# Classification

**Daniel Novák**

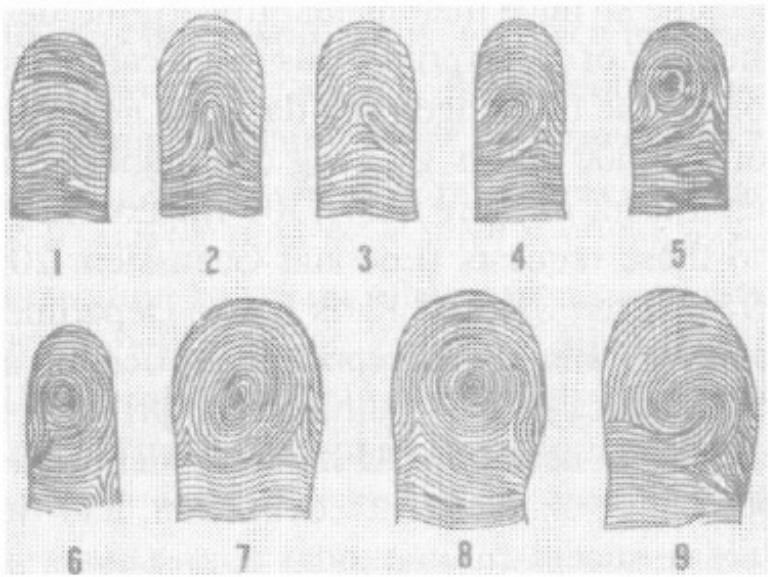
**17.10, 2024, Prague**

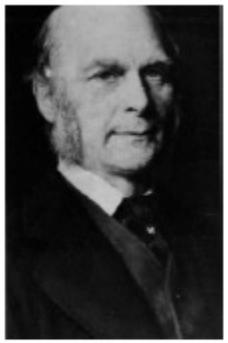
**Acknowledgments: Xavier Palathingal, Andrzej Drygajlo, Handbook of Fingerprint Recognition**

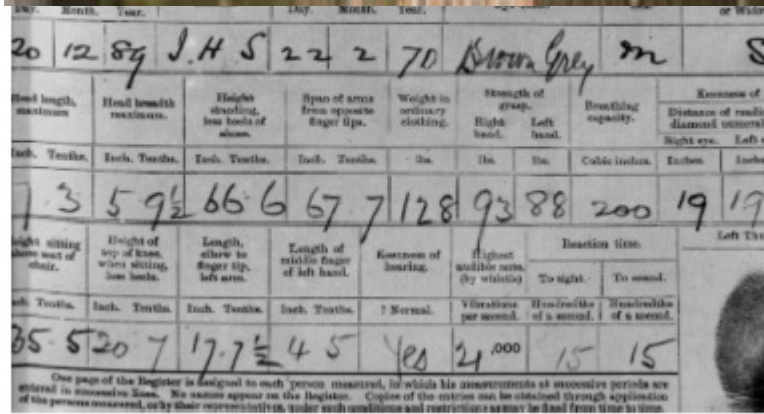
laboratory Gerstner

# History

- In 1823, Purkinji proposed the first fingerprint classification, which classified into nine categories: (transverse curve, central longitudinal stria, oblique stripe, oblique loop, almond whorl, spiral whorl, ellipse, circle, and double whorl)
- Sir Francis Galton introduced the minutae features for fingerprint matching in late 19th century





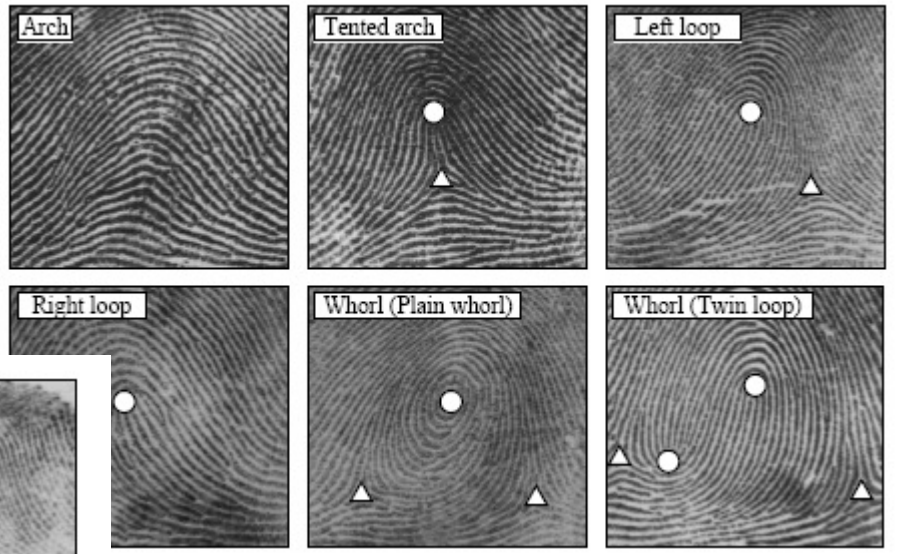
# Classes

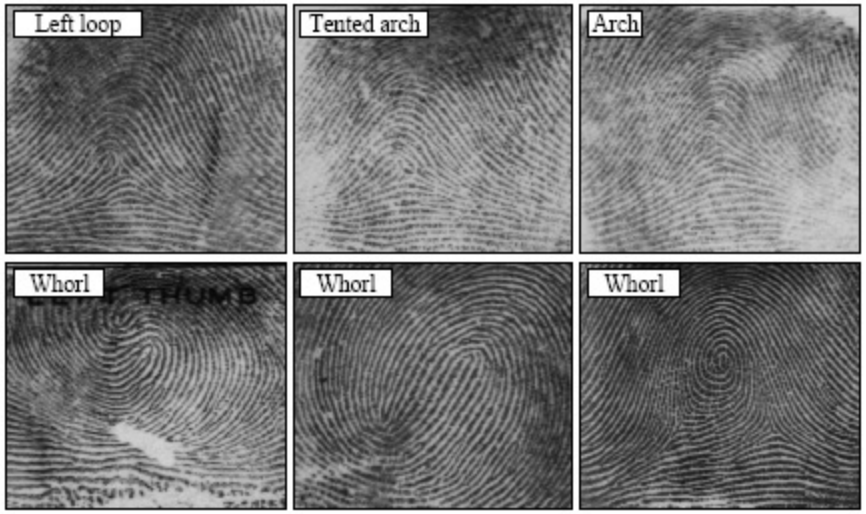
- **Arch: 3.7%, tented arch: 2.9%, left loop: 33.8%, right loop: 31.7%, whorl: 27.9%**
- Pattern recognition

PROBLEM

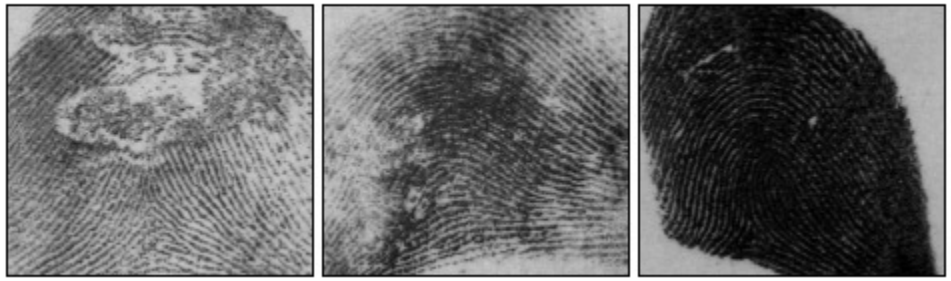
Small-inter class variability

Large intra-class variability




presence of noise

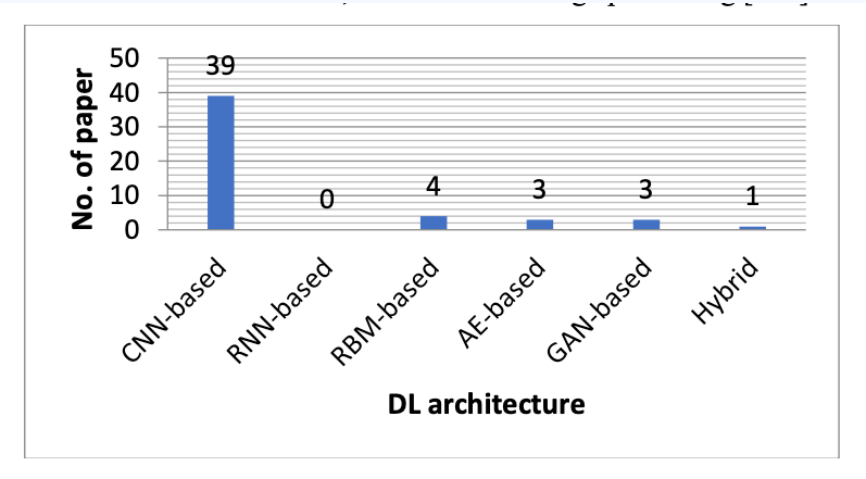


# Techniques

- Features:
  - O = orientation image
  - S = singularities
  - R = ridge flow,
  - G = Gabor
- classification technique
  - Rb = rule-based
  - Sy = syntactic
  - Str = structural,
  - Sta = statistical
  - Nn = neural network
  - Mc = multiple classifiers
  - **Deep learning**

| Fingerprint classification approach | *Features* | | | | *Classifier* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | O | S | R | G | Rb | Sy | Str | Sta | Nn | Mc |
| Moayer and Fu (1975) | √ | | | | | √ | | | | |
| Moayer and Fu (1976) | √ | | | | | √ | | | | |
| Rao and Balck (1980) | √ | | | | | √ | | | | |
| Kawagoe and Tojo (1984) | | √ | √ | | √ | | | | | |
| Hughes and Green (1991) | √ | | | | | | | | √ | |
| Bowen (1992) | √ | √ | | | | | | | √ | |
| Kamijo, Mieno, and Kojima (1992) | √ | | | | | | | | √ | |
| Kamijo (1993) | √ | | | | | | | | √ | |
| Moscinska and Tyma (1993) | √ | | | | √ | | | | √ | |
| Wilson, Candela, and Watson (1994) | √ | | | | | | | | √ | |
| Candela et al. (1995) | √ | | √ | | √ | | | | √ | √ |
| Omidvar, Blue, and Wilson (1995) | √ | | | | | | | | √ | |
| Halici and Ongun (1996) | √ | | | | | | | | √ | |
| Karu and Jain (1996) | | √ | | | √ | | | | | |
| Maio and Maltoni (1996) | √ | | | | | | √ | | | |
| Ballan, Sakarya, and Evans (1997) | | √ | | | √ | | | | | |
| Chong et al. (1997) | | | √ | | √ | | | | | |
| Senior (1997) | | | √ | | | | √ | | | |
| Wei, Yuan, and Jie (1998) | √ | | | | √ | | | | √ | √ |
| Cappelli et al. (1999) | √ | | | | | | √ | | | |
| Cappelli, Maio, and Maltoni (1999) | √ | | | | | | | √ | | |
| Hong and Jain (1999) | | √ | √ | | √ | | | | | √ |
| Jain, Prabhakar, and Hong (1999) | | | | √ | | | | √ | √ | √ |
| Lumini, Maio, and Maltoni (1999) | √ | | | | | | √ | | | |
| Cappelli, Maio, and Maltoni (2000a) | √ | | | | | | | √ | | √ |
| Cho et al. (2000) | | √ | | | √ | | | | | |
| Bartesaghi, Fernández, and Gómez (2001) | | √ | | | √ | | | | | |
| Bernard et al. (2001) | √ | | | | | | | | √ | |
| Marcialis, Roli, and Frasconi (2001) | √ | | | √ | | | √ | √ | √ | √ |
| Pattichis et al. (2001) | √ | | | | √ | | | | √ | √ |
| Senior (2001) | √ | | √ | | √ | | √ | | √ | √ |
| Yao, Frasconi, and Pontil (2001) | | | | √ | | | | √ | | √ |
| Cappelli, Maio, and Maltoni (2002a) | √ | | | | | | | √ | | √ |
| Jain and Minut (2002) | | | √ | | √ | | | | | |
| Cappelli et al. (2003) | √ | | | | | | | √ | | √ |
| Yao et al. (2003) | √ | | | √ | | | √ | √ | √ | √ |

# Deep learning




- 1) Traditional Method as Extractor and Deep network as Classifier
- 2) Deep network as Feature Extractor and Traditional Method as Classifier
- 3) Deep network in end-to-end learning

- Automated Feature Extraction
- Robustness to Variations:
- Handling Low-quality Images:
- Scalability and Speed
- End-to-end Learning
- Better Generalization

- Need for Large Datasets
- Computational Resources
- Overfitting

# Prize of GOLD

- Good large database> very expensive (FP,ECG,EEG, etc.)
  - DB4,DB14: STANDARDS for classification systems
  - ALREADY WITHDRAWN !!!
  - 8bit- grey level images of rolled FP scanned from cards,
  - Manual annotation by a human expert (A,L,R,T,W)
  - 2000 FPs: DB4
  - **27000 FPs: DB14**
  - All classes are distributed equally. However, Arch: 3.7%, tented arch: 2.9%, left loop: 33.8%, right loop: 31.7%, whorl: 27.9%

  - NIST Special Database 302
  - 10.000 FP from 200 people

  - OR : MSU PrintsGAN Dataset, 500k synthetised images

laboratory Gerstner

# Classification Evaluation

- Accuracy
  - Rejection can improve accuracy

$$\text{error rate} = \frac{\text{number of misclassified fingerprints} \times 100}{\text{total number of fingerprints}}\%$$

$$\text{accuracy} = 100\% - \text{error rate}.$$

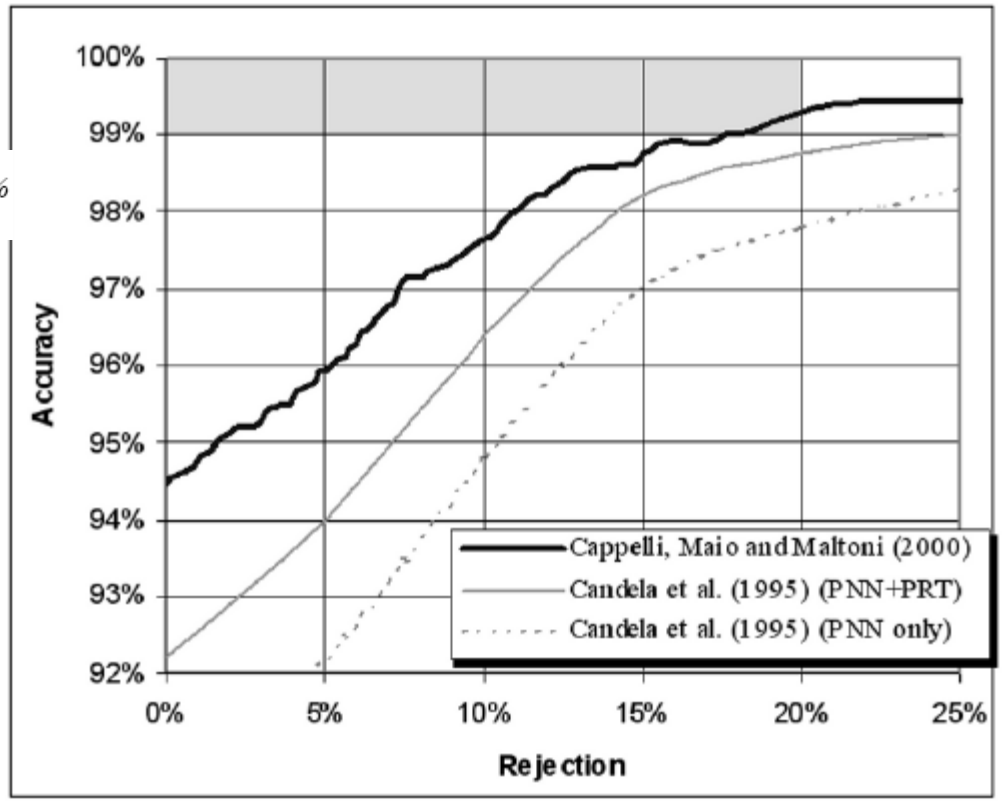
- Penetration rate: time constraint

$$\text{penetration rate} = \frac{\text{number of accessed fingerprints} \times 100}{\text{total number of fingerprints in the database}}\%$$

- Confusion matrix (DB4)

| True class | Hypothesized class | | | | |
|---|---|---|---|---|---|
| | A | L | R | W | T |
| A | **420** | 6 | 3 | 1 | 11 |
| L | 3 | **376** | 3 | 9 | 11 |
| R | 5 | 1 | **392** | 6 | 16 |
| W | 2 | 5 | 14 | **377** | 1 |
| T | 33 | 18 | 9 | 0 | **278** |

- Rejection can improve accuracy (DB14)
  - Unknown class
  - FBI target:shaded area
  - **NIST database -DB 4, DB14**



# Synthetic fingerprint generation

**Daniel Novák**

**17.10, 2024, Prague**

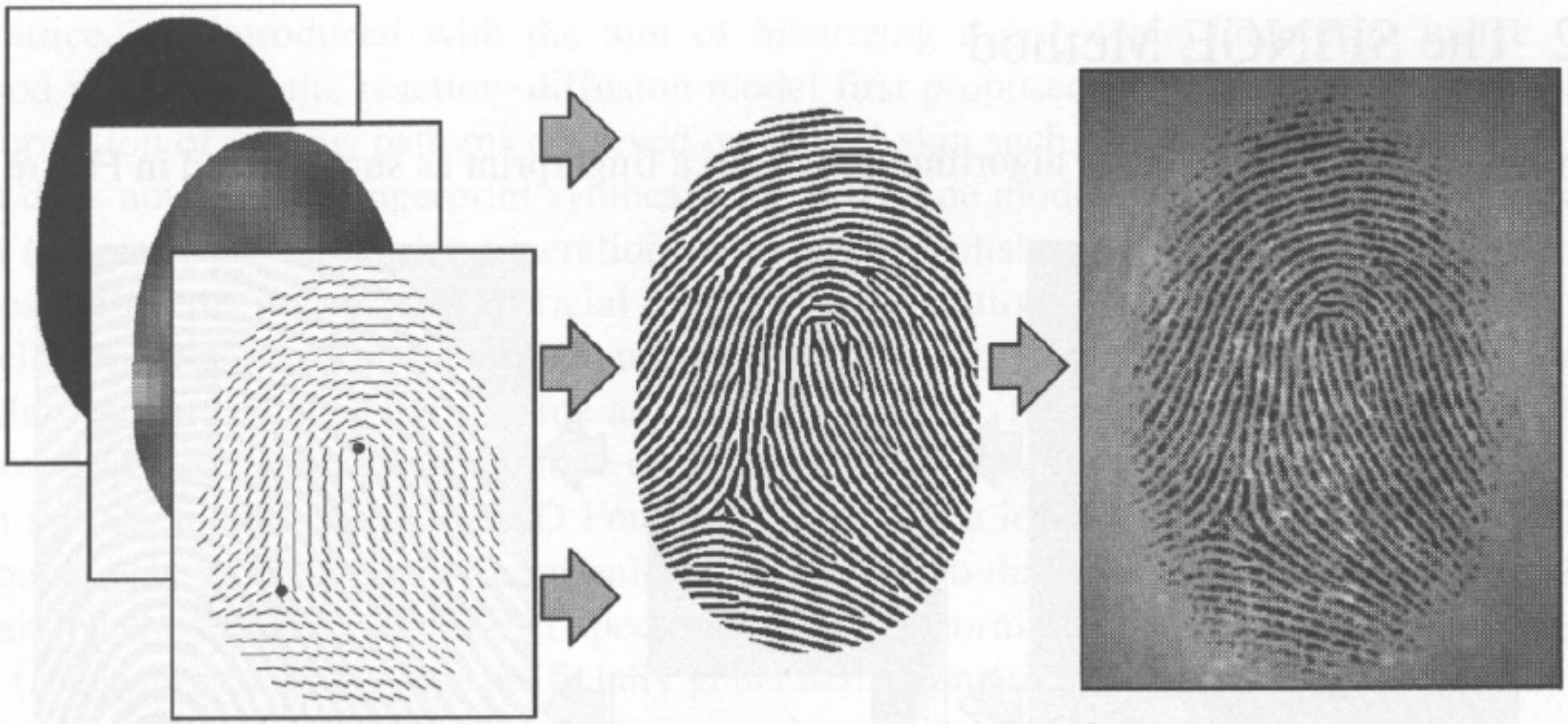
**Acknowledgments: Xavier Palathingal, Andrzej Drygajlo, Handbook of Fingerprint Recognition**

laboratory Gerstner

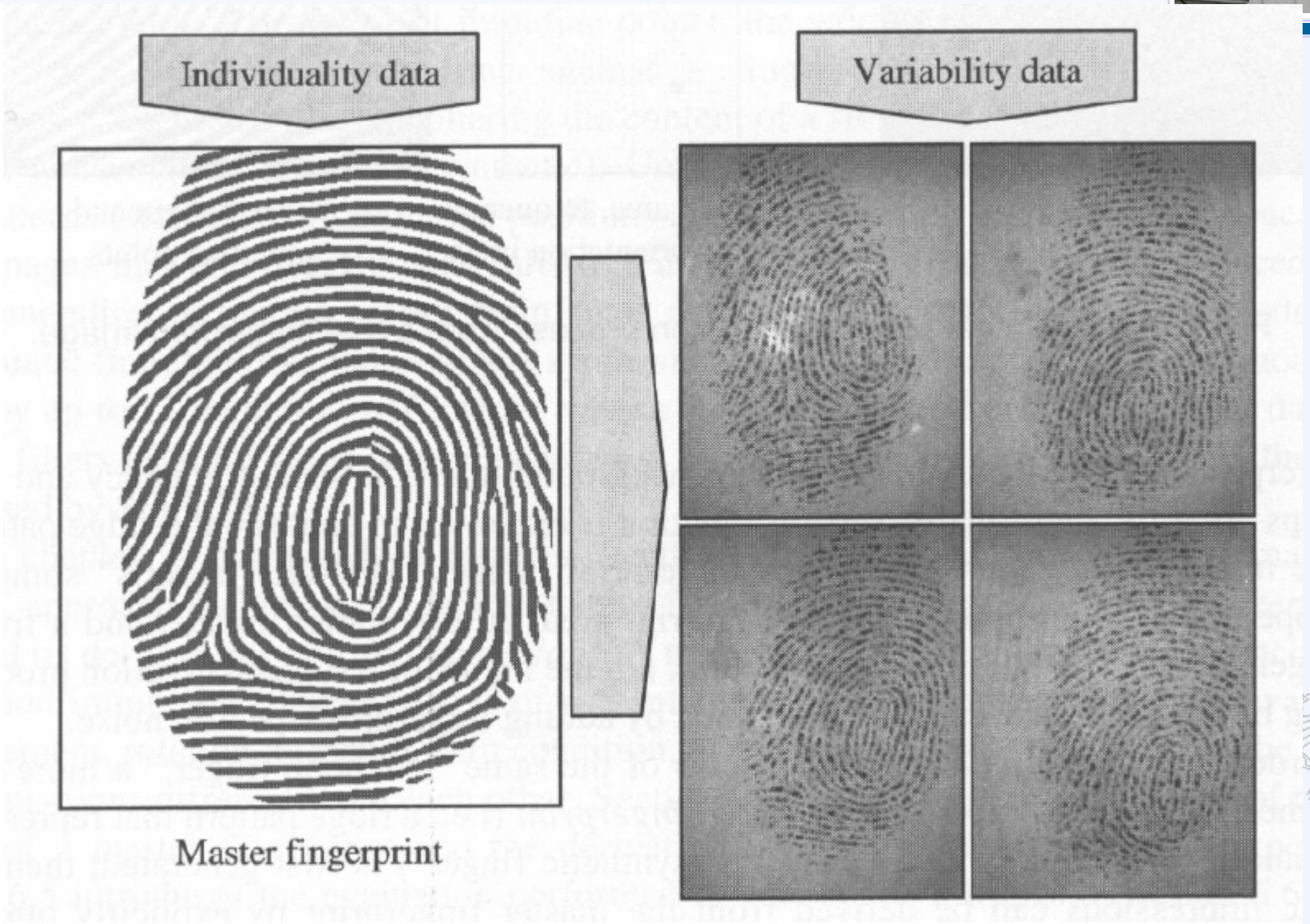
# Synthetic fingerprint generation

- Motivation
  - Accuracy of each algorithm is usually evaluated on relatively small proprietary databases
  - Evaluation on small databases makes the accuracy estimates highly data dependent
  - When the databases are proprietary, the accuracy of various fingerprint matching algorithms cannot be compared directly
- Synthetic fingerprint generation can be used to automatically create large databases of fingerprints, thus allowing fingerprint recognition algorithms to be effectively trained, tested, optimized, and compared

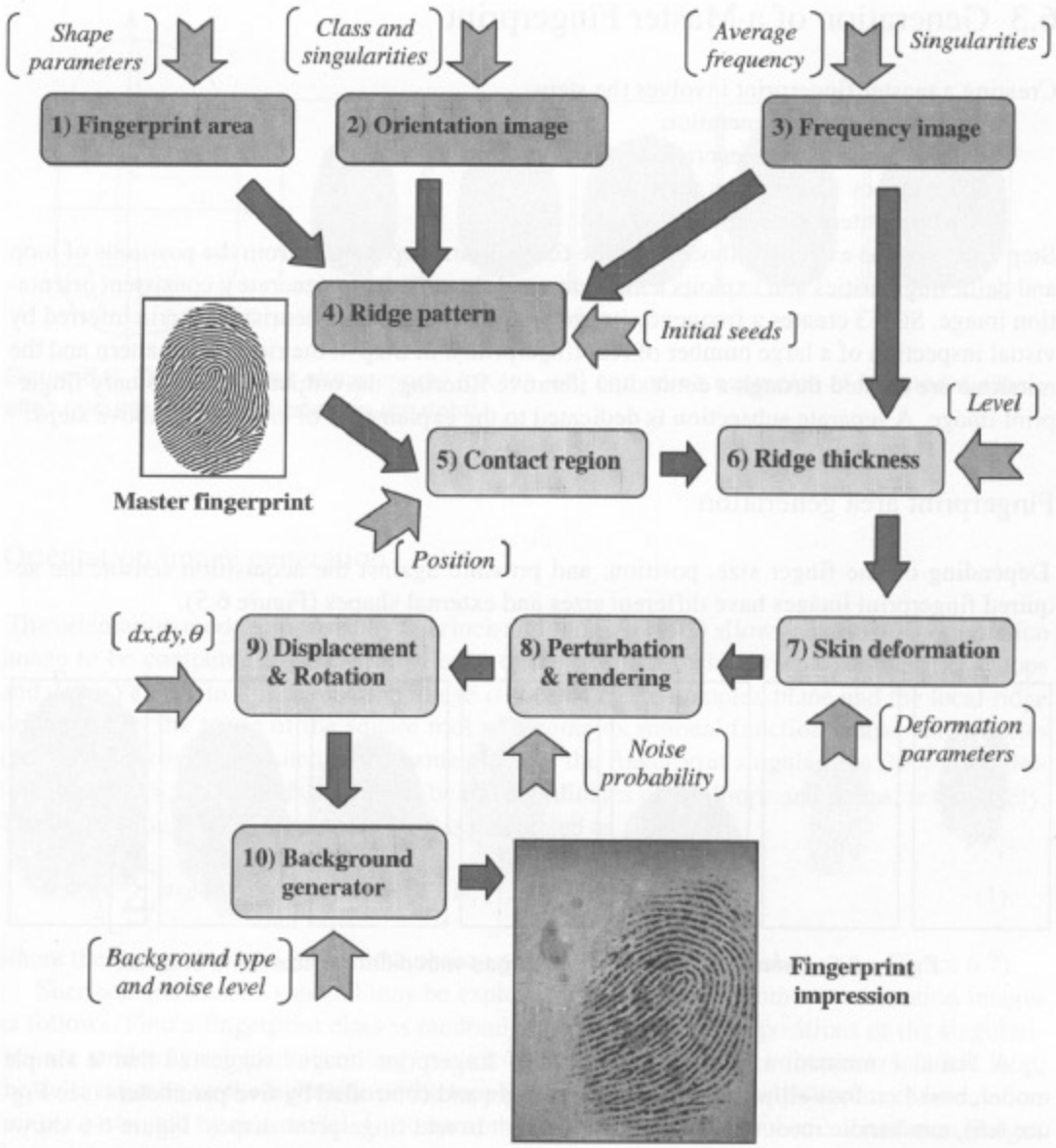
# Basic idea


Fingerprint area, frequency image, and orientation image
Ridge pattern and minutiae points
Fingerprint image

# From master to final impression



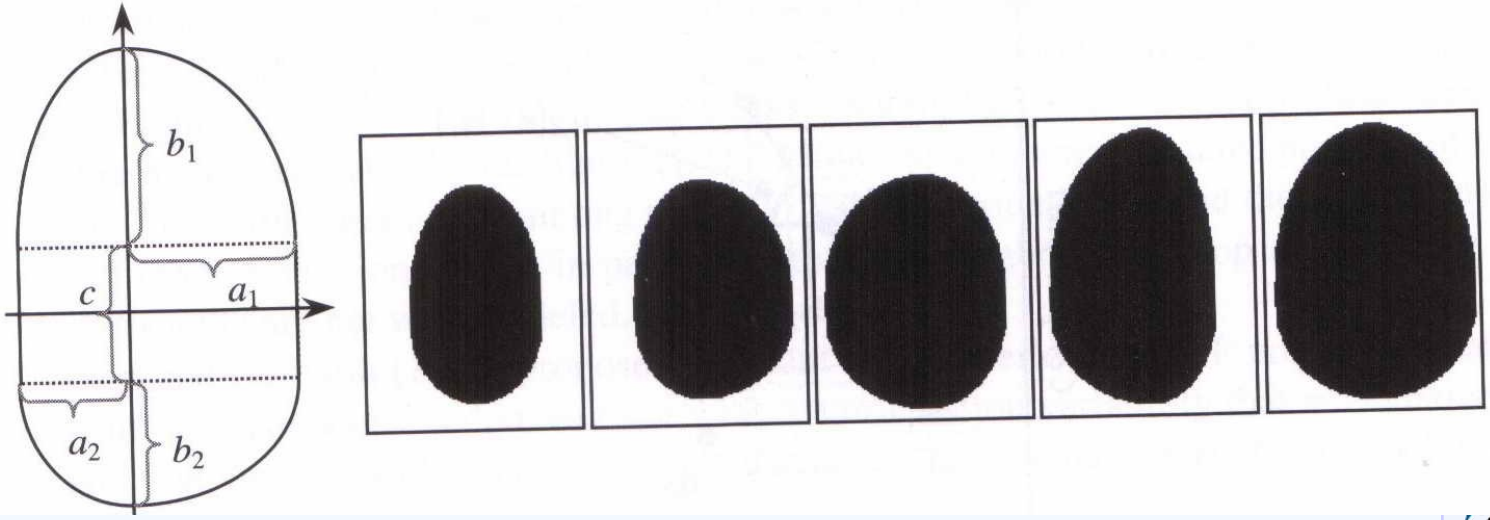
# SFINGE



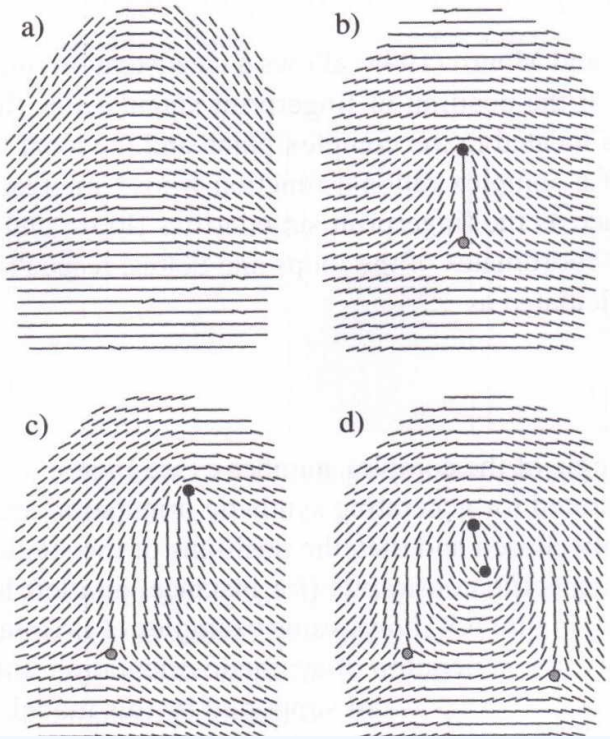

laboratory Gerstner

chnical University

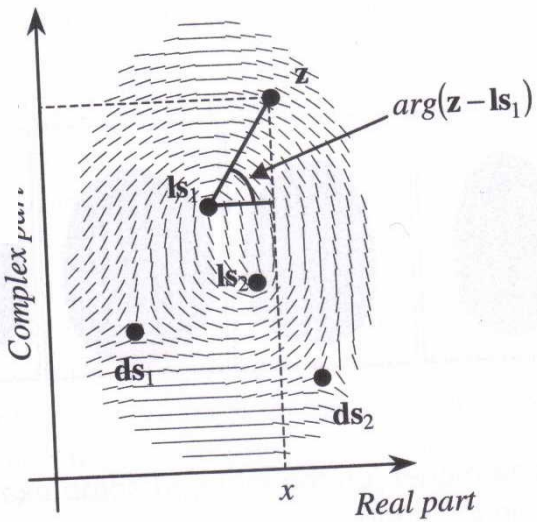
# FP area generation



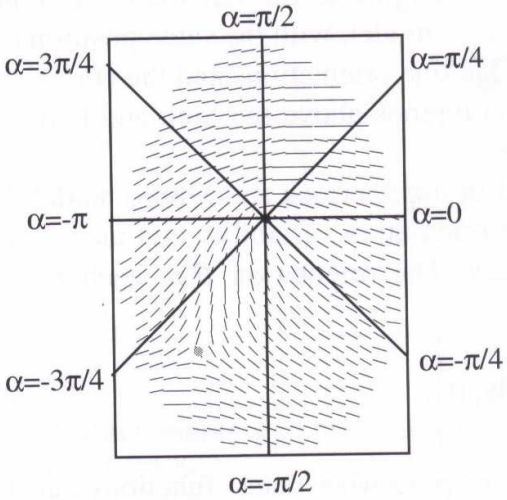
# Orientation

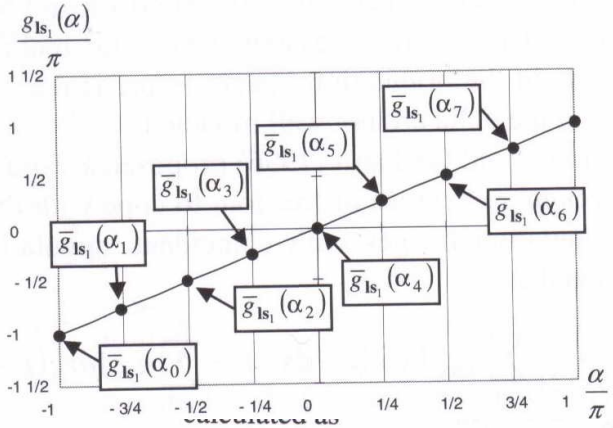






# Orientation



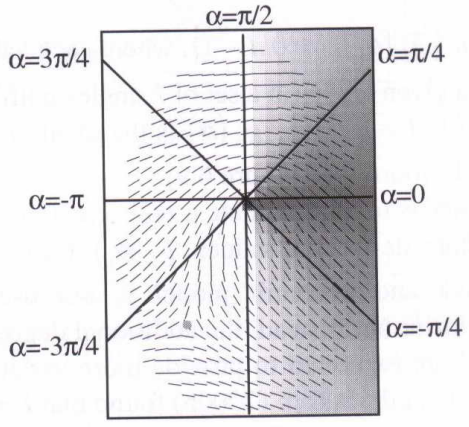



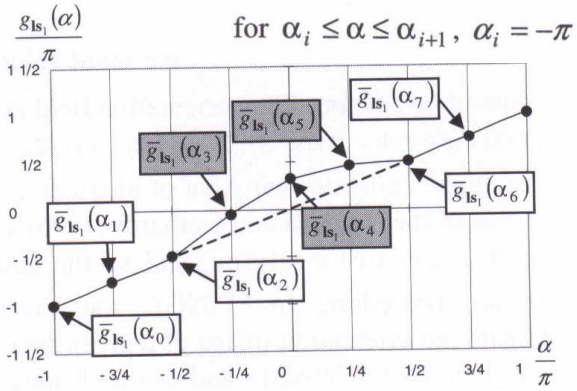

$$\theta = \frac{1}{2}\left[\sum_{i=1}^{n_d} g_{\mathbf{ds}_i}(arg(\mathbf{z}-\mathbf{ds}_i)) - \sum_{i=1}^{n_c} g_{\mathbf{ls}_i}(arg(\mathbf{z}-\mathbf{ls}_i))\right], \quad (2)$$

where $g_k(\alpha)$, for $k \in \{\mathbf{ls}_1,...,\mathbf{ls}_{n_c},\mathbf{ds}_1,...,\mathbf{ds}_{n_d}\}$, are piecewise linear functions capable of locally correcting the orientation field with respect to the value given by the Sherlock and Monroe model:

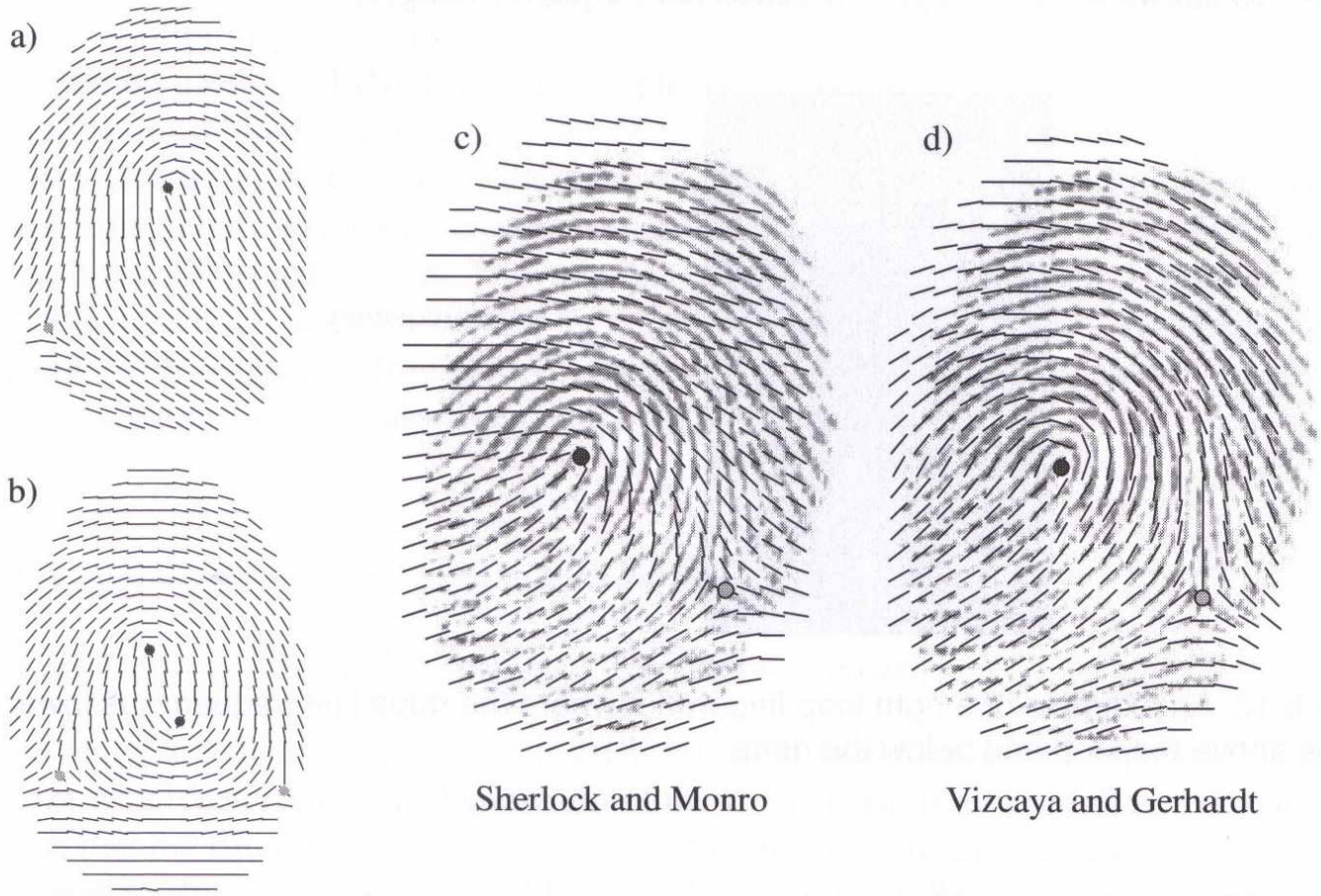
$$g_k(\alpha) = \bar{g}_k(\alpha_i) + \frac{\alpha-\alpha_i}{2\pi/L}(\bar{g}_k(\alpha_{i+1}) - \bar{g}_k(\alpha_i)), \quad (3)$$

for $\alpha_i \le \alpha \le \alpha_{i+1}$, $\alpha_i = -\pi + \frac{2\pi i}{L}$.

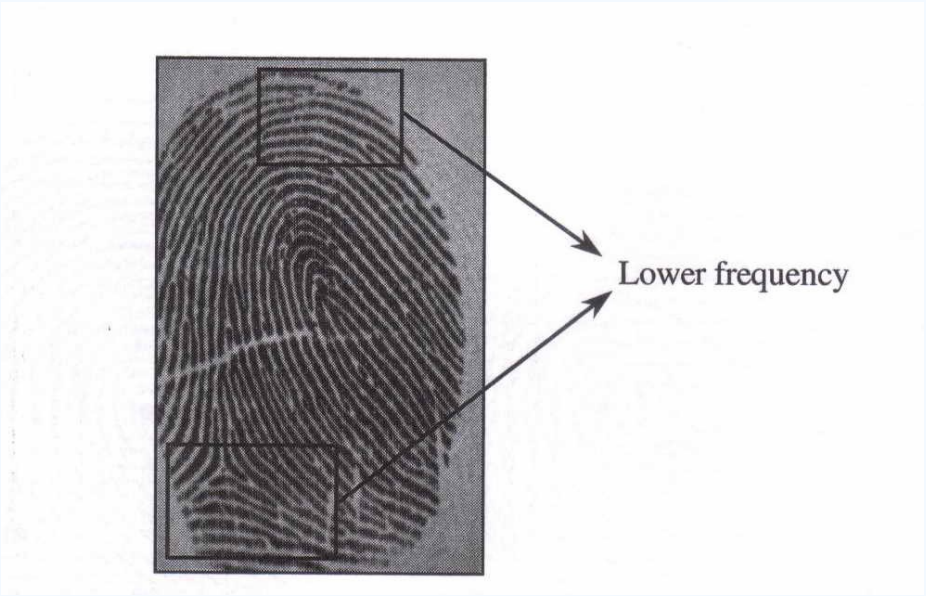


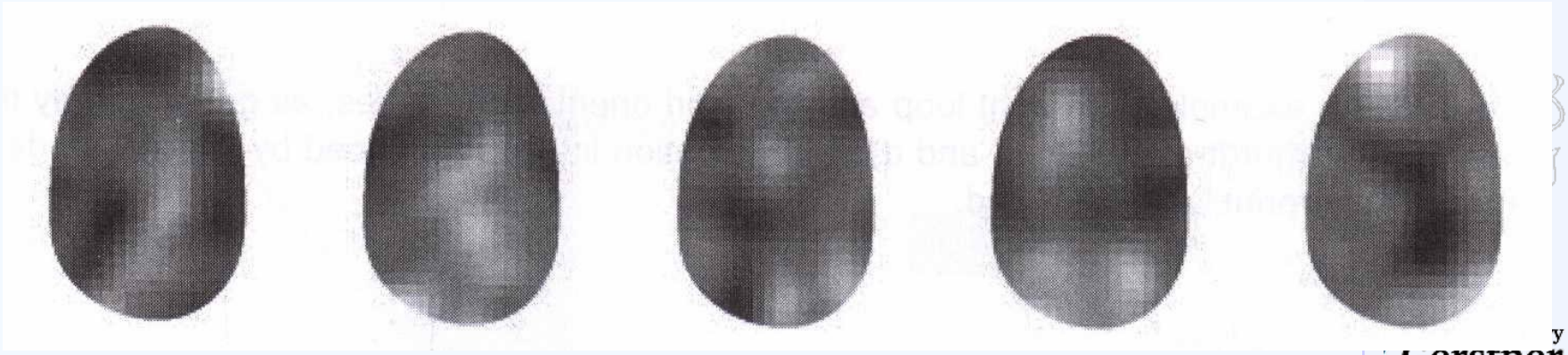



# Orientation



# Frequency

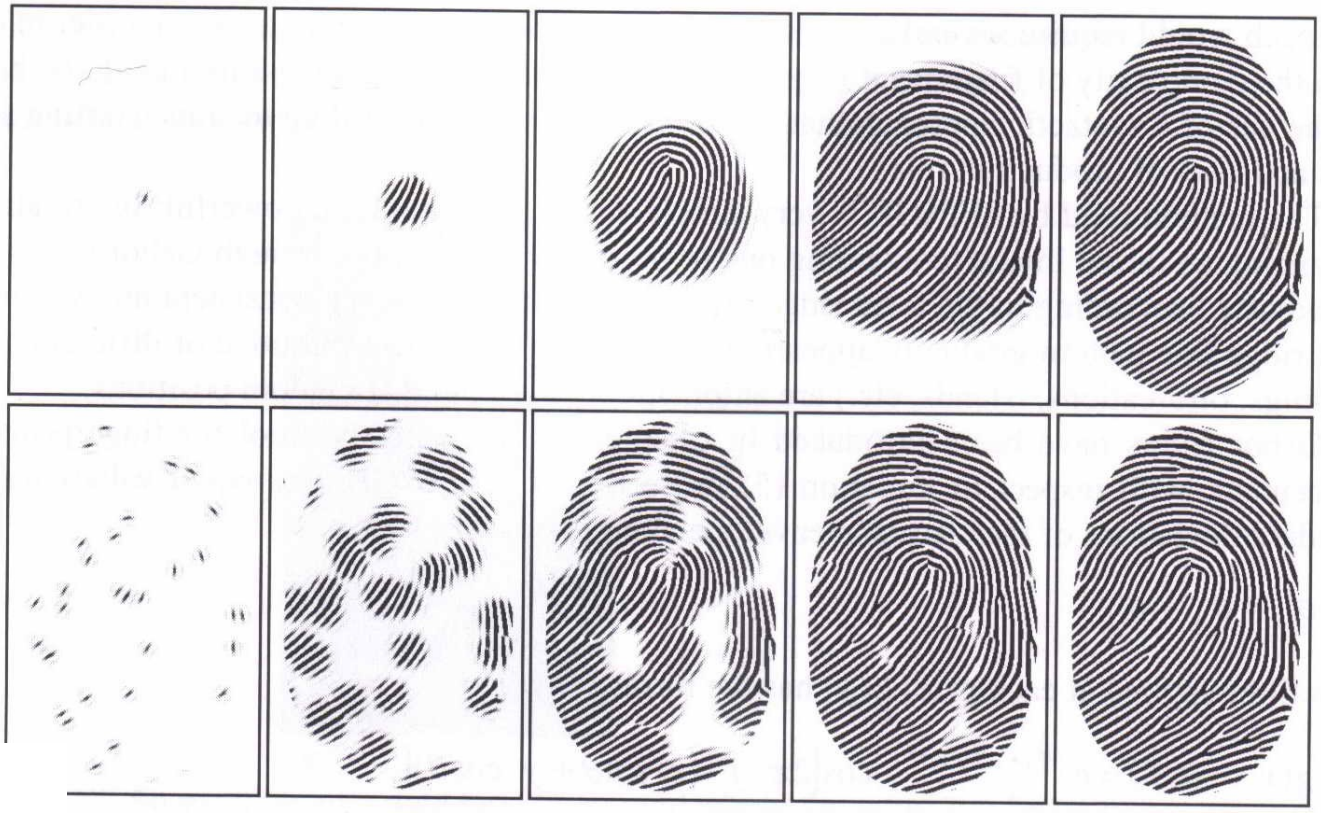


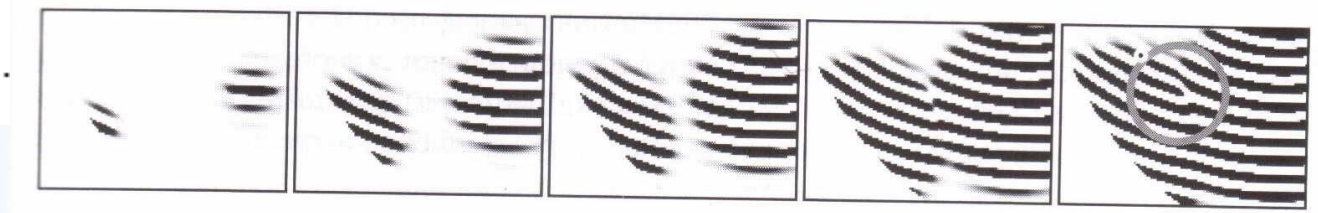
# Ridge line

- Gabor filter
- Seeds

$$\sigma_x = \sigma_y = \sigma$$

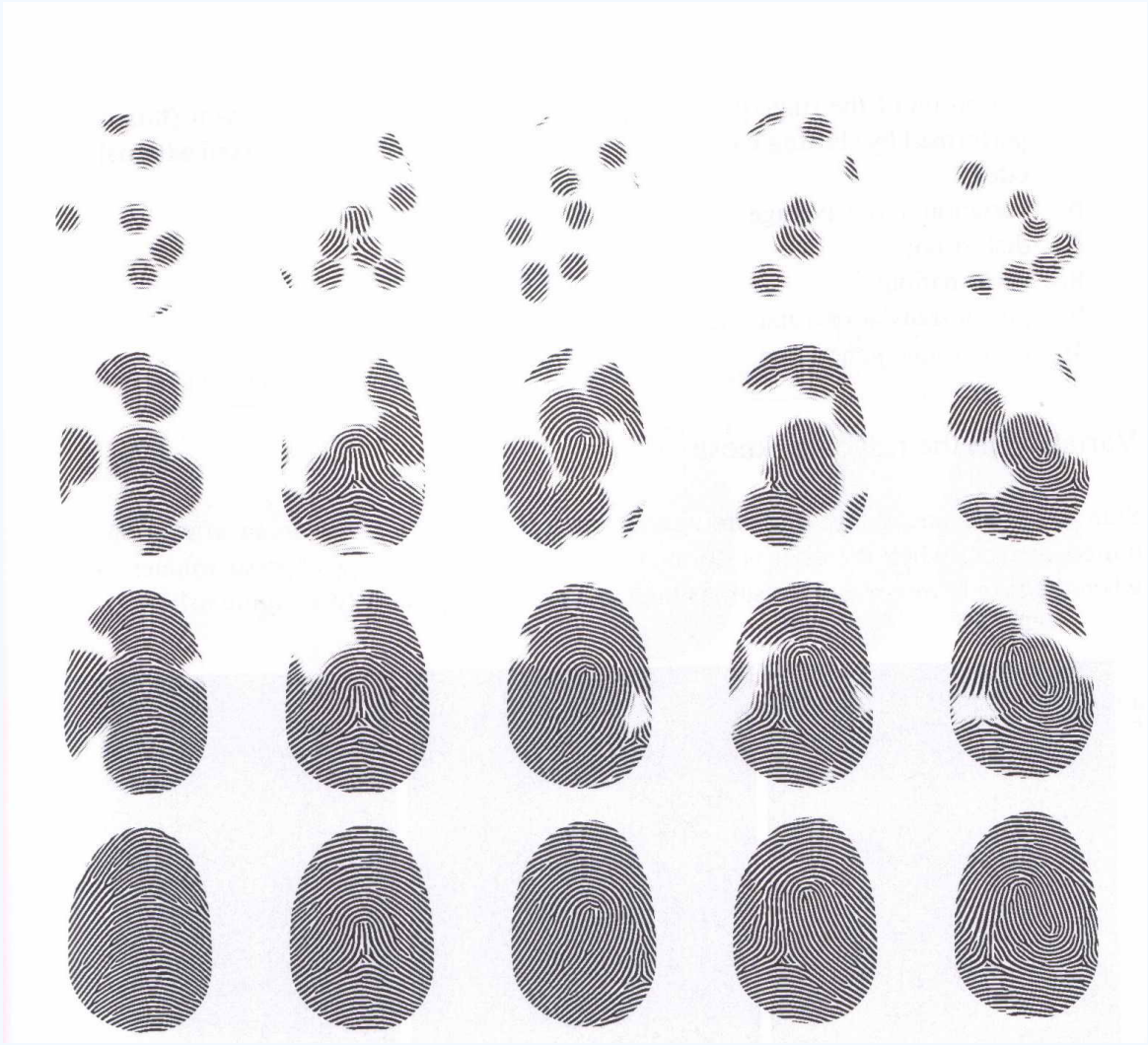


$$e^{-\left(\left(\frac{3}{2f}\right)^2 / 2\sigma^2\right)} = 10^{-3}.$$

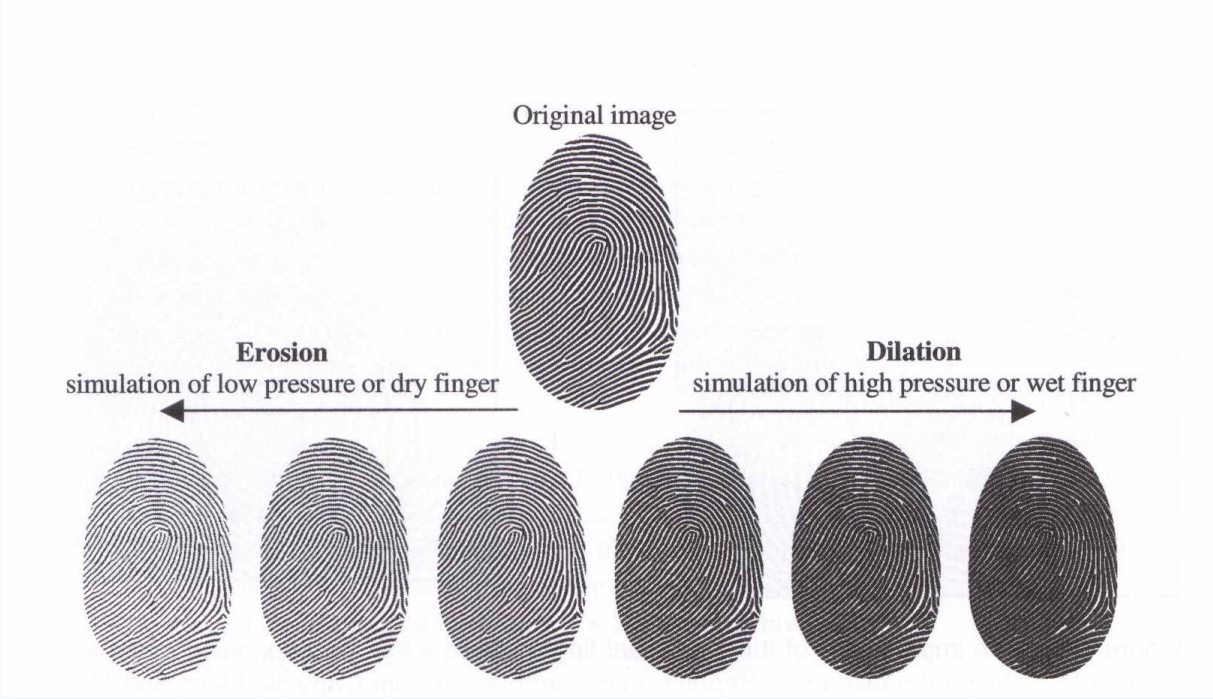


$$g(x, y : \theta, f) = e^{-\left((x^2+y^2)/2\sigma^2\right)} \cdot \cos\left[2\pi \cdot f \cdot (x \cdot \sin\theta + y \cdot \cos\theta)\right],$$

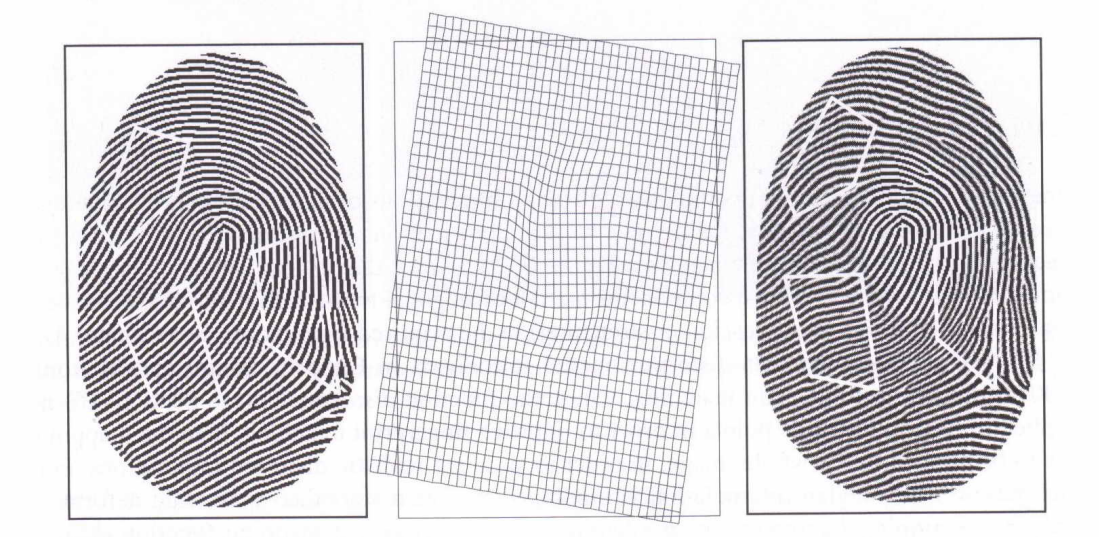
# Ridge line



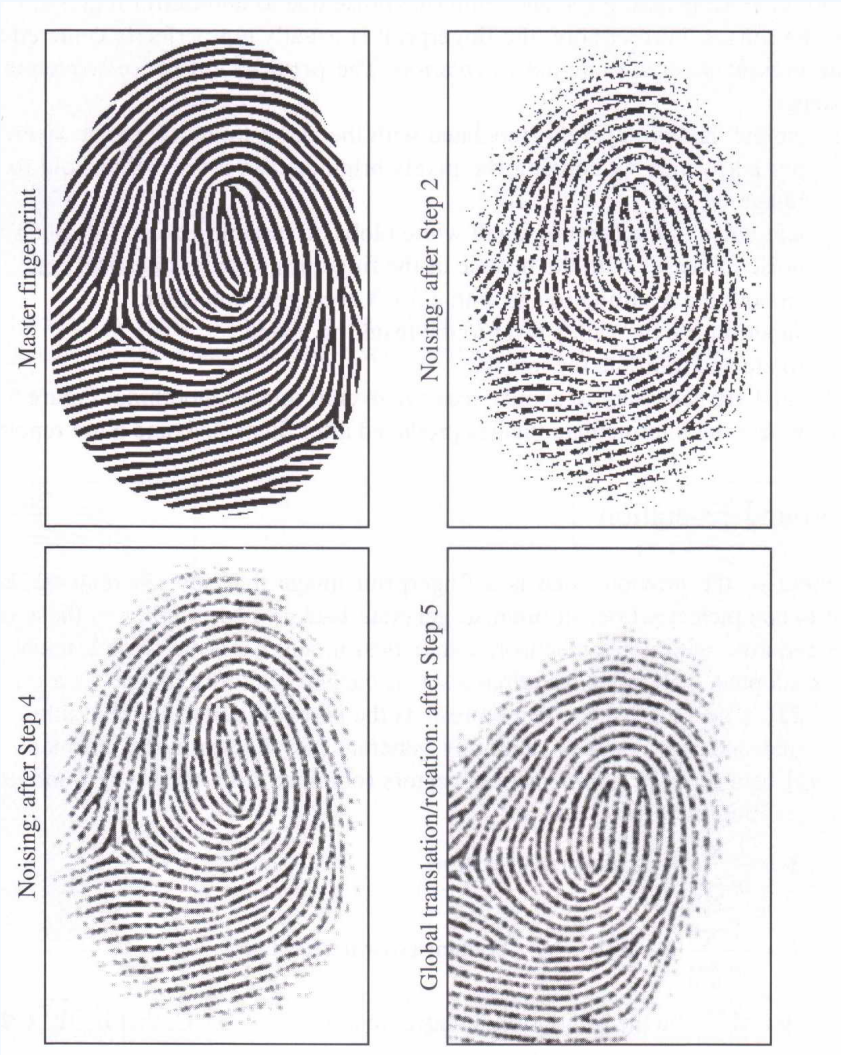
# Ridge thickness



# Distortion



# Perturbation & Translation



# Background

- $\bar{\mathbf{b}} = \frac{1}{m}\sum_{\mathbf{b}\in B}\mathbf{b}$ be their mean vector;
- $\mathbf{C} = \frac{1}{m}\sum_{\mathbf{b}\in B}(\mathbf{b}-\bar{\mathbf{b}})(\mathbf{b}-\bar{\mathbf{b}})^T$ be their covariance matrix;
- $\mathbf{\Phi} \in \Re^{n\times n}$ be the orthonormal matrix that diagonalizes $\mathbf{C}$; that is, $\mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} = \mathbf{\Lambda}$,

$\mathbf{\Lambda} = Diag(\lambda_1, \lambda_2, ..., \lambda_n)$, $\mathbf{\Phi} = [\varphi_1, \varphi_2, ..., \varphi_n]$,

where $\lambda_i$ and $\varphi_i$, $i = 1..n$ are the eigenvalues and the eigenvectors of $\mathbf{C}$, respectively.

1. a $k$-dimensional vector $\mathbf{y} = [y_1, y_2, ..., y_k]$ is randomly generated according to $k$ normal distributions: $y_j = N(0, \lambda_{i_j}^{1/2})$, $j = 1..k$;
2. the corresponding $n$-dimensional vector $\mathbf{b}$ is obtained as: $\mathbf{b} = \mathbf{\Phi}_k\mathbf{y} + \bar{\mathbf{b}}$.

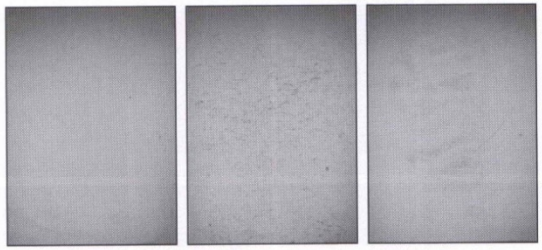


Figure 6.22. Examples of background-only images (acquired from an optical scanner) used for training the background generator.

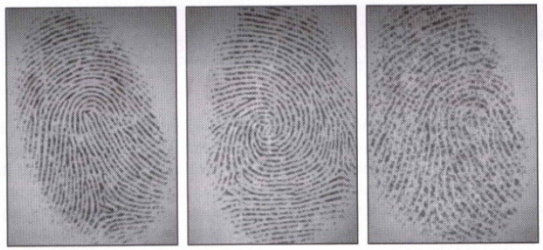
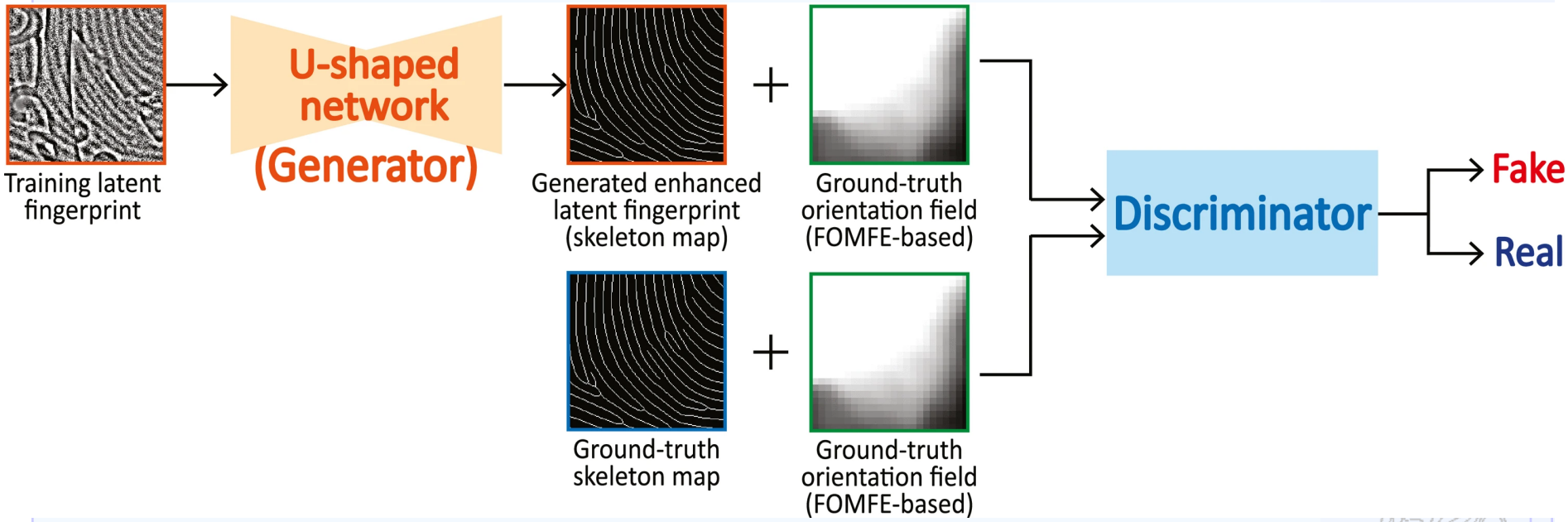
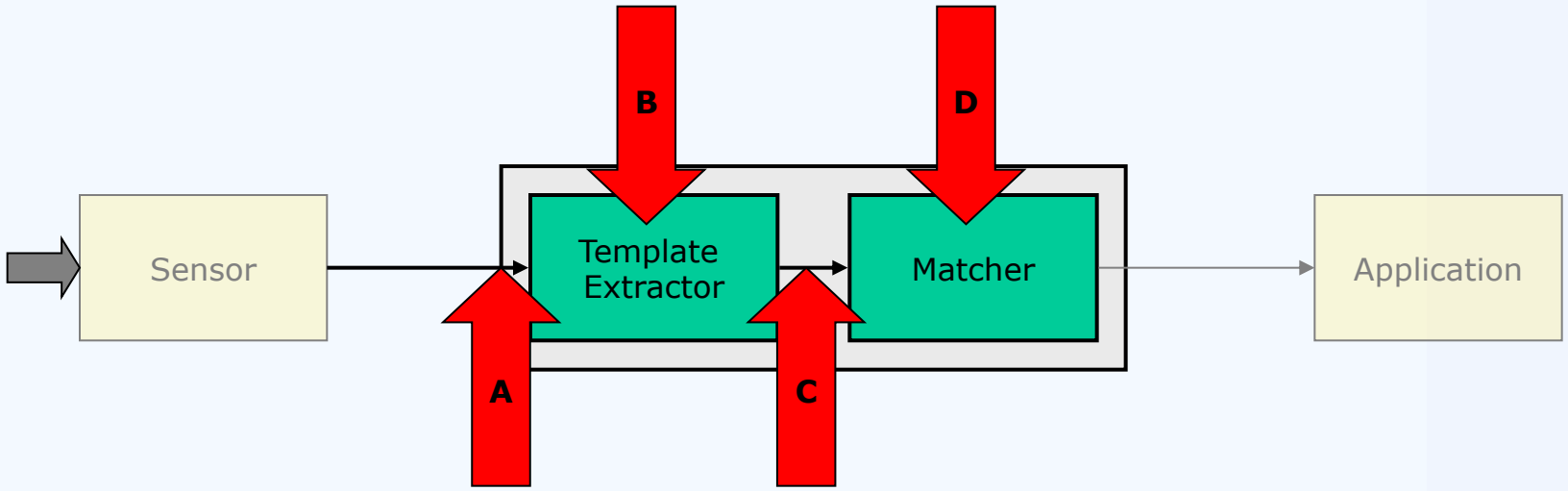


Figure 6.23. Three synthetic images with backgrounds generated according to the model (the parameters used for training are $m = 65$ and $k = 8$).

# Generative Adversarial Networks

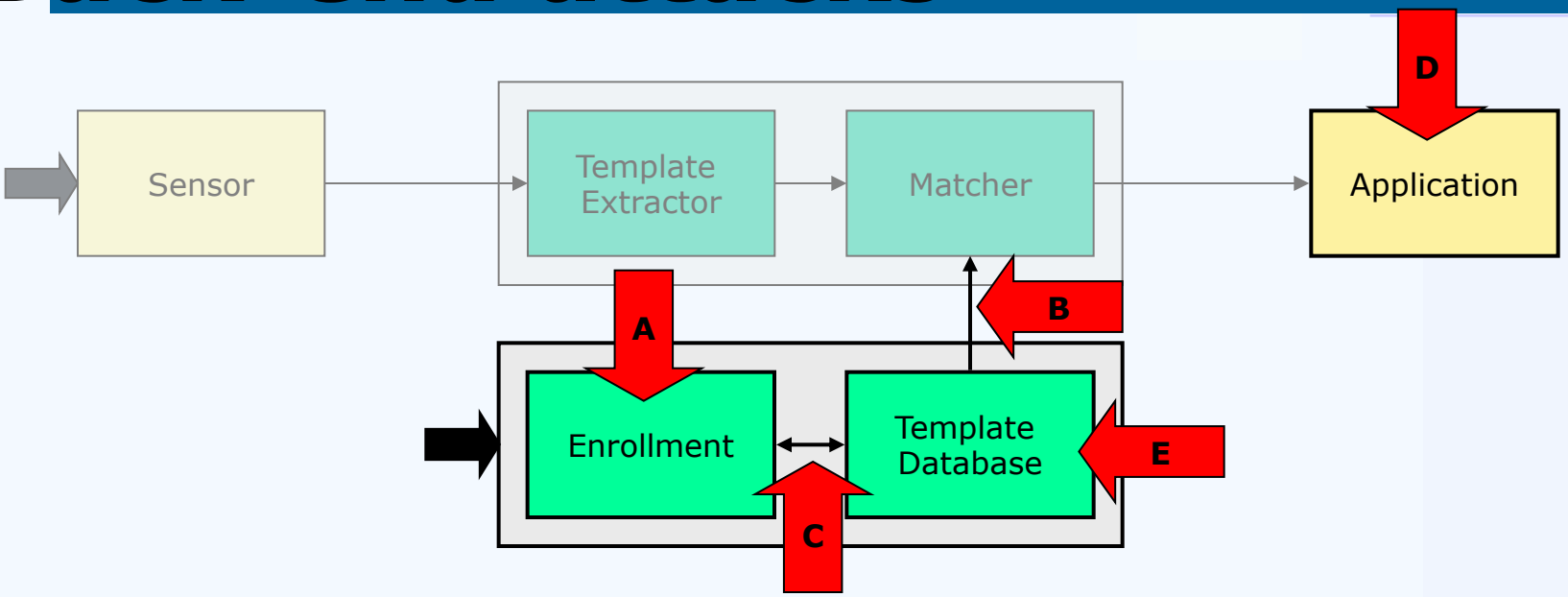


# Front-end attacks




| | |
|---|---|
| (A) Replay attack | A recording of true data is transmitted to Extractor; |
| (A) Electronic Impersonation | Injection of an image created artificially from extracted features; |
| (B) Trojan Horse | Extracted features are replaced; |
| (C) Communication | Attacks during transmission to remote matcher; |
| (D) Trojan Horse | Match decision is manipulated. |

# Back-end attacks




| | |
|---|---|
| (A) All seen so far | Enrollment has all the stages above; |
| (B) Communication Attack | Attacks during transmission between matcher and central or distributed database; |
| (C) Communication Attack | Attacks during transmission from enrollment stage to central or distributed database; |
| (D) Viruses, Trojans,... | |
| (E) Hacker's Attack | Modification or deletion of registers and gathering of information; |

# Threats


The Modern Burglar

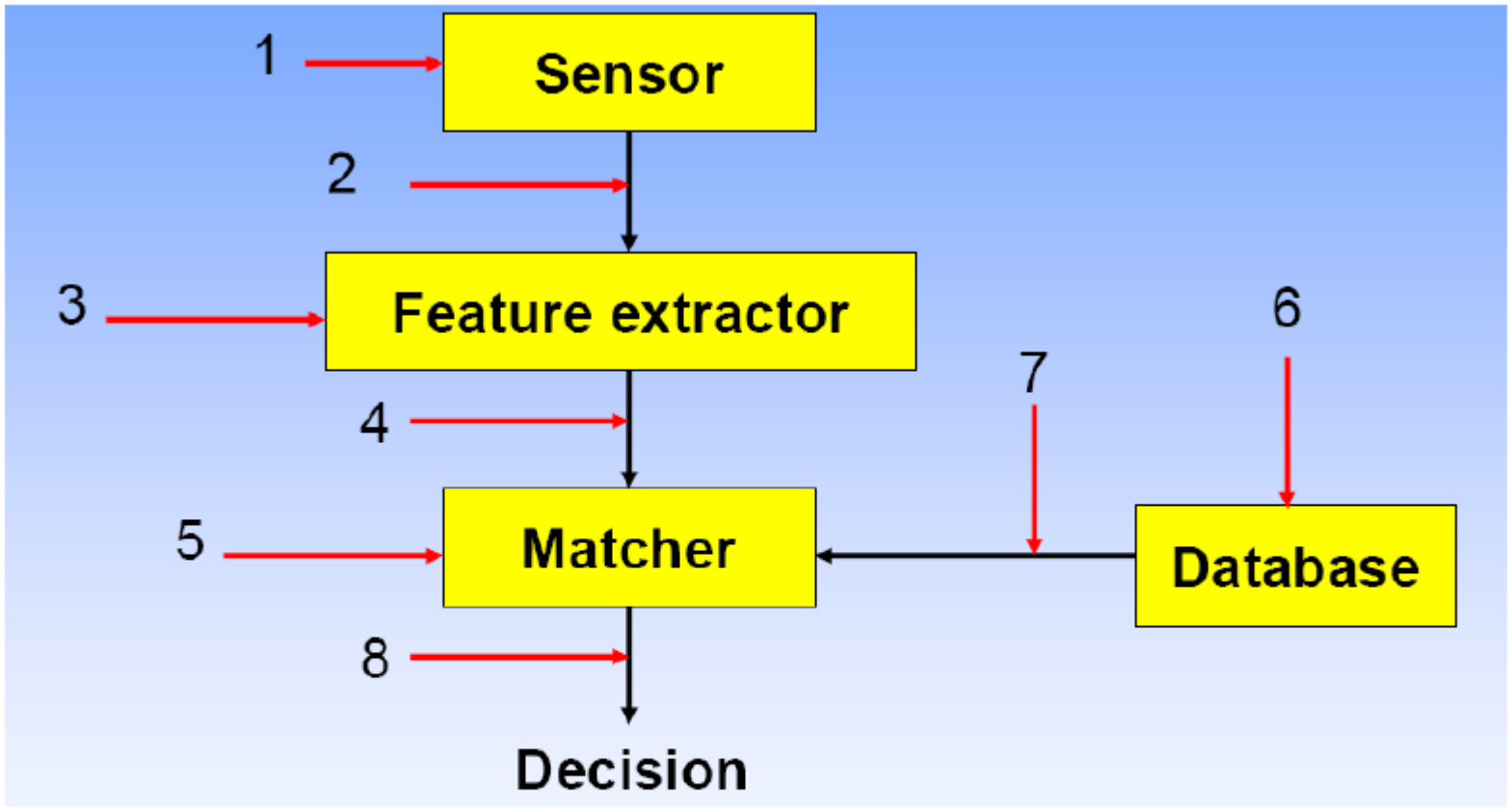
# Types of threats

- Circumvention: An attacker gains access to the system protected by biometric authentication
  - Privacy attack: Attacker accesses the data that she was not authorized (e.g., accessing the medical records of another user)
  - Subversive attack: Attacker manipulates the system (e.g., submitting bogus insurance claims)
- Repudiation: An attacker denies accessing the system
  - A bank clerk modifies the financial records and later claims that her biometric data was stolen and denies that she is responsible
- Contamination (covert acquisition): An attacker illegally obtains biometric data of genuine users and uses it to access the system
  - Lifting a latent fingerprint and constructing a synthetic finger

# Types of threats

- Collusion: A user with wide super user privileges (e.g., system administrator) illegally modifies the system
- Coercion: An attacker forces a legitimate user to access the system (e.g., using a fingerprint to access ATM at a gunpoint)
- Denial of Service (DoS): An attacker corrupts the biometric system so that legitimate users cannot use it
  - A server that processes access requests can be bombarded with many bogus access requests, to the point where the server's computational resources can not handle valid requests any more.

# Threats locations




Points of attack for a generic biometric system

# Threats locations

- Attack 1: A fake biometric (e.g., an artificial finger) is presented at the sensor
- Attack 2: Illegally intercepted data is resubmitted (replay)
- Attack 3: Feature detector is replaced by a Trojan horse program
  - It produces feature sets chosen by the attacker
- Attack 4: Legitimate features are replaced with a synthetic feature set
- Attack 5: Matcher is replaced by a Trojan horse program
  - It produces scores chosen by the attacker
- Attack 6: Templates in the database are modified, removed, or new templates are added
- Attack 7: The transferred template information is altered in the communication channel
- Attack 8: The matching result (e.g., accept/reject) is overridden

# Attack 1 Example

**Attack 1: Synthetic Biometric Submission**

- No detailed system knowledge or access privileges is necessary
- Digital protection mechanisms (e.g., encryption) are not applicable

*Putte, Keuning 2000*:

- 6 fingerprint verification systems attacked
- 5 out of 6 accepted the dummy finger in the first attempt

Dummy finger created with cooperation of the user in a few hours with liquid silicon rubber

Dummy finger created from a lifted impression of the finger without cooperation of the user in eight hours with silicon cement

# Attack 1 example

*Matsumoto et al. 2002*:

- 11 fingerprint verification systems attacked with artificial gelatin fingerprints
- Gelatin fingers accepted with a probability of 67-100%

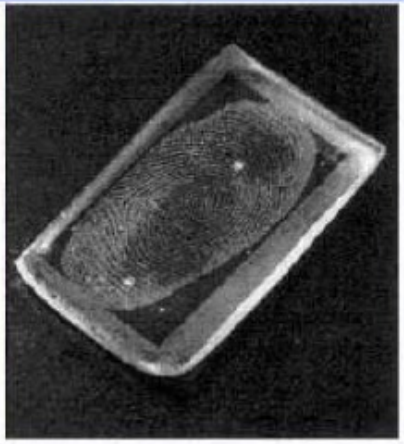
live

gelatin


With cooperation (finger pressed to plastic mold)

mold

gelatin



Without cooperation (residual fingerprint lifted from a glass)

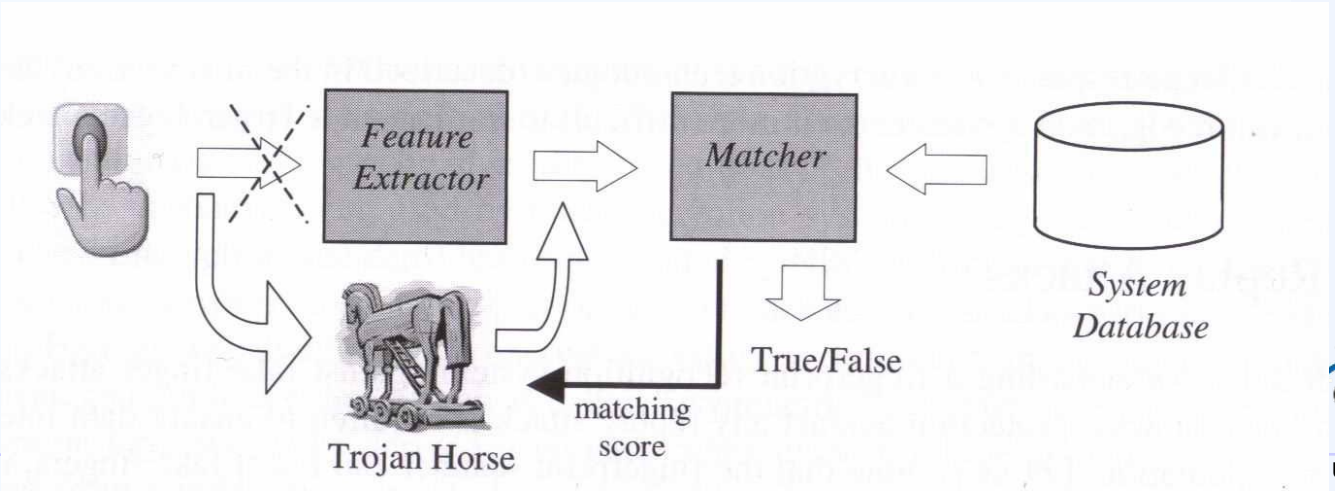
# Finger vitality detection

- Pattern classifier to discriminate between human and synthetic epidermis
- Vital signs
  - Temperature: at 20C epidermis higher by 8 to 10 degrees, fake finger 2 degrees less, cold coca-cola can
  - Conductivity: greatly varies, water or saliva is added to fake
  - Optical sensors: meassure absorption, reflection, scattering, refraction, gelation has similar optical properties to a live finger
  - Ultrasonic sensors: detec layer under epidermis, silicon rubber layer + silicon rubber finger
  - Increase resolution, detect sweet pores
  - Blood pressure, ECG meassurement
  - VIDEO

laboratory Gerstner

# Attack 5 – Trojan horse

- Sensor emulator, feature extraction, matcher, system database
- Digital signature
- Trust authorities – standard electronic commerce systems
- PIN example, 4 digits, 10^4 combinations
  - A) know password: easy
  - B) brute force attack
- FP example
  - FP representaion MUST be know: type of features, digital representation, quantization, spatial reference, ordering, etc.
  - Randomly generate FP representation, better synthetic generator, *apriori knowledge from latent fingerprints*
  - Gray-scale: use synthetic generator
- **Hill climbing**, match feedback available, iteratively details changing after positive feedback



# Other attacks

- **Hill Climbing:**
  - Repeatedly submit biometric data to an algorithm with slight differences, and preserve modifications that result in an improved score;
  - Can be prevented by
    - Limiting the number of trials;
    - Giving out only yes/no matches.
- **Swamping:**
  - Similar to brute force attack, exploiting weakness in the algorithm to obtain a match for incorrect data.
  - Submit a print with hundreds of minutiae in the hope that at least the threshold number of them will match the stored template;
  - Can be prevented by normalizing the number of minutiae.
- **Piggy-back:**
  - An unauthorized user gains access through simultaneous entry with a legitimate user (coercion, tailgating).