Visual Signature Verification using Affine Arc-length

Mario E. Munich† and Pietro Perona†‡

† California Institute of Technology, 136-93, Pasadena, CA 91125, USA

‡ Università di Padova, Italy

{mariomu,perona}@vision.caltech.edu

Abstract

Signatures can be acquired with a camera-based system with enough resolution to perform verification. This paper presents the performance of a visual-acquisition signature verification system, emphasizing on the importance of the parameterization of the signature in order to achieve good classification results. A technique to overcome the lack of examples in order to estimate the generalization error of the algorithm is also described.

1 Introduction and Motivation

One of the research areas that is receiving a lot of attention nowadays is the area of biometric techniques for personal identification. Signature verification belongs to this set of biometric techniques. In most systems, signature verification requires the use of electronic tablets or digitizers for on-line capturing and optical scanners for off-line conversion [17]. These interfaces have the drawback that are bulky (they need to have at least the minimum area required to sign) and complicated to use, increasing the complexity of the whole identification system. Cameras, on the other hand, are much smaller and simpler to handle, and are becoming ubiquitous in the current computer environment. We have demonstrated [11] the feasibility of using a visual interface that can be built with video technology and computer vision techniques in order to capture signatures to be used for personal identification. This visual interface allows the user to write on a normal piece of paper with a normal pen, providing him/her with a more natural and comfortable environment to interact with the computer. This vision-based personal identification system could be integrated as a component of a complete visual pen-based computer environment.

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 in getting error rates comparable to those of humans performing the same task. The literature on signature verification is quite extensive (see [7, 8, 14] 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 evaluated 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.

Two important problems of many on-line systems for signature verification are studied in this paper. First, we evaluate the issue of optimal parameterization of the signature in order to achieve the best classification performance. Second, we address the problem of having very small training and testing sets in order to predict the statistical performance of the algorithm, or, in other words, to estimate its generalization error. We also describe different similarity measures used to compare signatures.

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 block diagram of the system and the experimental setup. 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.

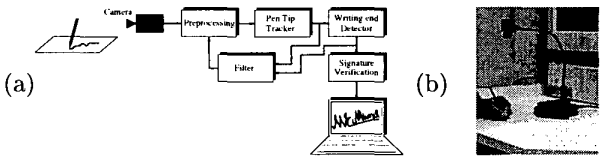

Figure 1: *(a) Block Diagram of the system. The camera feeds a sequence of images to the preprocessing stage. This block initializes the algorithm and selects the template to perform the tracking of the pen tip. The tip tracker obtains the position of the pen tip in each image of the sequence. The filter 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

2.1.1 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. We display a box at a particular location of the image. The user has to place the pen tip inside the box and the system

0-7695-0149-4/99 $10.00 © 1999 IEEE

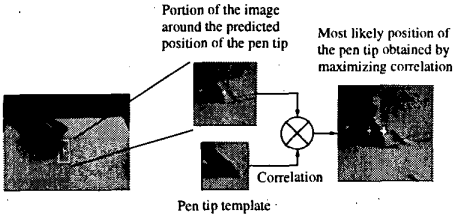

Figure 2: *Given the predicted location of the pen tip in the current frame, the most likely position of the pen is obtained by finding the place that has maximum correlation with the previously stored template of the pen tip.*

automatically grabs the template used to perform tracking.

2.1.2 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.

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.

2.1.3 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 [3], [5], [6] 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

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

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.

2.1.4 Real-time Implementation.

The implementation hardware 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 480x640 pixels per interlaced image. The frame grabber is a PXC200 manufactured by Imagination. The input camera image is digitized by the board and even and odd fields of the image are separated for future processing at 60 Hz and transferred to memory through the PCI bus. All further computations are performed with the PC. We achieved a total processing time of 14ms per frame.

2.2 Signature Verification

There is an extensive literature on the subject of signature verification [7, 8, 14]. Several researchers [4, 9, 11, 12, 13, 16, 18] have used Dynamic Time Warping (DTW) in order to perform the comparison between signatures. Depending on the particular system, the similarity between raw signature shapes and/or functions derived from the signatures (e.g., velocities, accelerations, moments of inertia, etc.) is evaluated with DTW.

The present implementation of DTW for signature verification attempts to perform the best alignment of the 2D shape of the signatures, i.e., we find the 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 [8] 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.

In most of the mentioned previous work, a time-based parameterization of the functions to be aligned was used. To our knowledge, only Nalwa [12] used an arc-length parameterization of the signatures for computing the characteristic functions proposed in his paper. The arc-length parameterization of the signature is loosely dependent on time and on the dynamics of signing, even though it keeps the causality of the signature's generation. This weak dependence on the 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 towards 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 [15]. In this paper, we will explore the performance of our verification system parameterizing the signature on time, arc-length and affine arc-length. For an extensive description of the implementation of DTW see reference [11]. Figure 3 shows an example of dynamic time warping applied to align the 2D shape of two signatures.

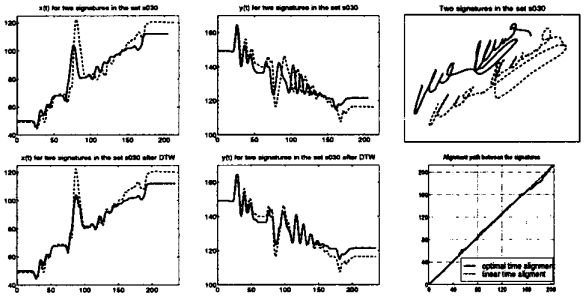

Figure 3: *Example of dynamic time warping applied to align the 2D shape of two realizations of the same signature. The first column shows $x(t)$ before and after time warping, the second column shows $y(t)$ before and after alignment. The upper plot of the third column shows the two examples of the signature. The lower plot of the third column shows the optimal time alignment path compared with a linear time alignment path. We note that the alignment is quite good regardless of the differences in the shapes of $x(t)$ and $y(t)$. These remaining mismatch between these signals accounts for the differences in shape of the signatures.*

2.2.1 Euclidean and affine arc-length.

Several studies (see [15] and references therein) show that the generation and perception of planar movements by humans follows a direct relationship between the tangential velocity of the hand and the radius of curvature of the planar curve. Experimental results exhibit 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 $1/3$ power of the radius of curvature. While the relationship between this two quantities is very intuitive, there is no clear explanation for the exact factor $1/3$ in the power law. Pollick and Sapiro [15] show that this power law precisely implies motion at a constant affine velocity. This means that curves with equal affine length will 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 [15] 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.

Let's define the relations used to re-parameterize the signatures on Euclidean and affine arc-lengths. A planar curve may be defined as the locus of points $C(p) = [x(p), y(p)] \in \mathbb{R}^2$, with $p \in [0,1]$. Given an increasing function $q(p) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, the curve defined by $C(q)$ 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 parameterizations used in our experiments is the Euclidean arc-length ν defined such as the curve is traveled with constant velocity, i.e., $\| \frac{\partial C}{\partial \nu} \| = 1$. Given our curve C, parameterized with an arbitrary parameterization p, in order to re-parameterize it in Euclidean arc-length, we use the relation $\nu(p) = \int_0^p \| \frac{\partial C(t)}{\partial t} \| \, dt$.

The second parameterization used in the experiments is the affine arc-length s 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. To re-parameterize the curve we use the following $s(p) = \int_0^p \left| \frac{\partial C}{\partial t} \times \frac{\partial^2 C}{\partial t^2} \right|^{\frac{1}{3}} dt$.

3 Experiments

The performance of the visual tracking system has been presented in reference [10] and some preliminary results on the topic of this paper have been described in reference [11]. We focus our experiments on the evaluation of the performance of the visually-based automatic personal identification system using different parameterization of the signature, enhancing the number of examples in order to better estimate the generalization error of the algorithm and employing different signature similarity measures.

We collected signatures from 56 subjects, 18 of them were women and 4 were left handed. Each of them was asked to provide 25 signatures, 10 of them to be used as the training set and the other 15 to be used as the test set. The data was collected in three sessions that took place in different days in order to get a sample of the variability of the subject's signatures while avoiding the distortion produced by the boredom of the repetitive task of signing. We should point out that the camera was not placed at a fixed position and height, it was changed from subject to subject and from session to session.

We also asked few of the signers to provide forg-

eries for each of the subjects in the database, as the ones shown in figure 4. Each set of forgeries for a particular subject was collected in one session. The naive forger was shown the ink trace of a set of real signatures and given enough time to practice the signature to be forged until feeling comfortable writing it. The set of forgeries was collected in two groups of 5 signatures each, giving the forger some rest in between. The visual tracker was set up such that the signer could not remain still in the same place for more than a few hundred milliseconds, not allowing the forger to copy the signatures at a very slow speed but rather forcing him/her to produce it at normal signing pace. The forger knew that the system was acquiring the full signing trajectory and he/she was given feedback on the success of his/her attempt.

There are two different errors that characterize the performance of the algorithm. The Type I error (or False Rejection Rate (FRR)), measures the number of true signatures classified as forgeries as a function of the classification threshold. The Type II error (or False Acceptance Rate (FAR)), evaluates the number of false signatures classified as real ones as a function of the classification threshold. The test set allows us to compute the FRR. We computed the FAR in two different ways. First, we used all the signatures from the other subjects as *random forgeries*, and second, we used the acquired forgeries.

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 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. Nevertheless, after a few trials, the user learns how to utilize the system without driving it to its limits.

3.1 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 in the case in which there are more than two examples in the training set, there is no clear way of aligning all of them at the same time. 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 it is difficult to define the elementary distance associated to the arc joining two nodes of this tensor. We propose a sub-optimal training procedure. We perform only pairwise alignment between all elements in the training set. The signature that yields minimum alignment cost with all the remaining ones is chosen to perform the final matching. All signatures are placed in correspondence with this particular one. The prototype that represents the training set is computed as the mean of the aligned signatures. The individual residual distances between each of the signatures in the training set and this reference signature are collected in order to estimate the statistics of the alignment process. This statistics is subsequently used for classification. In figure 4 we show several examples of signatures collected for our database, their corresponding training reference and one of the forgeries provided by other subject.

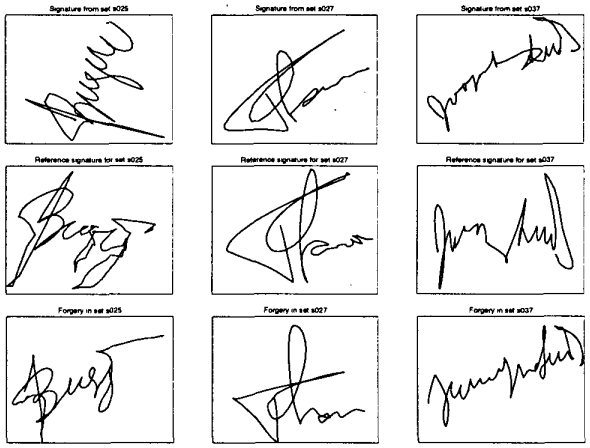

Figure 4: *Several examples of signatures in our database. On the first row we display signatures captured with the visual tracker, on the second row we show the corresponding reference signature of the training set and on the third row we display the forgeries provided by the subjects.*

3.2 Testing.

As we stated before, we used a test set of 15 signatures for computing the FRR and all the other signatures from other subjects or the forgeries, 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 curve of FAR as a function of FRR, using the classification threshold as a parameter, is called the error trade-off curve. It provides the behavior of the algorithm for any operation regime and it is the best descriptor of the performance of the algorithm. In practice, this curve is often characterized 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.

One common problem of many on-line system for signature verification is the lack of examples needed to build a reliable model for a signature and to asses the performance of the algorithm. This problem is inherent to the application since it is not feasible to ask a subject for all the examples of his/her signature required to perform these two tasks reliably. Thus, we have to build a model of the signature that will perform well in practice and we have to infer the generalization error of the algorithm, all with very few examples. We could increase the number of examples in both the training and test set by using *Duplicate Examples* as described by Y. Abu-Mostafa [1] if we know that the model that we are building should be invariant with respect to some transformation of the examples. In our particular case, one possible example of this transformation is time origin translation since our system should be insensitive to the particular instant of time in which we started acquiring the signature. Another possible transformation is given by affine deformation of the signatures, provided that the acceptable range of the parameters of this affine deformation could be estimated from the examples.

In experiments 1,2 and 3, we compute the performance of the system using the set of random forgeries in order to obtain the FAR. In experiment 4, we evaluate the performance of the algorithm using the set of real forgeries in order to calculate the FAR. In experiments 1 and 2, we evaluated two schemes in which we perform DTW alignment of the 2D shape of the signature. In one of them, we align the raw data and in the other, we align the signatures after having rotated them so that their main inertia axis coincide with the horizontal axis.

3.2.1 Experiment 1: Performance using different parameterizations of the signature (random forgeries)

In figure 5 we compare the error trade-off curves for each of the parameterizations and for each of the two schemes, using the residual distance between the signatures after alignment as the classification parameter. We only plot a section of the curve that is most informative. The best performance is achieved by the Euclidean arc-length parameterization since its corresponding error trade-off curve is below the curves corresponding to the other parameterizations as shown in figure 5.

3.2.2 Experiment 2: Performance using duplicate examples (random forgeries)

We generate duplicated examples both for training and for computing the FRR. We use two transformations in order to produce the duplicate examples.

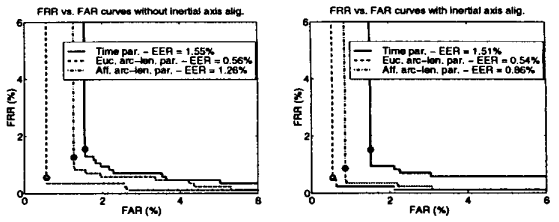

Figure 5: *Comparison of the error trade-off curves for each of the parameterizations and for each of the two schemes. The circle shows the equal error rate condition.*

One is time origin shifting, i.e., we re-sample the signature using linear interpolation, as if the time origin would have shifted from its original position to a point inside the inter-sample interval. The other is small scaling in x and y, where the range of scaling factors is estimated from the training examples. We end up with 200 examples for training and 300 examples for testing (a total of 500 examples obtained from the actual 25 signatures provided by the subject). We did not use duplicate examples to compute the FAR since we have enough random forgeries to estimate the FAR reliably. In figure 6 we compare the error trade-off curves for each of the parameterizations and for each of the two schemes. We observe that the error rates are bigger than the error rates of experiment 1. This increase in the error rates is expected since we have a bigger set of examples that provides a better statistical characterization of the problem. We should point out that the error rates are in the same order as in experiment 1, indicating that we did not introduce any wildly distorted example and that the algorithm seems to perform acceptably well. From experiments 1 and 2, we conclude that the best performance is obtained by rotating the main inertia axis of the signature in order to align it with the horizontal axis, therefore, we would use only this scheme for the following two experiments.

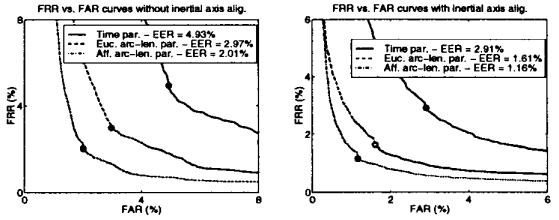

Figure 6: *Comparison of the error trade-off curves using duplicate examples, for each of the parameterizations and for each of the two schemes.*

3.2.3 Experiment 3: Performance using different distance measures (random forgeries)

In this experiment we evaluate the performance of the system using four different similarity measures and virtual examples. The first one is the residual distance between a given signature and the prototype, the second one is the correlation between

the signature under test and the prototype, and the third one is the weighted correlation between the signature and the prototype, all of these measures computed after establishing correspondence between the two curves with DTW. The weighting function represents the stability of each point of the prototype when aligned with each of the signatures in the training set. Finally, the last similarity measure is the harmonic mean of the other three ones [12]. In the first plot of figure 7, we compare the error-trade off curves for each of the parameterizations, using the mentioned harmonic mean as the classification parameter. We observe that the best performance is achieved by using affine arc-length parameterization of the signatures. So, in the second plot of the figure, we show the error trade-off curves for each of the different similarity measures for the affine arc-length parameterization. Given the error rates achieved with random forgeries, we could conclude that our verification method could be used for recognition since it has quite good discrimination capabilities.

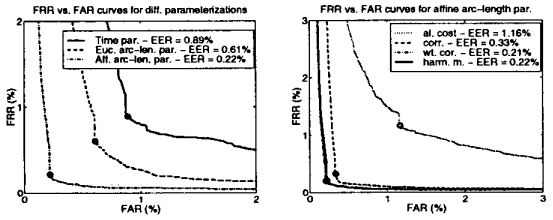

Figure 7: *Comparison of the error trade-off curves for each of the parameterizations, using the harmonic mean as the classification parameter, and comparison of the error trade-off curves for affine arc-length parameterization for each of the similarity measures. The affine arc-length parameterization provides the best performance.*

3.2.4 Experiment 4: Performance with intentional forgeries

This is the most important experiment since it will estimate how well the algorithm will perform in a real-life situation. The previous experiments were important in order to select the method that would be more likely to succeed when dealing with actual forgeries since a scheme that performs very well with random forgeries would probably perform well with actual forgeries. In this experiment is also when the use of duplicate examples is crucial in order to get good estimates of the generalization error since we have only 15 true signatures to compute the FRR and 10 forgeries to calculate the FAR. We use the same transformations that in experiments 2 and 3 in order to obtain a set of 300 signatures for computing the FRR and a set of 200 signatures for calculating the FAR. In figure 8, we show the error trade-off curves for all the parameterizations, using the harmonic mean of the similarity measures as the classification parameter, and the curves for affine arc-length parameterization for all the similarity measures described in experiment 3. We observe that the performance is considerably worst than in the previous case as one might expect, but it is still comparable to the best results presented in the literature [4, 12, 13, 16] under similar conditions.

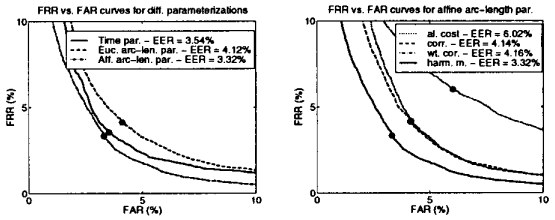

Figure 8: *Performance of the algorithm using forgeries in order to compute the FAR. Comparison of the error trade-off curves for each of the parameterizations, using the harmonic mean of the similarity measures as the classification parameter, and comparison of the error trade-off curves for affine arc-length parameterization for each of the similarity measures. The affine arc-length parameterization provides the best performance.*

In figure 9 we show one of the original signatures, the reference signature, a signature that is falsely rejected and a falsely accepted forgery for three cases whose equal error rate condition corresponds to a very low performance.

4 Conclusions and Further Work

We have presented the performance of a vision-based technique for personal identification. Our results show that the best performance is achieved parameterizing the signatures with affine arc-length, using the harmonic mean of several similarity measures. We could infer from this result that shape similarity and causality of the signature's generation are more important than matching the dynamics of signing. This result indicates that this dynamics is not stable enough to be used for signature verification since the subject is trying to reproduce a shape rather than a temporal pattern. However, the causality in which a signature is produced is still valuable and used in the DTW paradigm. This causality is the added information that our on-line system is using to out perform the bare comparison of a still picture of different signatures.

We also have shown that the use of duplicate examples provide 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 a time origin shifting and small scaling in x and y directions as the transformations. A full affine transformation could be used to generate duplicate examples given that a reasonable range of the parameters of this transformation could be estimated from the data.

We have shown a very good performance of the system when tested with random forgeries. This

Figure 9: *The cases in which the algorithm has biggest error. The first row shows signatures in the training set, the second row shows the reference signatures, the third row shows the signatures that were falsely rejected and, the last row shows signatures that were falsely accepted.*

result 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) and provides grounds to conclude that the algorithm could be used for signature recognition.

The signature verification algorithm could be made more robust by adding some global parameterization of the signatures that would 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. One problem that is unsolved with the present scheme is the problem of dramatic changes in scale and it is one of the areas of further research, as well as the development of a better similarity measure.

Comparing signature verification with another biometric technique for personal identification, such as fingerprint verification, we could observe that a forger with enough information about the true signature and enough training could deceive the algorithm. This weakness may or may not be important depending on the particular application.

Signature verification 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 figure 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 block possible forgers from knowing it. This system would overcome the mentioned weakness of the method but it would make the user feel a bit awkward since there would absolutely no visual feedback when signing.

References

[1] Yaser S. Abu-Mostafa. Hints. *Neural Computation*, 7:639–671, 1995.

[2] A.M. Bruckstein, R. Holt, A. Netravali, and T. Richardson. Invariant signatures for planar shape recognition under partial occlusion. *CVGIP: Image Understanding*, 58(1):49–65, 1993.

[3] R.S. Bucy. Non-linear filtering theory. *IEEE Trans. A.C. AC-10, 198*, 1965.

[4] K. Huang and H. Yan. On-line signature verification based on dynamic segmentation and global and local matching. *Optical Engineering*, 34(12):3480–3487, 1995.

[5] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[6] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME-Journal of basic engineering.*, 35-45, 1960.

[7] F. Leclerc and R. Plamondon. Automatic signature verification. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(3):643–660, 1994.

[8] G. Lorette and R. Plamondon. Dynamic approaches to handwritten signature verification. *Computer Processing of Handwriting*, pages 21–47, 1990.

[9] R. Martens and L. Claesen. On-line signature verification by dynamic time-warping. In *Proc. 13^{th} Int. Conf. Pattern Recognition*, pages 38–42, 1996.

[10] M.E. 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. Camera-based id verification by signature tracking. In *Proc. 5^{th} Europ. Conf. Comput. Vision, H. Burkhardt and B. Neumann (Ed.), LNCS-Series Vol. 1407-1408, Springer-Verlag*, pages 782–796, 1998.

[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 and G. Lorette. Automatic signature verification and writer identification, the state of the art. *Pattern Recognition*, 22(2):107–131, 1989.

[15] F.E. Pollick and G. Sapiro. Constant affine velocity predicts the 1/3 power law of planar motion perception and generation. *Vision Research*, 37(3):347–353, 1997.

[16] 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.

[17] 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.

[18] B. Wirtz. Stroke-based time warping for signature verification. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 179–182, 1995.