c((t_x - 1, t_y), (t_x, t_y)) \\ \mathbf{D}(t_x - 1, t_y - 1) + c((t_x - 1, t_y - 1), (t_x, t_y)) \\ \mathbf{D}(t_x, t_y - 1) + c((t_x, t_y - 1), (t_x, t_y)) \end{cases} \quad (6)$$

3. Termination: $\mathbf{D}(X, Y) = \mathbf{D}(T_x, T_y)$

The only missing element in the algorithm is the specification of the elementary cost of the arc joining nodes $(t'_x, t'_y)$ and $(t_x, t_y)$, $c((t'_x, t'_y), (t_x, t_y))$. Assume that the point $\mathbf{x}(t'_x)$ in the first signature is matched with point $\mathbf{y}(t'_y)$ in the

second signature, and that point $\mathbf{x}(t_x)$ in the first signature is matched with point $\mathbf{y}(t_y)$ in the second signature. The translation vectors associated with each match are $\mathbf{t}(t'_x, t'_y) = \mathbf{x}(t'_x) - \mathbf{y}(t'_y)$ and $\mathbf{t}(t_x, t_y) = \mathbf{x}(t_x) - \mathbf{y}(t_y)$. The error between matches is $d((t'_x, t'_y), (t_x, t_y)) = \mathbf{t}(t_x, t_y) - \mathbf{t}(t'_x, t'_y)$. We choose the elementary cost to be the Euclidean norm of the error between matches: $c((t'_x, t'_y), (t_x, t_y)) = \|d((t'_x, t'_y), (t_x, t_y))\|^2$, as shown in figure 4(b). This cost function tends to give minimum elementary cost to pair of places in the warping plane that corresponds to pairs of points in the signatures that have the same displacement vectors, providing zero cost in the case in which one signature is a translated version of the other.

## 3 Experiments

The performance of our tracking system has been presented in references [10], [11]. In this paper, we focus our experiments on the results of the automatic personal identification. In section 3.1 we present the performance of the time warping algorithm tested on a database of signatures collected with a tablet digitizer. In section 3.2 we show the results of the visual identification system.

### 3.1 Performance of the Signature Verification Algorithm

We collected signatures from 38 subjects. Each of them was asked to provide 20 signatures, 10 of them to be used as the training set and the other 10 to be used as the test set. The test set allows us to evaluate the Type I error (or False Rejection Rate (FRR)). We did not collect any real forgery so we used all the signatures from the other subjects as *random forgeries* in order to obtain the Type II error (or False Acceptance Rate (FAR)). The tablet is a WACOM Digitizer, Active area: 153.6x204.8 mm, Resolution: 50 lpmm, Accuracy: $\pm$ 0.25 mm and Maximum report rate: 205 points/second. In our experiments we used the full signing trajectory, in order to be consistent with the experiments that we will perform with the real-time system.

**Training.** During training the system must *learn* a representation of the training set that will yield minimum generalization error. The dynamic time warping algorithm provides the optimal alignment of two signatures, so we could compute the mean signature of these two along the warping path. This mean signature would provide a more robust representative for the class since the inherent noise in capturing the signatures will be averaged. In the case in which there are more than two examples in the training set, there is no clear way of defining the mean signature. In principle, one could think of performing the simultaneous alignment of all the examples at the same time, working on an $N$-dimensional tensor instead of a matrix. The disadvantage of this approach is that there is no clear way of defining the elementary cost of the arc joining two nodes of this tensor. We propose a sub-optimal training procedure. We perform only pairwise alignment in order to find all the pairwise mean signatures out of all the possible

pairs of elements in the training set. The mean signature that yields minimum alignment cost with all the remaining signatures in the training set is the one that represents the training set. The individual costs of aligning each of the signatures in the training set with this reference signature are collected in order to estimate the statistics of the alignment process. This statistics are subsequently used for classification. In figure 5 we show several examples of signatures collected for our database and their corresponding training reference.

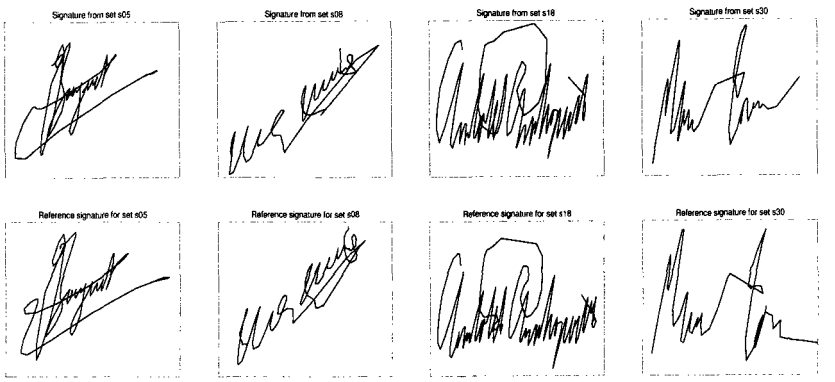


**Fig. 5.** Several examples of signatures in our database. In the first row we display signatures captured with the tablet while in the second row we show the corresponding reference signature of the training set.

We evaluated several possible alignment schemes in our experiments. Lorette and Plamondon [1] show evidence that the best representation space for a 2D signature verification system is the velocity domain, in concordance with the kinematic theory of rapid human movements developed by Plamondon [14], [15]. Therefore, one of the schemes that we tested was time alignment of the 2D velocity shape of the signatures. The $x$ and $y$ velocities were computed using first-order central finite differences. We examined two more schemes in which we perform time alignment of the 2D shape of the signature. In one of them, we align the raw data acquired with the tablet and in the other, we align the signatures after having rotated them so that their main axis coincide with the horizontal axis.

**Testing.** As we stated before, we used a test set of 10 signatures for computing the FRR and all the 740 signatures from other subjects for computing the FAR, both of them as a function of the classification threshold. Clearly, we can trade off one type of error for the other type of error. As an extreme example, if we accept every signature, we will have a 0% of FRR and a 100% of FAR, and if we reject every signature,we will have a 100% of FRR and a 0% of FAR. The simplest characterization of the FAR-FRR tradeoff is given by the *equal error rate*, i.e., the error rate at which the percentage of false accepts equal the percentage of false rejects. This equal error rate provides an estimate of

the statistical performance of the algorithm, i.e., it provides an estimate of its generalization error. We calculate the value of the equal error rate by intersecting the FAR and FRR curves that we computed, considering them to be piecewise linear.

In figure 6 we plot the value of the equal error rate for the 38 signers, for each of the three schemes under test. We observe that the alignment of 2D shapes of the signatures using the raw data gives the best performance of the three. We note that, for this case of best performance, there are three signatures for which the equal error rate is 10% and one for which the equal error rate is 20%, that, in our present experimental setting, accounts for having one or two signatures being falsely rejected. In figure 7 we show one of the original signatures, the reference signature, the signature that is falsely rejected and one of the falsely accepted signatures for each of these worst performance cases.

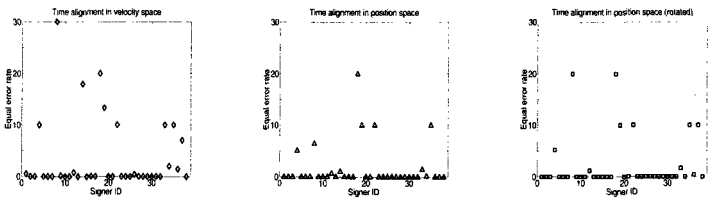


**Fig. 6.** Value of the equal error rate for each of the 38 subjects. The first plot corresponds to time alignment in velocity space, the second plot corresponds to time alignment of the 2D shape of the signatures and the third plot corresponds to time alignment of the 2D shape of the signatures, having previously aligned their main axis with the horizontal axis.

Table 3.1 summarizes the results shown in figure 6 for the best case, i.e., the one corresponding to the alignment of the 2D shape of the signatures. There is a noticeable difference between the numbers on table 3.1 and the values plotted on figure 6. The difference is due to the fact that in figure 6 we plot the value of equal error rate, that was computed as the intersection of the FAR and FRR curves and corresponds to an estimate of the performance of the algorithm in average, while in table 3.1 we show the values of FAR and FRR that correspond to a particular realization of the performance of the algorithm given by our test set. In the first four columns of the table, we show the number of signatures falsely rejected and falsely accepted, as well as the FRR and FAR for the condition of equal error rate. In the last four columns of the table, we present the number of signatures falsely rejected and falsely accepted, as well as the FRR and FAR for the condition of FAR $< 1\%$. We note the mentioned trade off between FRR and FAR comparing the results obtained under these two conditions. In a verification system used for credit card transactions, it is very reasonable to operate with FAR $< 1\%$ since accepting a false signature could cause a substantial financial loss while rejecting a true signature could produce a little annoyance in the customer that would have to repeat the signature. For this case, the algorithm has an average FRR of 4 %. We should note that the overall performance of the

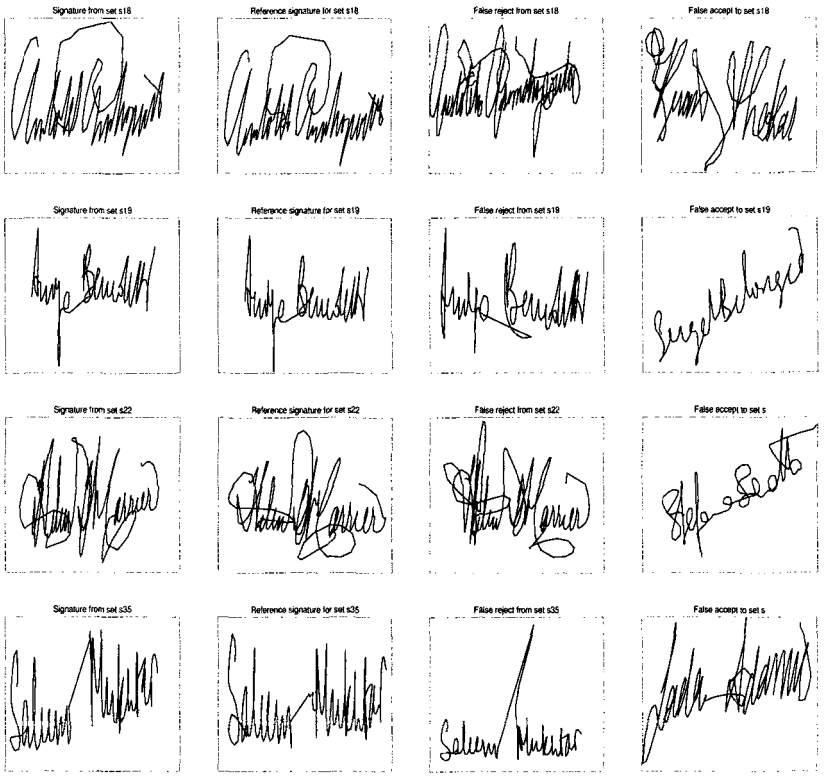


**Fig. 7.** The 4 cases in which the algorithm has biggest error. The first column shows signatures in the training set, the second column shows the reference signatures, the third column shows the signatures that were falsely rejected and, the last column shows signatures that were falsely accepted.

algorithm is quite good, with error rates comparable to the best presented in the literature [5], [12], [13], [19] under similar conditions.

### 3.2 Performance of Visual Signature Verification System

We collected signatures from 15 subjects. They were asked to train the system with 5 signatures and test it with at least 2 genuine signatures and 2 quasi-random forgeries (some piece of handwriting that has similar writing pattern as the genuine signature and it is written by the same subject).

Given the results showed in the previous section, we only implemented the 2D signature shape alignment in our real time system. The training is performed in the same way as it was described in section 3.1, right after the signatures have been captured. The training time is variable, depending on the duration of the signatures, experimentally we observe a maximum training time of 5 seconds. We should point out that our visual tracker is working in real-time at 30Hz. As shown in our previous work [10], we are able to track the pen tip in conditions of normal cursive or printed handwriting and drawings. However, in the case of

**Table 1.** Performance of the algorithm tested on the database of signatures collected with the tablet.

| Signer ID | Equal error condition | | | | FAR<1% condition | | | |
|---|---|---|---|---|---|---|---|---|
| | # FRs | # FAs | FRR (%) | FAR (%) | # FRs | # FAs | FRR (%) | FAR (%) |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 42 | 10 | 5.6757 | 1 | 7 | 10 | 0.9459 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 55 | 10 | 7.4324 | 2 | 3 | 20 | 0.4054 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 1 | 5 | 10 | 0.6757 | 0 | 5 | 0 | 0.6757 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 1 | 8 | 10 | 1.0811 | 1 | 2 | 10 | 0.2703 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 2 | 153 | 20 | 20.6757 | 3 | 2 | 30 | 0.2703 |
| 19 | 1 | 77 | 10 | 10.4054 | 4 | 2 | 40 | 0.2703 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 1 | 77 | 10 | 10.4054 | 1 | 2 | 10 | 0.2703 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 33 | 1 | 11 | 10 | 1.4865 | 1 | 7 | 10 | 0.9459 |
| 34 | 1 | 1 | 10 | 0.1351 | 0 | 6 | 0 | 0.8108 |
| 35 | 1 | 83 | 10 | 11.2162 | 1 | 1 | 10 | 0.1351 |
| 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 11 | 512 | 2.8947 | 1.8208 | 14 | 37 | 3.6842 | 0.1316 |

signatures, we observe that the system occasionally loses track of the pen tip when the subject produces an extremely fast stroke. This problem of losing track of the pen tip could be solved in the future by using a more powerful machine or dedicated hardware. However, after a few trials, the user learns how to utilize the system without driving it to its limits.

The results of our tests were very encouraging and they are summarized in table 3.2. After the training, we obtain the individual costs of aligning each signature in the training set with the reference signature. We compute the mean and the standard deviation of these individual costs in order to use them for classification. When a new signature is captured for testing the system, the cost of aligning this signature with the reference signature is computed. The signature will be classified as genuine if this cost is smaller than a threshold and classified as forgery otherwise. In all our experiments, the threshold was taken to be the mean plus twice the standard deviation of the training costs.

We observe that the performance of subject 6 is not very good. Figure 8 shows signatures corresponding to this subject. We believe that this low performance is due to the fact that he was writing Greek letters with a big change in the inter-letter hesitation, generating in this way a series of sequences that are not very consistent time-wise when we compare them with signatures. We should point out that these results from the visual system need to be verified

**Table 2.** Results obtained by testing the algorithm in real time.

| Signer ID | False Rejects | False Accepts | FRR(%) | FAR (%) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 33 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 |
| 6 | 4 | 0 | 80 | 0 |
| 7 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 |
| Total | 5 | 0 | 7.5 | 0 |

by doing a bigger set of experiments, in order to fully characterize the performance of the algorithm. Nevertheless, we note that these results correspond to the most pessimistic scenario since the subjects were writing their signatures without any feedback from the system on the consistency of their signatures. In a more realistic scenario in which the subjects receive feedback from the system, one could expect that they would adapt, learning to sign in a way that would be considered consistent for the system.

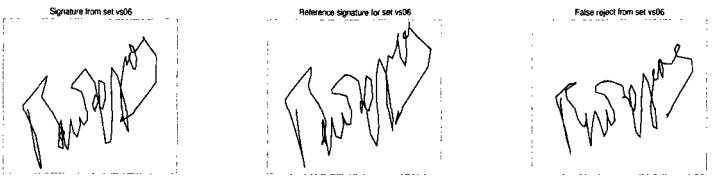


**Fig. 8.** Signature from subject 6, reference signature and false rejected signature.

In figure 9 we show some original signatures, the corresponding reference signature and one of the random forgeries provided by the same subject.

The last part of our experiment was to compute the the FAR and FRR in the same way as we calculated it in section 3.1, i.e., the FRR was obtained by using the 2 signatures that we had for testing and the FAR was obtained by using all the other signatures provided by the other signers and all the forgeries. The result provided by the algorithm were perfect with 0% of FAR and FRR, probably due to the fact that our test set was very small. Nevertheless, these results are very encouraging in order to further pursue the development of a visual password verification system.

## 4 Conclusions and Further Work

We have presented a novel vision-based technique for personal identification. The system does not require any special hardware, unlike fingerprint verification, iris or retina scanning systems. It is comparable to face recognition systems in terms of hardware since it uses a camera for tracking the signature. We demonstrated

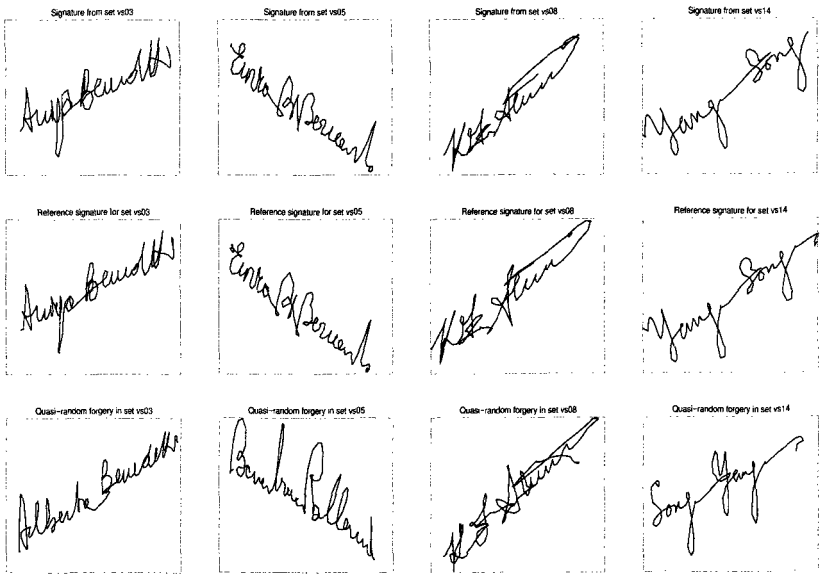


**Fig. 9.** Sequences acquired with the visual tracker. The first row shows signatures in the training set, the second row shows the reference signatures and the third row shows the quasi-random forgeries provided by the subjects.

the feasibility of having such a system working in real time with a high degree of accuracy in verifying signatures.

We consider it important to increase the robustness of the signature verification algorithm by adding some global parameterization of the signatures that will allow the system to discard coarse forgeries. The dynamic time warping algorithm could be improved by using different constraints or even using continuous time warping. We would also like to explore ways of training the system by aligning all the signatures at the same time, so that the resulting mean signature will be a more robust representative of its class. More extensive testing of the algorithm is required in order to understand its week points and correct them. In particular, we should include more actual forgeries in our database.

We note that the user would need some minimal training in order to master the system. When the signatures have extremely fast strokes, the system loses track of the pen tip. We asked the users to slow down a bit the motion of these strokes so that the tracker could acquire the signature. We believe that the use of a more powerful machine or the implementation of our tracking algorithm in hardware using FPGA's will eliminate this problem, allowing users to sign at normal speed.

## References

1. Dynamic approaches to handwritten signature verification. G. lorette and r. plamondon. *Computer Processing of Handwriting*, pages 21–47, 1990.
2. R.S. Bucy. Non-linear filtering theory. *IEEE Trans. A.C. AC-10, 198*, 1965.

3. J.L. Crowley, F. Bernard, and J. Coutaz. Finger tracking as an input device for augmented reality. *Int. Work. on Face and Gecture Recog.*, pages 195–200, 1995.
4. J.G. Daugman. High confidence visual recognition of persons by a test of a statistical independence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.
5. K. Huang and H. Yan. On-line signatuer verification based on dynamic segmentation and global and local matching. *Optical Engineering*, 34(12):3480–3487, 1995.
6. A.K. Jain, L. Hong, S. Pankanti, and E. Bolle. An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 85(9):1365–1388, 1997.
7. A.H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.
8. R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME-Journal of basic engineering.*, 35-45, 1960.
9. F. Leclerc and R. Plamondon. Automatic signature verificationand. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(3):643–660, 1994.
10. M. Munich and P. Perona. Visual input for pen-based computers. In *Proc. $13^{th}$ Int. Conf. Pattern Recognition*, 1996.
11. M.E. Munich and P.Perona. Visual input for pen based computers. *CNS Technical Report CNS-TR-95-01, California Institute of Technology*, 1995.
12. Vishvjit S. Nalwa. Automatic on-line signature verification. *Proceedings of the IEEE*, 85(2):215–239, 1997.
13. 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*, 12(7):710–717, 1990.
14. R. Plamondon. A kinematic theory of rapid movements. part i: Movement representation and generation. *Biological Cybernetics*, 72:295–307, 1995.
15. R. Plamondon. A kinematic theory of rapid movements. part ii: Movement time and control. *Biological Cybernetics*, 72:309–320, 1995.
16. R. Plamondon and G. Lorette. Automatic signature verification and writer identification, the state of the art. *Pattern Recognition*, 22(2):107–131, 1989.
17. L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, Inc., 1993.
18. H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech, Signal Processing*, 26(1):43–49, 1978.
19. Y. Sato and K. Kogure. On-line signature verification based on shape, motion and writng pressure. *Proc. 6th Int. Conf. on Patt. Recognition*, pages 823–826, 1982.
20. 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*, 12:787–808, 1990.
21. C.J. Taylor, T.F. Cootes, A. Lanitis, G. Edwards, and P. Smyth et al. Model-based interpretation of complex and variables images. *Philosophical transactions of the Royal Society of London*, 352(1358):1267–1274, 1997.
22. M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neurosci.*, 3(1):71–86, 1991.
23. L. Wiskott, J.M. Fellous, N. Kruger, and C. Von der Malsburg. Face recogniton by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.