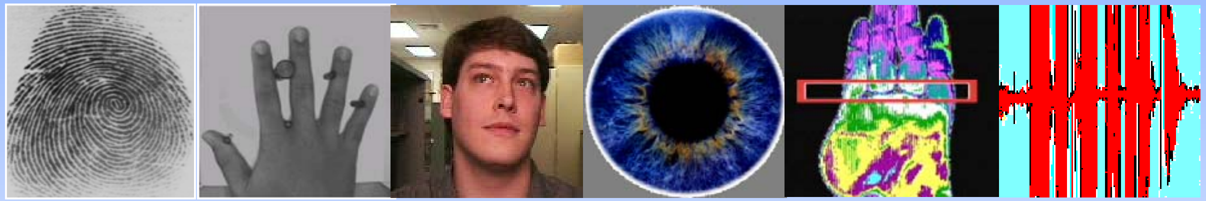
Fingerprint Recognition



Anil K. Jain
Michigan State University
jain@cse.msu.edu

http://biometrics.cse.msu.edu

Outline

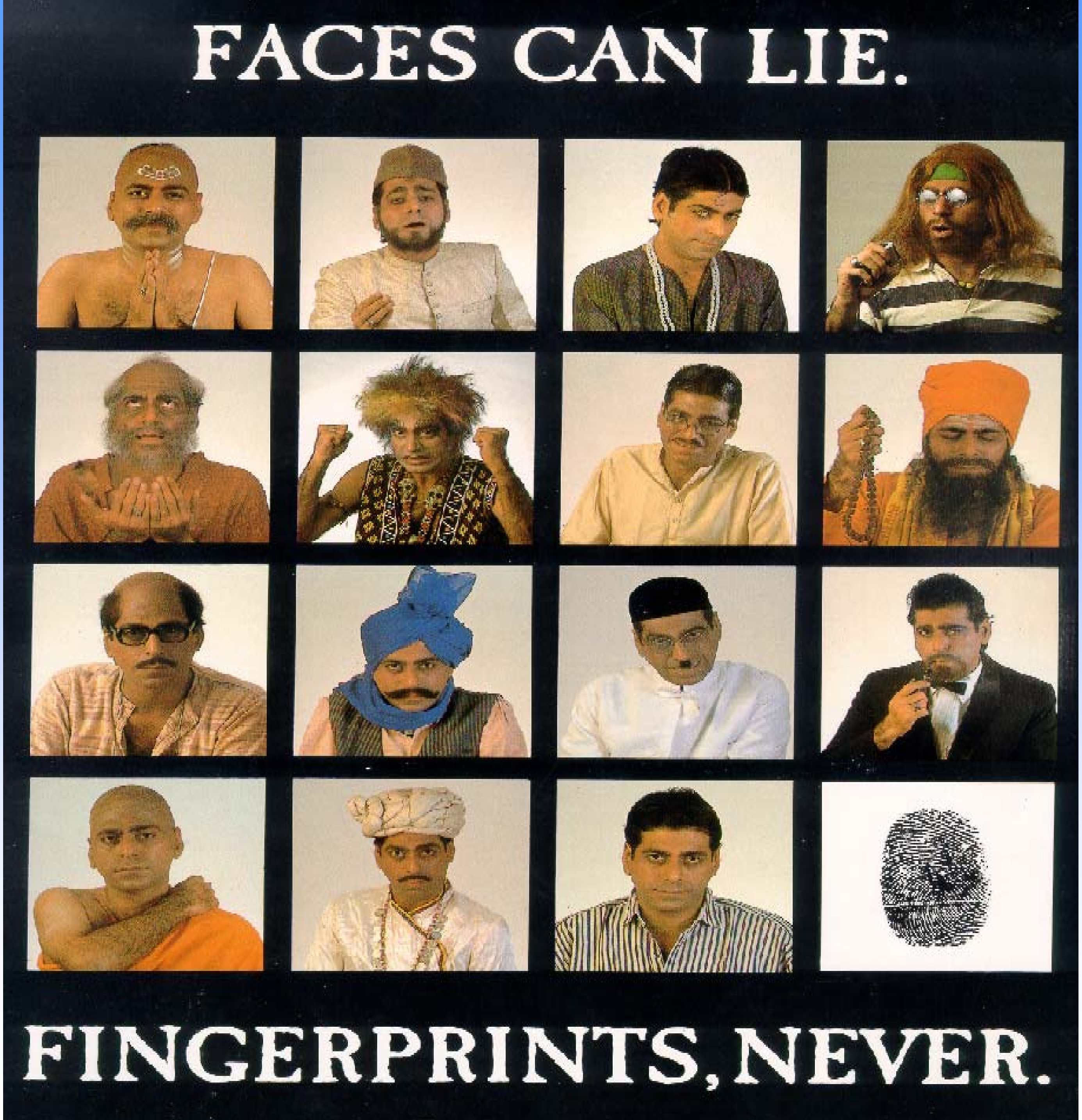
- Brief History
- Fingerprint Representation
- Minutiae-based Fingerprint Recognition
- Fingerprint Enhancement
- Other methods of Fingerprint Recognition
  - Ridge Feature Maps
  - Correlation
- Fingerprint Classification
- Fingerprint Individuality
- Estimating non-linear deformation

Fingerprints

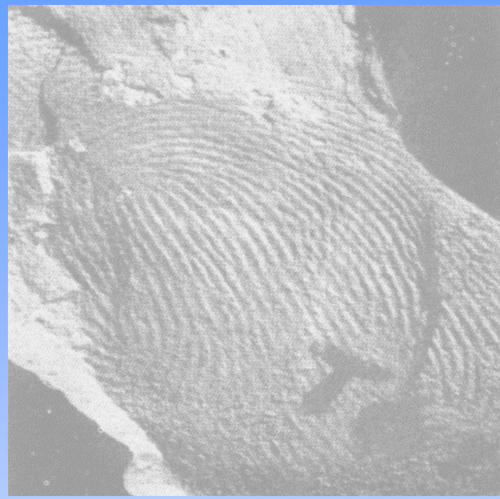
- Description: graphical flow like ridges present in human fingers
- Formation: during embryonic development
- Permanence: minute details do not change over time
- Uniqueness: believed to be unique to each finger
- History: used in forensics and has been extensively studied

Biological Principles of Fingerprints

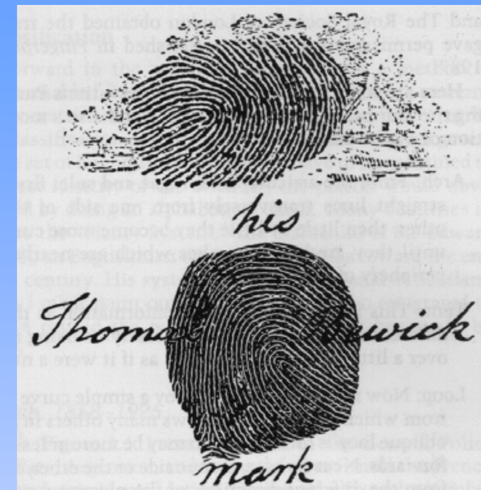
- Individual epidermal ridges and valleys have different characteristics for different fingerprints
- Configurations and minute details of individual ridges and valleys are permanent and unchanging (except for enlargement in the course of bodily growth)
- Configuration types are individually variable, but they vary within limits which allow for systematic classification


FACES CAN LIE.
FINGERPRINTS, NEVER.

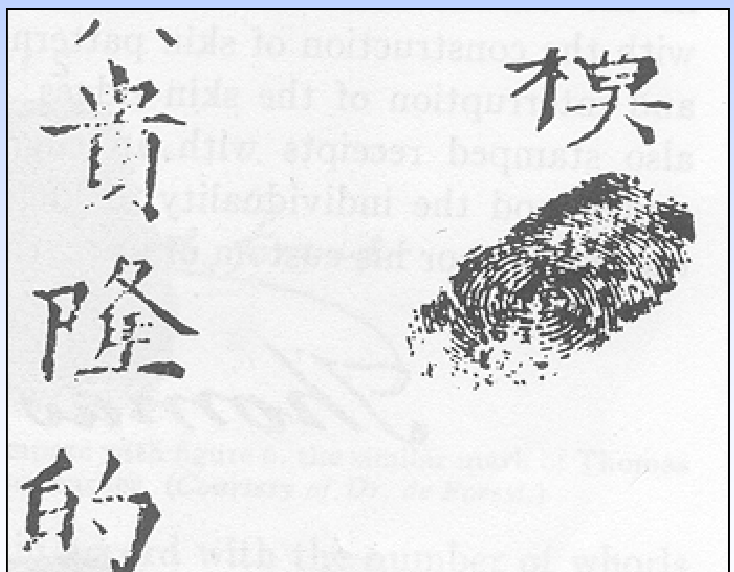
History of Fingerprints



Fingerprint on Palestinian lamp (400 A.D.)



Bewick’s trademark (1809)



A Chinese deed of sale (1839) signed with a fingerprint

History of Fingerprints

- Many impressions of fingers have been found on ancient pottery
- Grew (1684): first scientific paper on ridges, valleys & pore structures
- Mayer (1788): detailed description of anatomical formation of fingerprints
- Bewick (1809): used his fingerprint as his trademark
- Purkinje (1823): classified fingerprints into 9 categories based on ridges
- Herschel (1858): used fingerprints on legal contracts in Bengal
- Fauld (1880): suggested "scientific identification of criminals" using fingerprints
- Vucetich (1888): first known user of dactylograms (inked fingerprints)
- Scotland Yard (1900): adopted Henry/Galton system of classification
- FBI (1924) set up a fingerprint identification division with a database of 810,000 fingerprints
- FBI (1965): installed AFIS with a database of 810,000 fingerprints
- FBI (2000): installed IAFIS with a database of 47 million 10 prints; conducts an average of 50,000 searches/ day; ~15% of searches are in lights out mode. Response time: 2 hours for criminal search and 24 hours for civilian search

Fingerprint Matching

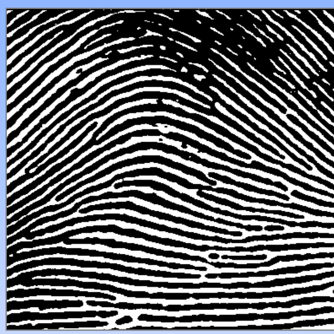
- Find the similarity between two fingerprints

Fingerprints from the same finger

Fingerprints from two different fingers

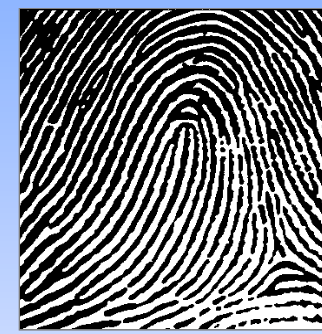
Fingerprint Classification

- Assign fingerprints into one of pre-specified types


Plain Arch


Tented Arch


Right Loop


Left Loop


Accidental


Pocket Whorl


Plain Whorl


Double Loop

THE HENRY SYSTEM

- DIVIDES ALL FINGERPRINT FORMS INTO 1,024 BUNDLES.

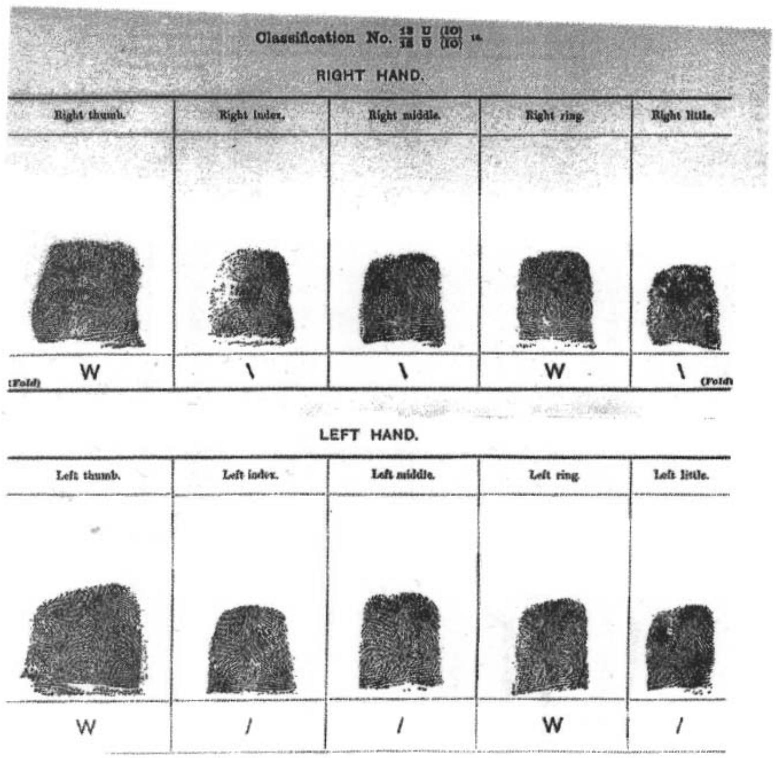


Figure 1: A Fingerprint Slip with Henry Code

1	2	3	4	5
RT 16	RF 16	RM 8	RR 8	RL 4
LT 4	LF 2	LM 2	LR 1	LL 1
6	7	8	9	10

1

W	W			W
	W	W	W	

$= \frac{19}{24}$

- WHORL PATTERNS ONLY HAVE VALUES.
- EVEN NUMBERS ON THE FORM CONSTITUTE THE NUMERATOR, ODD NUMBERS PROVIDE THE DENOMINATOR. ADD '1' TO BOTH NUM. & DENOM.

Fingerprint Sensors

- Optical, capacitive, ultrasound, pressure, thermal, electric field

Types of Fingerprint Images

Flat Fingerprint

(One-touch print from a single-finger livescan device)

Unsegmented Slap Fingerprint

(4-finger simultaneous impression from livescan devices or scanned from paper FP cards)

Rolled Fingerprint

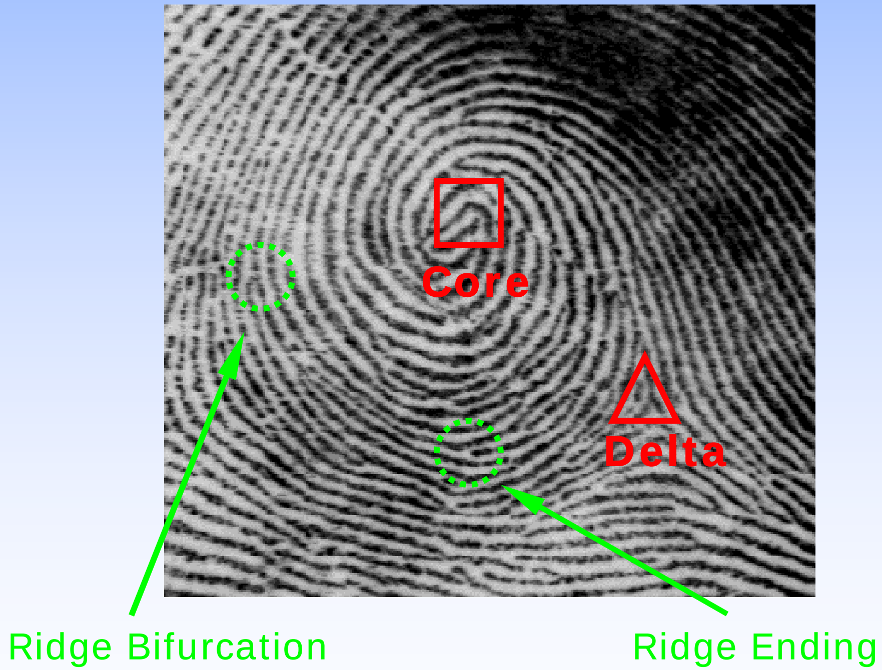
(Image collected by rolling the finger across the livescan platen or paper from nail to nail)

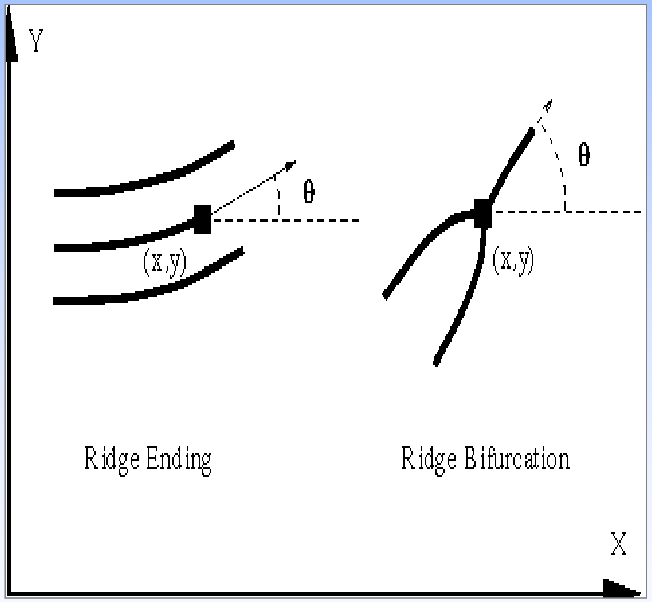
Terminology

- Fingerprint – Impression of a finger
- Minutiae – Ridge bifurcations,endings and many other features (52 types listed, 7 are usually used by human experts and two by automated systems)
- Core – uppermost point on the innermost ridge
- Delta – separating point between pattern area and non-pattern area
- Pattern class – determined by ridge flow characteristics

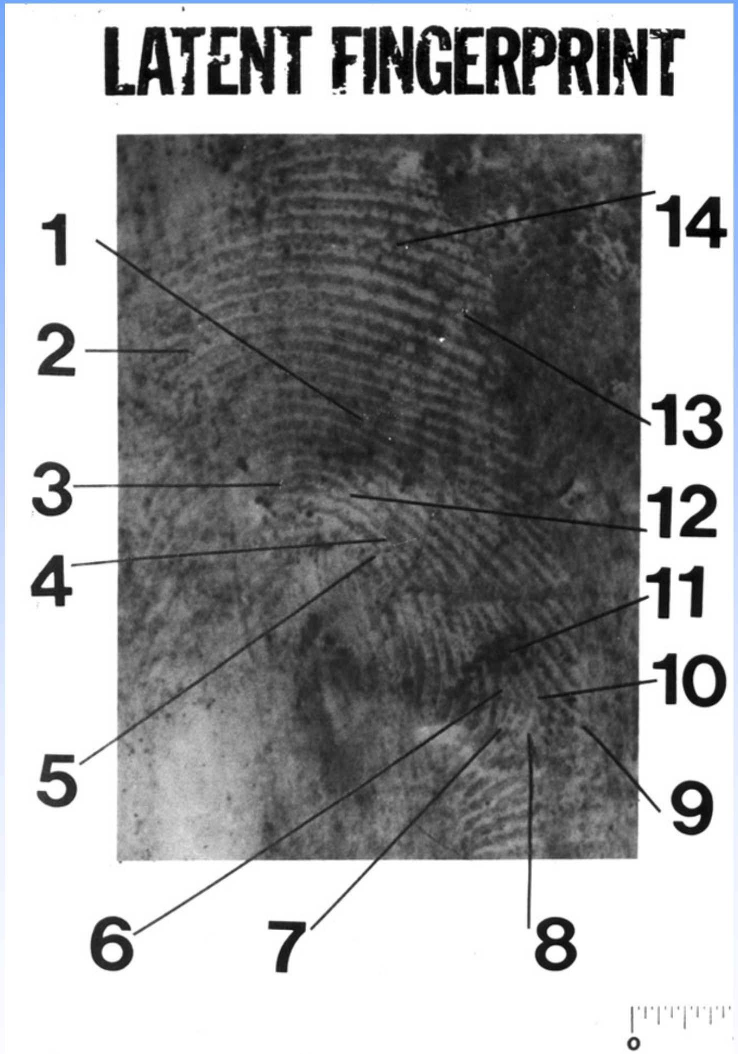
Fingerprint Representation

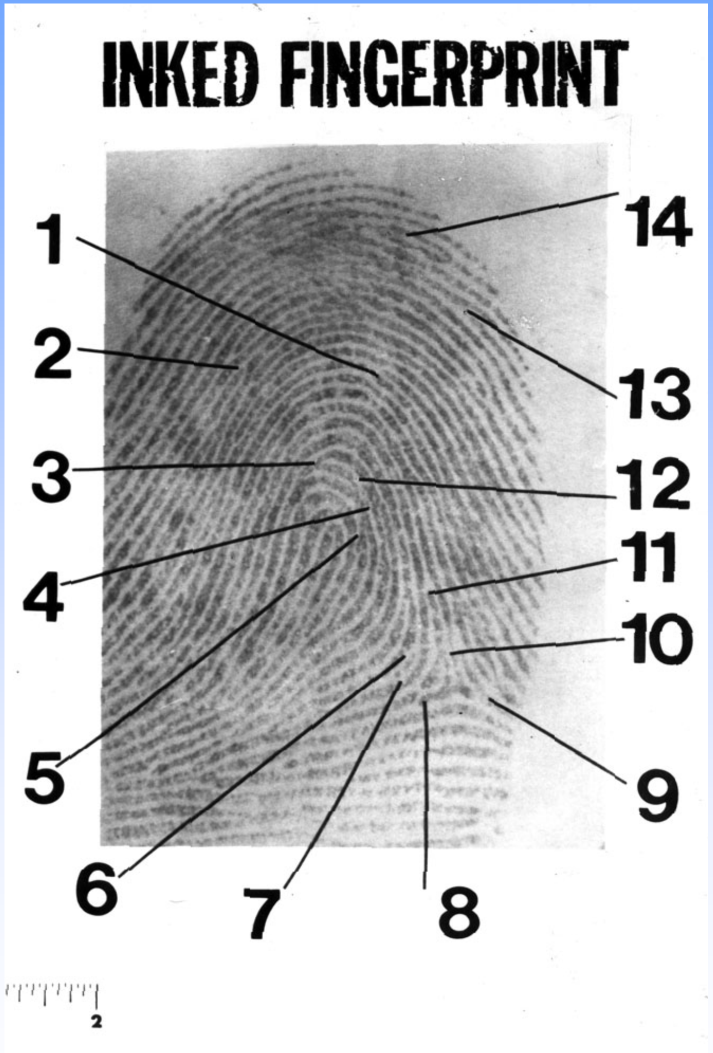
- Local ridge characteristics (minutiae): ridge ending and ridge bifurcation
- Singular points: Discontinuity in ridge orientation





Minutiae-based Representation

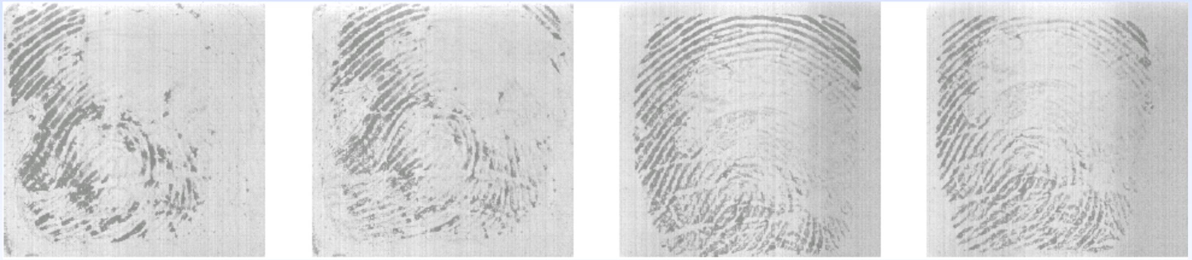




Challenges in Fingerprint Identification

- Cuts and bruises on finger; dry or oily fingers; ~3% of fingerprints are not of “good” quality
- Wear and tear of sensor; no proven contactless fingerprint sensor technology is currently available
- New compact solid-state sensors capture only a small portion of the fingerprint
- Fingerprint impression is often left on the sensor
- Non-universality of fingerprint
- Changes in sensor technology; sensor interoperability

Fake fingerprint

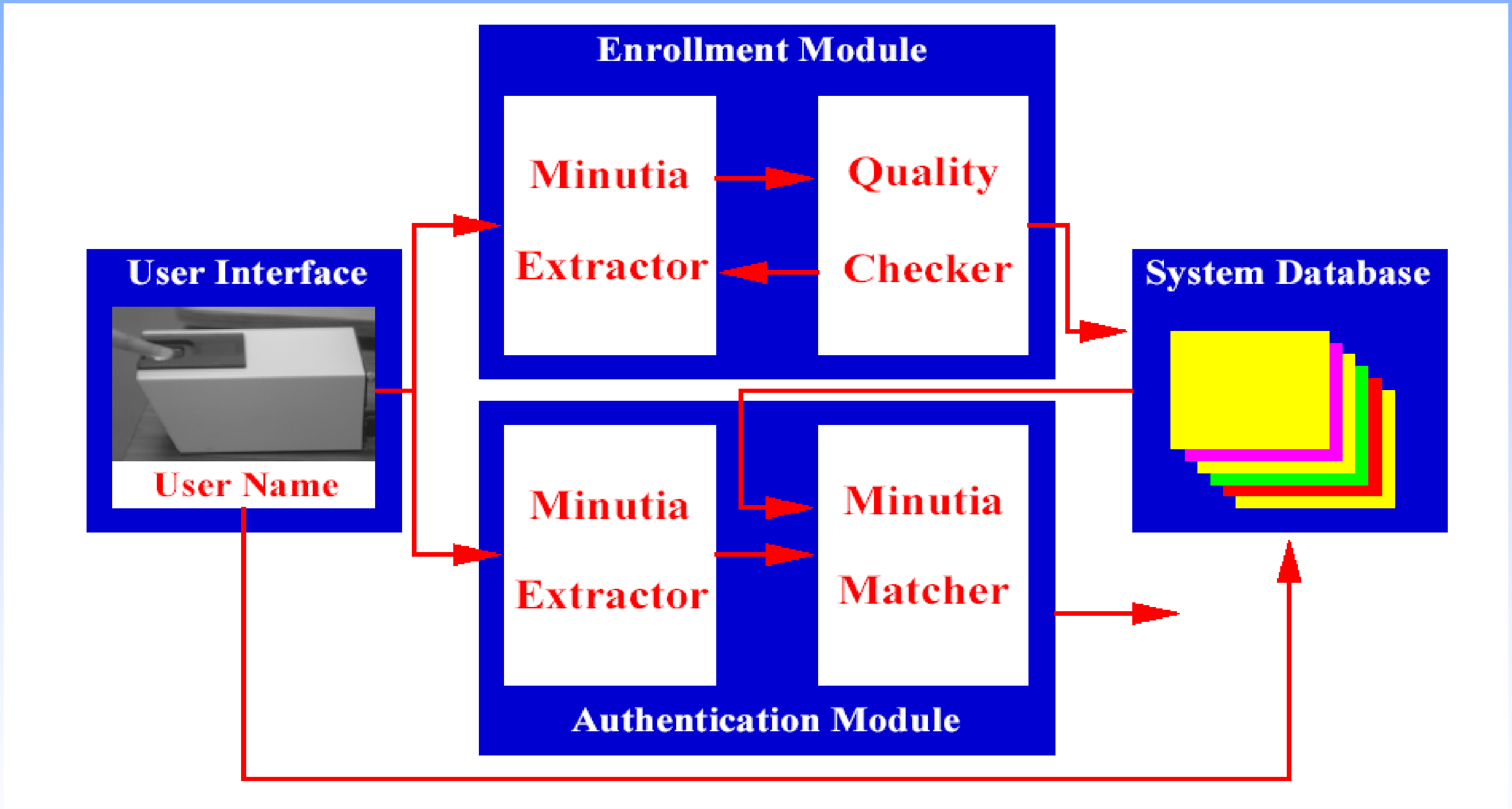


Non-universality of fingerprint

Fingerprint Matching Techniques

- Minutiae-based
- Correlation-based
- Ridge Feature-based

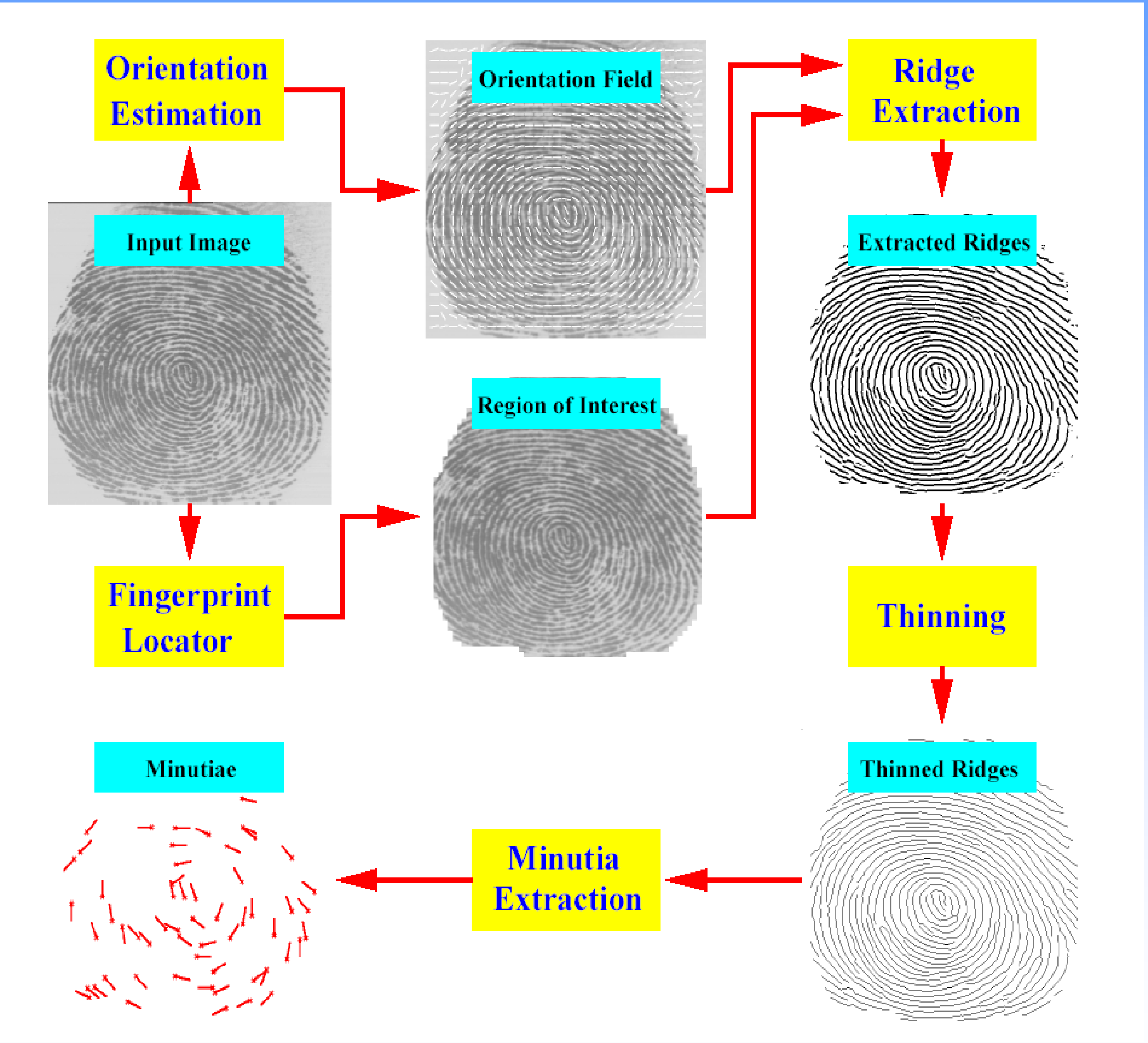
Minutiae-based Fingerprint Verification System



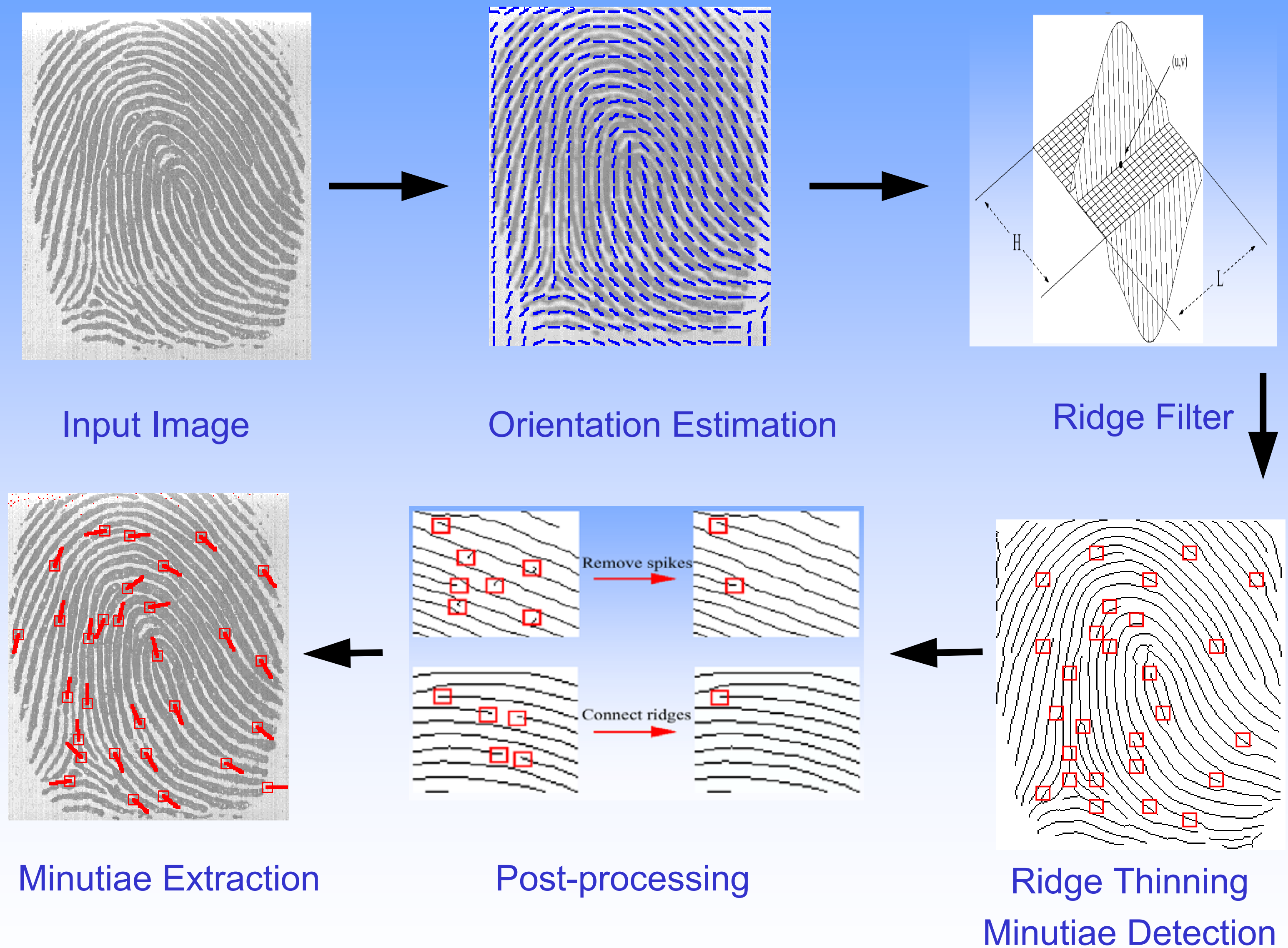
Steps in Minutiae Extraction

- Orientation field estimation
- Fingerprint area location
- Ridge extraction
- Thinning
- Minutia extraction

Minutiae Extraction Algorithm



Minutiae Extraction Algorithm



Orientation Field

- Angle formed by the ridges with horizontal axis
- Usually computed at the block level and not at the pixel level (more resistant to noise)
- The gradient direction is perpendicular to the ridge direction
- An arithmetic average of gradient angles is not well defined
- Doubling the angles allows local gradient estimates to be averaged

Orientation Field Estimation Algorithm

- Divide the input fingerprint image into non-overlapping blocks of size W x W
- Compute the gradients G_x (i, j) and G_y (i, j) at each pixel (i, j) using Sobel or Marr-Hildreth operator
- Least squares estimate of the local orientation of the block centered at (i, j) is

$$\theta(i,j) = \left(\frac{1}{2}\right)\tan^{-1}\left(\sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \frac{2G_x(u,v)G_y(u,v)}{G_x^2(u,v)-G_y^2(u,v)}\right)$$

Orientation Field Estimation Algorithm

- Local ridge orientation varies slowly in a neighborhood where no singular points appear
- A low-pass filter can be applied to modify the local ridge orientation
- To apply low pass filter the orientation image is converted into a continuous vector field

$$\Phi_x(i, j) = \cos(2\theta(i, j)), and$$
$$\Phi_y(i, j) = \sin(2\theta(i, j))$$

Orientation Field Estimation Algorithm

- Apply a 2-D low pass filter as follows

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} h(u,v)\Phi_x(i-uw, j-vw) \quad and$$

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} h(u,v)\Phi_y(i-uw, j-vw),$$

where $w_\Phi \times w_\Phi$ is the size of the filter, default is 5 x 5 blocks

Local ridge orientation at (i, j) is $O(i,j) = \left(\frac{1}{2}\right)\tan^{-1}\left(\frac{\Phi'_x(i,j)}{\Phi'_y(i,j)}\right)$

Orientation Field Consistency

- Consistency (actually inconsistency) level of the orientation field in the local neighborhood is

$$
\begin{aligned}
\mathcal{C}(i,j) &= \frac{1}{n}\sqrt{\sum_{(i',j')\in D} |\mathcal{O}(i',j') - \mathcal{O}(i,j)|^2}, \\
|\mathcal{O}(i',j') - \mathcal{O}(i,j)| &= \begin{cases} d & if\, d < 180, \\ d - 180 & otherwise, \end{cases} \\
d &= (\mathcal{O}(i',j') - \mathcal{O}(i,j) + 360)\ mod\ 360
\end{aligned}
$$

If C (i, j) is greater than T_c, re-estimate the local orientations in this block at a lower resolution level, until C (i, j) is less than T_c

Fingerprint Region Localization

- Separation of the fingerprint area from the background
- In the fingerprint area, the variance is very high in a direction orthogonal to the ridge orientation
- In the background regions, the variance has no directional dependence
- Local variance of gray level can be used to locate the fingerprint area
- Assumption: Only one fingerprint is present in the image

Fingerprint Location

Certainty level is defined as

$$\varepsilon(i, j) = \sqrt{\left(\frac{1}{W^2}\right)\left(\frac{G_x^2(i, j) + G_y^2(i, j)}{G_e^2(i, j)}\right)},$$

$$G_e(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \left(G_x^2(u,v) + G_y^2(u,v)\right).$$

For a block, if the certainty level is below a threshold T_k, then all pixels in that block are marked as background

Ridge Extraction

- Ridges and furrows run parallel to one another forming 2-D sine wave
- Gray-level values on the ridges attain their local maxima along a direction that is orthogonal to the local orientation
- The fingerprint image is convolved with two masks h_t (i, j; u, v) and h_b (i, j ; u, v) that are 180º out of phase

Ridge Extraction Filter

$$h_t(i,j;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}}, & \text{if } u = \left(v\cot(\mathcal{O}(i,j)) - \frac{H}{2\cos(\mathcal{O}(i,j))}\right), v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}}, & \text{if } u = (v\cot(\mathcal{O}(i,j))), v \in \Omega \\ 0, & \text{otherwise,} \end{cases}$$

$$h_b(i,j;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}}, & \text{if } u = \left(v\cot(\mathcal{O}(i,j)) + \frac{H}{2\cos(\mathcal{O}(i,j))}\right), v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}}, & \text{if } u = (v\cot(\mathcal{O}(i,j))), v \in \Omega \\ 0, & \text{otherwise,} \end{cases}$$

$$\Omega = \left[-\left|\frac{L\sin(\mathcal{O}(i,j))}{2}\right|, \left|\frac{L\sin(\mathcal{O}(i,j))}{2}\right|\right],$$

Ridge Extraction

- On average, L x H is 11 x 7. Ideally the mask width must be equal to the width of the local ridge
- Pixel (i, j) is labeled as ridge pixel if both the gray-level values of the pixel (i, j) of the convolved images are greater than T_{ridge}
- The masks also do some smoothing, determined the parameter δ
- For computational efficiency, δ is made large enough to make the filter coefficient degenerate to a single value

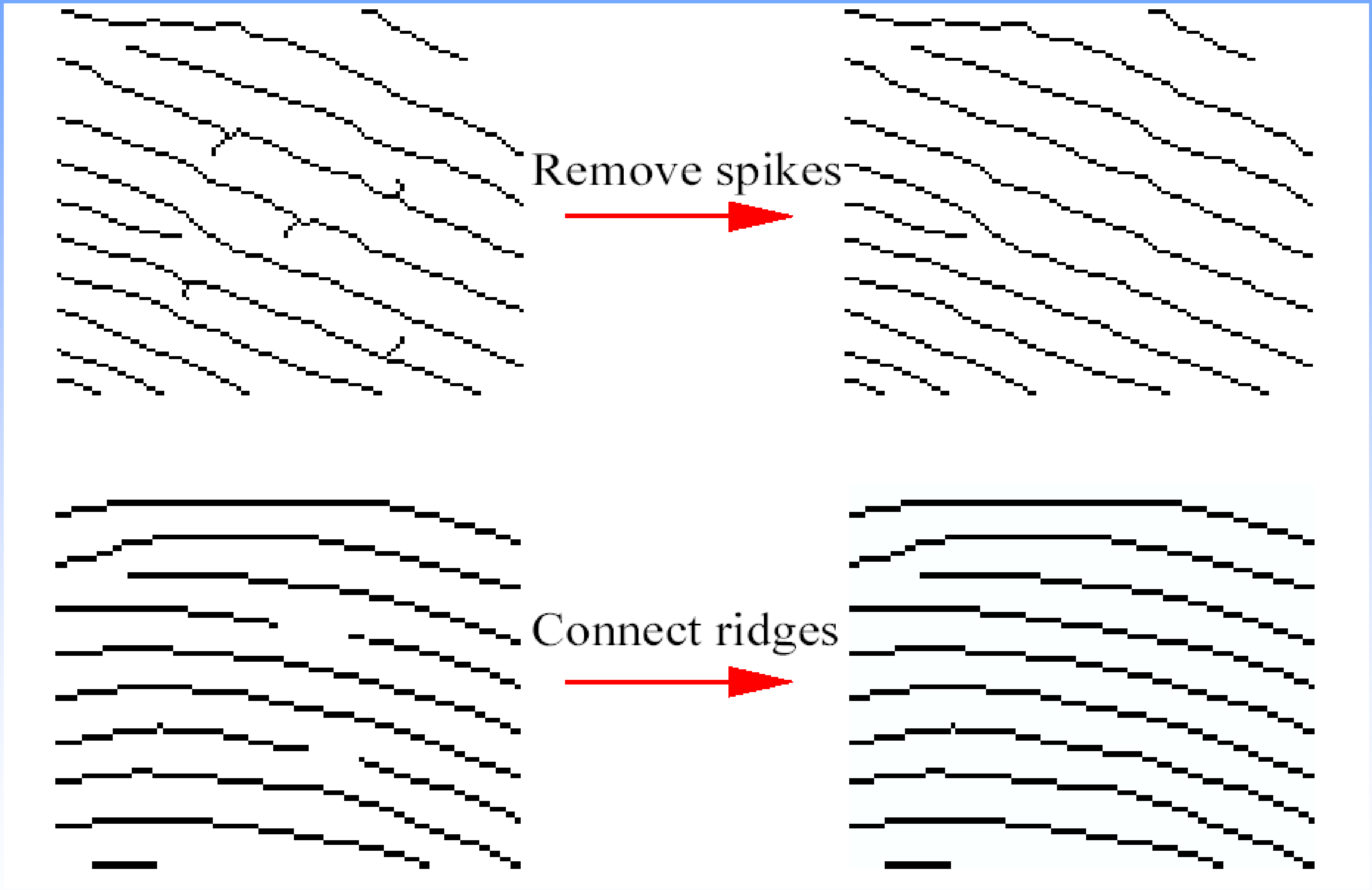
Removing Holes and Speckles

- The binary ridge map has many holes and speckles due to noise, smudges, breaks, etc.
- Connected component algorithm is used to remove holes and speckles
- Remove the connected components having less than 50 pixels
- Thinning is applied to obtain 8-connected thinned ridges of one pixel width

Minutiae Detection

- Spikes and breaks in the thinned ridge map leads to detection of spurious minutiae
- Heuristics are applied to post-process the thinned ridge map
- If the angle between the branch and the trunk is greater than 70º and less than 110º, then the branch is removed
- If the break in the ridges is less than 15 pixels and no other pixel passes through it, then the break is connected

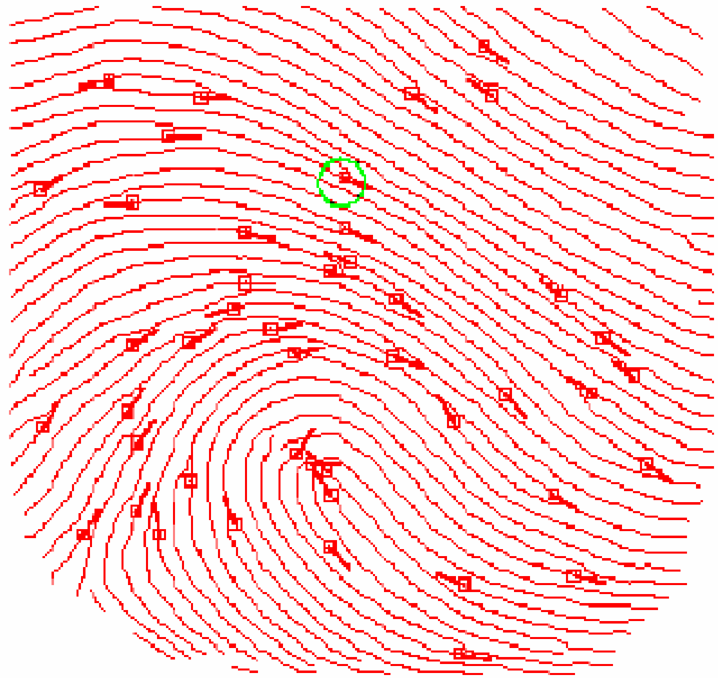
Minutiae Detection

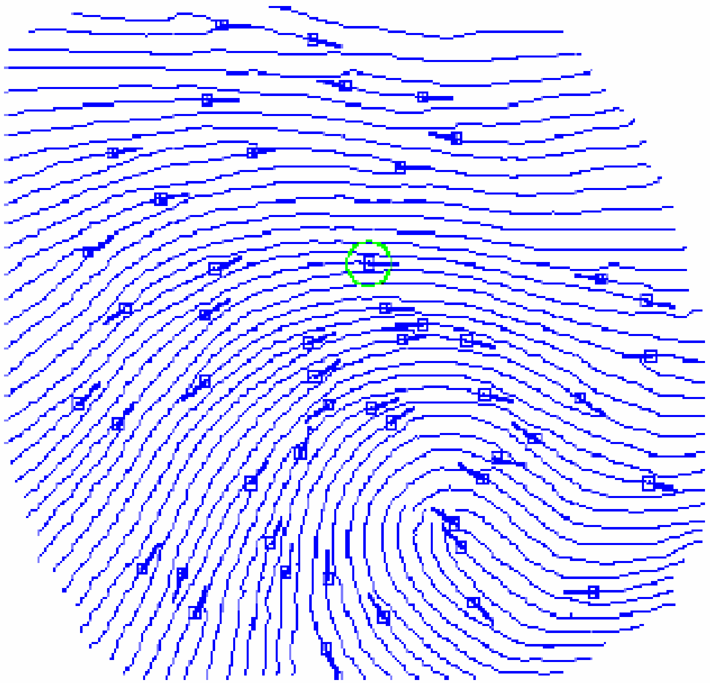


Minutiae Type Detection

- A ridge pixel is a ridge ending, if the number of ridge pixels in the 8-neighborhood is 1
- A ridge pixel is a ridge bifurcation, if the number of ridge pixels in the 8-neighborhood is greater than or equal to 3
- A ridge pixel is a intermediate ridge pixel, if the number of ridge pixels in the 8-neighborhood is 2
- [x, y, θ, associated ridge] are stored for each minutia

Minutiae Point Pattern Matching





Minutiae Matching

- Point pattern matching problem
- Let $P = \left\{\left(x_1^P, y_1^P, \theta_1^P\right), \ldots, \left(x_M^P, y_M^P, \theta_M^P\right)\right\}$

  be the set of M minutiae in the template image

- Let $Q = \left\{\left(x_1^Q, y_1^Q, \theta_1^Q\right), \ldots, \left(x_N^Q, y_N^Q, \theta_N^Q\right)\right\}$

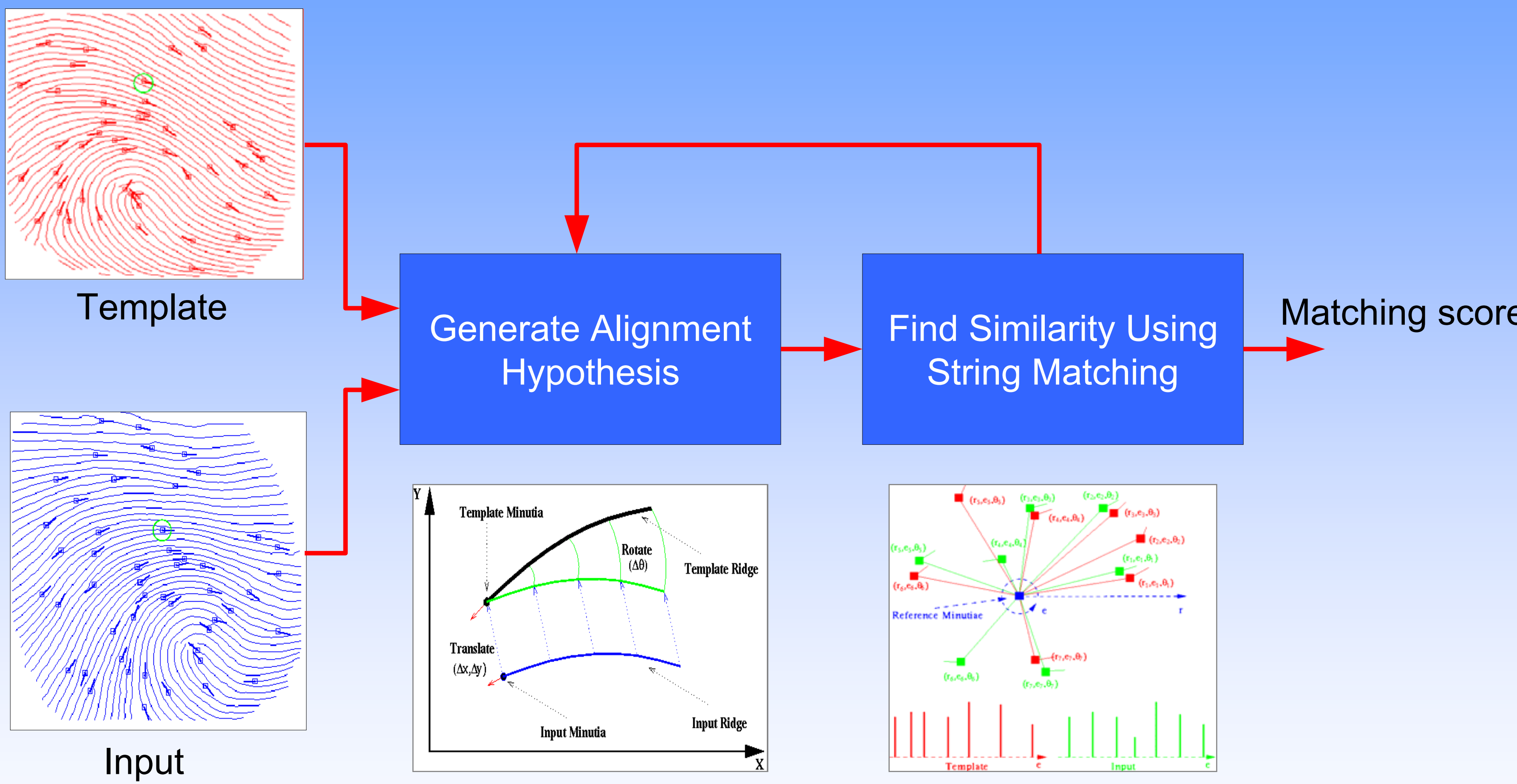
  be the set of N minutiae in the input image

- Find the number of corresponding minutia pairs between P and Q and compare it against a threshold

Stages in Minutiae Matching

- Alignment
  - Estimate rotation, translation, and distortion
  - Input fingerprint is aligned with the template
- Matching
  - Compute the similarity between the pre-aligned input and the template using the number of matching minutiae

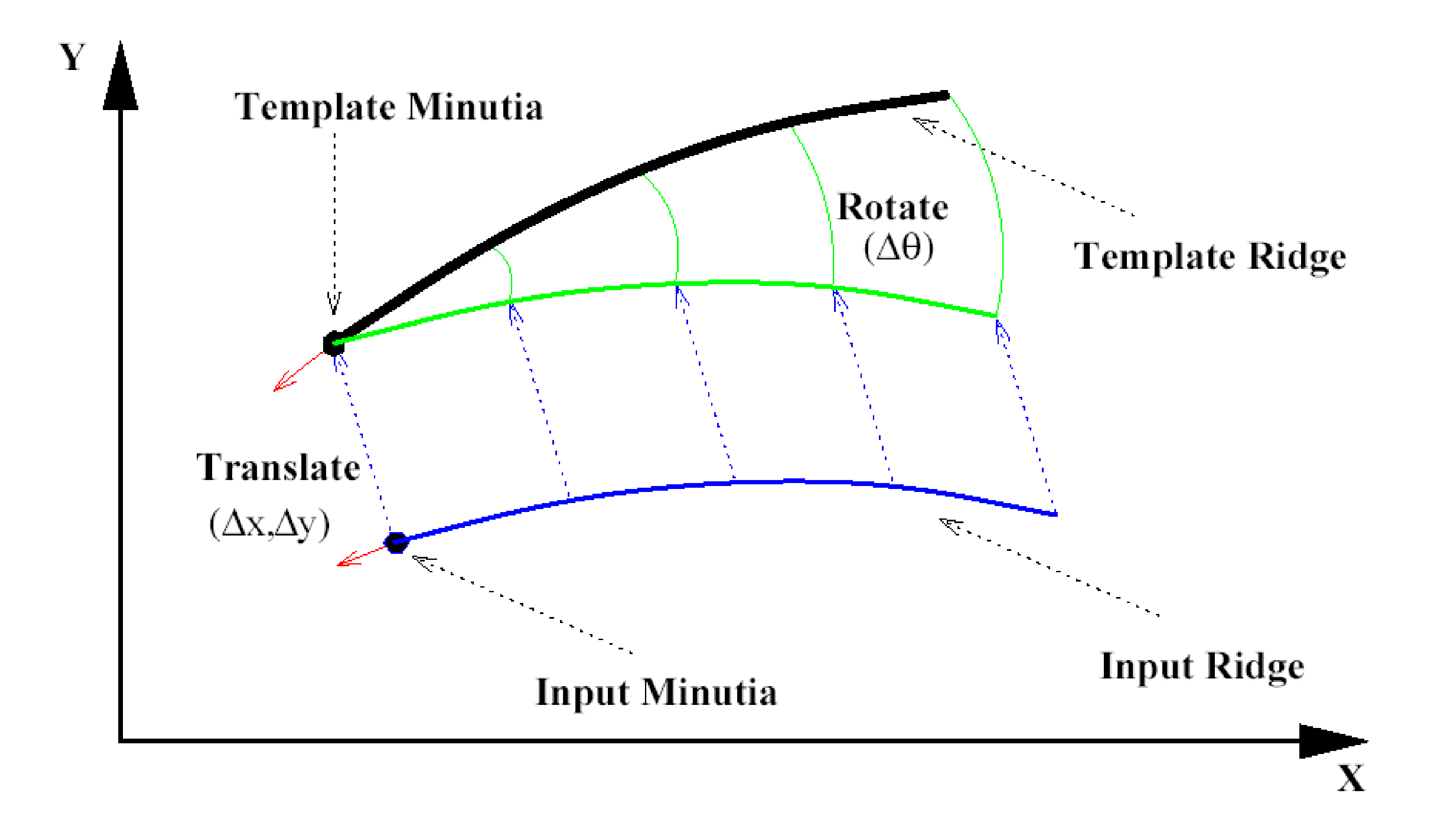
Minutiae Matching



Alignment of point patterns

- Corresponding point pairs are not known
- Due to noise and deformations, the input minutiae cannot be aligned exactly with respect to the template minutiae
- Ridges corresponding to the minutia are aligned

Alignment of Input and Template Ridges



Alignment Algorithm

- Ridges are represented as 1-D discrete signals, normalized wrt the inter-ridge distance
- Let R^d and R^D be the set of ridges in the input and template images
- Each ridge $d \in R^d$ is matched with $D \in R^D$ as

$$S = \frac{\sum_{i=0}^{L} d_i D_i}{\sqrt{\sum_{i=0}^{L} d_i^2 D_i^2}},$$

L is the minimal length of the two ridges, d_i and D_i represent the distances from point i to the x-axis

Alignment Algorithm

- If S is greater than a threshold (0.8), go to the next step. Otherwise continue to match the next pair of ridges
- Compute the translation vector and rotation angle as follows:

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x^d \\ y^d \end{pmatrix} - \begin{pmatrix} x^D \\ y^D \end{pmatrix},$$

$$\Delta\theta = \frac{1}{L}\sum_{i=0}^{L}(\gamma_i - \Gamma_i),$$

γ_i, and Γ_i are the radial angles of the i^{th} point with respect to the reference minutia

Alignment Algorithm

- Let $(x^d, y^d, \theta^d)^T$ be the reference minutia (based on which the transformation parameters are estimated). The N input minutiae are transformed as follows:

$$\begin{pmatrix} x_i^A \\ y_i^A \\ \theta_i^A \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{pmatrix} + \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta & 0 \\ \sin\Delta\theta & -\cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i - x^d \\ y_i - y^d \\ \theta_i - \theta^d \end{pmatrix}$$

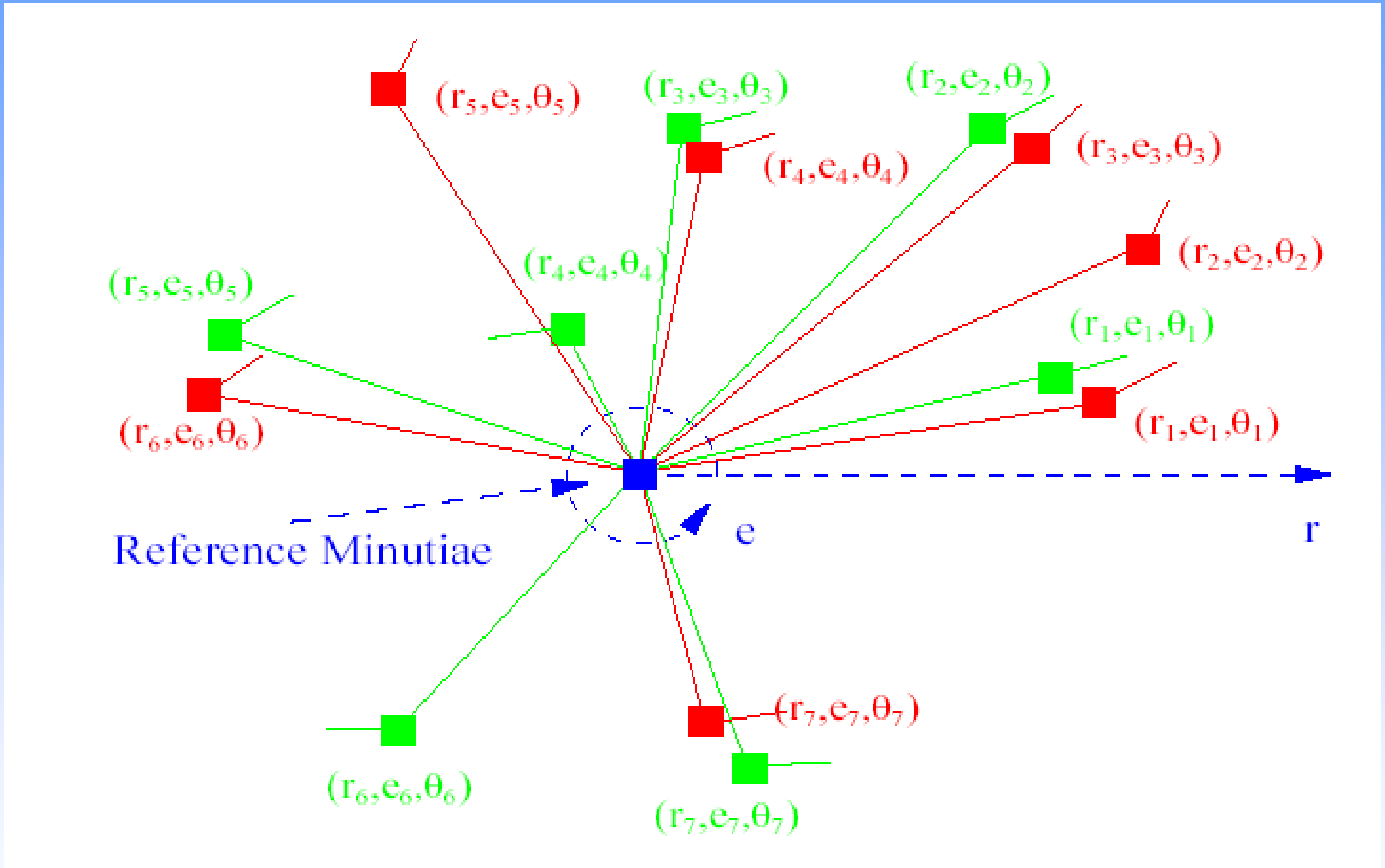
Matching Algorithm

- Estimate the rotation and translation parameters and align the two minutiae patterns
- Convert the template and input patterns into the polar coordinate system with respect to the reference minutiae and represent them as strings by concatenating each minutia in an increasing order of radial angles

$$P_p = \left\{ \left(r_1^P, e_1^P, \theta_1^P \right), \ldots, \left(r_M^P, e_M^P, \theta_M^P \right) \right\}$$

$$Q_p = \left\{ \left(r_1^Q, e_1^Q, \theta_1^Q \right), \ldots, \left(r_N^Q, e_N^Q, \theta_N^Q \right) \right\}$$

String Formation



String Matching

- Match the strings P_p and Q_p with a modified dynamic programming algorithm to find the edit distance
- Edit distance must have the elastic property of string matching
- Edit distance is defined recursively as follows:

$$C(m,n) = \begin{cases} 0 & \text{if } m=0 \text{ and } n=0 \\ \min\left\{ \begin{array}{c} C(m-1,n)+\Omega \\ C(m,n-1)+\Omega \\ C(m-1,n-1)+w(m,n) \end{array} \right\} & 0 < m \le M \text{ and } 0 < n \le N, \end{cases}$$

where

$$w(m,n) = \begin{cases} \alpha\left|r_m^P - r_n^Q\right| + \beta\Delta e + \gamma\Delta\theta & \text{if } \left|r_m^P - r_n^Q\right| < \delta,\ \Delta e < \epsilon \text{ and } \Delta\theta < \rho \\ \Omega & \text{otherwise,} \end{cases}$$

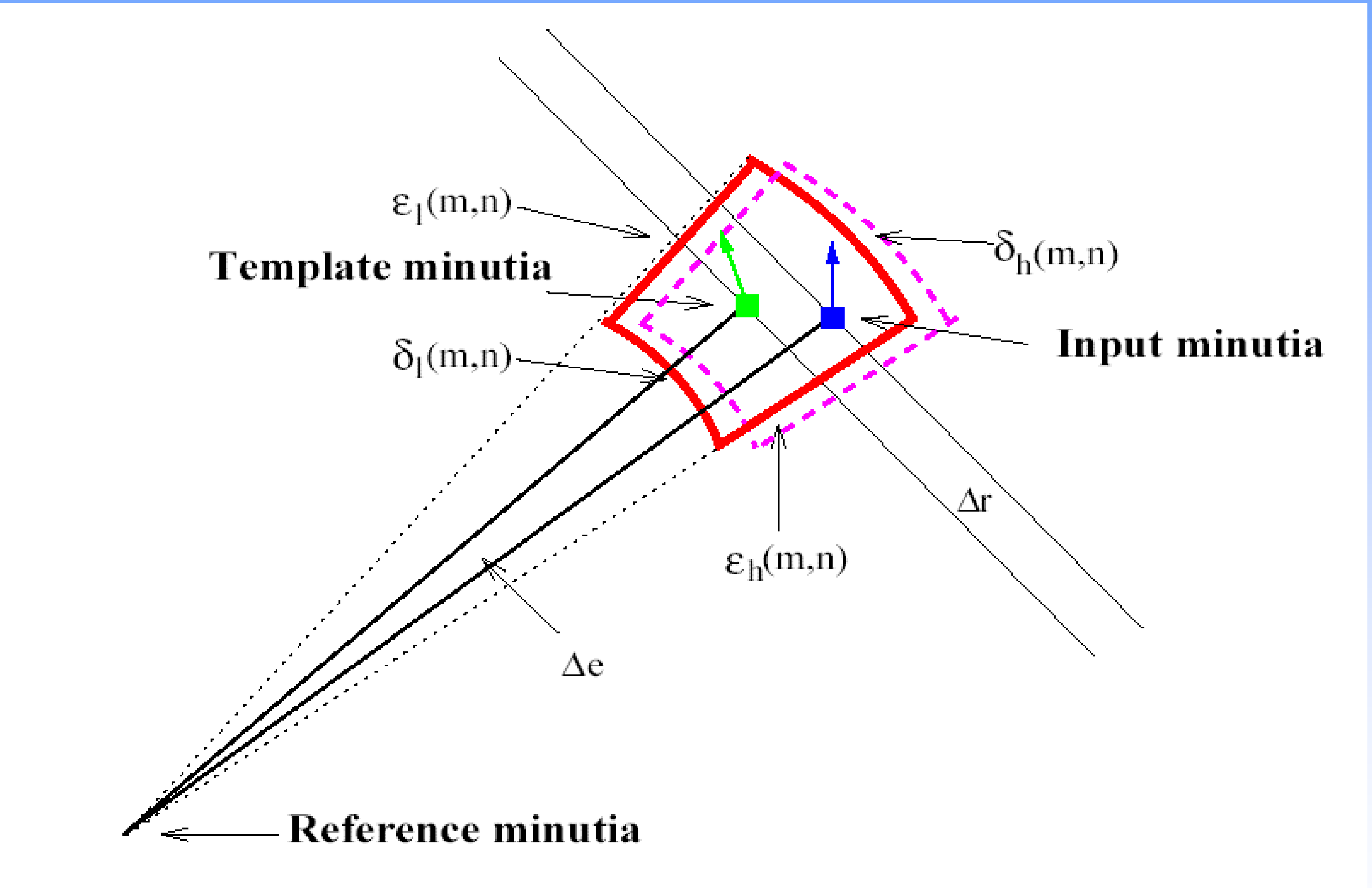
$$\Delta e = \begin{cases} a & \text{if } (a = (e_m^P - e_n^Q + 360) \bmod 360) < 180 \\ a - 180 & \text{otherwise,} \end{cases}$$

$$\Delta\theta = \begin{cases} a & \text{if } (a = (\theta_m^P - \theta_n^Q + 360) \bmod 360) < 180 \\ a - 180 & \text{otherwise,} \end{cases}$$

String Matching

- α, β, and γ are the weights associated with radius, radial angle, and orientation components
- δ, ε, and ρ specify the bounding box
- Ω is the pre-specified penalty for mismatch
- This scheme can tolerate non-linear deformations and inexact transformations to some extent, but cannot compensate for it
- An adaptive bounding box can be used

Bounding Box Adjustment



$$
w'(m,n) = \begin{cases} \alpha \left| r_m^P - r_n^Q \right| + \beta \Delta e + \gamma \Delta\theta & \text{if} \begin{cases} \delta_l(m,n) < (r_m^P - r_n^Q) < \delta_h(m,n) \\ \epsilon_l(m,n) < \Delta e < \epsilon_h(m,n) \\ \Delta\theta < \rho \end{cases} \\ \Omega & \text{otherwise,} \end{cases}
$$

where

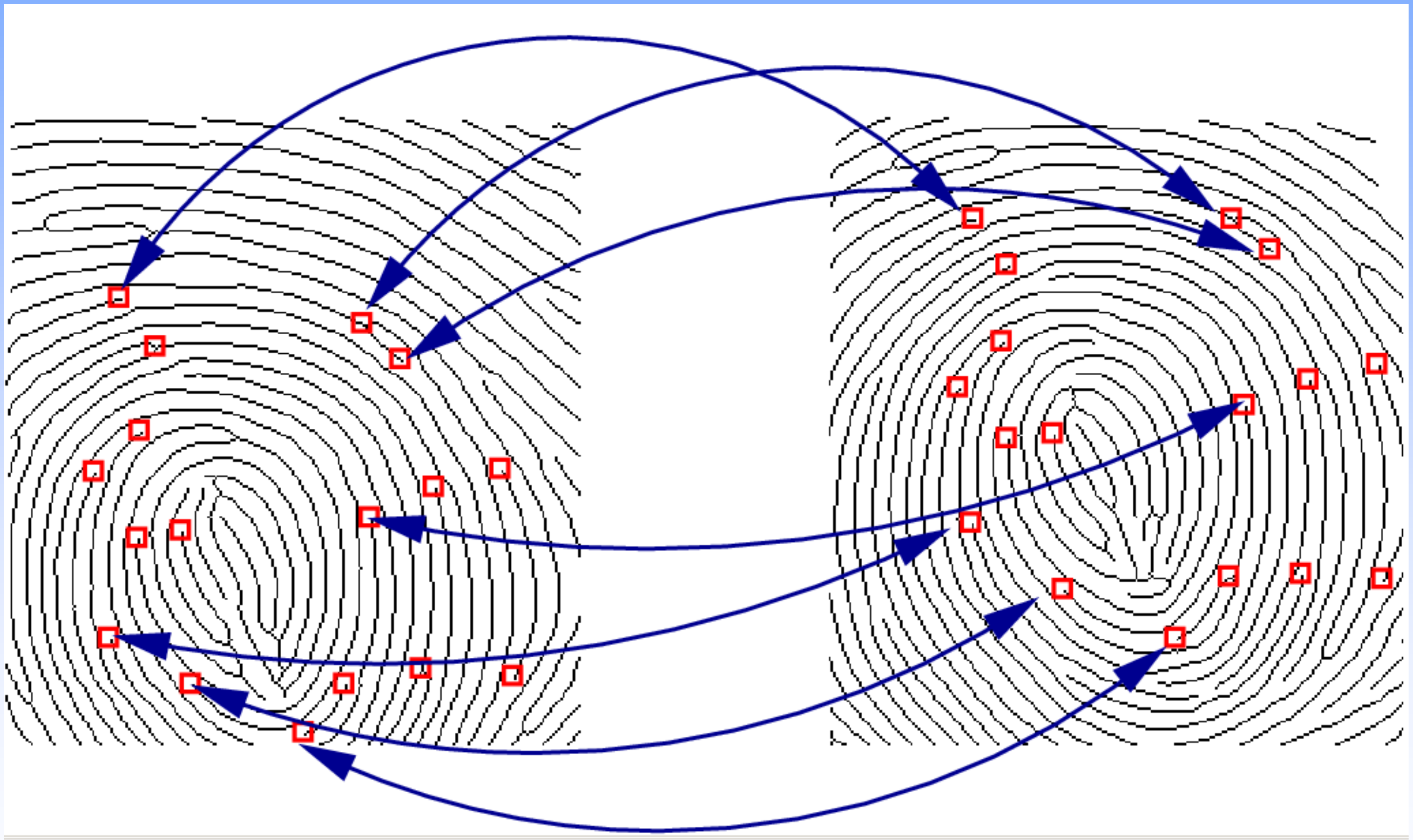
$$
\begin{pmatrix} \Delta r_a \\ \Delta e_a \end{pmatrix} = \begin{cases} \begin{pmatrix} r_m^P - r_n^Q \\ \Delta e \end{pmatrix} & \text{if} \begin{cases} \delta_l(m,n) < (r_m^P - r_n^Q) < \delta_h(m,n) \\ \epsilon_l(m,n) < \Delta e < \epsilon_h(m,n) \\ \Delta\theta < \rho \end{cases} \\ 0 & \text{otherwise,} \end{cases}
$$

$$
\begin{aligned}
\delta_l(m+1,n+1) &= \delta_l(m,n) + \eta \Delta r_a, \\
\delta_h(m+1,n+1) &= \delta_h(m,n) + \eta \Delta r_a, \\
\epsilon_l(m+1,n+1) &= \epsilon_l(m,n) + \eta \Delta e_a, \\
\epsilon_h(m+1,n+1) &= \epsilon_h(m,n) + \eta \Delta e_a,
\end{aligned}
$$

String Matching

- $\delta_l(m,n)$, $\delta_h(m,n)$, $\varepsilon_l(m,n)$, and $\varepsilon_h(m,n)$ specify the adaptive bounding box; initial values depend on the resolution of the fingerprint image
- η is the learning rate
- Parameter values are determined empirically

Minutiae Correspondences

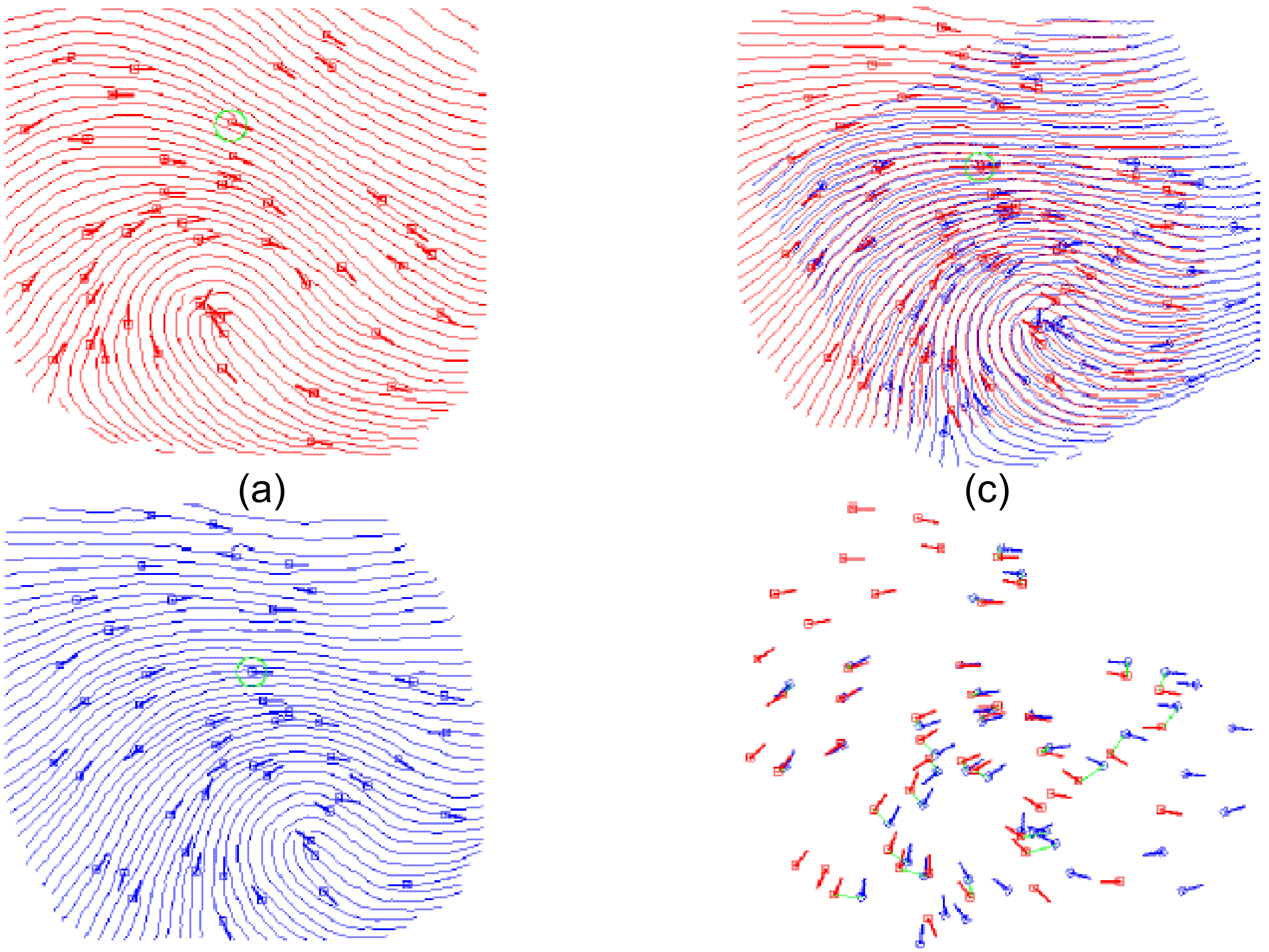


Matching Score

- The edit distance is used to establish the correspondence of the minutiae between P_p and Q_p
- Total number of corresponding minutiae is computed as M_{PQ}
- The matching score is

$$S = \frac{100 M_{PQ} M_{PQ}}{MN}$$

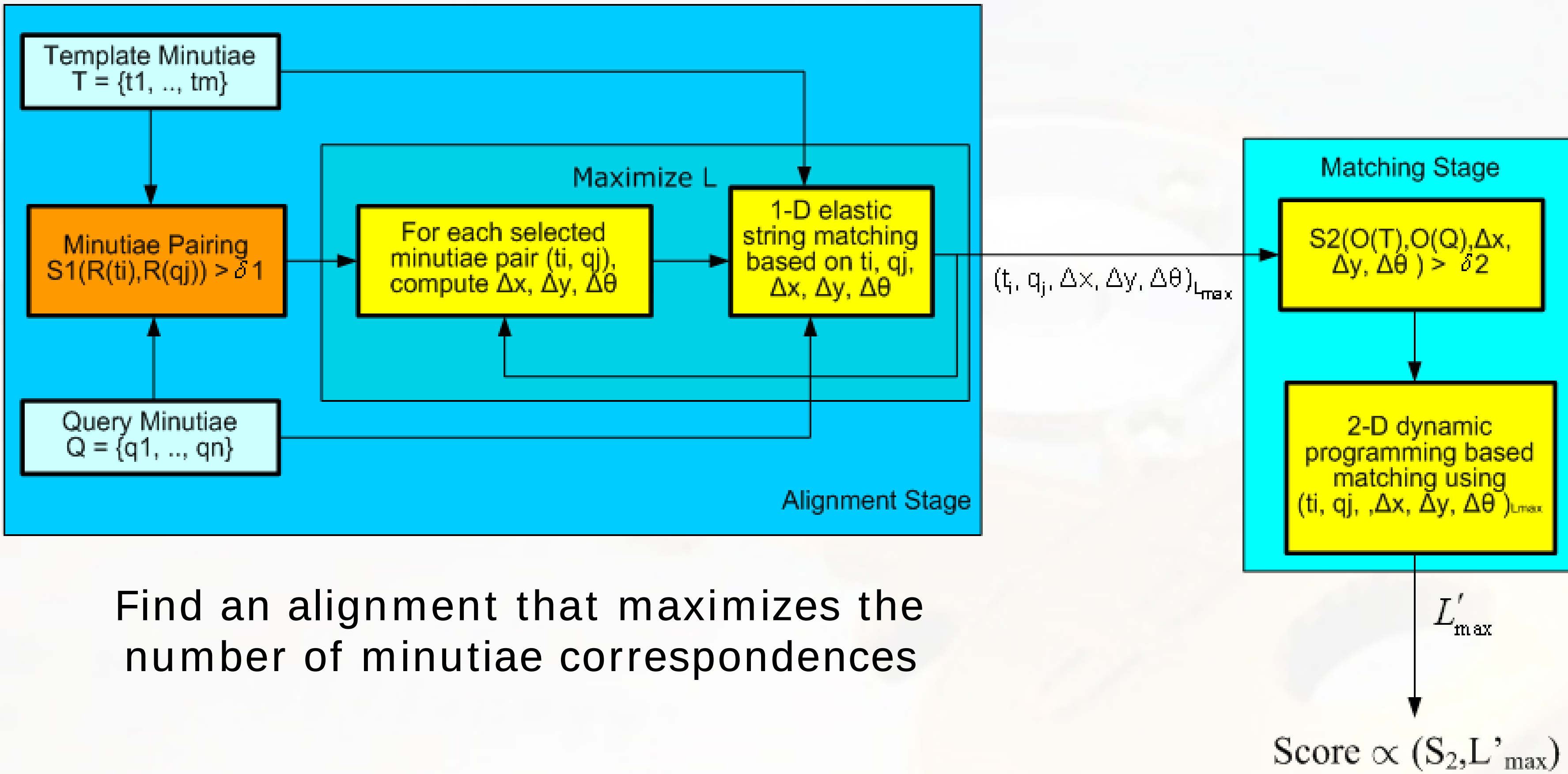
Minutiae Matching Result



(a) (c)

(b) (d)

2-D Dynamic Programming based Minutiae Matching



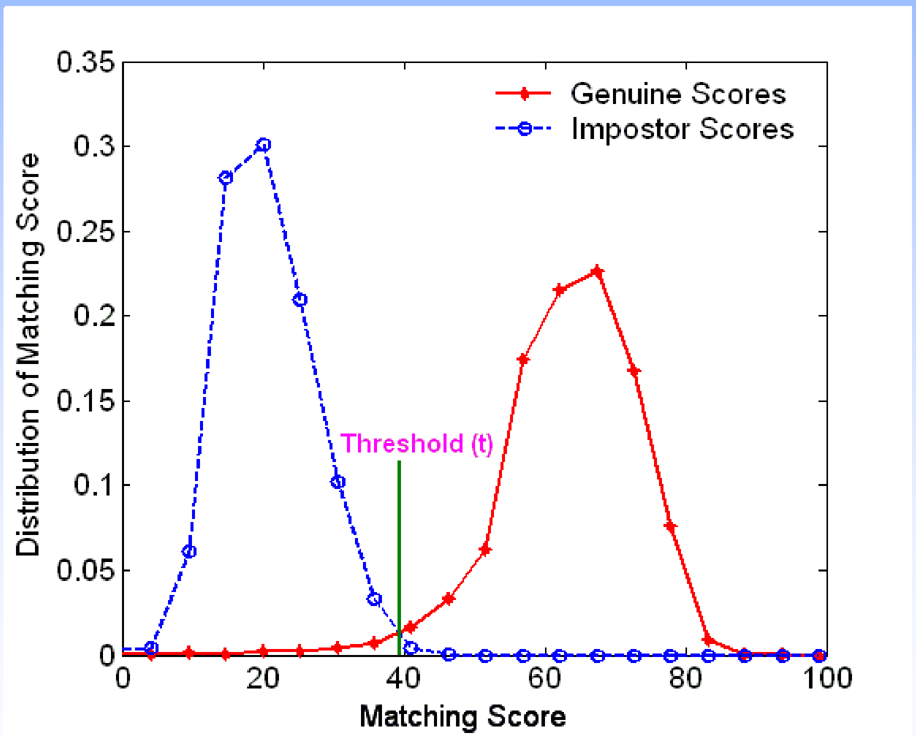
Find an alignment that maximizes the number of minutiae correspondences

S_1 → Ridge similarity measure, S_2 → Orientation similarity measure,
R(t) → 1-D representation of ridge points of minutia t, O(T) → Orientation field

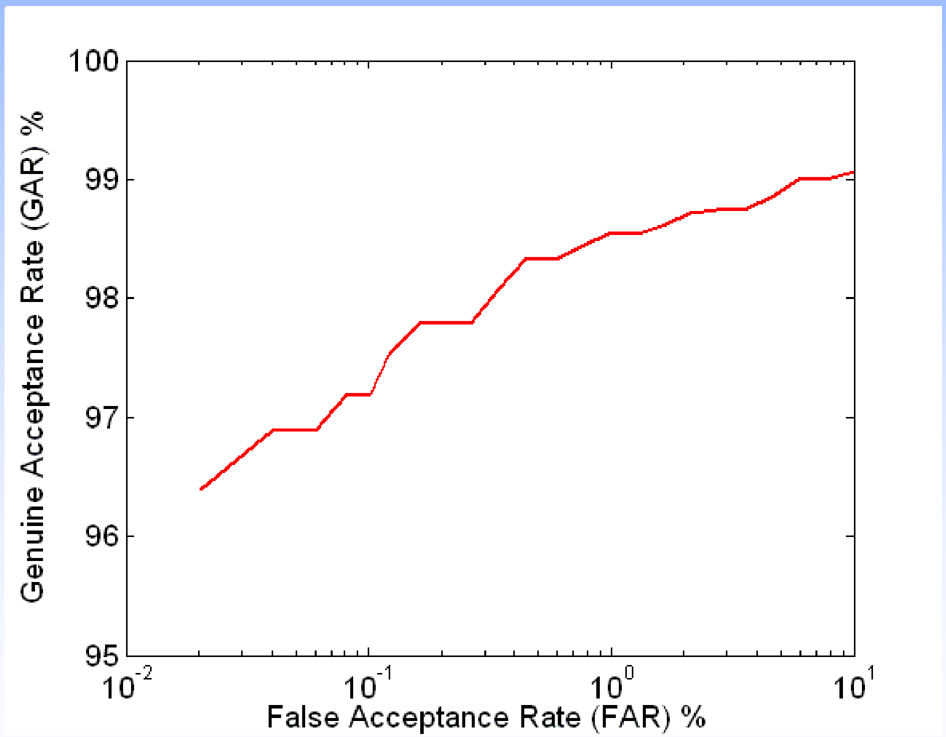
Matching time ~0.1 sec.

Matching Score Distributions

- Performance depends on the database. FVC2002 Database 1 (100 users, 8 impressions/user)
- For FAR = 0.1% (1 in 1000), GAR = 97.1%
- EER = 1.65%; at 0% False Accept, FRR = 4%



Matching Score Distribution



ROC Curve

Analysis of Errors

- Minutiae Extraction
  - Extraction stage does not extract all minutiae and their ridges
  - There may be no corresponding minutiae having ridge points for finding the correct alignment
- Alignment
  - Corresponding minutiae with ridge points exist
  - Alignment step fails due to small number of correspondences
- Matching
  - Estimated alignment is correct
  - But, the matching score is low because the number of correspondences is low compared to the number of minutiae
  - Reasons: deformation, spurious and missing minutiae

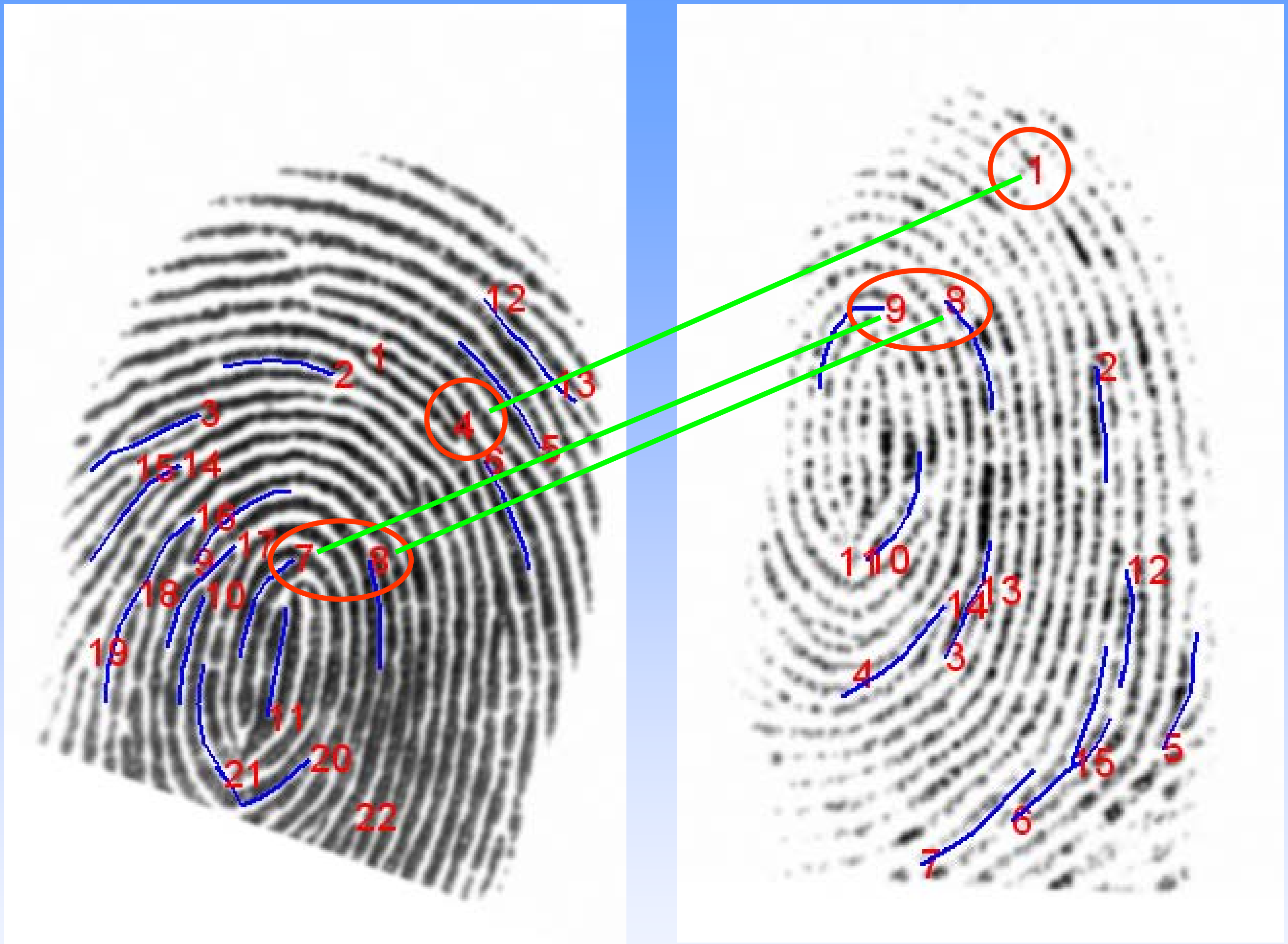
Minutiae Extraction Failure

True Minutiae Matches: A1→B3, A18→B9, A19→B7

A1, B9 and B7 were detected, but the associated ridges were not detected because they are close to the boundary

Alignment Failure

A



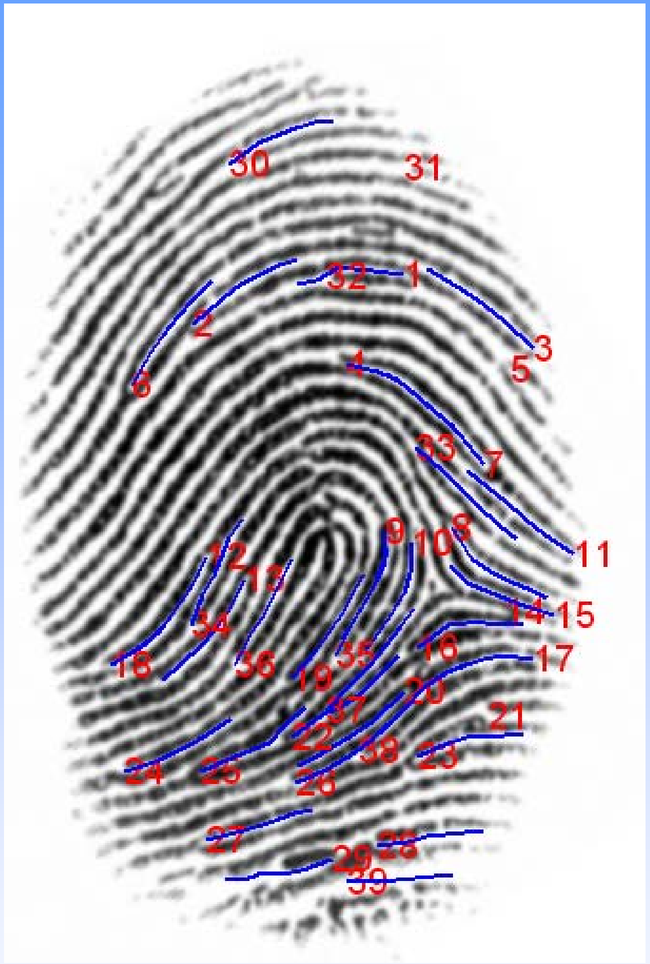
B

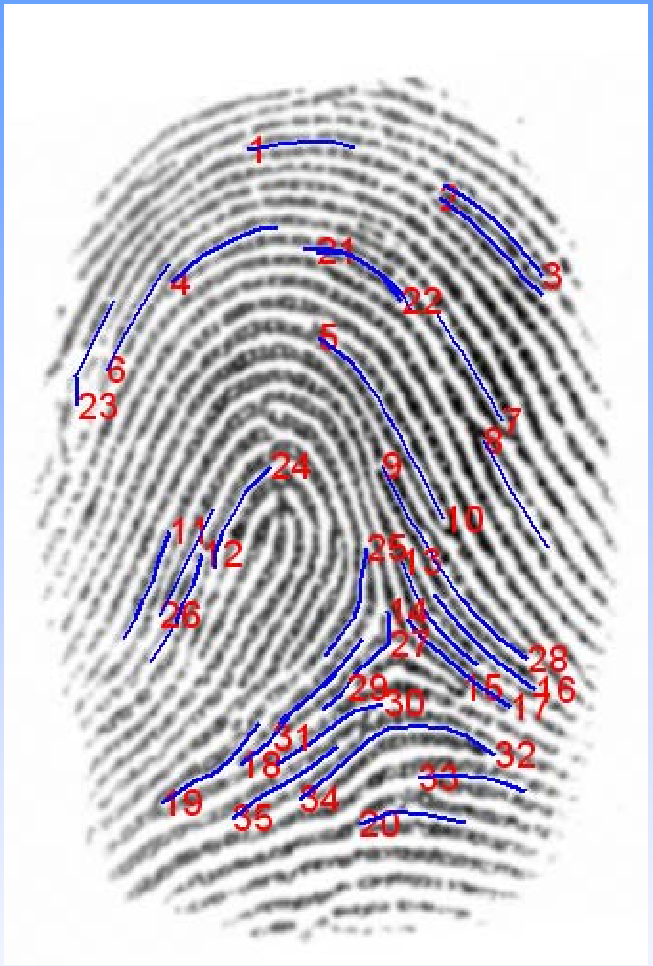
True Minutiae Matches: A7→B9, A8→B8, A4→B1

A7→B9 and A8→B8 pairs have ridge points; however, there exists a false alignment that results in more than three matches

Matching Failure

A





B

No. of matching minutiae identified by the matcher = 10

No. of minutiae in A = 38; No. of minutiae in B = 34

Spurious minutiae and large deformation leads to small score