BioLab - Biometric Systems Lab
University of Bologna - ITALY

http://bias.csr.unibo.it/research/biolab

SFinGe

Synthetic **Fin**gerprint **Ge**nerator

Dr. Raffaele Cappelli

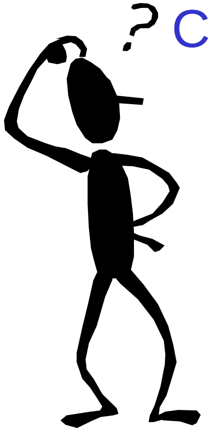
Why generating synthetic fingerprints? (1)

- Testing fingerprint recognition algorithms requires large databases of fingerprints, due to the small errors which have to be estimated

> Example: if an algorithm is quoted at 0.01% FAR (that is, the probability of falsely accepting an impostor is 1 on 10,000), then about 1,000,000 attempts of matching against impostor fingerprints are necessary to claim, with a 95% confidence, that the true error lies in the range [0.006%..0.014%]

- Once a database has been “used” (to test and optimize algorithms) it expires, since for a successive fair testing stage a new unknown database is mandatory

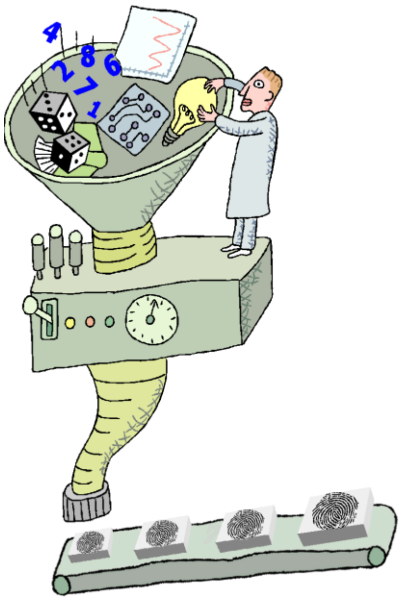
Why generating synthetic fingerprints? (2)



Collecting large databases of fingerprint images is:

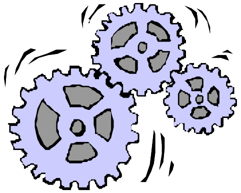
- expensive both in terms of money and time
- boring for both the people involved and for the volunteers, which are usually submitted to several acquisition sessions at different dates
- problematic due to the privacy legislation which protects such personal data

A method able to *artificially* generate realistic fingerprint-images could be used in several contexts to avoid collecting databases of real fingerprints



Outline

Fingerprint anatomy



How SFinGe works

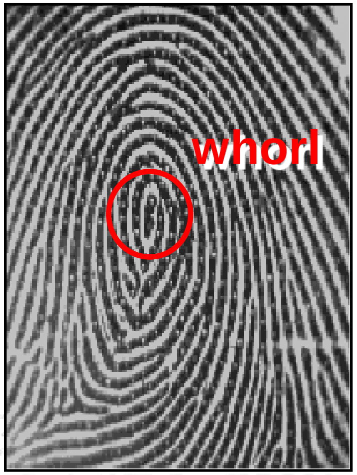
The main steps of the generation method

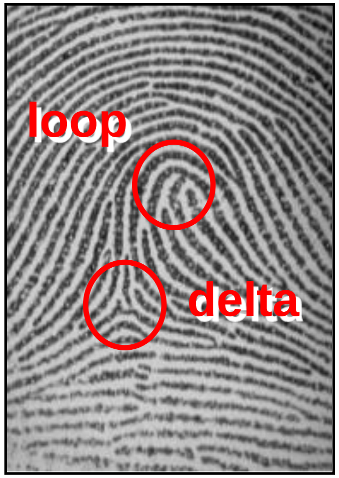
Examples and applications

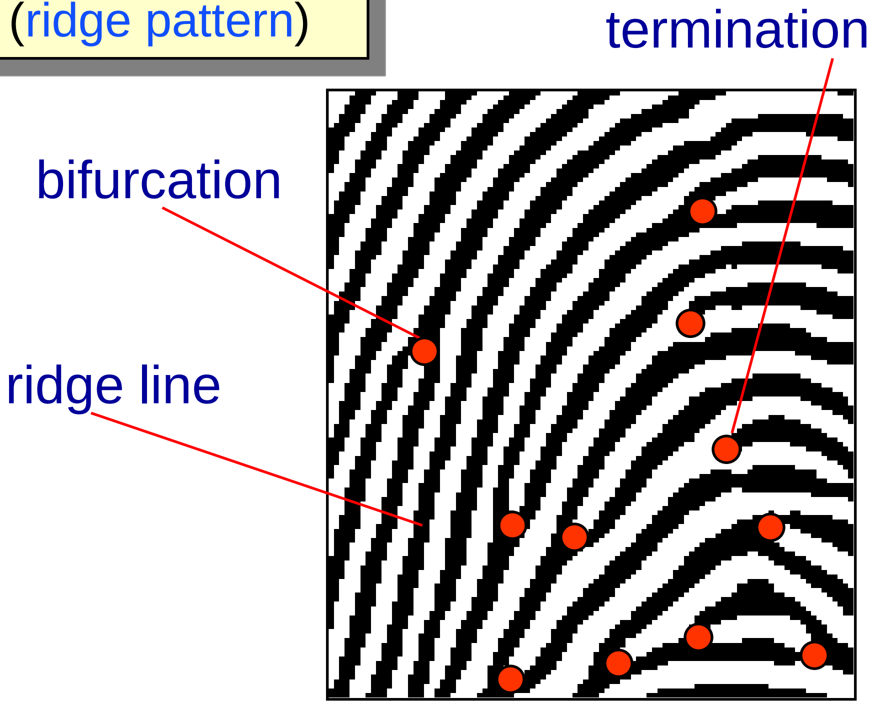
Fingerprint Anatomy (1)

A fingerprint is composed of a set of lines (ridge lines), which mainly flow parallel, making a pattern (ridge pattern)

Sometimes the ridge lines produce local macro-singularities, called whorl (O), loop (U) and delta (Δ)

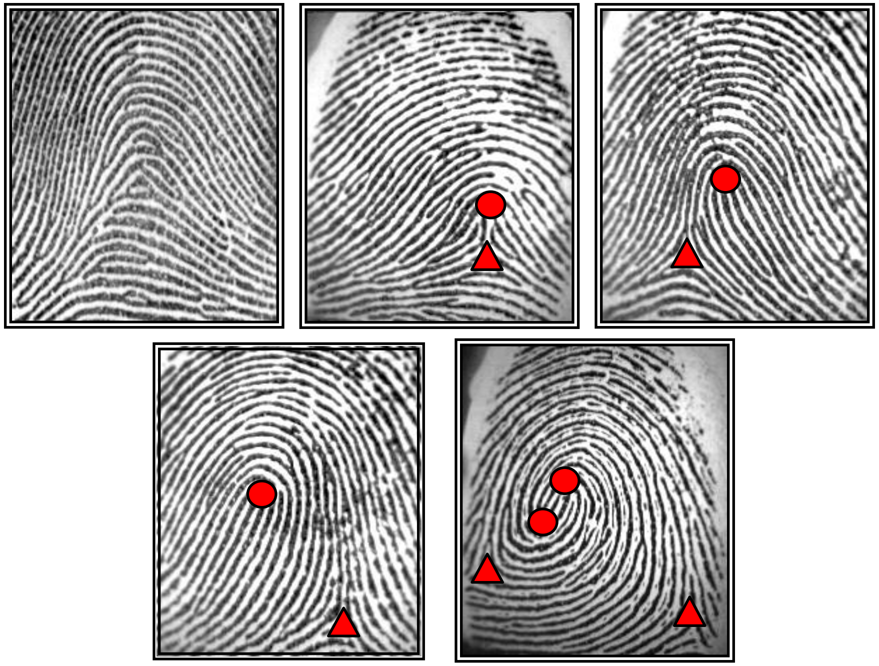






The minutiae, or Galton’s characteristics, are determined by the termination or the bifurcation of the ridge lines

Fingerprint Anatomy (2)


arch

tented arch

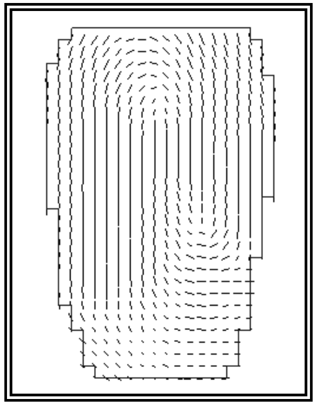
right loop

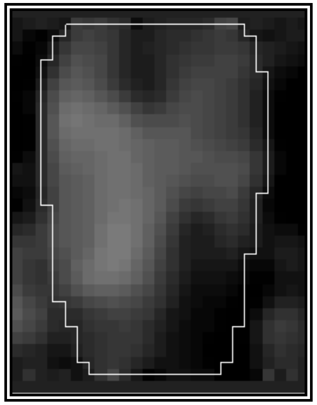
left loop

whorl

The five main fingerprint classes

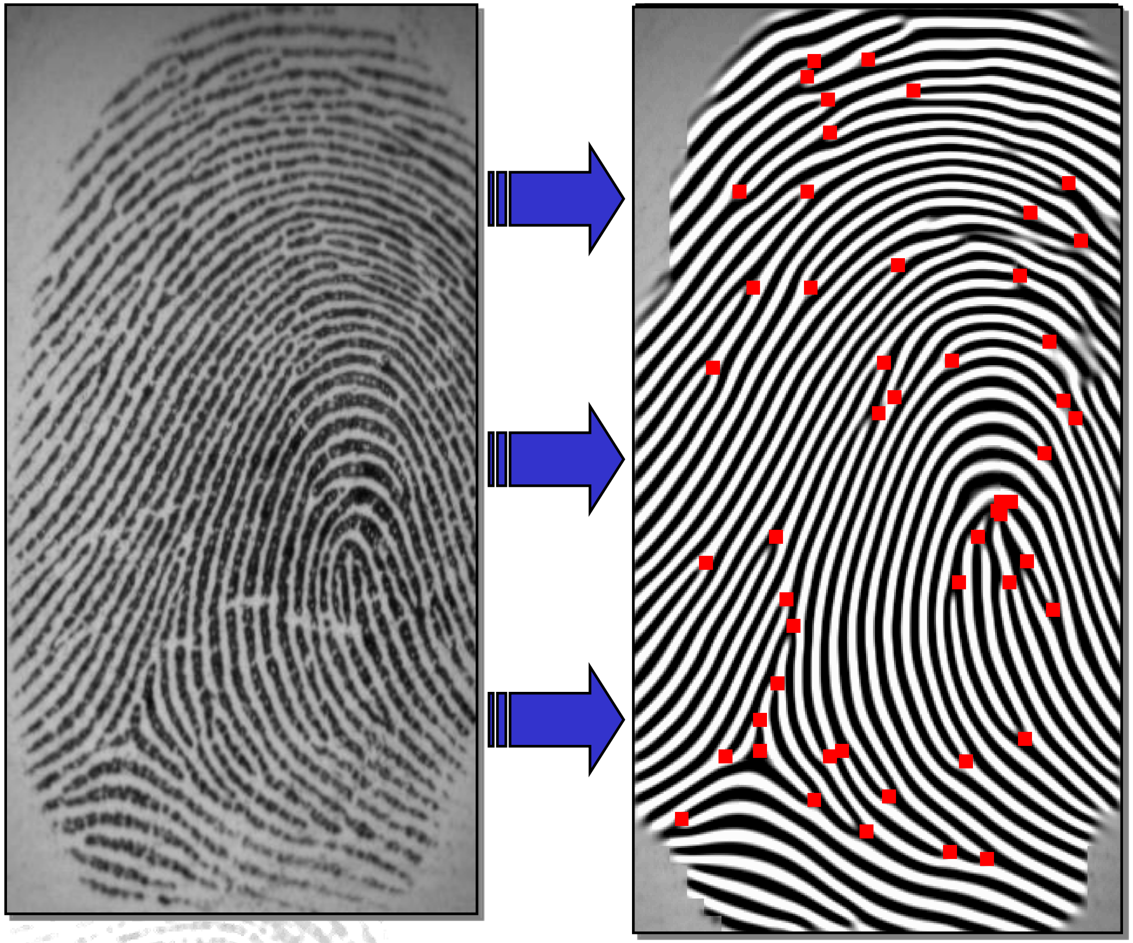
fingerprint

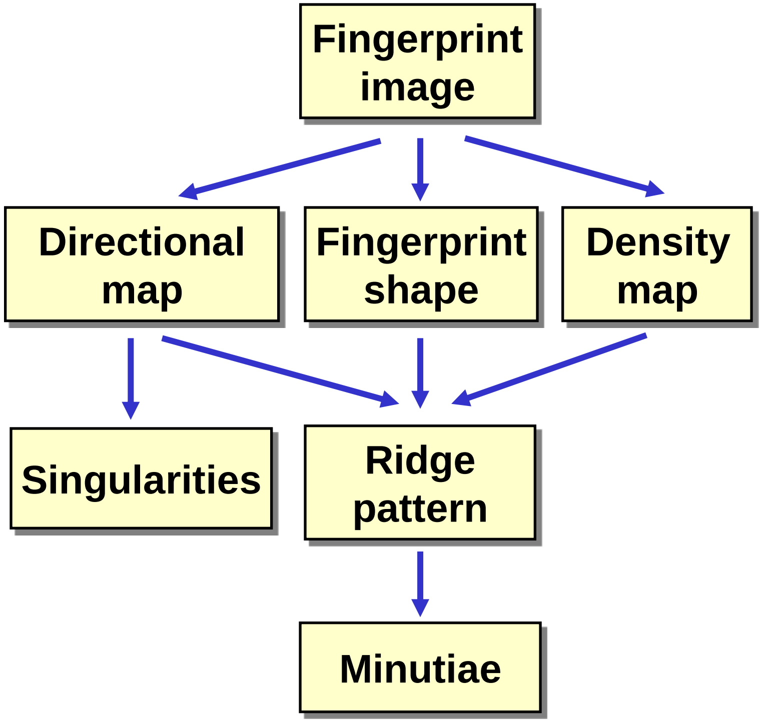

directional map


density map

- The ridge-line flow can be effectively described by a directional map
- The ridge line density can be summarized by using a density map

Fingerprint feature extraction





Fingerprint image

Directional map

Fingerprint shape

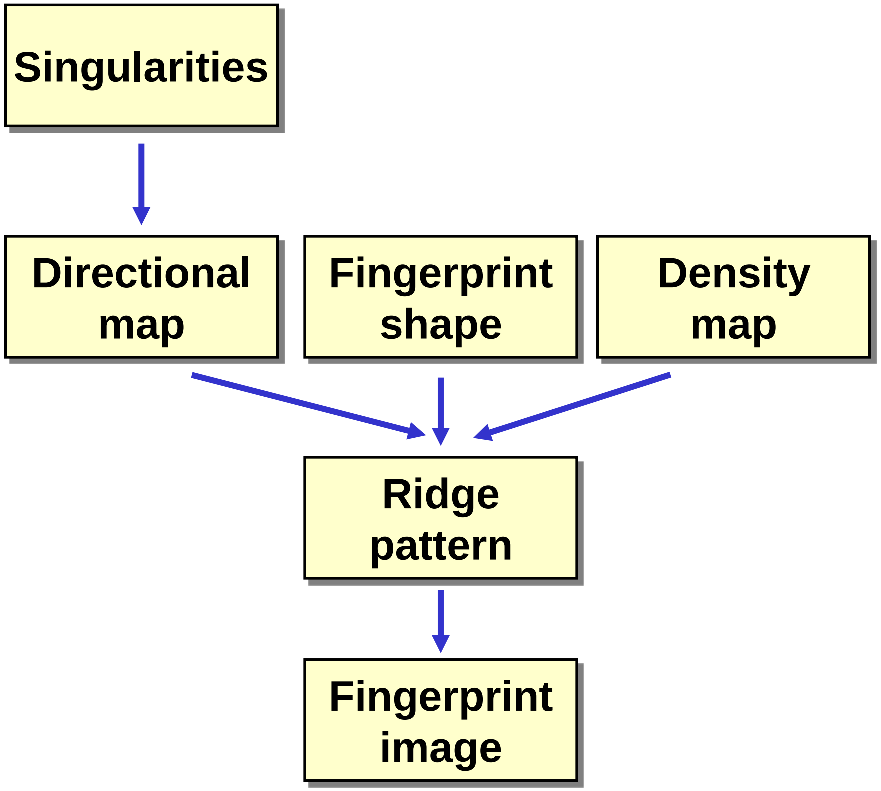
Density map

Singularities

Ridge pattern

Minutiae

How SFinGe works (1)



Singularities

Directional map

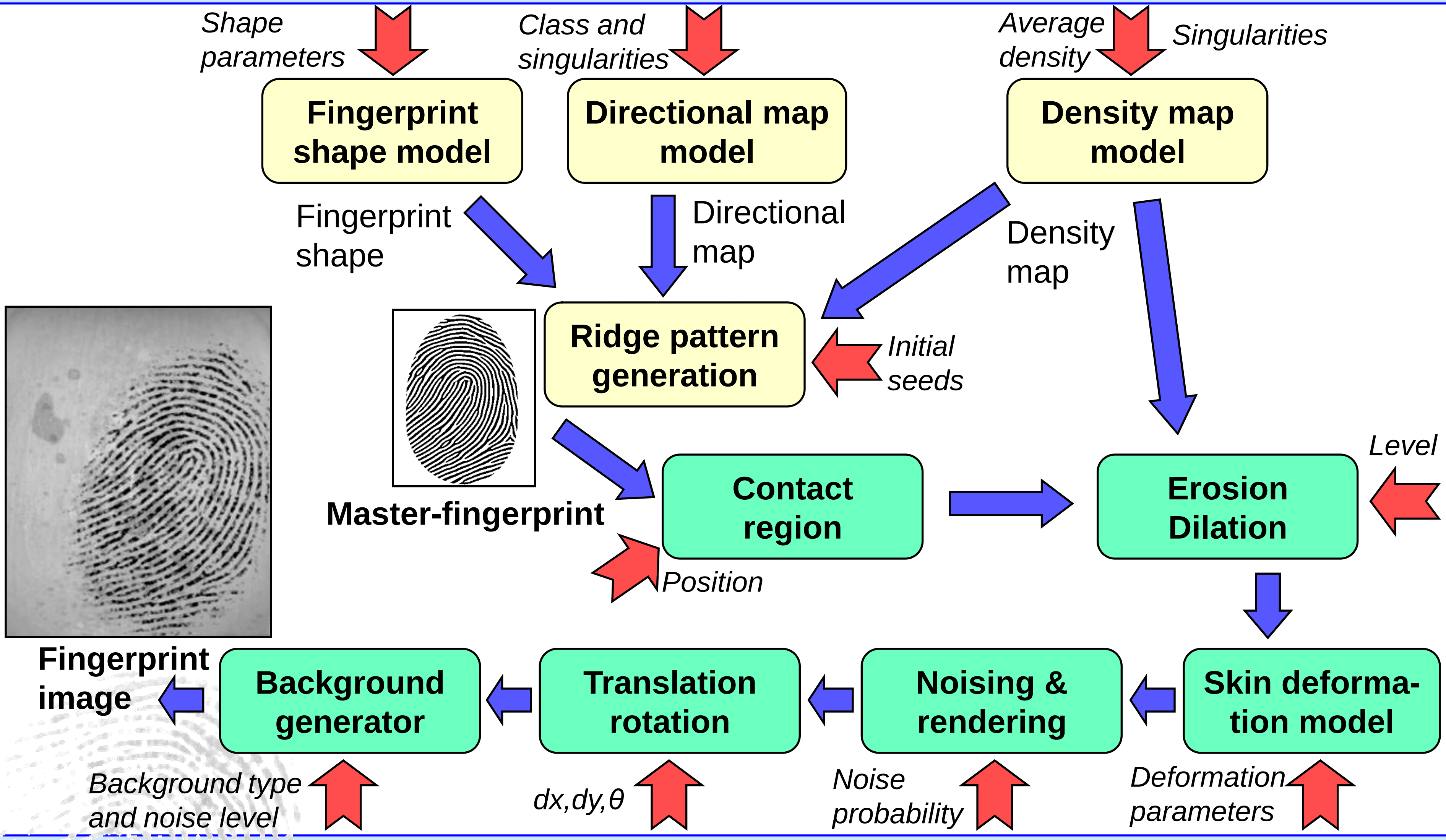
Fingerprint shape

Density map

Ridge pattern

Fingerprint image

How SFinGe works (2)



Shape parameters

Class and singularities

Average density

Singularities

Fingerprint shape model

Directional map model

Density map model

Fingerprint shape

Directional map

Density map

Ridge pattern generation

Initial seeds

Master-fingerprint

Contact region

Position

Erosion Dilation

Level

Skin deformation model

Noising & rendering

Translation rotation

Background generator

Fingerprint image

Background type and noise level

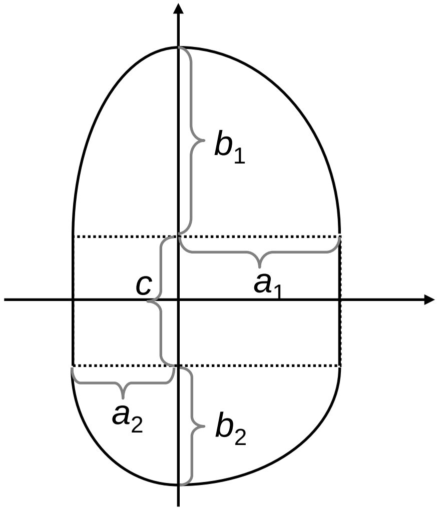
dx,dy,θ

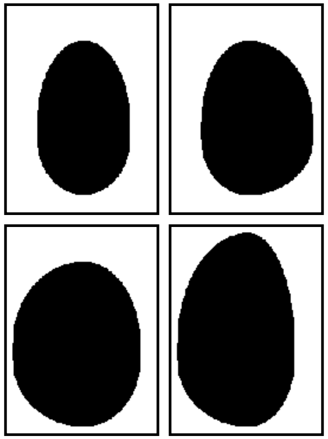
Noise probability

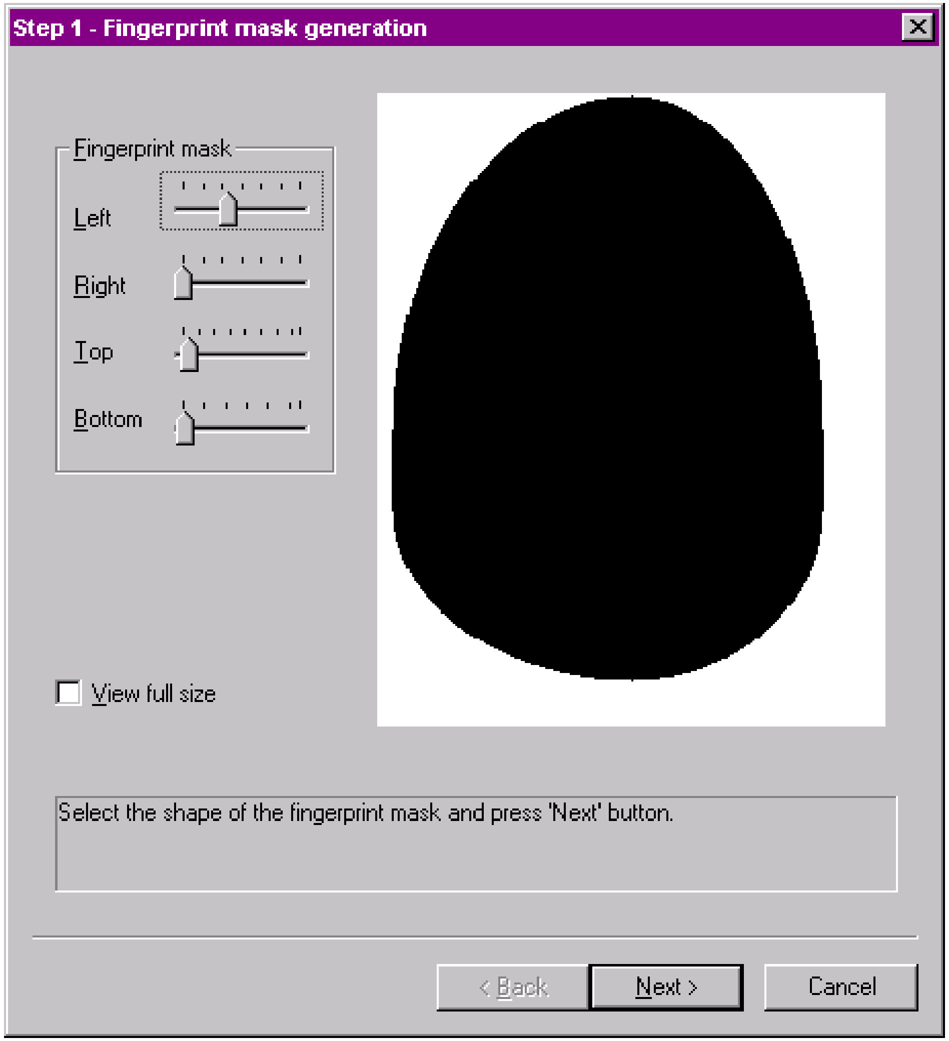
Deformation parameters

Fingerprint shape generation

SFinGe adopts a simple shape model based on elliptical segments

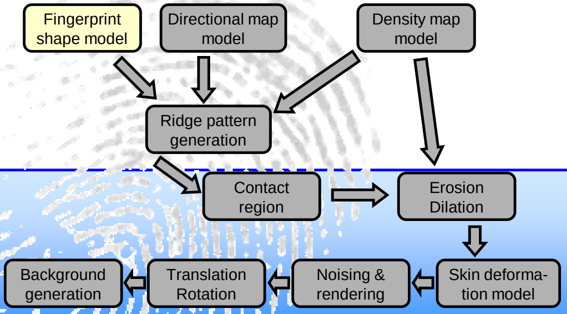






Step 1 - Fingerprint mask generation

Select the shape of the fingerprint mask and press 'Next' button.



Fingerprint shape model

Directional map model

Density map model

Ridge pattern generation

Contact region

Erosion Dilation

Skin deformation model

Noising & rendering

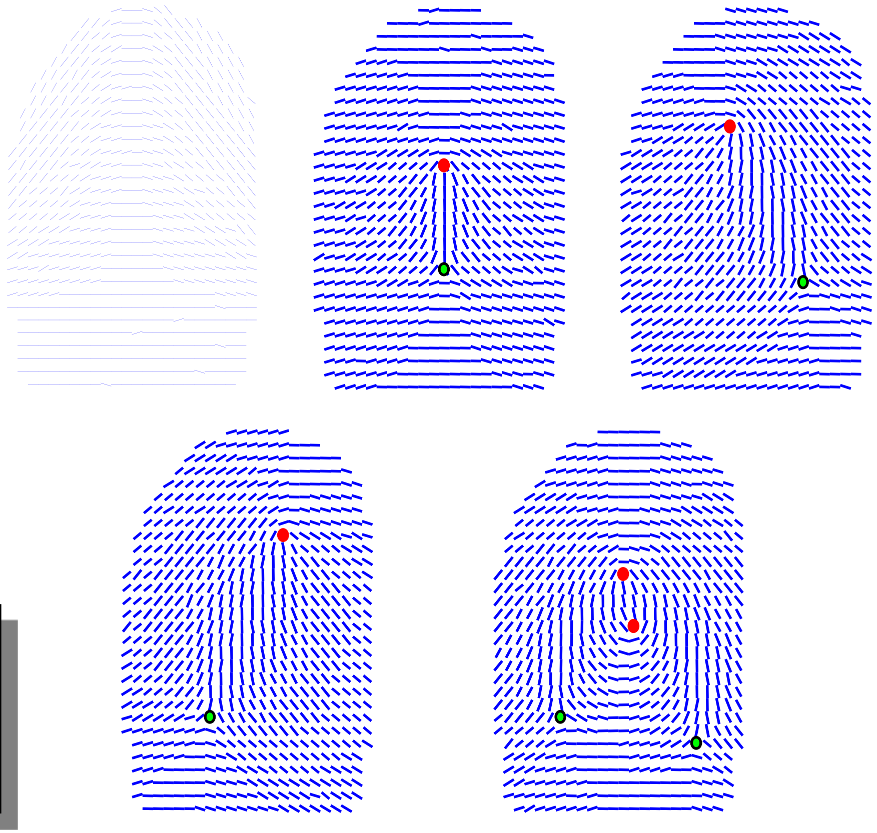
Translation Rotation

Background generation

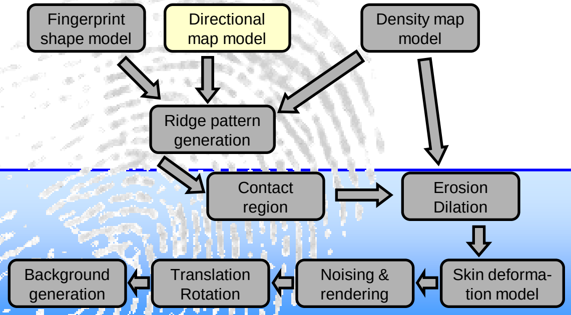
Directional map generation

Starting from the positions of the singularities, a mathematical ridge-flow model [Sherlock and Monro ’93] is applied to generate a consistent directional map

$$O(z)=O_0+\frac{1}{2}\left[\sum_{i=1}^{n_d} arg(z-\mathbf{d}_{\mathrm{i}})-\sum_{i=1}^{n_c} arg(z-\mathbf{c}_{\mathrm{i}})\right]$$



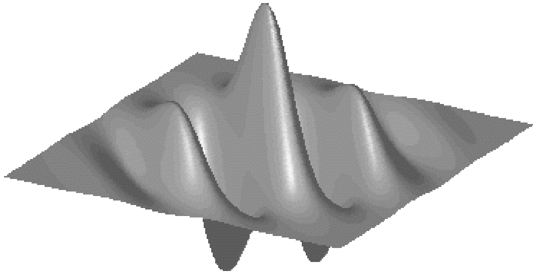
For the arch class, SFinGe uses a sinusoidal function whose frequency and amplitude control the arch curvature



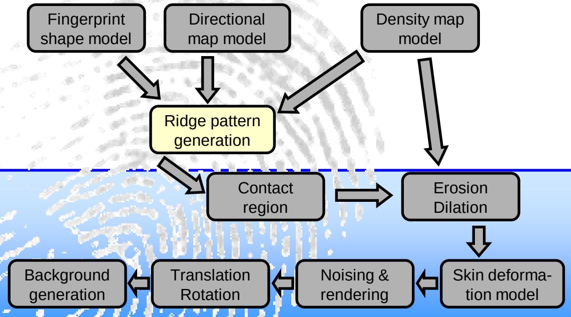
Ridge pattern generation (1)

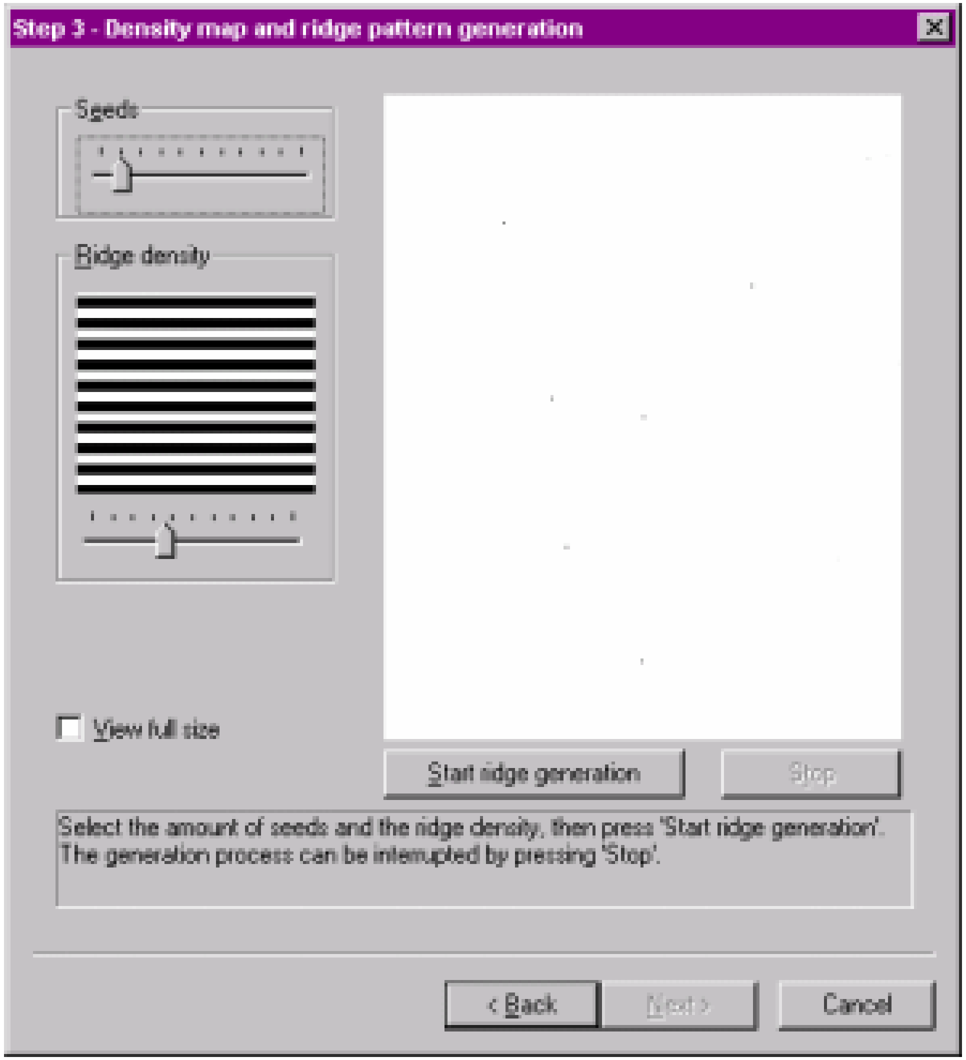
Gabor-like filters are iteratively applied to an initially-white image, enriched with few random points.

The filters orientation and frequency are locally adjusted according to the directional and density maps.



Realistic minutiae appear at random positions

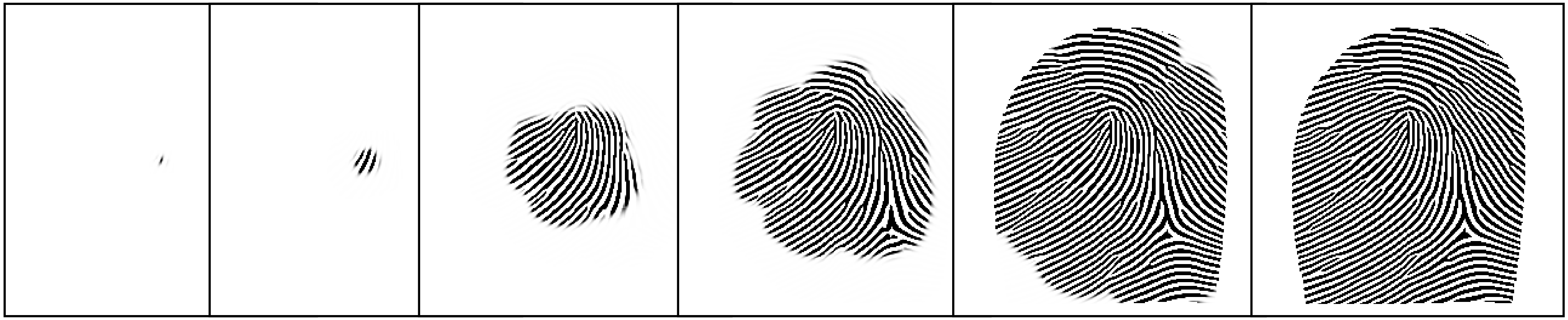


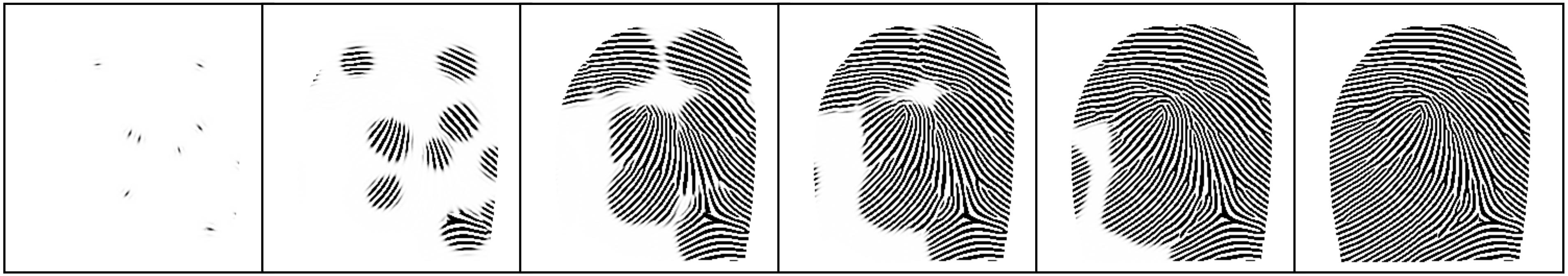


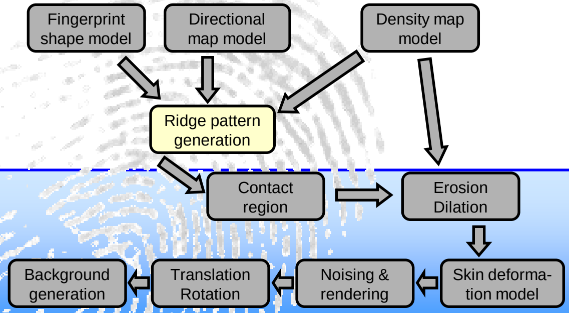
☐ View full size

Select the amount of seeds and the ridge density, then press 'Start ridge generation'.
The generation process can be interrupted by pressing 'Stop'.

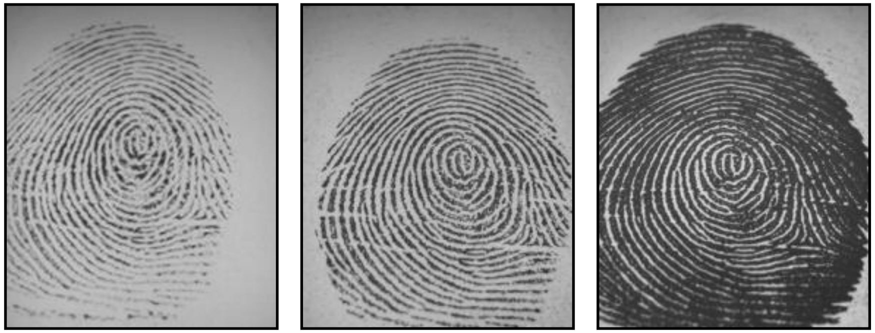
Ridge pattern generation (2)







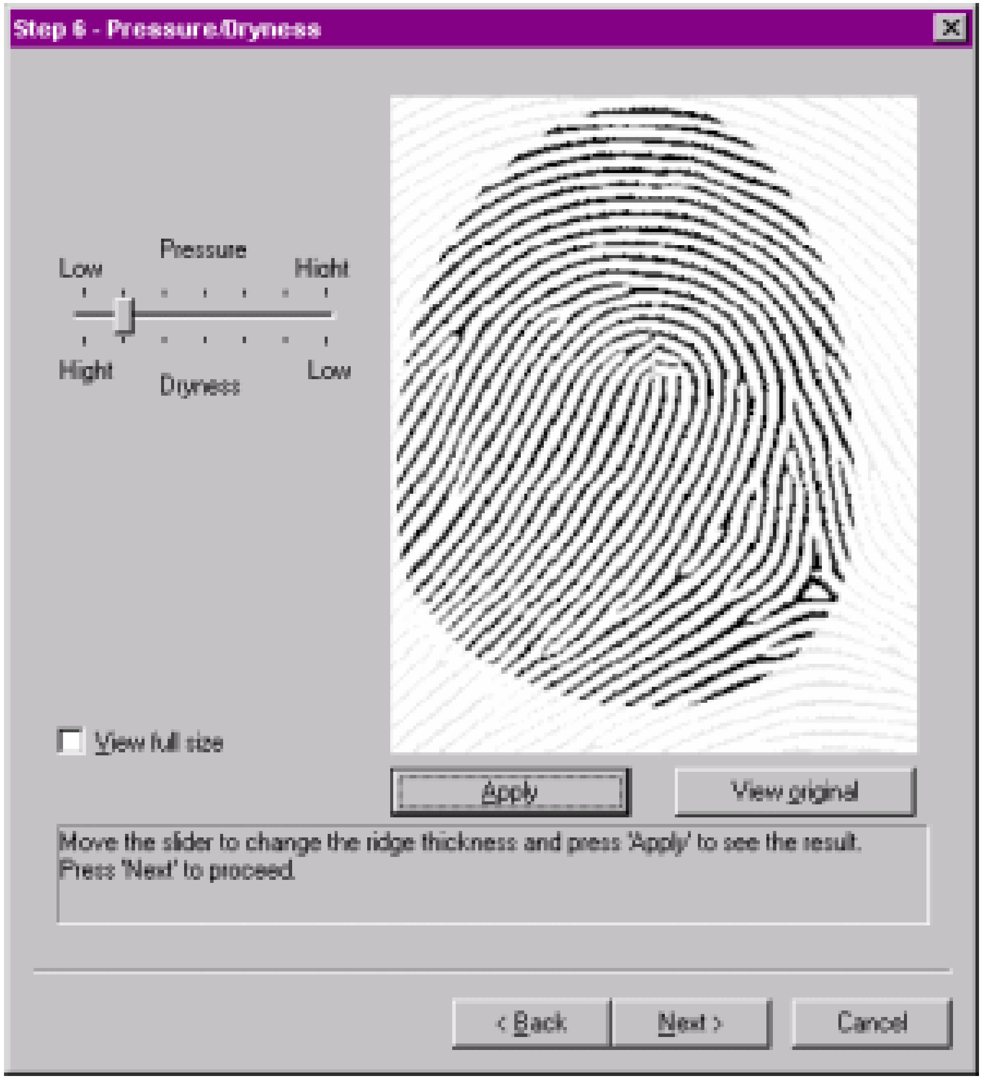
Ridge-line erosion and dilation (1)



Dry fingerprint

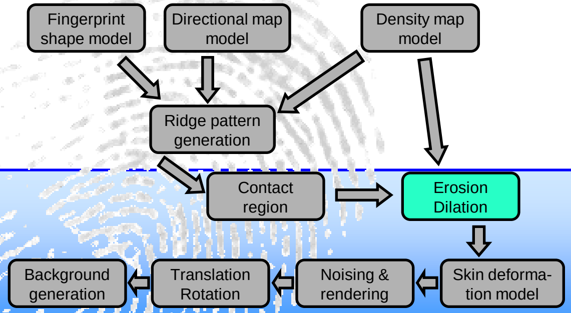
Wet fingerprint

- The erosion operator is applied to simulate dry skin
- The dilation operator is adopted to simulate wet skin



Step 6 - Pressure/Dryness

Move the slider to change the ridge thickness and press 'Apply' to see the result.
Press 'Next' to proceed.



Fingerprint shape model

Directional map model

Density map model

Ridge pattern generation

Contact region

Erosion Dilation

Skin deformation model

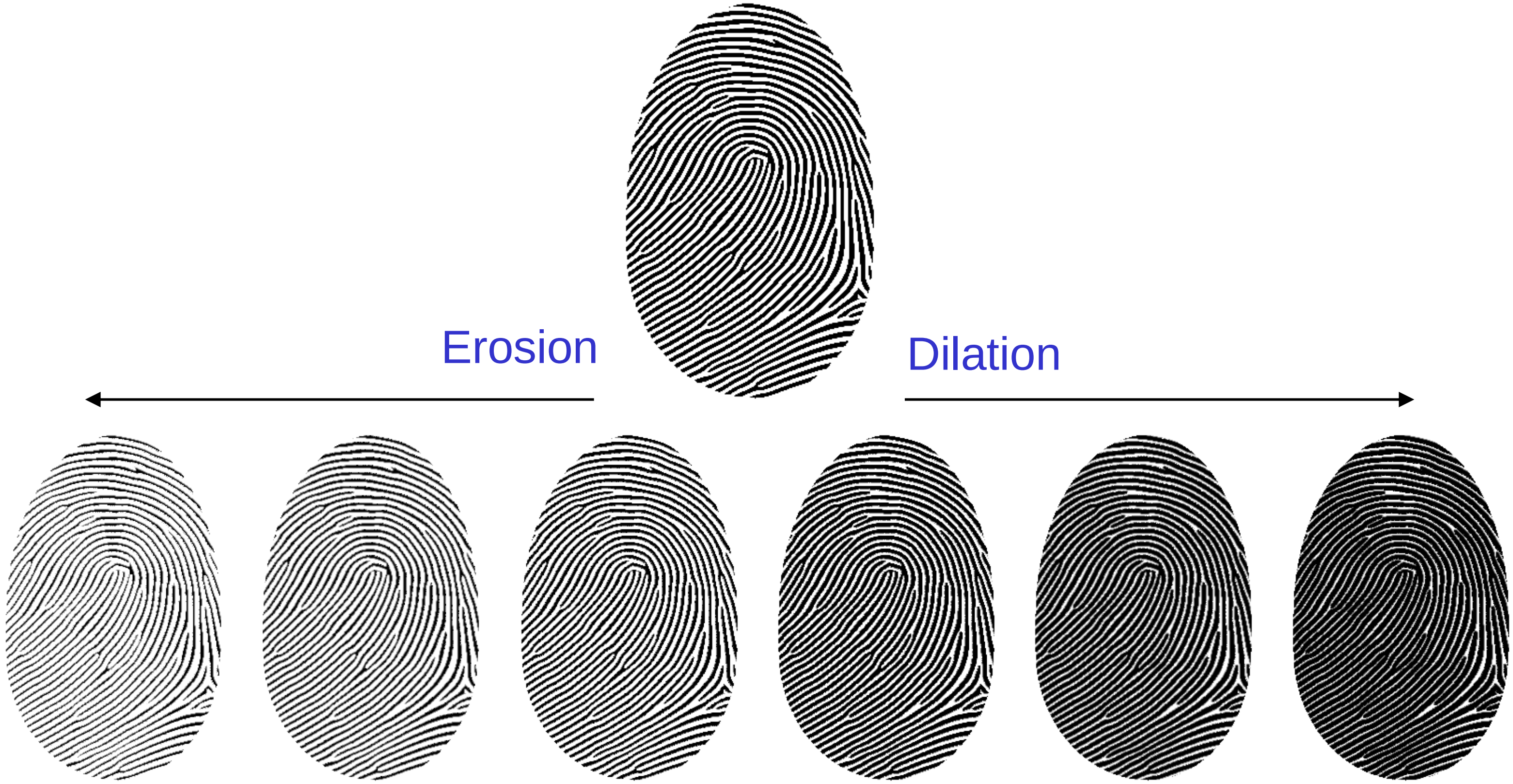
Noising & rendering

Translation Rotation

Background generation

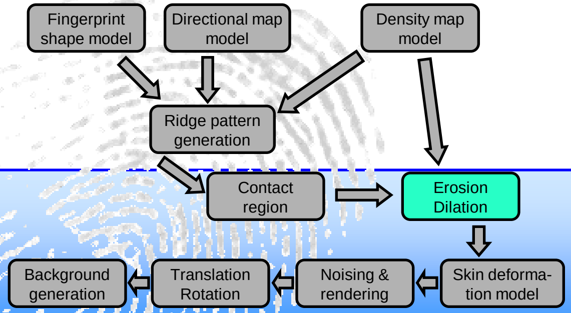
Ridge-line erosion and dilation (2)

Original image



Erosion

Dilation



Fingerprint shape model

Directional map model

Density map model

Ridge pattern generation

Contact region

Erosion Dilation

Skin deformation model

Noising & rendering

Translation Rotation

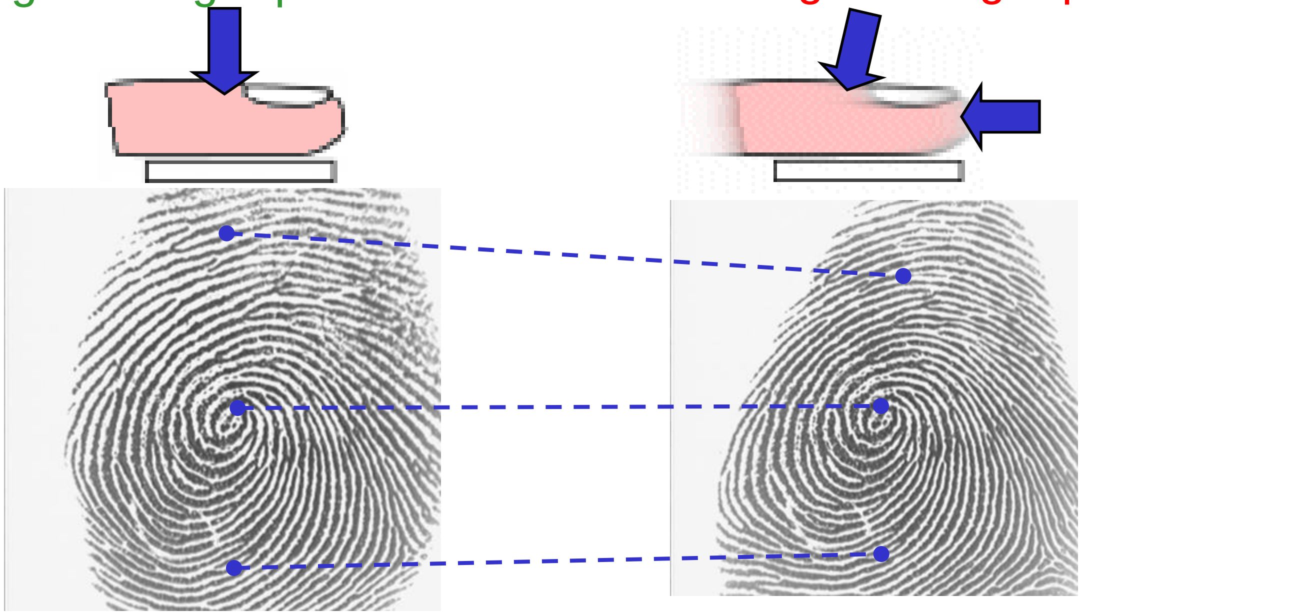
Background generation

Skin distortion

One of the main factors that contribute to make substantially different the impressions of a given finger is skin distortion

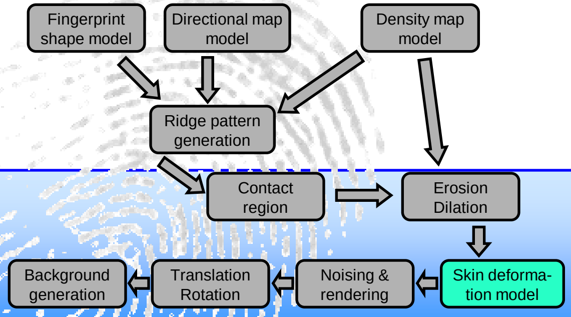
Orthogonal finger placement

Non-orthogonal finger placement



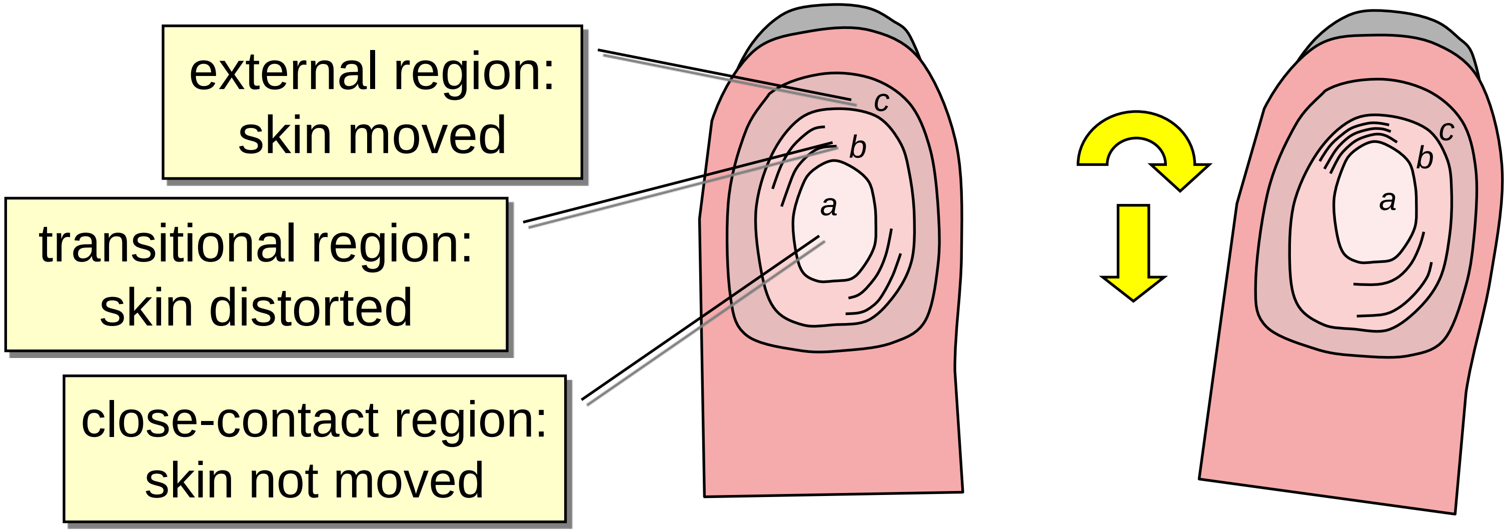
Non-distorted fingerprint

Distorted fingerprint



Skin distortion model (1)

The finger pressure against the sensor is not uniform, but decreases moving from the center towards the borders.



external region: skin moved

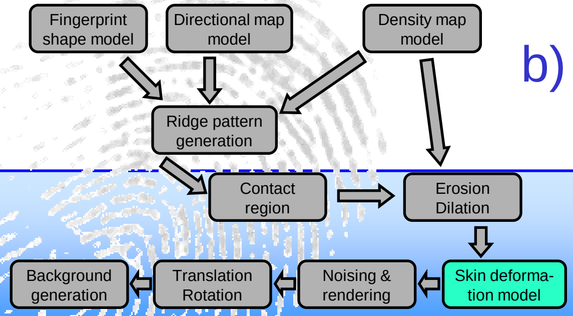
transitional region: skin distorted

close-contact region: skin not moved

a) close-contact region, where the high pressure and the surface friction does not allow any skin slippage

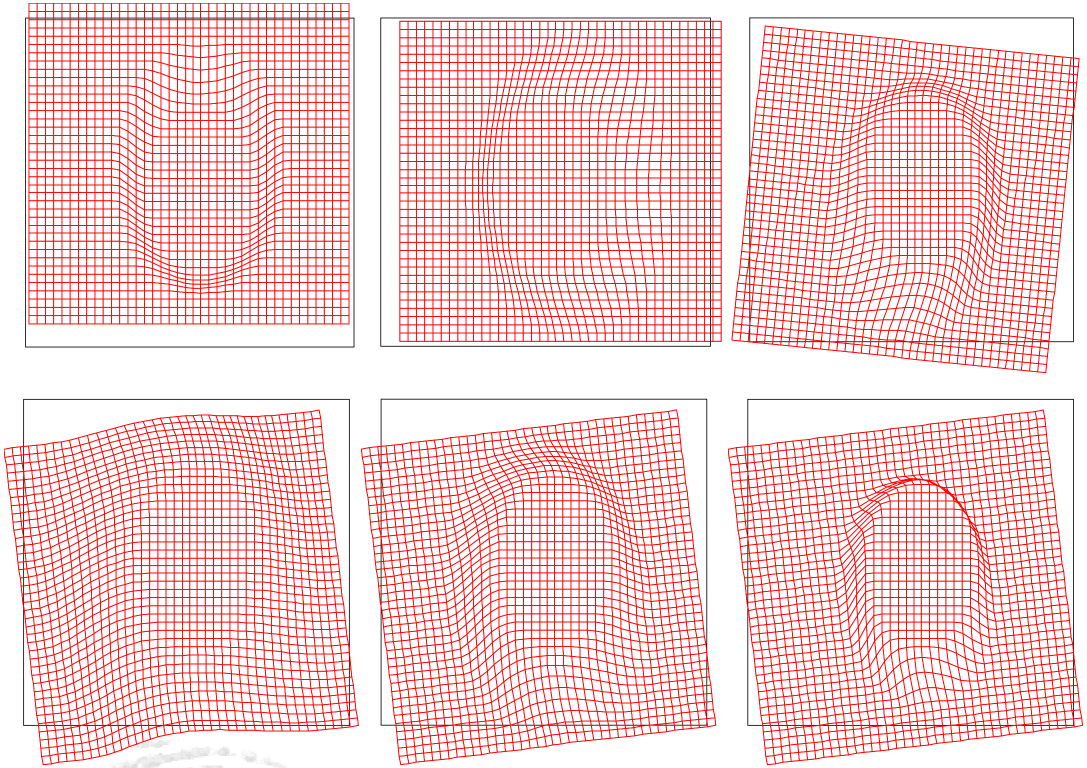
c) external region, where the low pressure allows the skin to be dragged by the finger movement

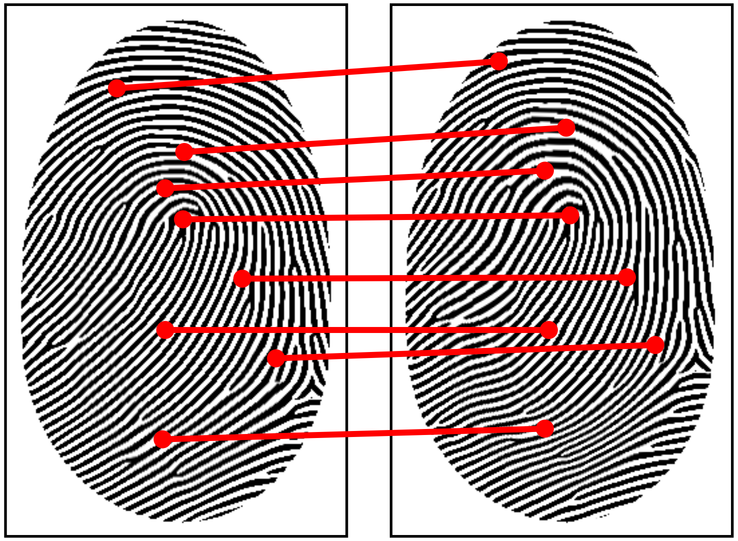
b) transitional region where an elastic distortion is produced to smoothly combine regions *a* and *c*



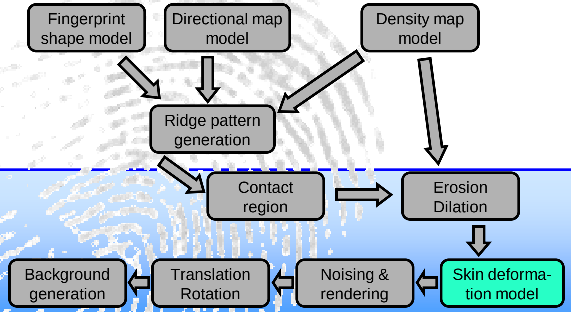
Skin distortion model (2)

$$distortion : \Re^2 \rightarrow \Re^2 \,,\ distortion(\mathbf{v}) = \mathbf{v} + \Delta(\mathbf{v}) \cdot brake(shapedist_a(\mathbf{v}), k)$$



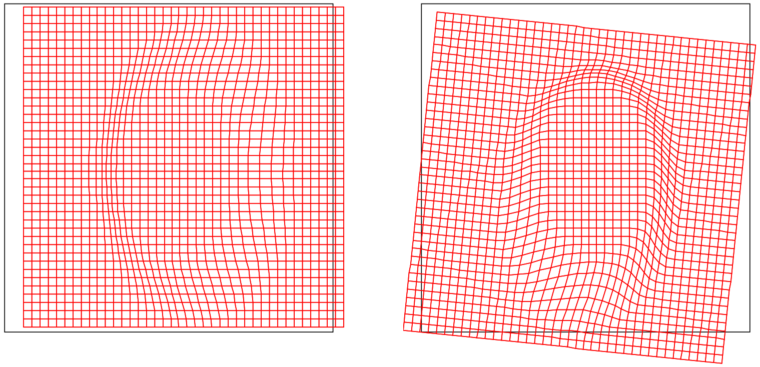


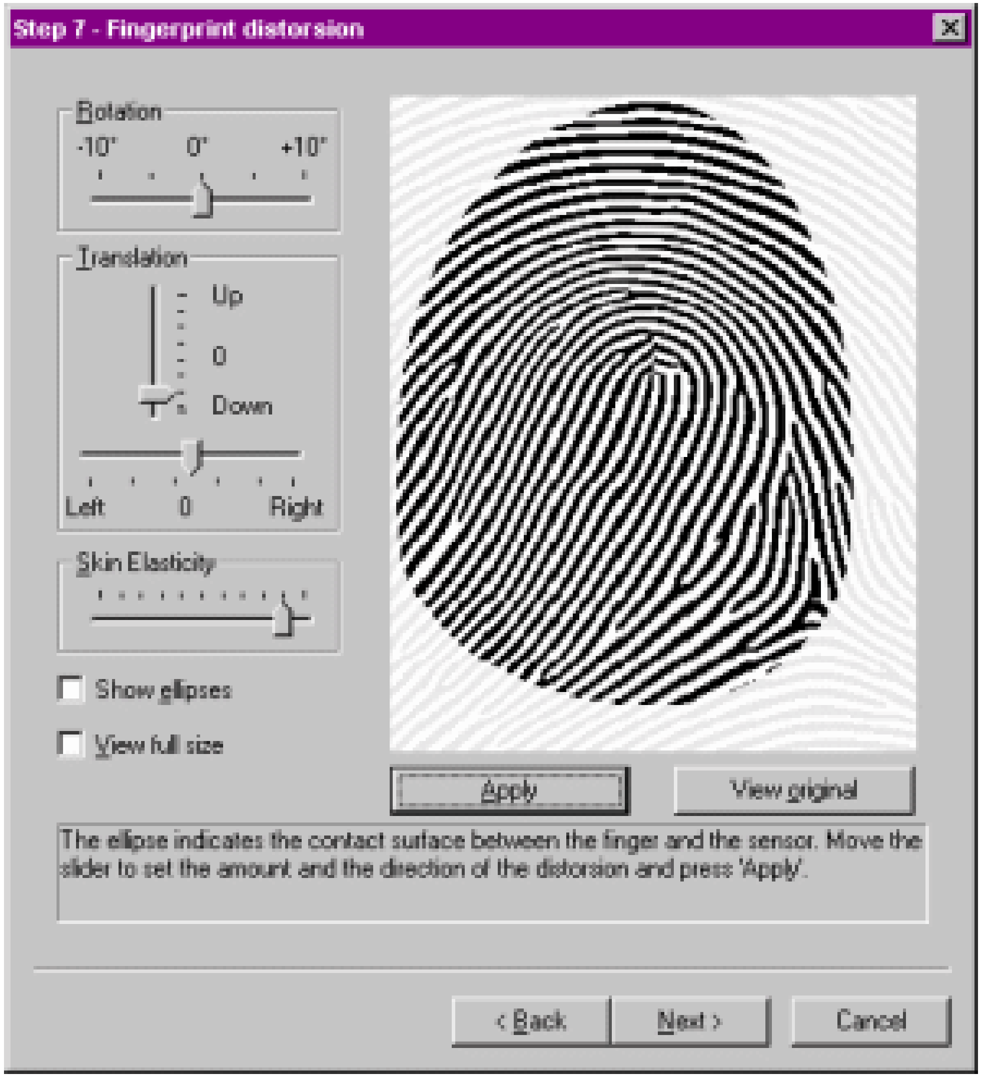
Original image Distorted image



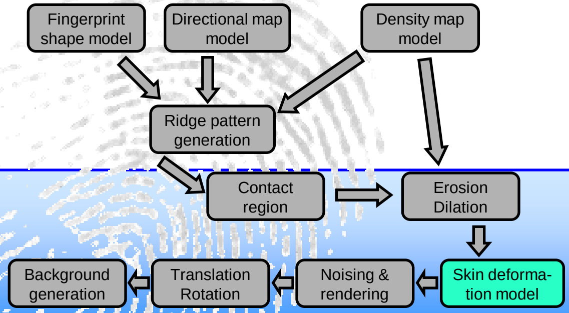
Skin distortion model (3)

The skin distortion model is applied to randomly generate realistic impressions of the same "synthetic finger"





The ellipse indicates the contact surface between the finger and the sensor. Move the slider to set the amount and the direction of the distorsion and press 'Apply'.

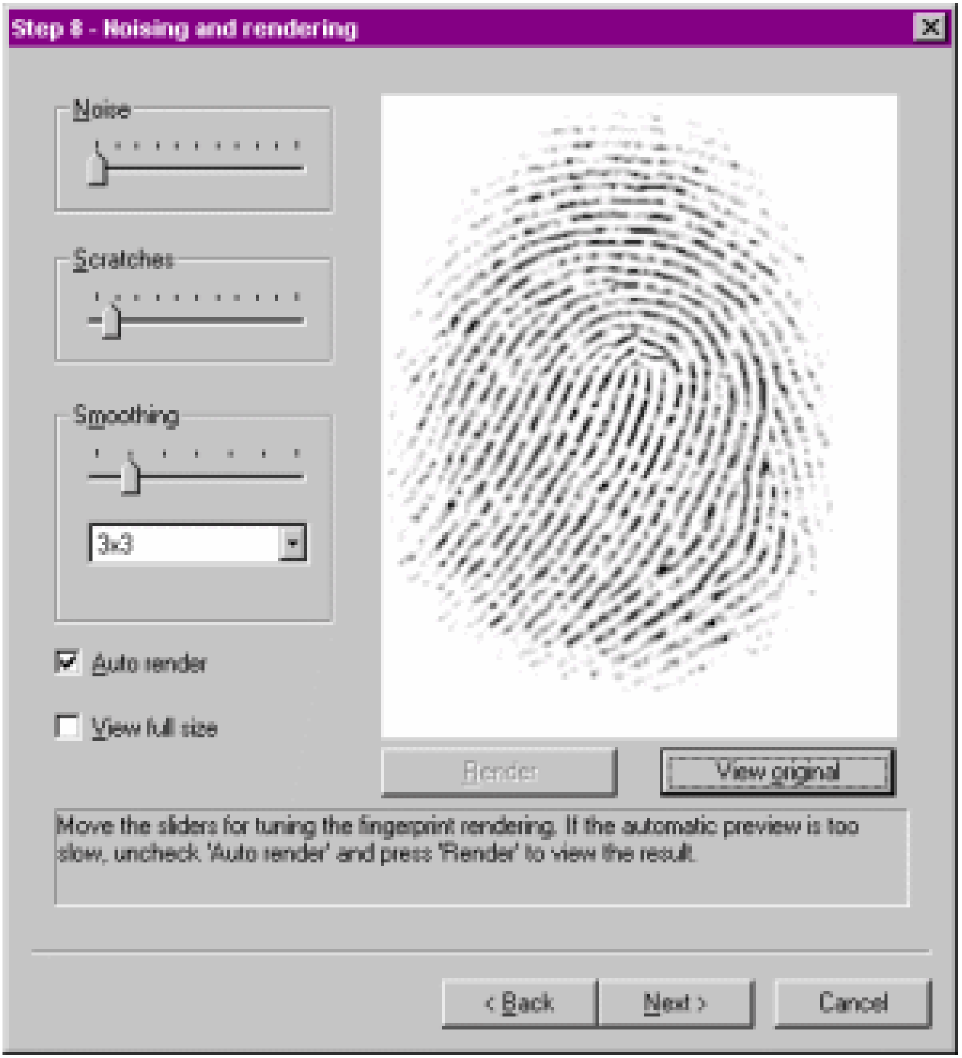


Noising and rendering

Several factors contribute to deteriorate the quality of real fingerprints:

- irregularity of the ridges and their different contact with the sensor surface
- small cuts or abrasions on the fingertip
- presence of small pores within the ridges

SFinGe adds specific noise and applies an ad-hoc smoothing process to simulate real-fingerprints irregularities



Step 8 - Noising and rendering

Noise

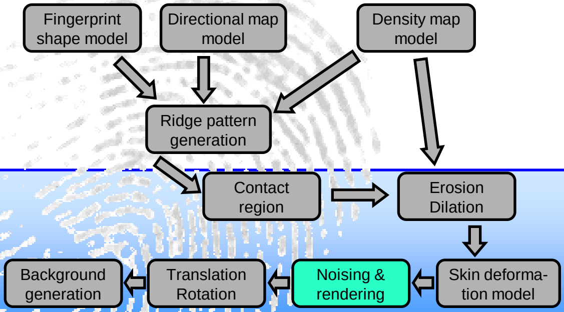
Scratches

Smoothing: 3x3

☑ Auto render

□ View full size

Move the sliders for tuning the fingerprint rendering. If the automatic preview is too slow, uncheck 'Auto render' and press 'Render' to view the result.



Fingerprint shape model

Directional map model

Density map model

Ridge pattern generation

Contact region

Erosion Dilation

Skin deformation model

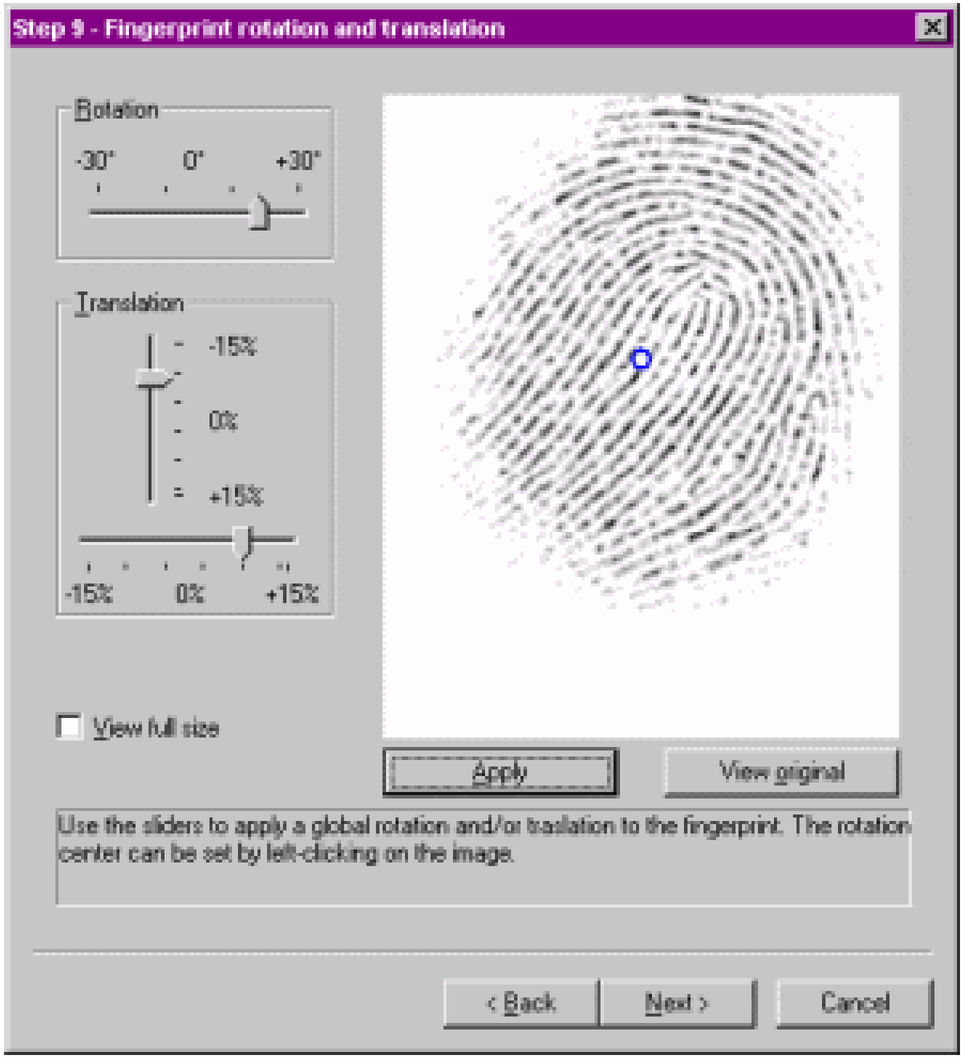
Noising & rendering

Translation Rotation

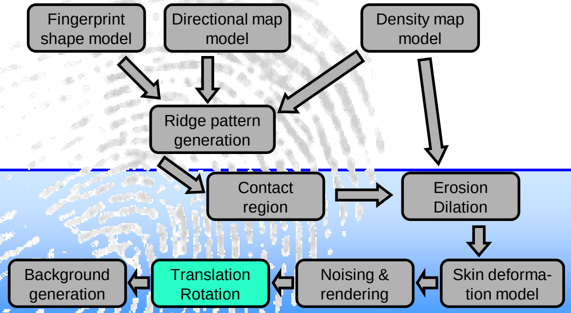
Background generation

Rotation and translation

The image generated is randomly rotated and translated, to simulate real fingerprints, which usually are not perfectly centered and can present a certain amount of rotation



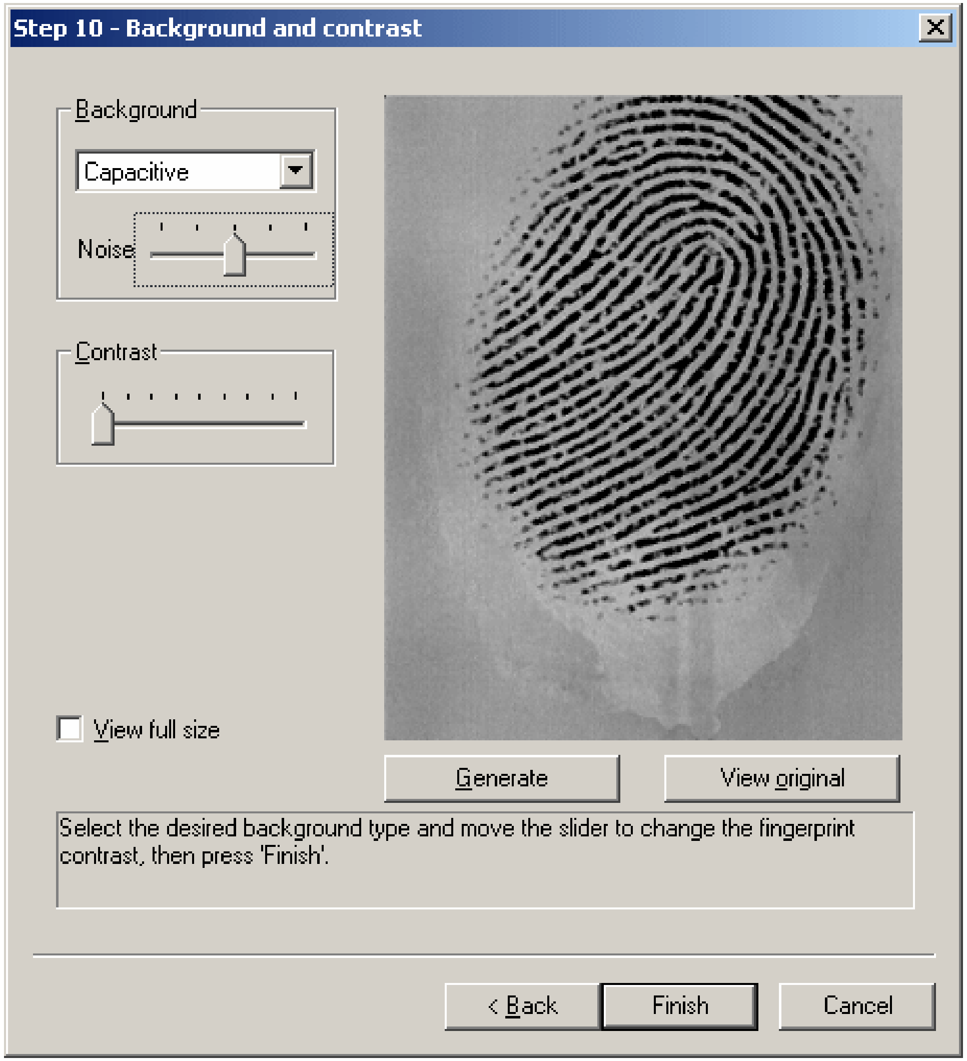
Use the sliders to apply a global rotation and/or traslation to the fingerprint. The rotation center can be set by left-clicking on the image.



Background generation

A mathematical model based on the KL transform is applied to generate a realistic background, which is placed "behind" the fingerprint

Different background models can be created to simulate different acquisition technologies (e.g. optical, capacitive, ...)



Step 10 - Background and contrast

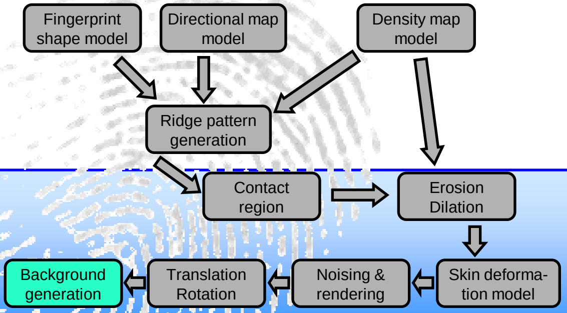
Background: Capacitive

Noise

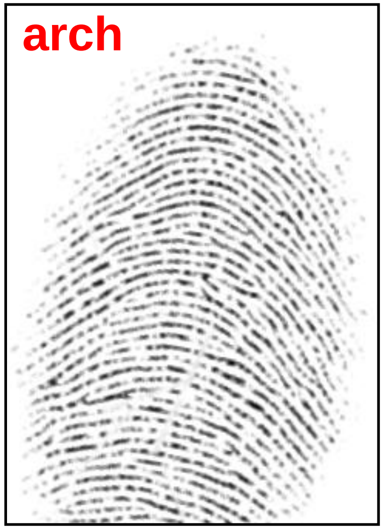
Contrast

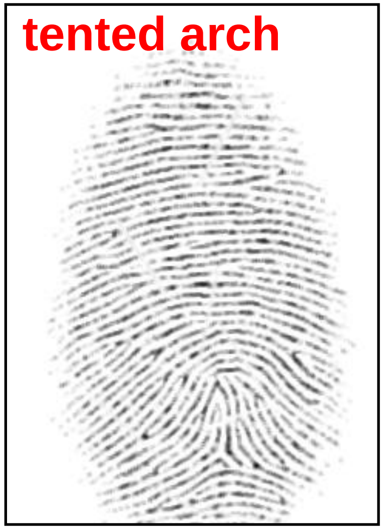
□ View full size

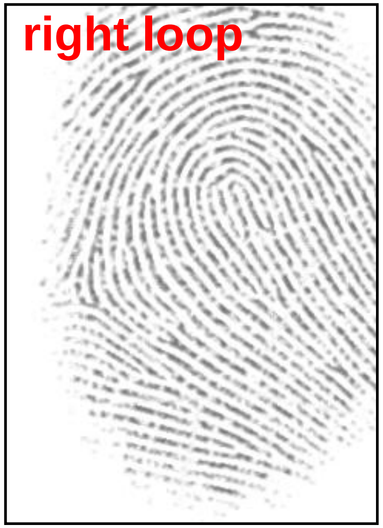
Select the desired background type and move the slider to change the fingerprint contrast, then press 'Finish'.

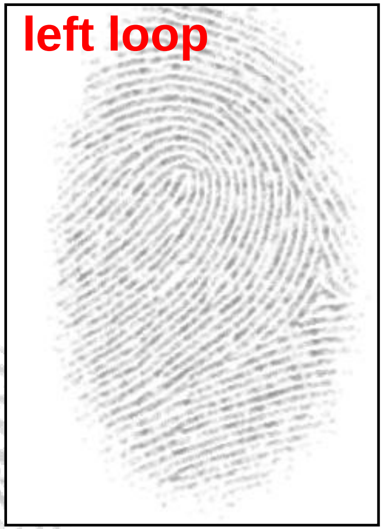


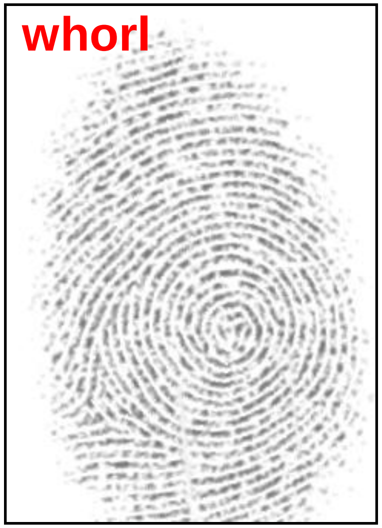
Examples (1)

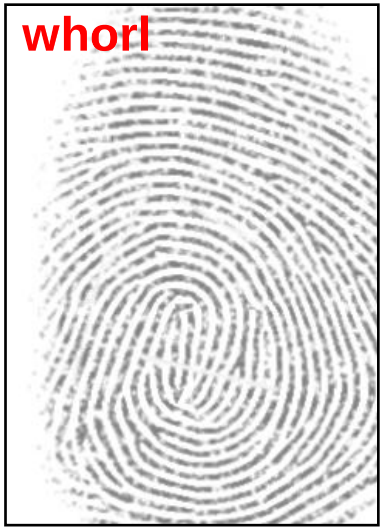

arch


tented arch


right loop


left loop

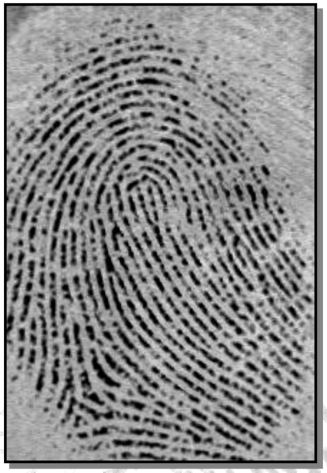

whorl


whorl

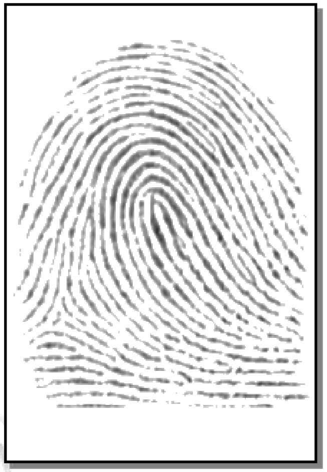
Model validation (1)

Fingerprint images generated by SFinGe appear very realistic

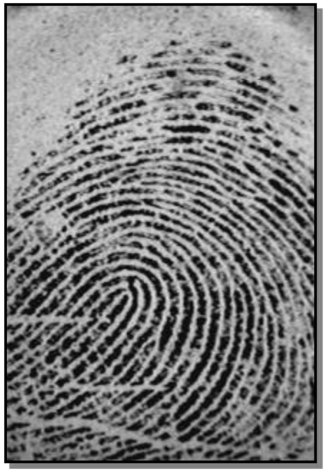
About 90 people (many of them having a good background in fingerprint analysis) have been asked to find a synthetic fingerprint image among 4 images (3 of which were real fingerprints).
The synthetic image proved to be not distinguishable from the others



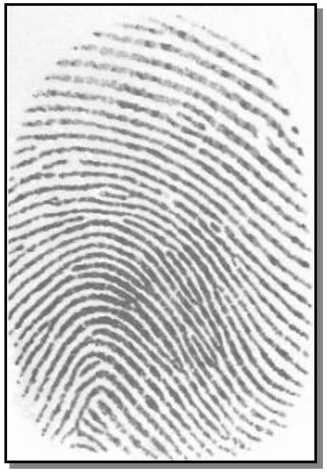
A



B



C



D

Poll results	
A	23%
B	27%
C	21%
D	29%

Model validation (2)

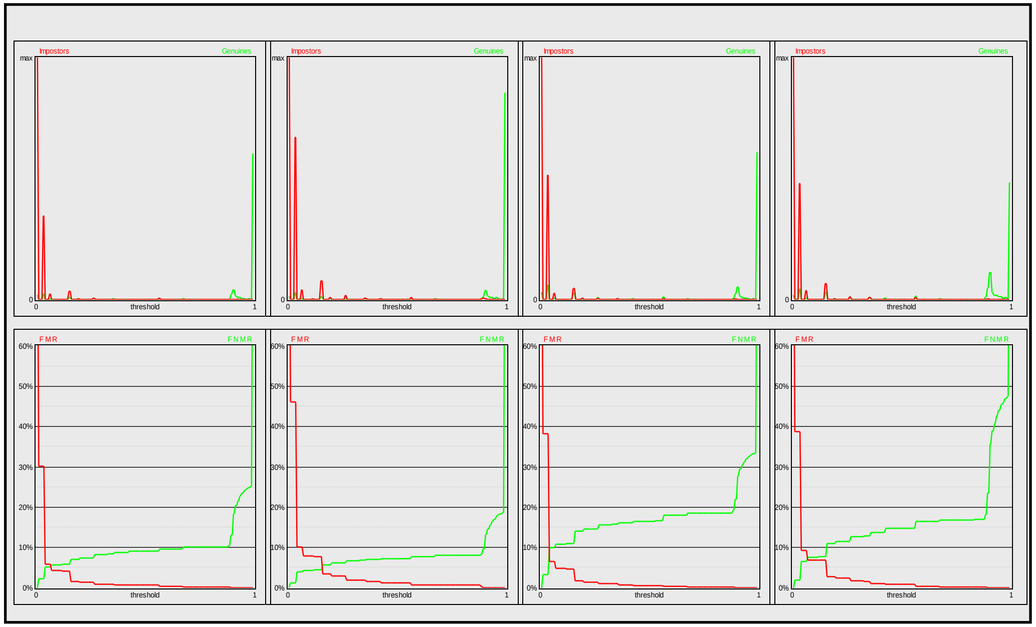
FVC2000

Fingerprint Verification Competition

A test has been performed in conjunction with FVC2000, where one of the four DB used (DB4) was synthetically generated by SFinGe:

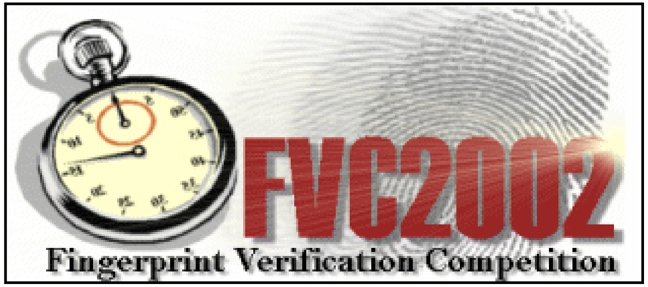
- The participant algorithms performed on DB4 similarly to the other DBs
- The genuine/impostor distributions and the ROC curves are also very close

DB1 (real) DB2 (real) DB3 (real) DB4 (Synthetic)



This proves that the main inter-class and intra-class variation of fingerprints in nature are well captured by SFinGe

Model validation (3)



A more systematic analysis was performed on FVC2002 results.

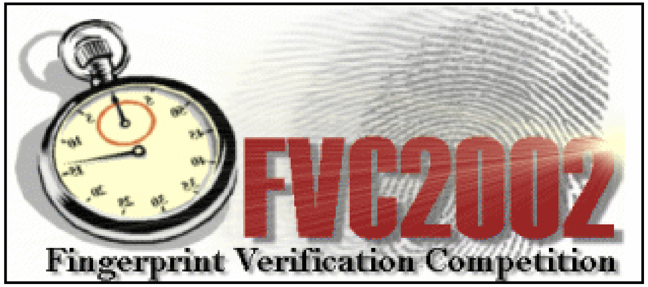
$$RRD_i(j) = \frac{\left|R_{i1}(j) - R_{i2}(j)\right| + \left|R_{i1}(j) - R_{i3}(j)\right| + \left|R_{i2}(j) - R_{i3}(j)\right|}{3}$$

is the *average ranking difference of algorithm i according to indicator j, among the three real databases; indicates how stable is the performance of algorithm i (according to indicator j) over the three databases*

$$SRD_i(j) = \frac{\left|R_{i4}(j) - R_{i1}(j)\right| + \left|R_{i4}(j) - R_{i2}(j)\right| + \left|R_{i4}(j) - R_{i3}(j)\right|}{3}$$

is the *average ranking difference of algorithm i according to indicator j, between the synthetic database and each of the real database; denotes the amount of variation between synthetic and real databases*

Model validation (4)

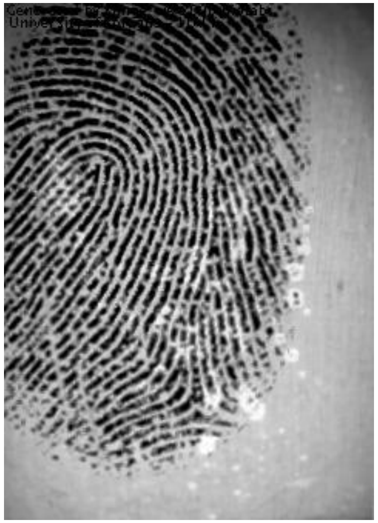


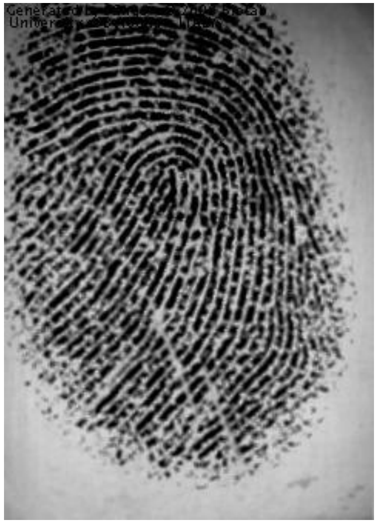
The results are quite surprising !

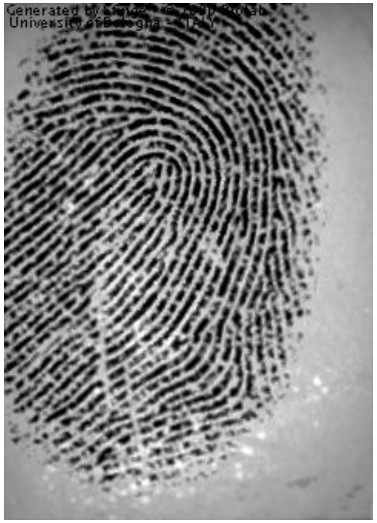
The difference between DB4 (the synthetic DB) and the others are even smaller than the inter-difference among the three real databases.

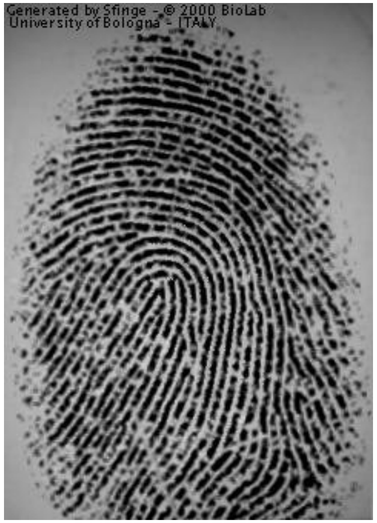
	EER		ZeroFMR		FMR1000		FMR100	
	$RRD_i^{(1)}$	$SRD_i^{(1)}$	$RRD_i^{(2)}$	$SRD_i^{(2)}$	$RRD_i^{(3)}$	$SRD_i^{(3)}$	$RRD_i^{(4)}$	$SRD_i^{(4)}$
Average	2.84	2.65	3.14	2.74	2.58	2.58	2.69	2.59
Max	8.67	11.33	11.33	7.67	7.33	5.67	8.00	10.67
Min	0.00	0.00	0.67	0.33	0.00	0.33	0.00	0.33
St. Dev.	2.51	2.43	2.35	1.76	1.94	1.45	2.15	2.36

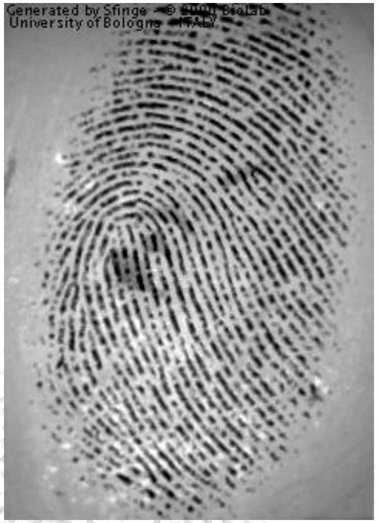
Examples (2)

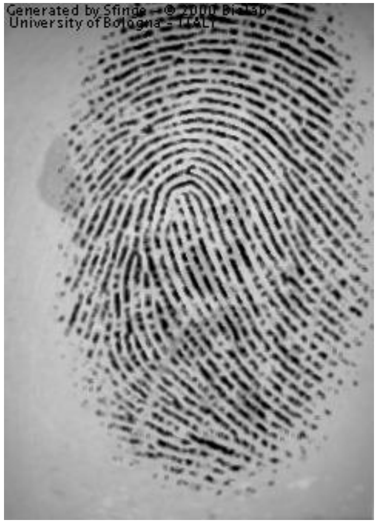

Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY

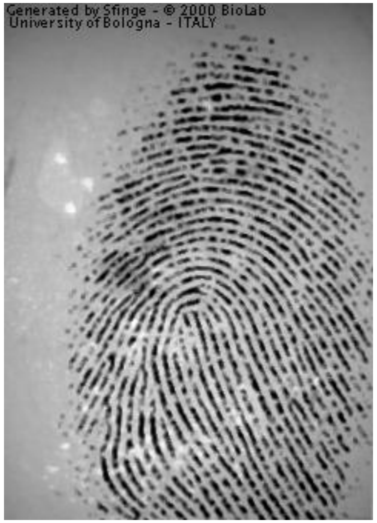

Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY

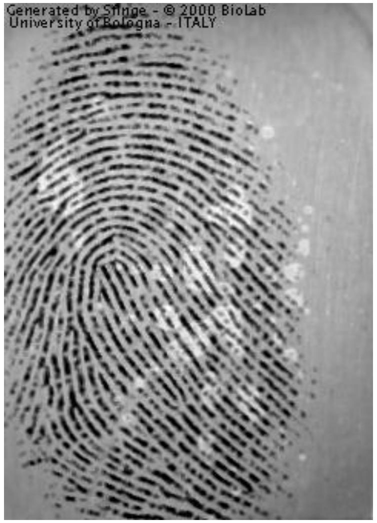

Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY


Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY

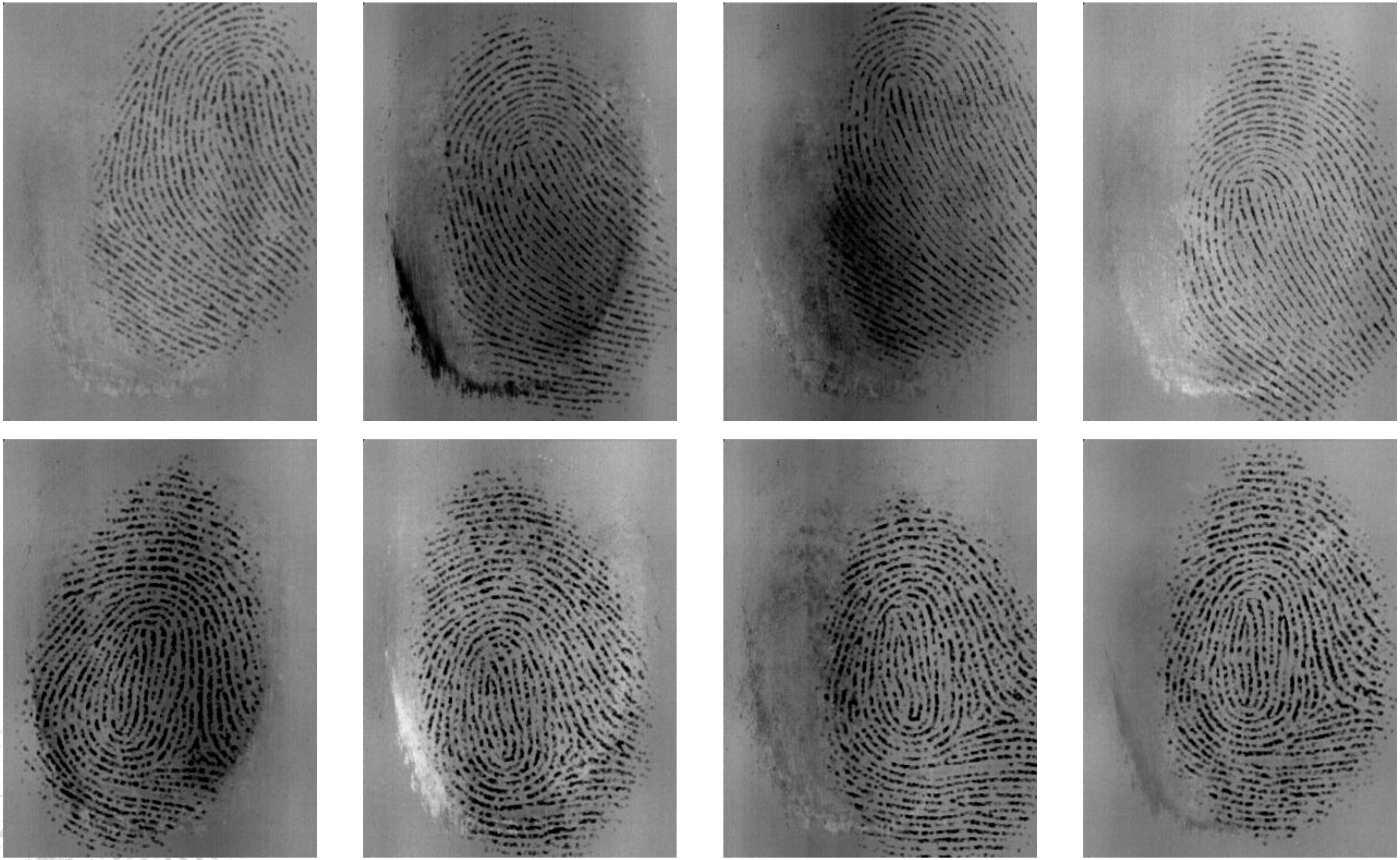

Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY


Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY


Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY

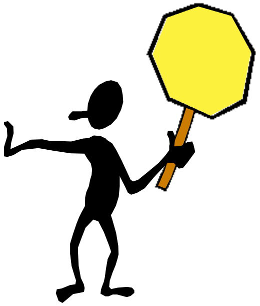

Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY

Examples (3)



Applications (1)

SFinGe is an effective technique to overcome the problem of collecting large fingerprint databases for training and testing purposes



When the performance of a fingerprint recognition system has to be measured referring to a given real environment, synthetic fingerprints cannot be used and real-fingerprint databases must be collected.

Several research groups and companies are currently using SFinGe to:

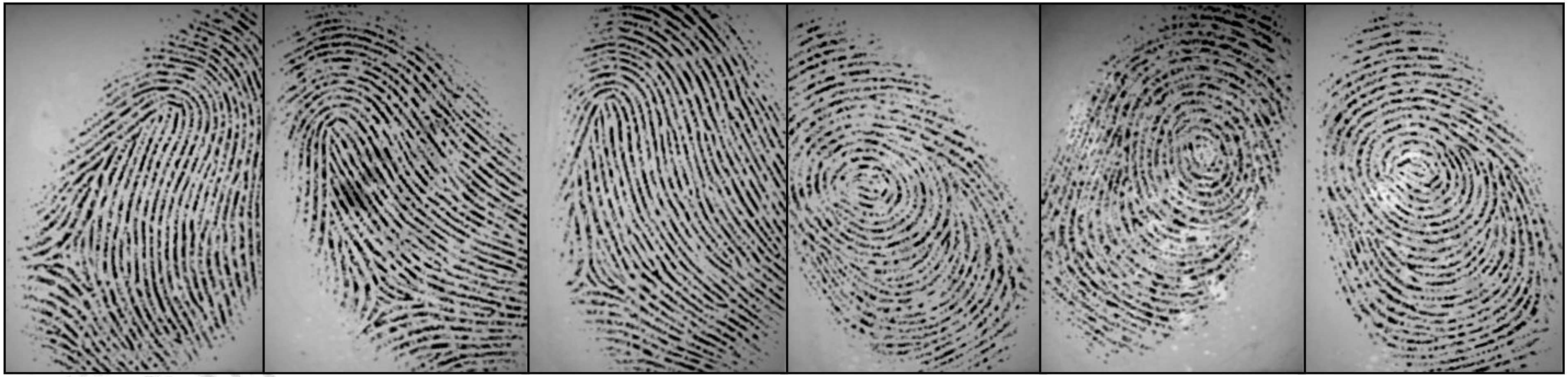
- compare different fingerprint matching algorithms
- train pattern recognition techniques that require large learning-sets (e.g. neural network, PCA,...)
- easily generate a large number of “virtual users” to develop and test medium/large-scale fingerprint-based systems (e.g. AFIS)

Applications (2)

Some possible scenarios:

1. A new fingerprint verification algorithm has to be tested to measure its robustness against fingerprint rotation

SFinGe can be used to generate some databases with an increasingly amount of rotation among different samples of the same fingers

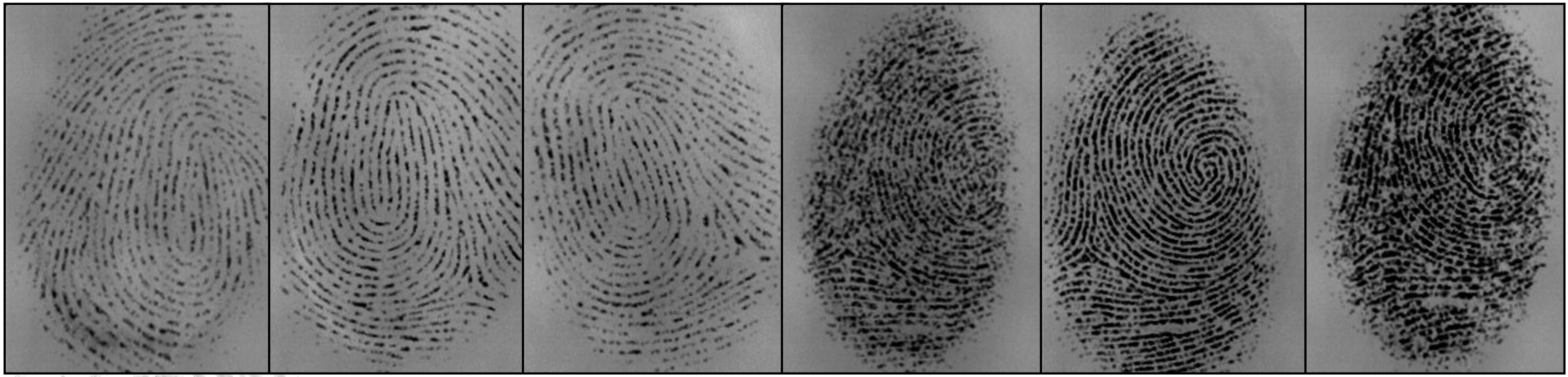


Applications (3)

Some possible scenarios:

2. Several algorithms have to be compared to evaluate which is the least affected by skin distortion and noise

SFinGe is able to automatically generate databases of highly distorted and/or noisy fingerprints

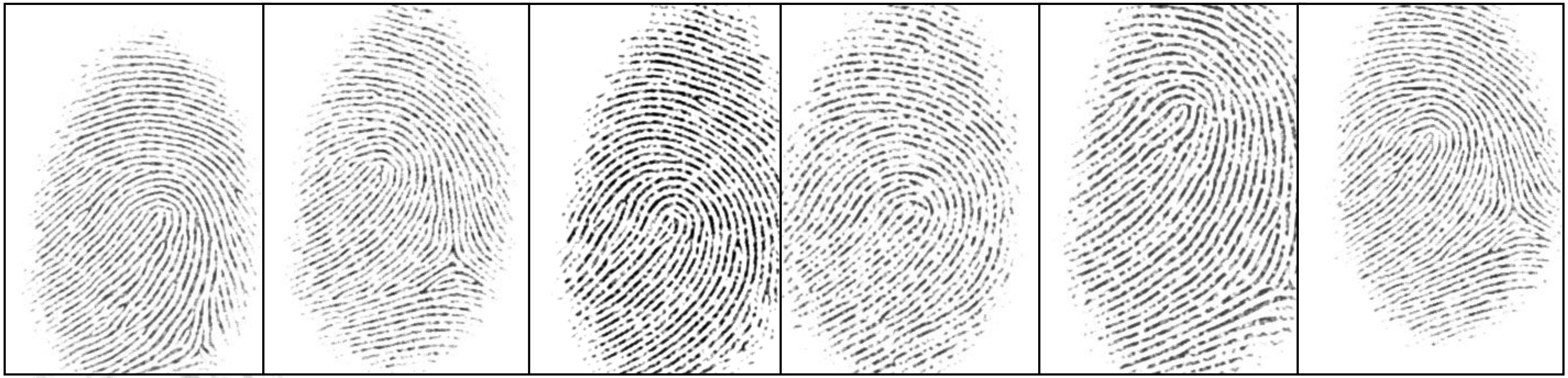


Applications (4)

Some possible scenarios:

3. In order to improve the performance of a fingerprint classification approach, a large set of high-quality fingerprints belonging to the left-loop class is needed

SFinGe can be tuned to generate a database of high-quality fingerprints belonging to a single class



Improving SFinGe noise model (1)

Noise is added in the form of small blobs of variable size and shape.

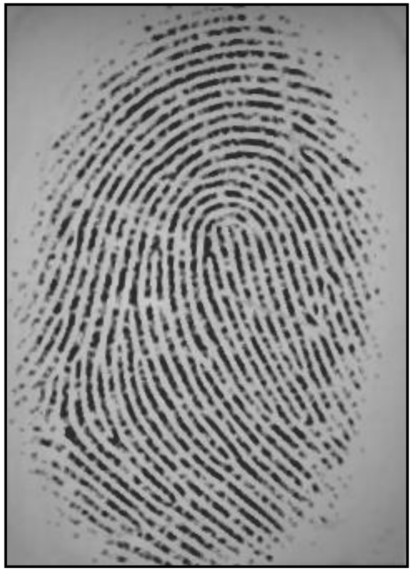
The probability of adding a noise blob at pixel (x,y) is:

$$p_n(x,y) = p_L \cdot \left(1 + d(x,y)^3\right)$$

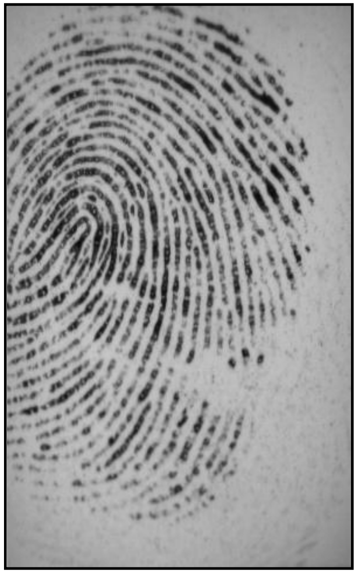
$$d(x,y) = \begin{cases} 0 & \text{if } d_B(x,y) > t_B \\ \left(t_B - d_B(x,y)\right)/t_B & \text{otherwise} \end{cases}$$

Main limitation:

➔It distributes the noise uniformly over the entire fingerprint area, except for the borders where the amount of noise gradually increases.



Synthetic

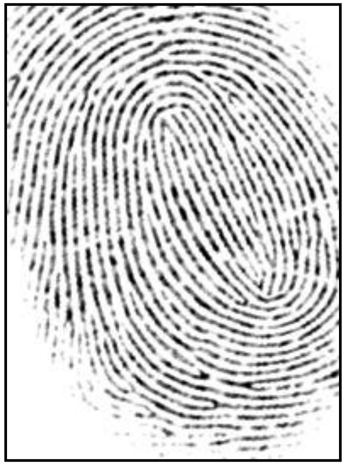


Real

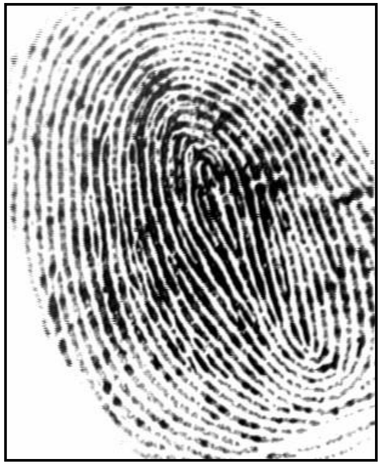
Improving SFinGe noise model (2)

Generating uniform noise using a simple random generator is not appropriate:

➔the result is a noise function that changes too abruptly over the space (non-coherent noise)



Synthetic



Real

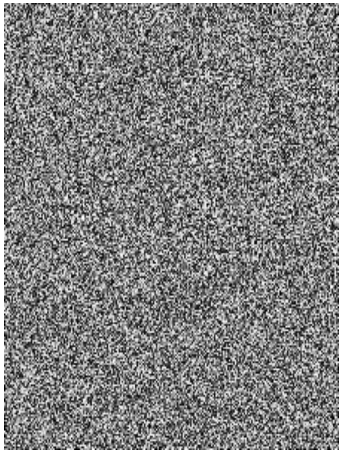
We need a noise function that:

➔is able to produce coherent noise

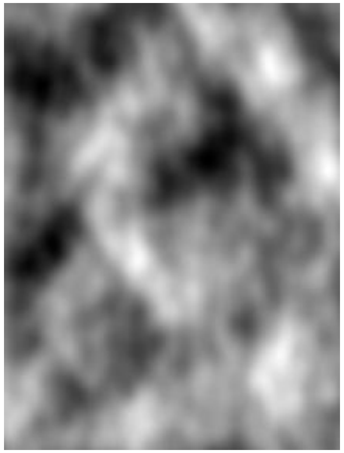
➔can be efficiently computed

A practical solution:

Perlin Noise function



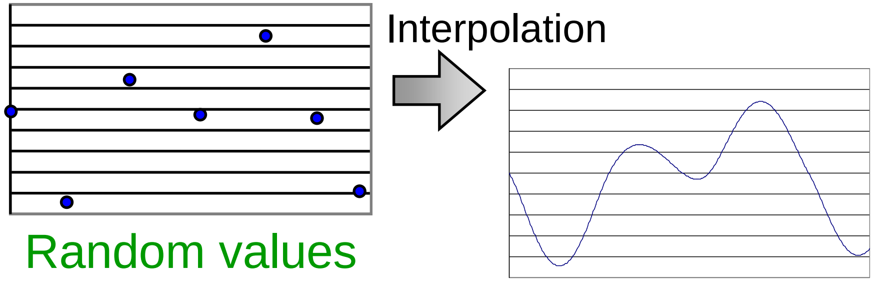
Non-coherent noise



Coherent noise

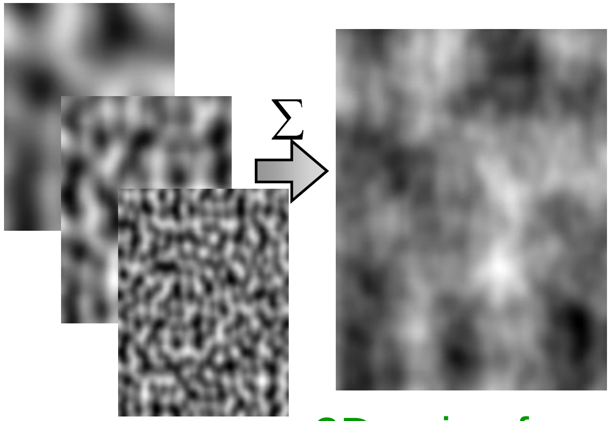
Perlin noise

Perlin Noise (K. Perlin, 1985)


Interpolation

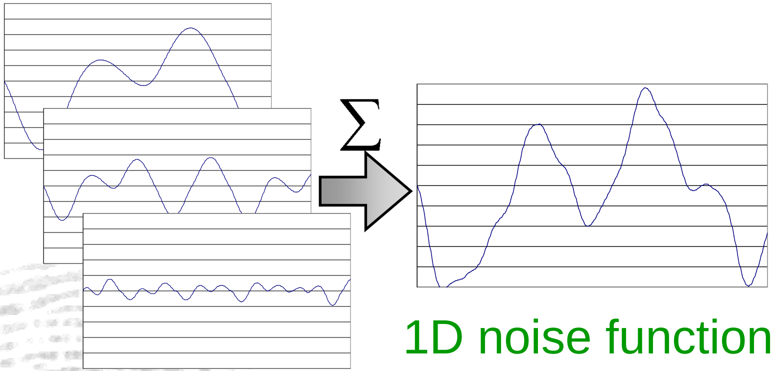
Random values

Random wave



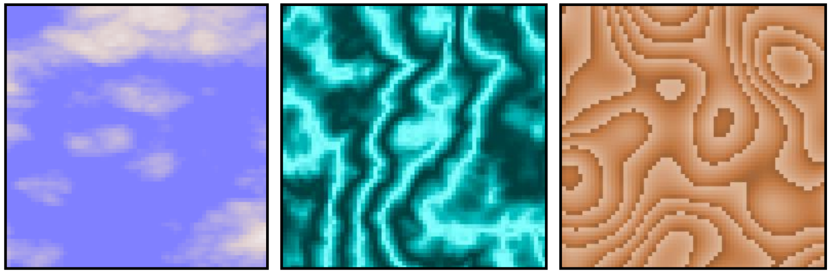
2D random waves

2D noise function



1D random waves

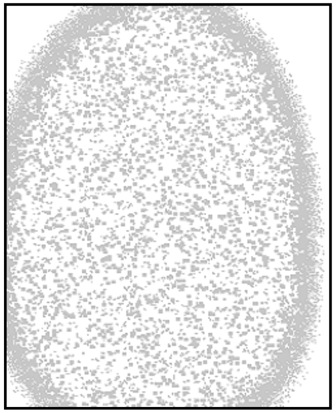
1D noise function


Sample textures generated by Perlin noise

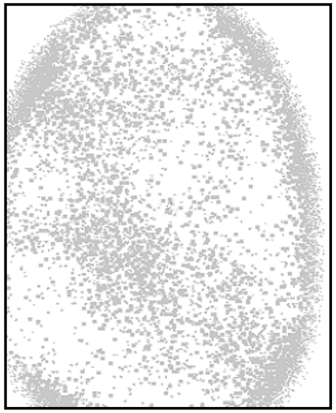
The improved noise model

Probability of adding a noise blob at pixel (x,y):

$$p'_n(x,y) = p_n(x,y) \cdot (1+noise_{01}(x,y))^k$$

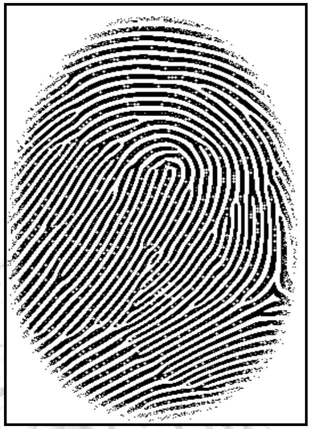


Previous model



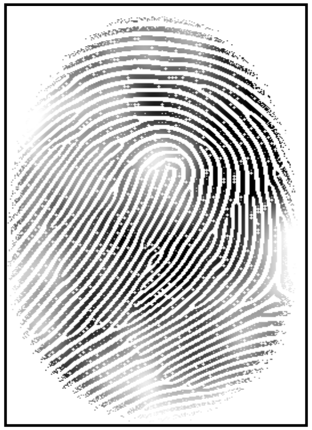
New model

Further noising step:



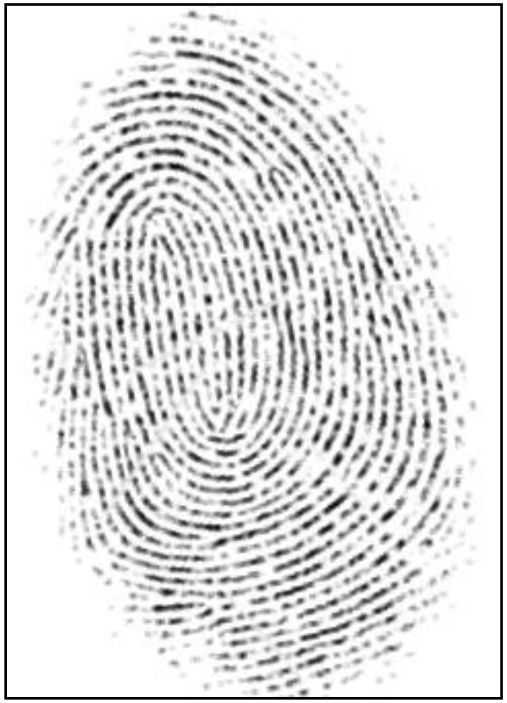
Before

$$g(x,y) = min(255, g(x,y) + 255 \cdot noise_{01}(x,y))$$

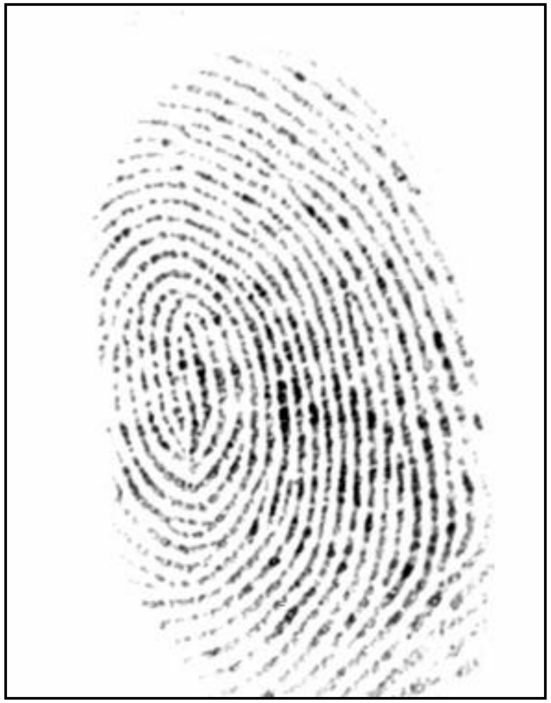


After

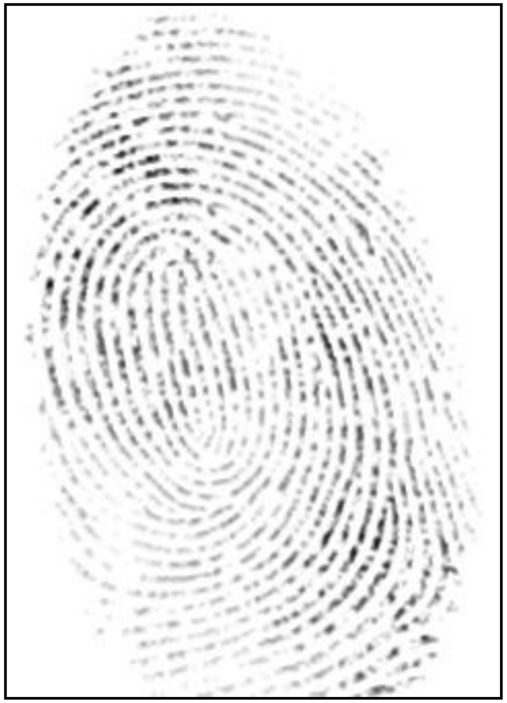
Examples (1)



Previous model

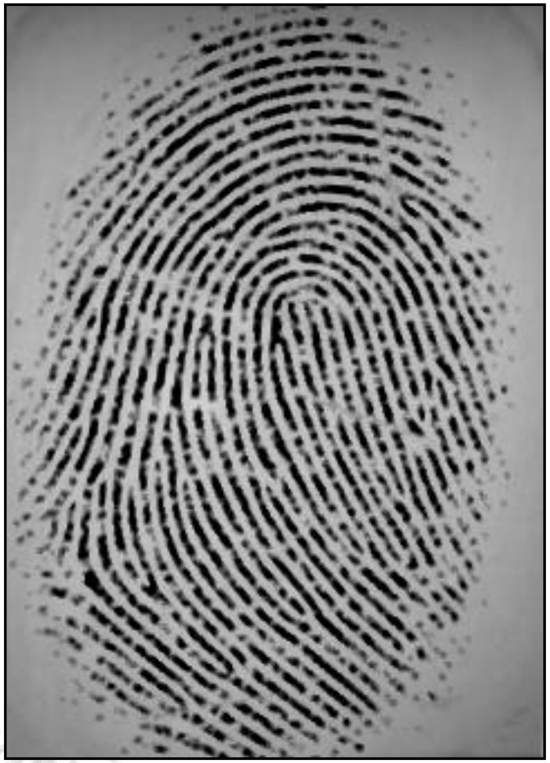


Real fingerprint

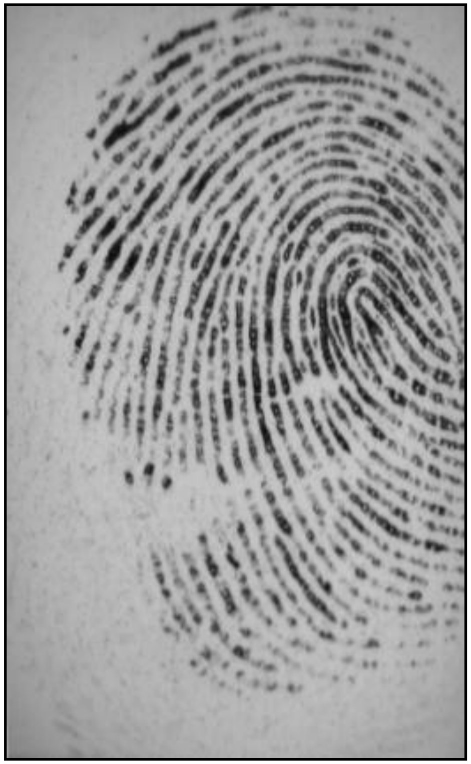


Improved model

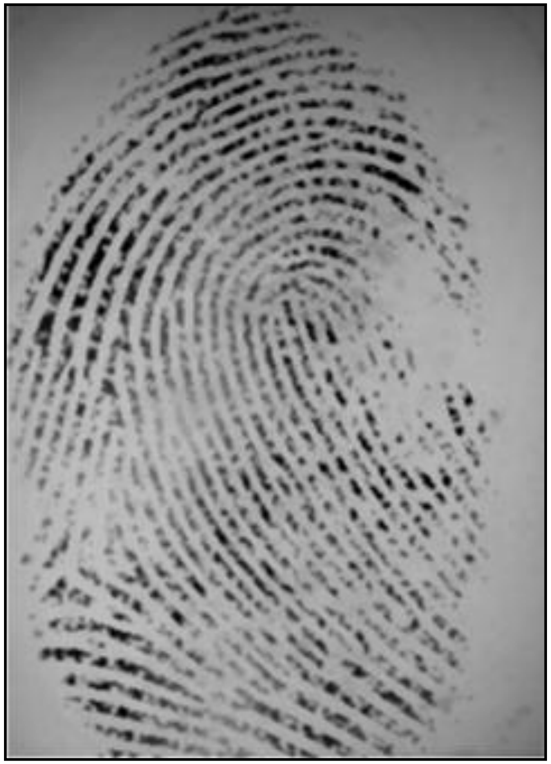
Examples



Previous model

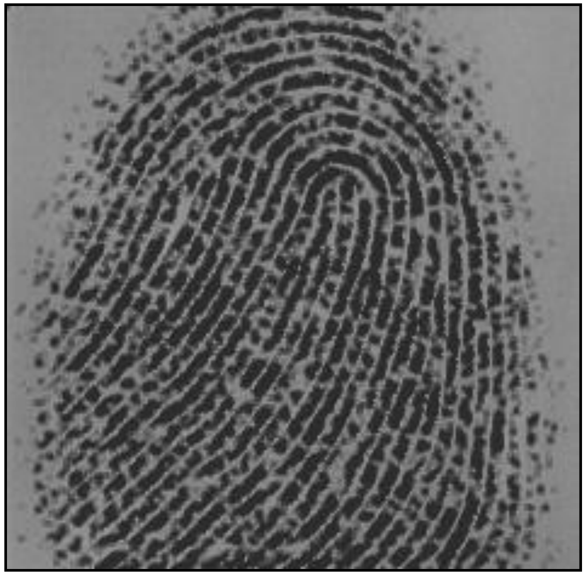


Real fingerprint

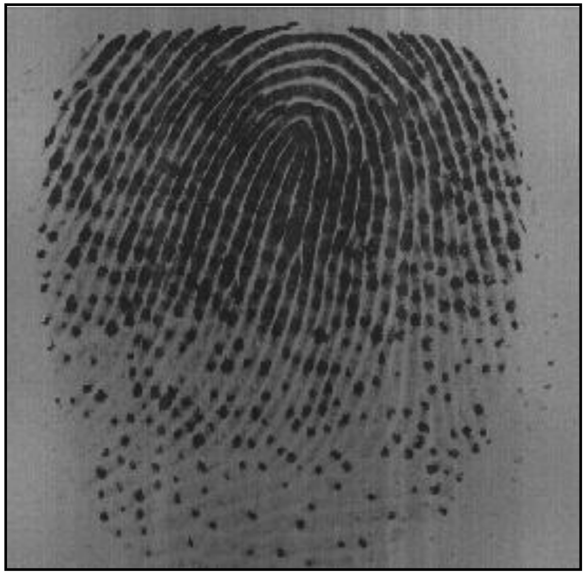


Improved model

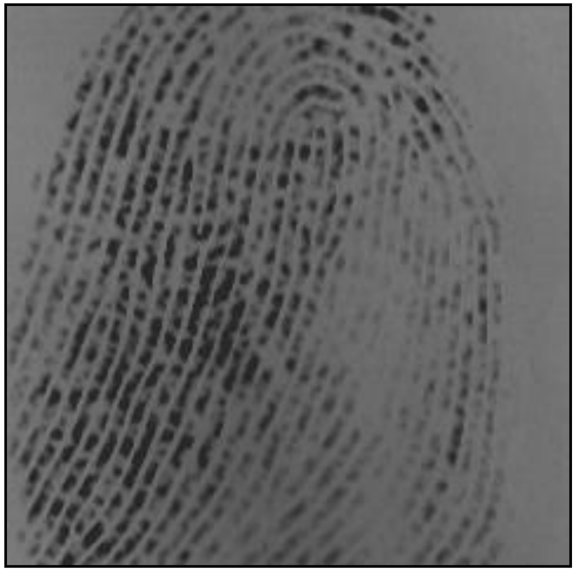
Examples



Previous model



Real fingerprint



Improved model

References

- BioLab web site
  - **http://biolab.csr.unibo.it**

- Book chapter

  - R. Cappelli,"Synthetic fingerprint generation", in D. Maltoni, D. Maio, A.K. Jain and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer (New York), 2003.

- Papers

  - R. Cappelli, A. Erol, D. Maio and D. Maltoni, "Synthetic Fingerprint-image Generation", in Proc. *International Conference on Pattern Recognition (ICPR2000)*, Barcelona, vol.3, pp.475-478, September 2000.
  - R. Cappelli, D. Maio and D. Maltoni, "Modelling Plastic Distortion in Fingerprint Images", in Proc. *Second International Conference on Advances in Pattern Recognition (ICAPR2001)*, Rio de Janeiro, pp.369-376, March 2001.
  - R. Cappelli, D. Maio and D. Maltoni, "Synthetic Fingerprint-Database Generation", in Proc. *International Conference on Pattern Recognition (ICPR2002)*, Quebec City, August 2002.
  - R. Cappelli, D. Maio and D. Maltoni, "An Improved Noise Model for the Generation of Synthetic Fingerprints", in Proc. *Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV2004)*, Kunming, China, December 2004.