Visual Signature Verification using Affine Arc-length

Mario E. Munich and Pietro Perona
California Institute of Technology
www.vision.caltech.edu/mariomu

Outline

- Biometric ID
- Visual Signature Acquisition
  - Description of the system
- Signature Verification
  - Dynamic Time Warping
  - Signature parameterization
- Experimental Results
- Conclusions and further work

Biometric ID

Identification based on measurements of human biological characteristics

Vision-based:

- Face Recognition [Taylor et. al.,Turk & Pentland, Wiskott et.al.]
- Fingerprint Recognition [Jain et. al.]
- Iris Scanning [Daugman]
- Retina Scanning
- ...

Other:

- Voice Recognition
- Signature Verification

Why visual signature acquisition?

• Current I/O computer interfaces have limitations for decreasing their size.

tablet digitizer

screen

keyboard

Cell phone + PDA

PDA

mouse

• Future new I/O computer interfaces will involve Audio and Visual techniques.

camera

microphone

Advantages: • smaller size implementing them in VLSI.
• more *natural* way for people to communicate with computers.

Visual signature acquisition

Visual signature acquisition

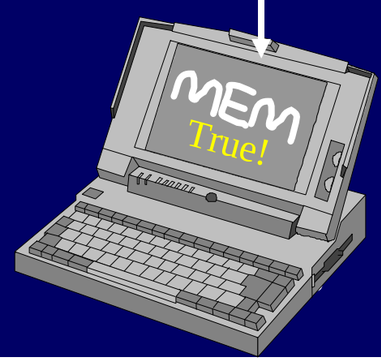
Preprocessing

Pen Tip Tracker

Ballpoint Detector

Filter

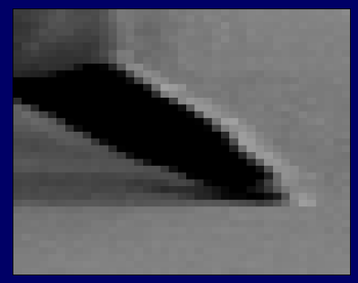
Signature Verification



Preprocessing

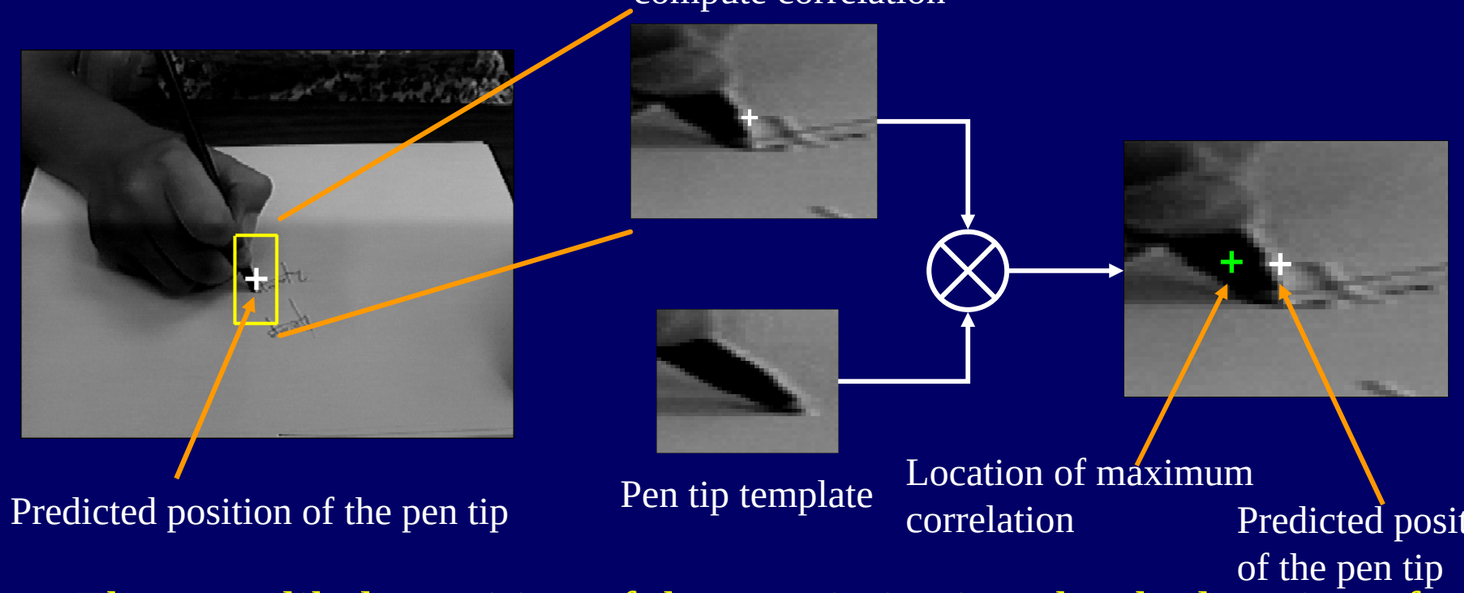
- Get the template of the pen tip
  - Mouse-clicking (manual)
  - Pen familiar to the system
  - Unknown pen

(size = 25x25 pixels)



- Get the portion of image where the pen tracker looks for the pen tip (size = 31x31 pixels)

Pen tip tracker



The most likely position of the pen tip is given by the location of maximum correlation

Filter

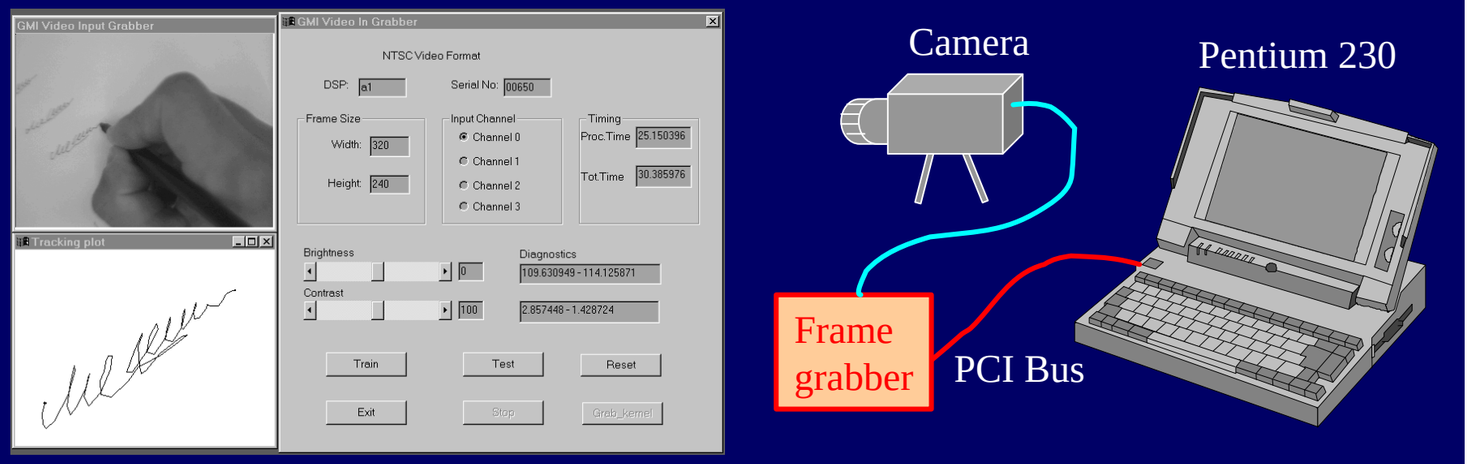
- Predicts the position of the pen tip in the following image
  - Speeds up computations
  - Smoothes out the trajectory
  - Estimates the position of the pen tip for missing frames
- Model of pen tip's dynamics

$$\begin{cases} \frac{dx}{dt} = v(t) \\ \frac{dv}{dt} = a(t) \\ \frac{da}{dt} = n_a(t) \\ y(t) = x(t) + n_y(t) \end{cases}$$

where:

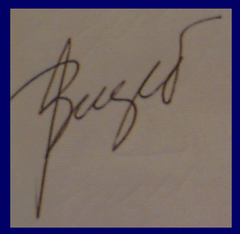
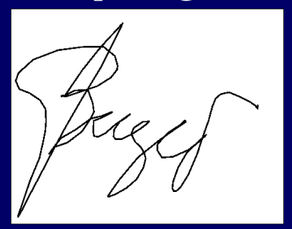
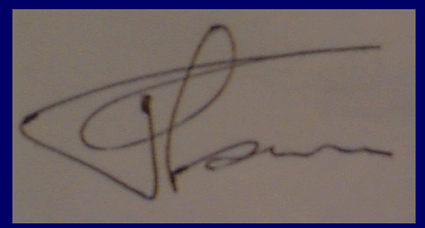
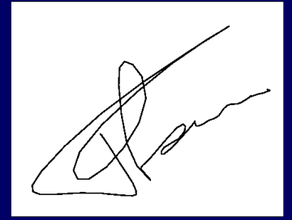
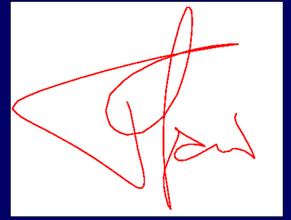
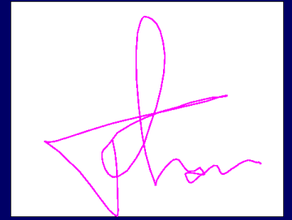
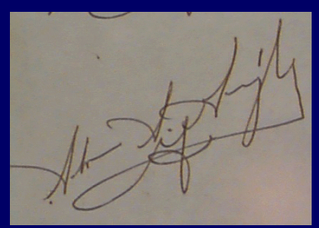
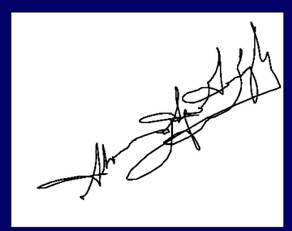
x(t): 2D pen tip's position
v(t): 2D pen tip's velocity
a(t): 2D pen tip's acceleration
y(t): location of max. correlation

Real Time Implementation



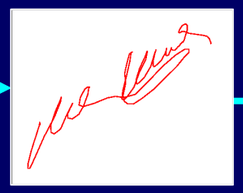
- The frame grabber is a PXC200 from Imagination.
- The system runs at 60 Hz with a total processing time of 15ms per frame.
- No calibration needed.

Acquired signatures

Picture	Example signature	Prototype	Forgery
			
			
			

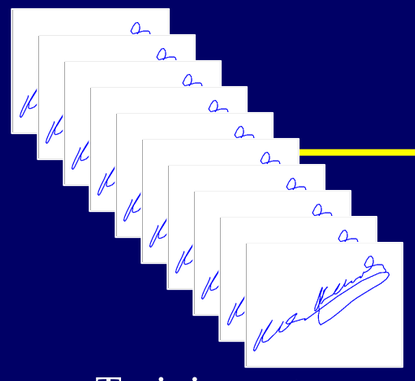
Signature Verification

New signature



Comparison

True signature or forgery?



Training set

Prototype

Signature Verification

- Off-line Signature Verification
  - Works on a static image of the signature, i.e., the result of the act of signing.
- On-line Signature Verification
  - Works on the dynamic process of generation of the signature, i.e., the action of signing itself.

("Automatic signature verification and writer identification, the state of the art", by M.Parizeau and R. Plamondon is a good survey paper)

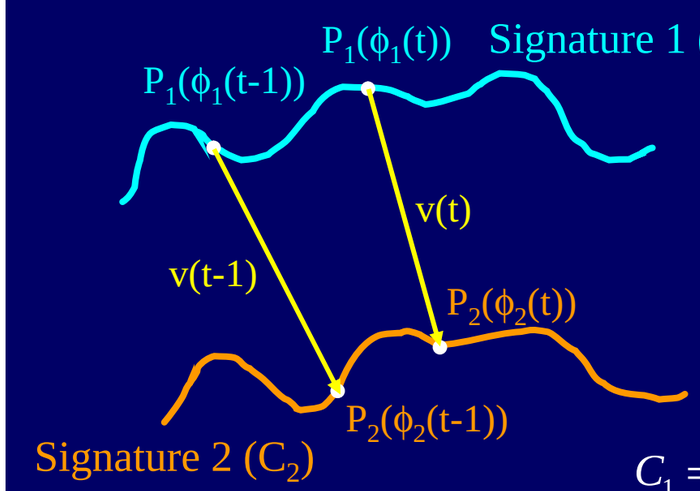
Signature Verification

Previous work on Signature Verification using Dynamical Time Warping:

- Sato and Kogure 82
- Parizeau and Plamondon 90
- Huang and Yang 95
- Wirtz 95
- Nalwa 97
- Munich and Perona 98, 99
- ...

Signature Verification

Comparison of two signatures



Elementary distance of matching $P_1(\phi_1(t-1))$ with $P_2(\phi_2(t-1))$ and $P_1(\phi_1(t))$ with $P_2(\phi_2(t))$.

$$d((P_1(f_1(t-1)), P_2(f_2(t-1))), (P_1(f_1(t)), P_2(f_2(t)))) = \left\| \overrightarrow{P_1(f_1(t))P_2(f_2(t))} - \overrightarrow{P_1(f_1(t-1))P_2(f_2(t-1))} \right\|^2$$

Matching function

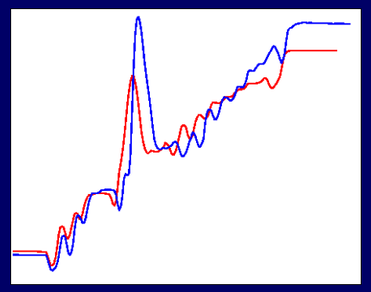
$$C_1 = (P_1(1), \ldots, P_1(T_1)), C_2 = (P_2(1), \ldots, P_2(T_2)), f = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}$$

Similarity measure ("distance" between C_1 and C_2)

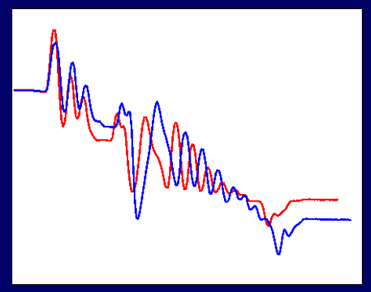
$$D(C_1, C_2) = \sum_{t=2}^{T} \left\| \overrightarrow{P_1(f_1(t))P_2(f_2(t))} - \overrightarrow{P_1(f_1(t-1))P_2(f_2(t-1))} \right\|^2$$

Dynamical Time Warping

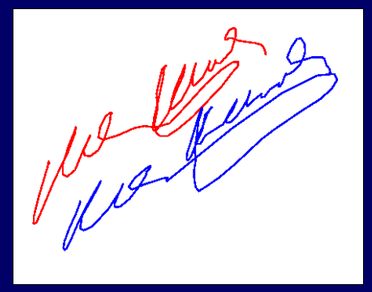
x(t)



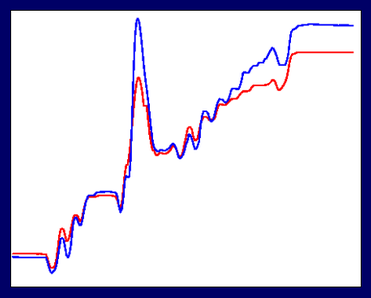
y(t)



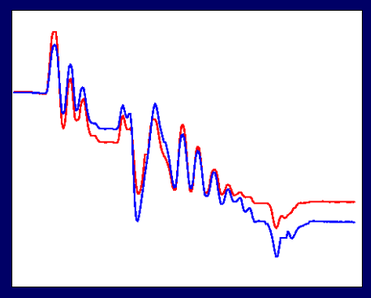
2 examples from s030



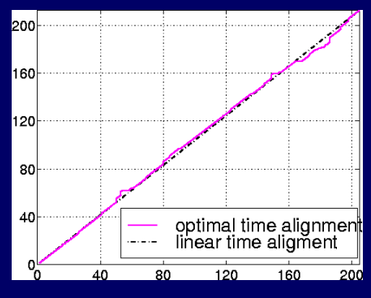
x(t) after DTW



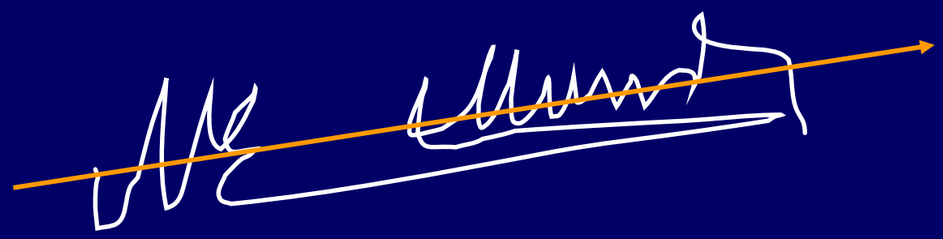
y(t) after DTW



Alignment Path



Invariance w.r.t. rotation



In order to obtain some degree of invariance w.r.t. rotations, align the main inertia axis of the signature with the horizontal axis before performing DTW.

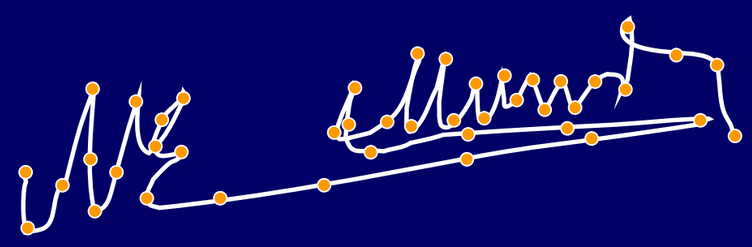
Signature parameterization

The acquisition system provides us a set of samples {x(t), y(t)} that represents the signature.

We could think the signature as a “planar” curve, so the question is whether the parameterization of this curve in time is the “proper” thing to do!?

Signature parameterization

C(p) = [x(p), y(p)]



Reparametrization: q(p): $R^+ \to R^+$
s.t. C(p) and C(q) have the same trace

Euclidean arc-length

Parameterization such that $\left\| \frac{\partial C(v)}{\partial v} \right\| = 1$

(velocity along the curve)

$$v(p) = \int_0^p \left\| \frac{\partial C(q)}{\partial q} \right\| dq$$

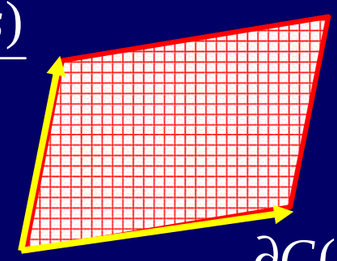
Re-parameterization transformation (Nalwa 97)

This parameterization defines a length that is preserved in the case of Euclidean transforms

Affine arc-length

Parameterization such that

$$\left| \frac{\partial C(s)}{\partial s} \text{ x } \frac{\partial^2 C(s)}{\partial s^2} \right| = 1$$



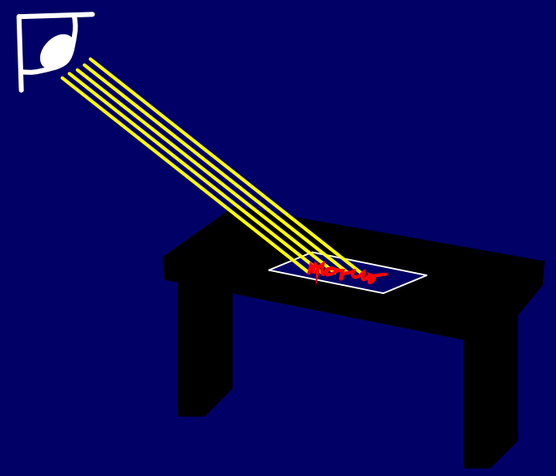
The area of the parallelogram is constant

$$s(p) = \int_0^p \left| \frac{\partial C(q)}{\partial q} \text{ x } \frac{\partial^2 C(q)}{\partial q^2} \right|^{\frac{1}{3}} dq$$

Re-parameterization transformation

This parameterization defines a length that is preserved in the case of affine transforms

Why affine?



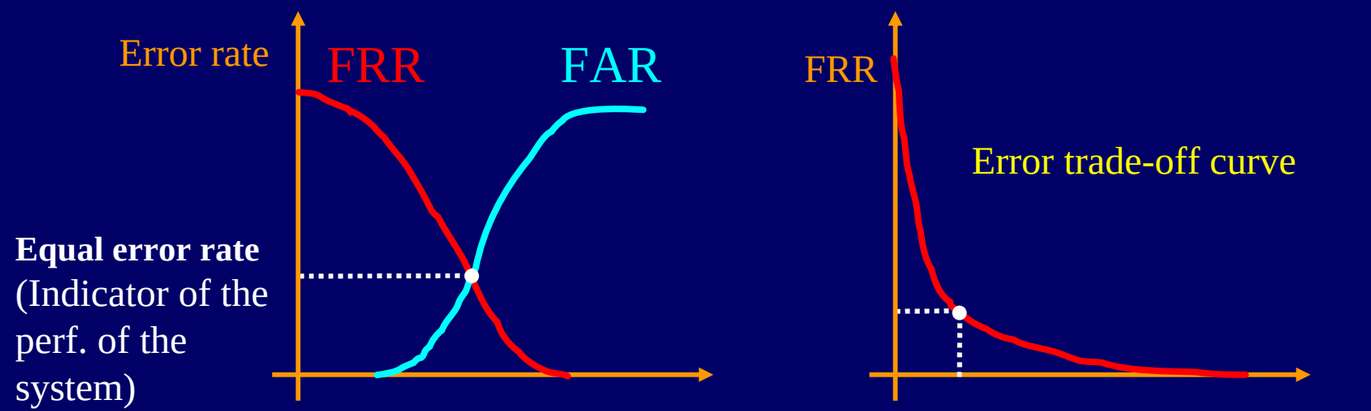
Affine transformations appear when a planar object is rotated and translated in space, and then projected into the eye via parallel projection
(Pollick and Sapiro 97)

Evaluation of the SV system

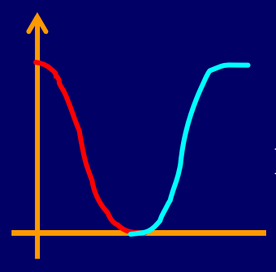
There are two types of errors to evaluate in order to asses the performance of the system:

- FAR (False Acceptance Rate): percentage of false signatures classified as true.
- FRR (False Rejection Rate): percentage of real signatures classified as false.

Evaluation of the SV system



Equal error rate
(Indicator of the perf. of the system)



Ideal case: FAR = FRR = 0%

Experiments

Collected a database of signatures

- 56 subjects
- 25-30 sample signatures per subject
- 10 signatures were collected per session, and each session took place on a different day
- 10 forgeries per subject
- each signature acquired with the real-time pen tracking system

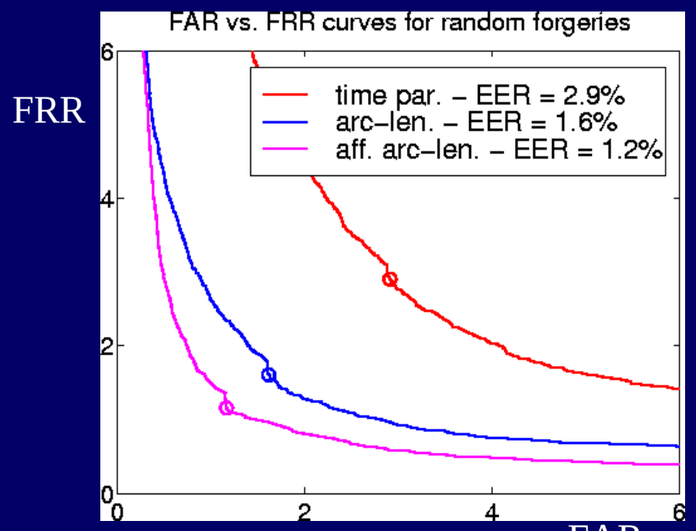
Generated a set of duplicated examples (Abu-Mustafa 95) for each signature using time origin shifting, scaling and shearing.

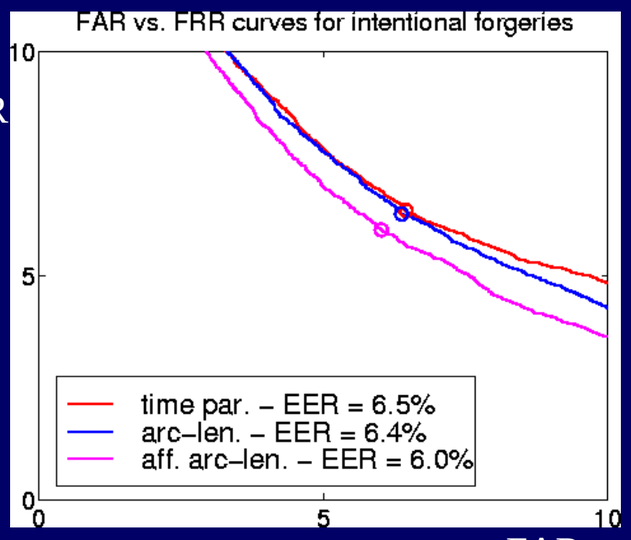
Experiments

The algorithm was tested as follows:

- Training set: 10 signatures per subject, augmented to 200 with the duplicated examples.
- Test set (true): remaining 15 signatures per subject, augmented to 300 with the duplicated examples.
- Test set (false): all 1375 signatures from all other subjects (“random forgeries”).
- Test set (false): 10 intentional forgeries, augmented to 200 with the duplicated examples.

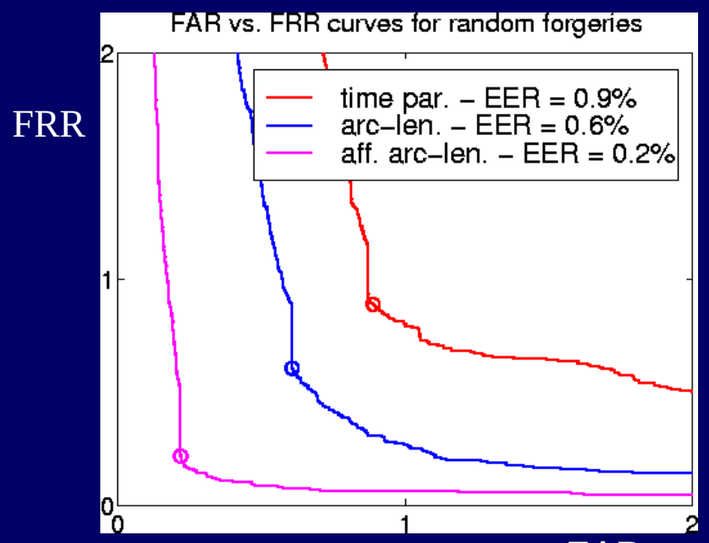
Results

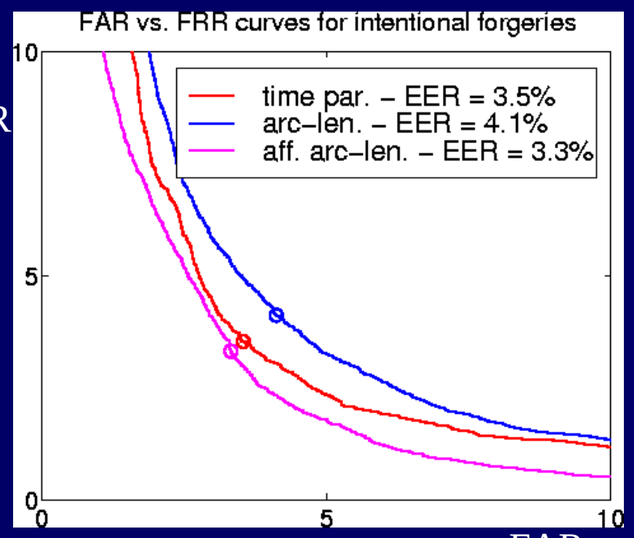




Classification performed using the similarity measure provided by DTW.

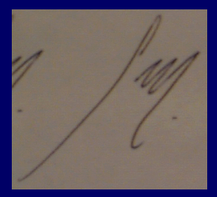
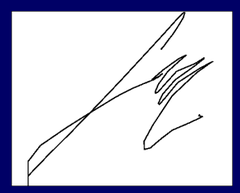
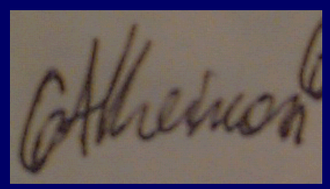
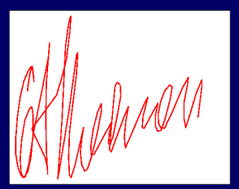
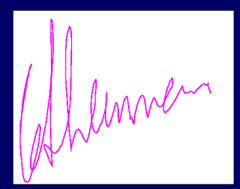
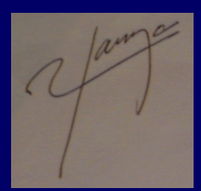
Results





Classification performed using various distance measures (DTW similarity measure, correlation and weighted correlation).

Results

pictures	examples	references	False rejects	False accepts
				
				
				

Conclusions

- Presented the performance of a vision-based technique for personal identification.
- Demonstrated the feasibility of having such a system working in real-time with high performance in verification.
- Evaluated different representations of the signature and different similarity measures in order to improve performance.
- Overcame the lack of examples in order to extract more meaningful estimates of the generalization error.

Further work

- Match all examples in a way such that the prototype would be a more robust representative of its class (sub-sample matching?).(ICCV99).
- Collect a new, clean and bigger database of examples.
- Explore different distance functions.