Chapter 2: Fingerprint Recognition

Jain, Ross, Nandakumar. *Introduction to Biometrics.*

Chapter Guide/Notes

2.1 INTRODUCTION

- Skin on palms/soles exhibits flow-like pattern of ridges and valleys
- Friction ridges on fingers assist in grasping objects; appear as soon as 4th month of gestation
- Two layers: Dermis (inner layer) and Epidermis (outer layer where unique -even between twins! - ridges emerge)
- Used to identify repeat offenders, perform background checks, and identify partial, or latent, prints left at crime scenes

TYPES OF FINGERPRINTS

(a) (b) (c)

Fig. 1. Three types of fingerprint impressions. Rolled and plain fingerprints are also called full fingerprints. (a) Rolled; (b) plain; (c) latent.

FINGERPRINT RECOGNITION ON SMARTPHONES…WITHOUT A DESIGNATED SENSOR…

Vivo X20 Plus UD in-display fingerprint scanner survives scratches

0

JC Torres - Feb 26, 2018

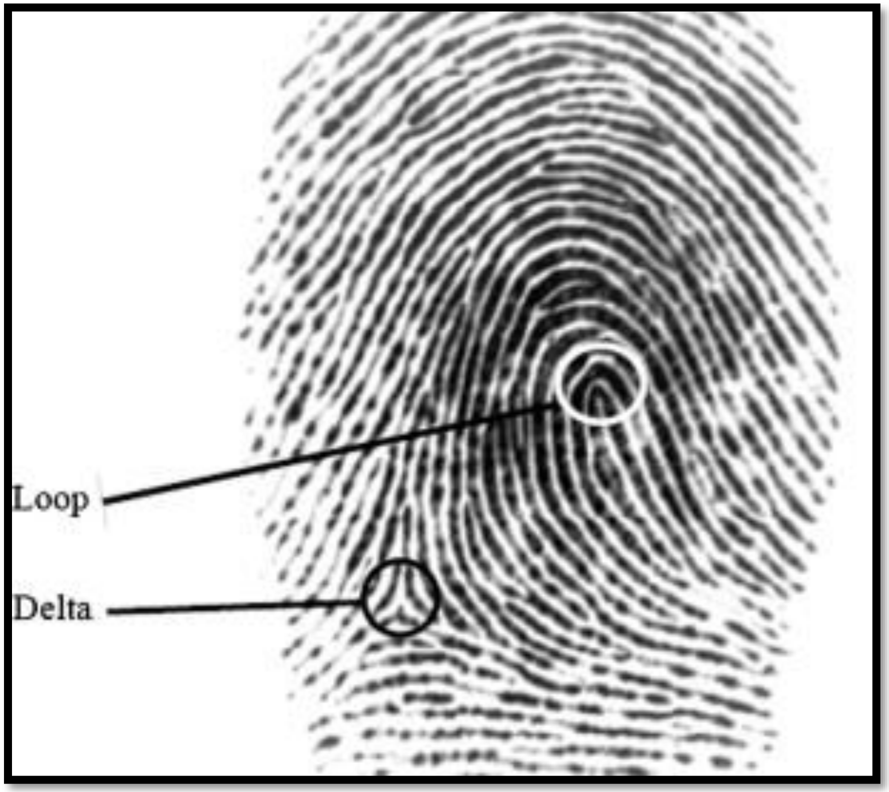
In-display fingerprint scanners have been hailed as the optimal solution to bezel-less screens. At least over face or iris recognition or even placing the sensor on the phone's back. But by putting it under the display's glass, the

2.2 FRICTION RIDGE PATTERN

- Fingerprint recognition is feature-based instead of image-based

Level 1 Features:

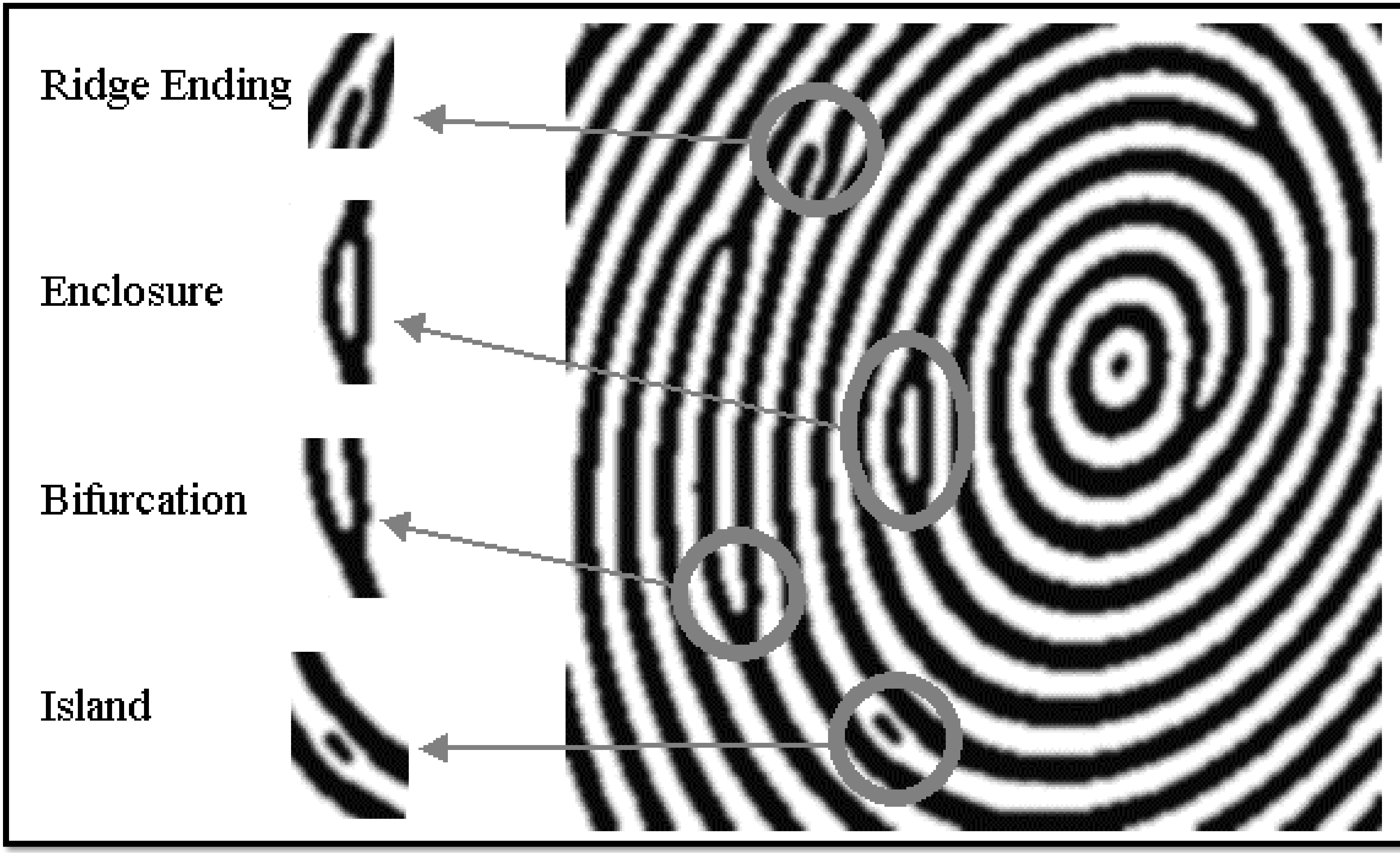
- Coarsest level: ridge orientation map
    - Local ridge orientation and frequency
    - Don't care about location
    - Ridge orientation at (x,y) = tangential direction of ridges at (x,y) defined in range $[0, \pi)$ → **Local direction**
    - There may be points where the ridge direction abruptly changes = singular points
        - Loops -> Ridges enter in one direction and exit in the same
        - Delta -> Three ridge systems meet
        - Different combinations of loops, deltas, and their spatial relationship yields **abstract representations of the ridge orientation map**



Level 2 Features:

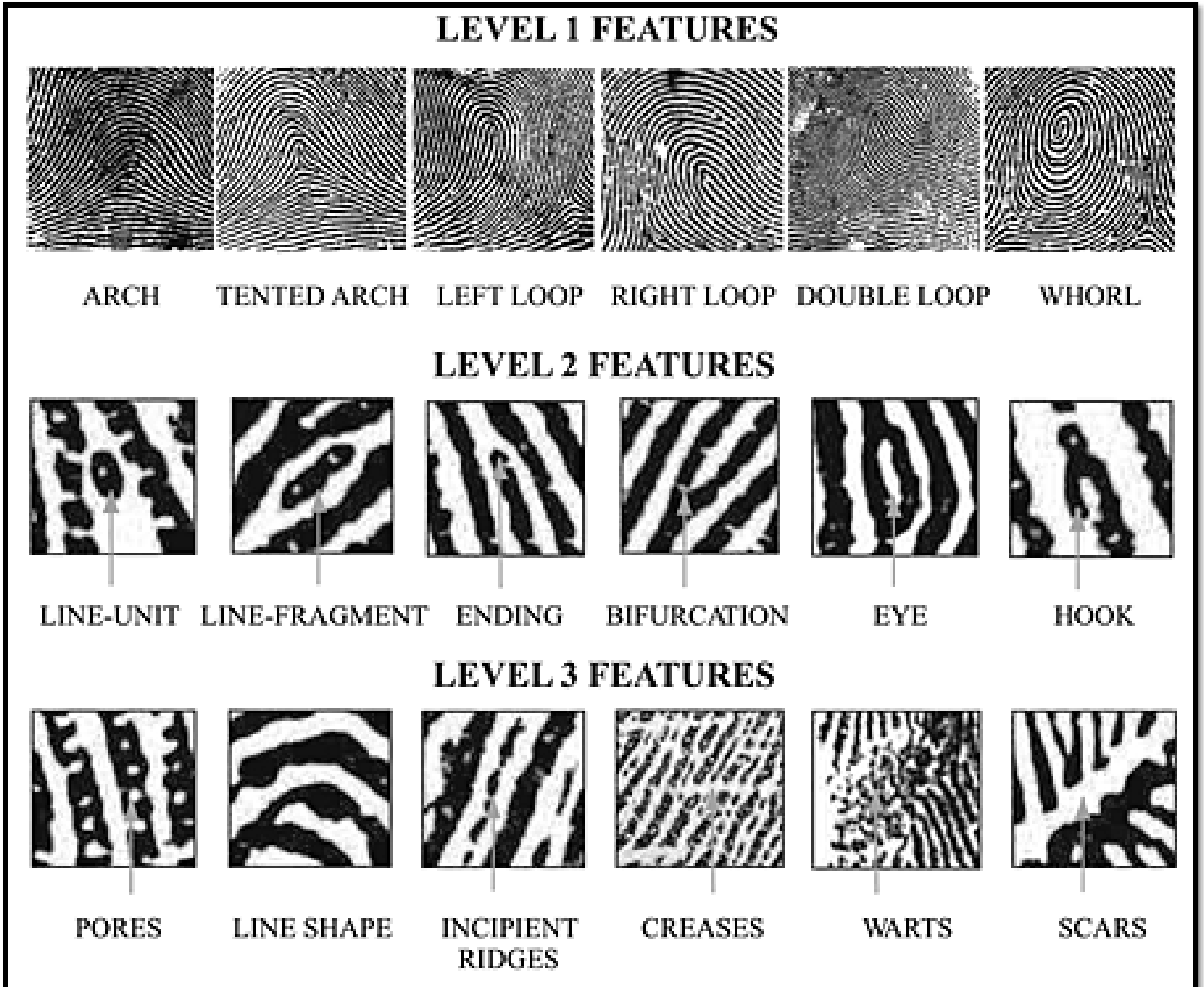
- Fingerprint represented as a ridge skeleton
    - Each ridge is **1 pixel wide and you know exact locations**
- Important: A minutiae point is where a ridge emerges, ends, splits, or merges!
- Two basic types: ending (termination) / bifurcation
- A set of minutiae yields **an abstract representation of the ridge skeleton**

How can we characterize a minutia point?



Level 3 Features:

- Fingerprint represented with inner holes (sweat pores) and outer contours (ridge edges)
- Incipient ridges – thinner with no sweat pores
- Dot – short ridges


LEVEL 1 FEATURES
ARCH
TENTED ARCH
LEFT LOOP
RIGHT LOOP
DOUBLE LOOP
WHORL
LEVEL 2 FEATURES
LINE-UNIT
LINE-FRAGMENT
ENDING
BIFURCATION
EYE
HOOK
LEVEL 3 FEATURES
PORES
LINE SHAPE
INCIPIENT RIDGES
CREASES
WARTS
SCARS

2.3 FINGERPRINT ACQUISITION

Offline Methods:

- Inked on paper to digital format
- Lifted latent prints left at a crime scene
- Doesn't work well for non-forensic applications
    - Imagine rolling your finger on paper to access your smartphone!

Online Methods:

- Directly retrieve digital format from fingerprint = **live-scan fingerprint**
    - Optical Frustrated Total Internal Reflection (FTIR)
    - Capacitance
    - Ultrasound Reflection
    - Piezoelectric Effect
    - Temperature Differential

Image Quality

- Factors affecting image quality:
    - Resolution
    - Finger area
    - Clarity of ridge pattern

LET ME IN People born today will be LAST generation to use computer passwords – here's why

> Say goodbye to typing in confusing and easy-to-forget passwords as the sci-fi future of biometric scanning becomes a reality

By Sean Keach, Digital Technology and Science Editor
27th February 2018, 2:26 pm | Updated: 27th February 2018, 2:47 pm

COMMENT NOW

PASSWORDS for your computer and phone could soon be a thing of the past.

An industry expert reckons that people born today will be the last generation of people to use computer passwords.

2.3 FEATURE EXTRACTION

Four main steps:

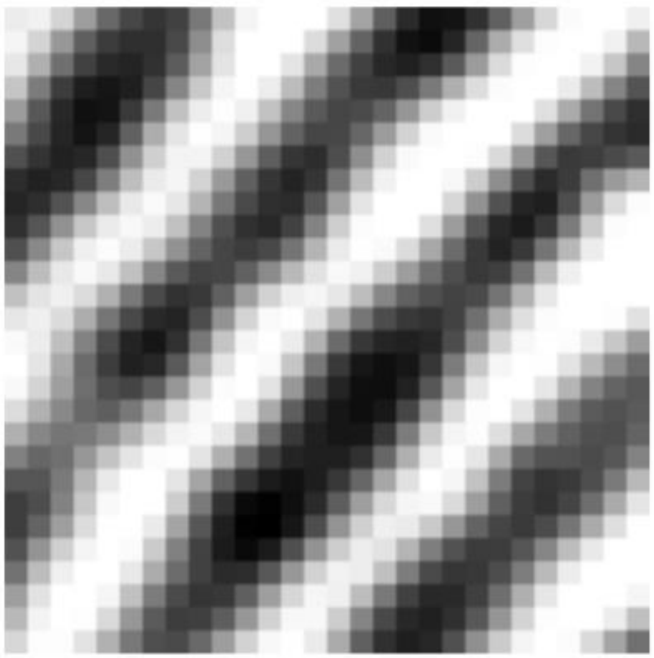
1. Ridge Orientation and Frequency Estimation
2. Ridge Extraction
3. Singularity Extraction
4. Minutiae Extraction

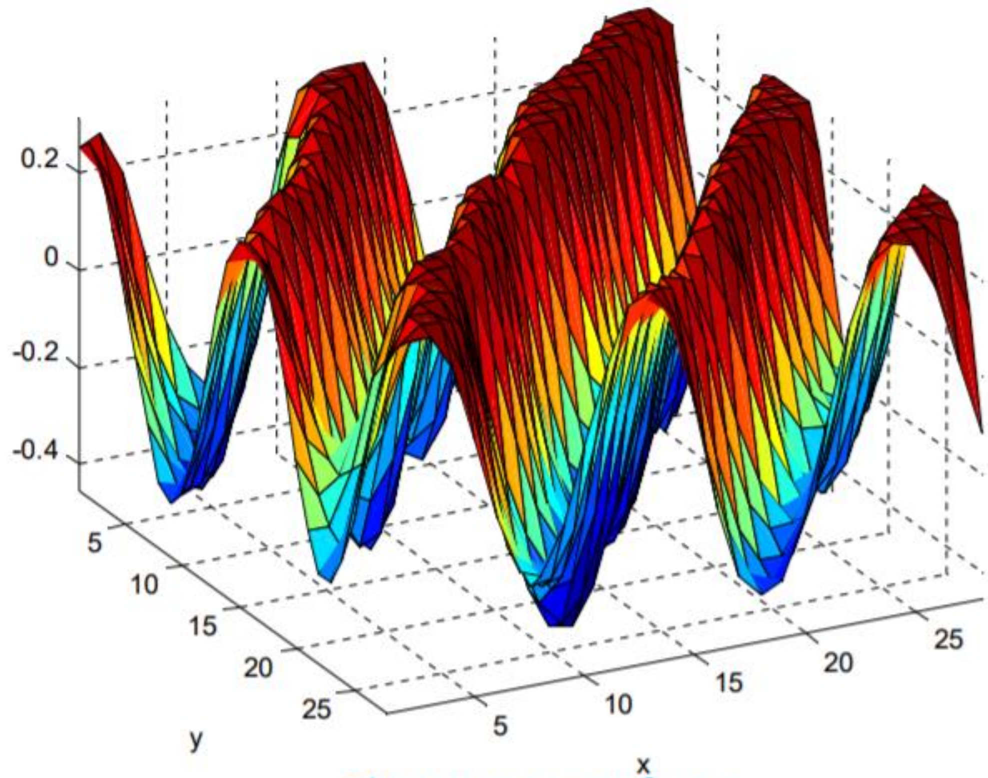
Ridge Orientation and Frequency Estimation:

Ridge pattern in a local area of a fingerprint can be approximated by a cosine wave

$$w(x, y) = A\cos\big(2\pi f\,(x\cos\theta + y\sin\theta)\big)$$

amplitude (A) frequency (f) orientation (θ)


Local fingerprint region



Shown as surface

- Estimate those parameters using Fourier Transform

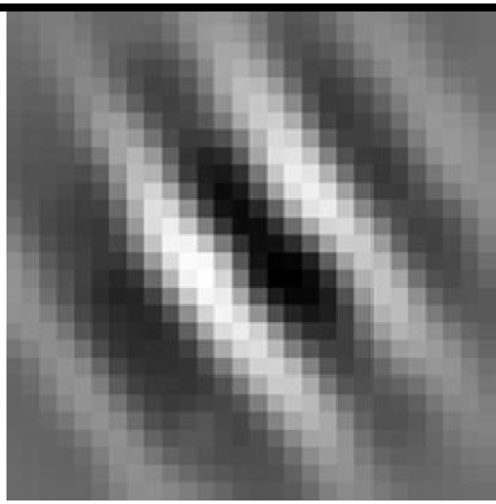
4.1 Feature Extraction: Ridge parameters estimation

- **Ridge pattern in a local area of a fingerprint can be approximated by a cosine wave**

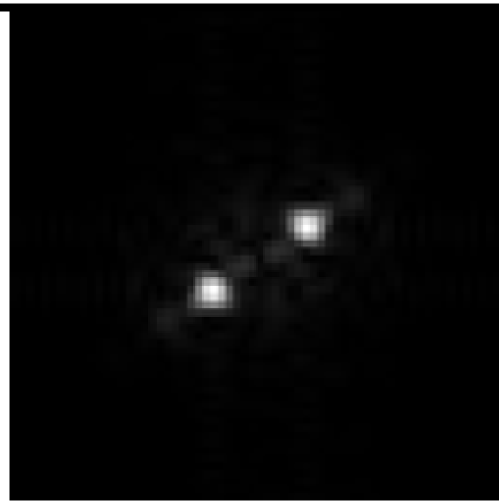
$$w(x,y) = A\cos(2\pi f_0(x\cos\theta + y\sin\theta))$$

- **2D Fourier transform of cosine wave**

$$W(u,v) = \frac{A}{2}[\delta(u - f_0\cos\theta, v - f_0\sin\theta) + \delta(u + f_0\cos\theta, v + f_0\sin\theta)]$$



local ridge pattern



magnitude spectrum

- **Let $(\hat{u}, \hat{v})$ denote the location of the maximum magnitude, then**

$$\hat{A} = |W(\hat{u}, \hat{v})|,$$

$$\hat{\theta} = \arctan\left(\frac{\hat{u}}{\hat{v}}\right),$$

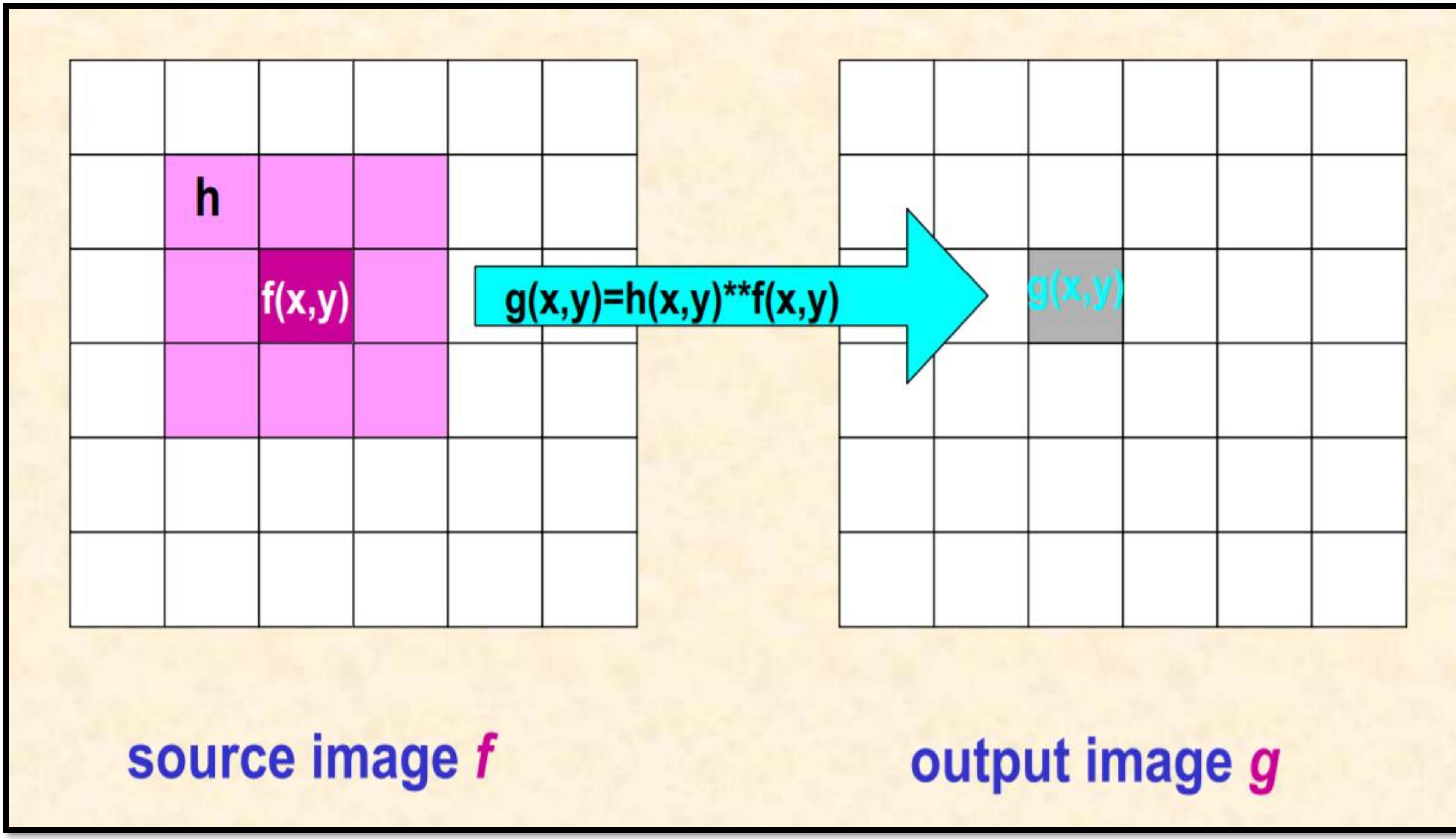
$$\hat{f}_0 = \sqrt{\hat{u}^2 + \hat{v}^2}.$$

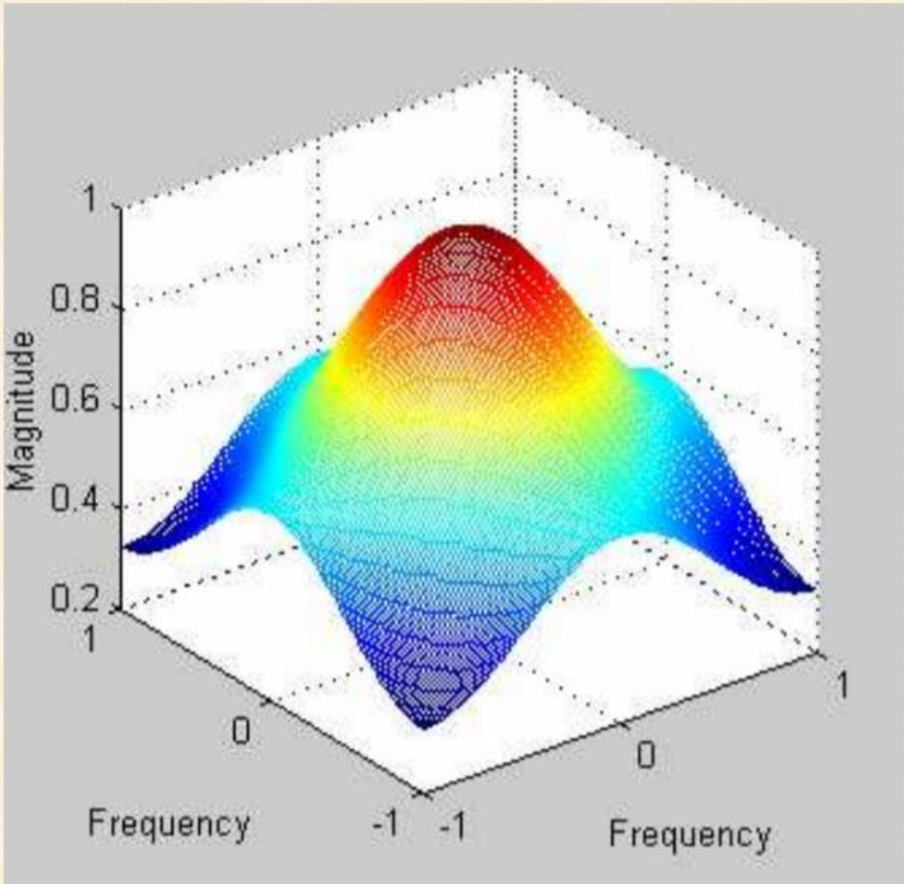
- Smoothing the magnitude spectrum with a **low pass filter** is suggested since the ridge is not exactly approximated by a cosine wave
    - Presence of noise creates erroneous local regions

Will simple averaging of local orientations work?

4.1 Feature Extraction: Ridge parameters estimation

- To deal with noise, we should **smooth** the orientation field.
- Special consideration on ridge orientation:
  - in the range $[0, \pi)$.
  - θ and $(\theta + \pi)$ is the **same** orientation.
  - the average value between 1° and 179° should be **0°** **rather than 90°** !


h
f(x,y)
g(x,y)=h(x,y)**f(x,y)
g(x,y)
source image f
output image g



$$h = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\boxed{h(x,y) = e^{\frac{-\pi(x^2+y^2)}{{d_0}^2}}}$$

$$\boxed{H(u,v) = e^{\frac{-\pi {d_0}^2 (u^2+v^2)}{N}}}$$

- **3 steps to smooth orientation field:**
  - **Construct a vector field $V = (V_x, V_y) = (\cos 2\theta, \sin 2\theta)$;**
  - **Perform low pass filtering on the two components of the vector field separately to obtain the smoothened vector field $V' = ({V_x}', {V_y}')$;**
  - **The smoothened orientation field is given by $\frac{1}{2}\arctan(\frac{{V_x}'}{{V_y}'})$.**

Fingerprint image **Initial (noisy) orientation field** **Smoothed orientation field**

RECALL: Level 1 Features:

- Coarsest level: ridge orientation map
  - Local ridge orientation and frequency
  - Don't care about location
  - Ridge orientation at (x,y) = tangential direction of ridges at (x,y) defined in range [0, π) -> Local direction
  - There may be points where the ridge direction abruptly changes = singular points
    - Loops -> Ridges enter in one direction and exit in the same
    - Delta -> Three ridge systems meet

SINGULARITY EXTRACTION (FINDING THOSE LOOPS/DELTAS NOW THAT WE HAVE THE RIDGE ORIENTATION MAP THROUGH THE USE OF THE FOURIER TRANSFORM)

- Singularities are found through the Poincare index
  - Cumulative change of orientations
  - Use the 8 neighbors of a pixel

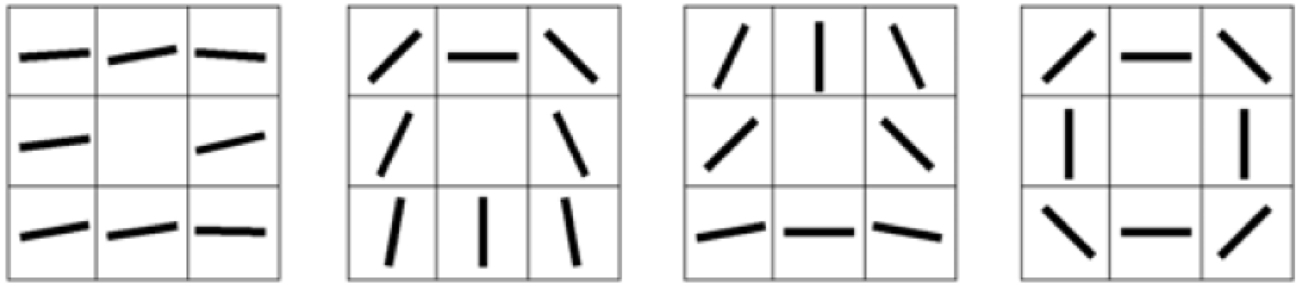
- **Poincaré index method:**
  - **Let $O[i] \epsilon [0, \pi), i = 0, \cdots, 7$,denote the orientations at eight neighbors of point i.**
  - **Poincaré index (PI) at this point is:**

$$PI = \frac{1}{\pi} \sum_{i=0}^{7} \delta(O[(i+1)_{\bmod 8}] - O[i]),$$

where

$$\delta(\theta) = \begin{cases} \theta - \pi, & \text{if } \theta > \pi/2 \\ \theta, \text{if } -\pi/2 \le \theta \le \pi/2 \\ \theta + \pi, & \text{if } \theta < -\pi/2 \end{cases}$$

- **Poincaré index in 4 cases:**
  - 0 (non-singular)
  - 1 (loop)
  - -1 (delta)
  - 2 (whorl)



5	10	-5
7		13
10	8	-1

Non-singularity

45	0	-45
65		-65
80	90	-80

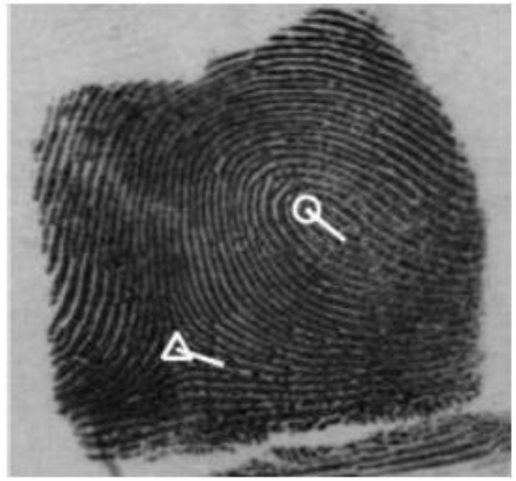
Loop

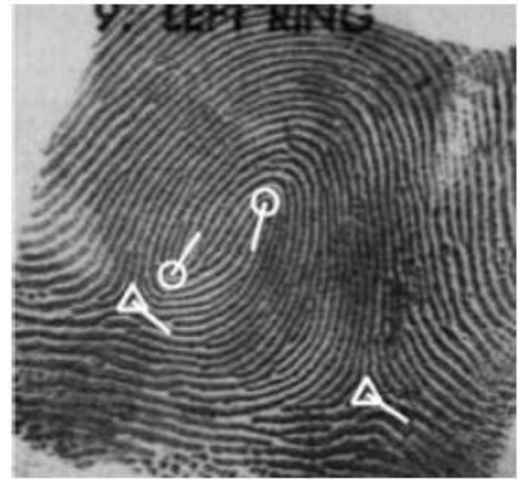
65	90	-65
45		-45
10	0	-10

Delta

45	0	-45
90		90
-45	0	45

Whorl

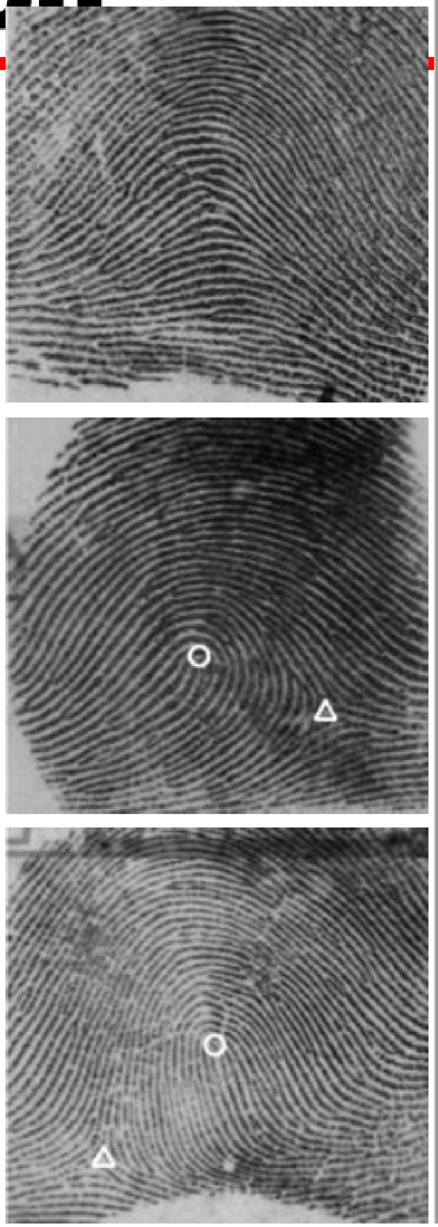




Singularity extraction

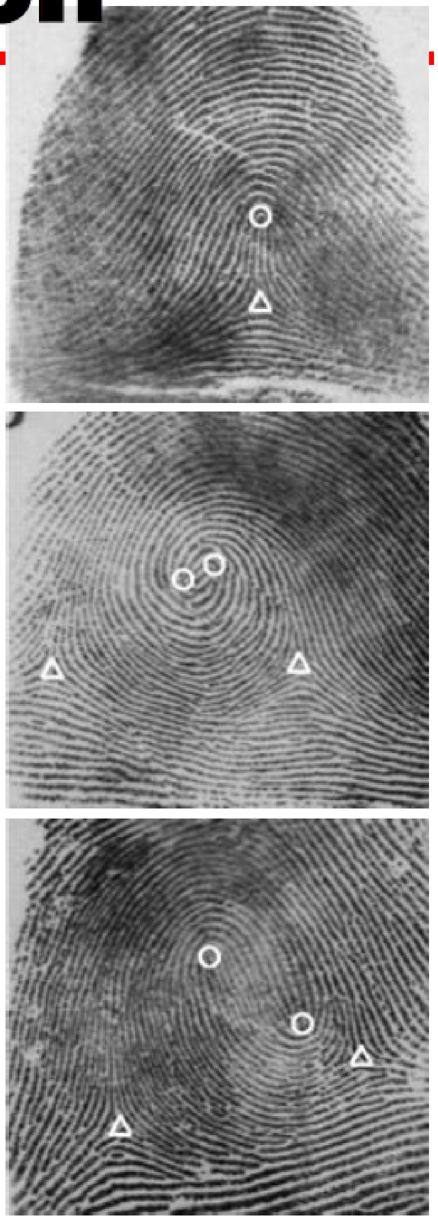
We can classify a fingerprint into one of 6 major pattern types based on singular points (SP):

- plain arch: contains no SP.
- left loop: contains 1 delta and 1 loop whose direction points to the left side of the delta.
- right loop: contains 1 delta and 1 loop whose direction points to the right side of the delta.



4.2 Feature Extraction: Singularity extraction

- **tented arch: contains 1 delta and 1 loop whose direction points toward the delta.**
- **whorl: contains at least 2 loops and 2 deltas where ridge orientation field around the two loops form a circular orbit.**
- **twin loop: contains at least 2 loops and 2 deltas where the ridge orientation field around the two loops does not form a circular orbit.**



Recap:

We've examined level 1 features → ridge orientation map!

We've figured out how to approximate the ridge using the cosine wave through the Fourier transform which encodes

magnitude, orientation, and frequency for us!

We then were able to look at neighboring pixels to find singularities such as loops or deltas. These singularities will be useful later for aligning gallery and probe fingerprints when we want to

perform matching. But first, we must learn how to extract minutiae!

RIDGE EXTRACTION

- Minutiae are those special areas on the ridges, so we want to extract the ridges

Shouldn't we be able to apply a threshold; any pixel darker than *t* is a ridge?

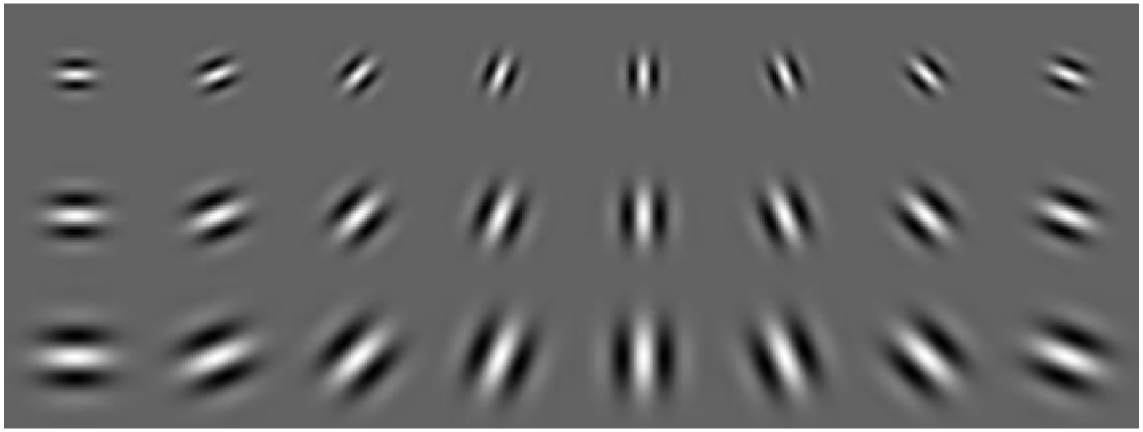
NO!

- Pores in ridges alter darkness
- Cuts/creases break ridges
- Moisture makes adjacent ridges appear joined
- We should instead enhance the image with a **contextual** filter
    - Use real part of 2D Gabor filter
    - Tune orientation and frequency to local region
- Now, using this enhanced image, we can **thin** the ridge to one pixel thick

2D Gabor wavelet:

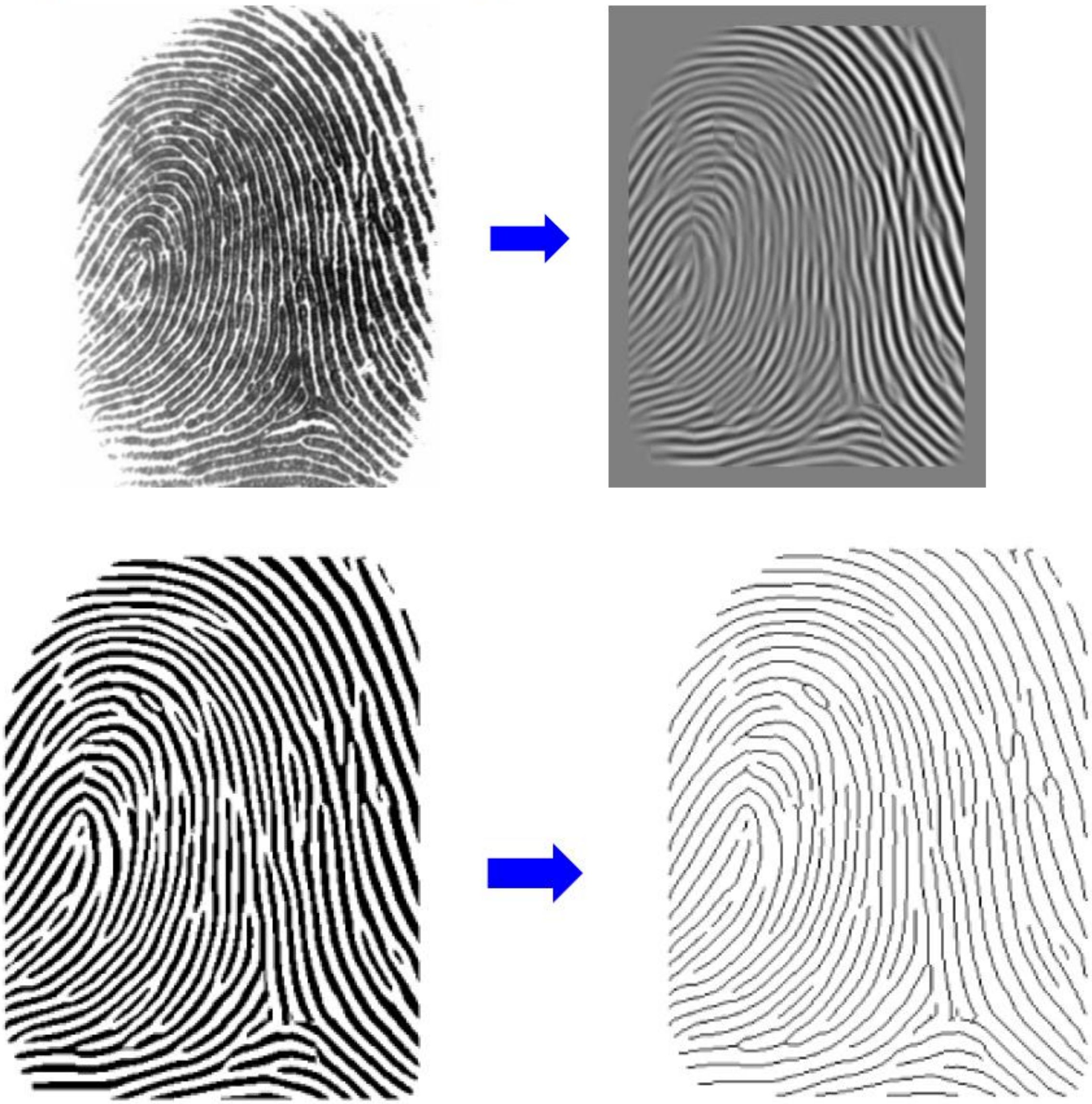
$$G(x,y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2]} e^{-2\pi i[u_0(x-x_0)+v_0(y-y_0)]}$$

where (x_0, y_0) denote the position in the image, (α, β) denote the effective width and length, and (u_0, v_0) denote the wave direction with a spatial frequency $\omega_0 = \sqrt{u_0{}^2 + v_0{}^2}$.



Real parts of Gabor filters (8 orientations and 3 scales)

- **An example of ridge enhancement by Gabor filtering.**

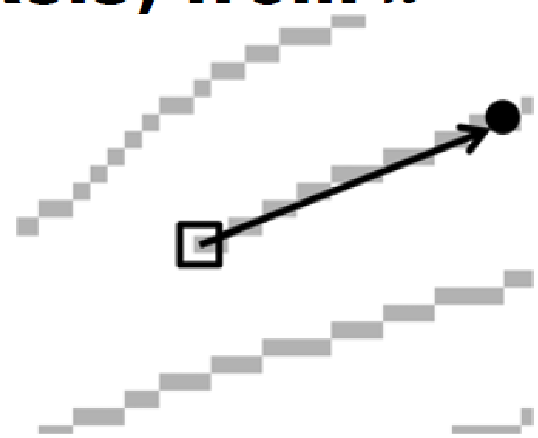


MINUTIAE EXTRACTION

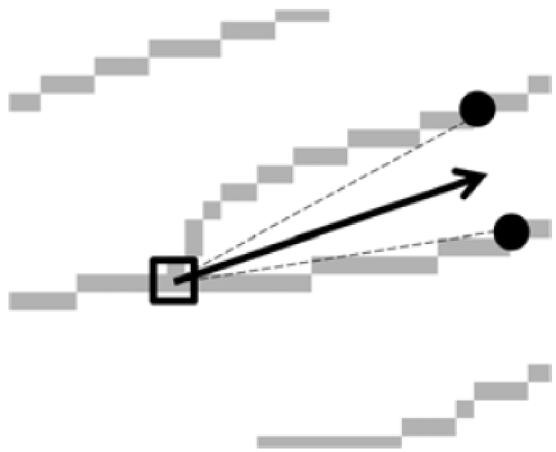
- 3 ridge neighbors → bifurcation
- 1 ridge neighbor → ending

What about direction?

- **Direction of a ridge ending:**
  - **Trace the associated ridge with a fixed distance (say 10 pixels) from x to a.**
  - **The direction xa is the minutia direction.**

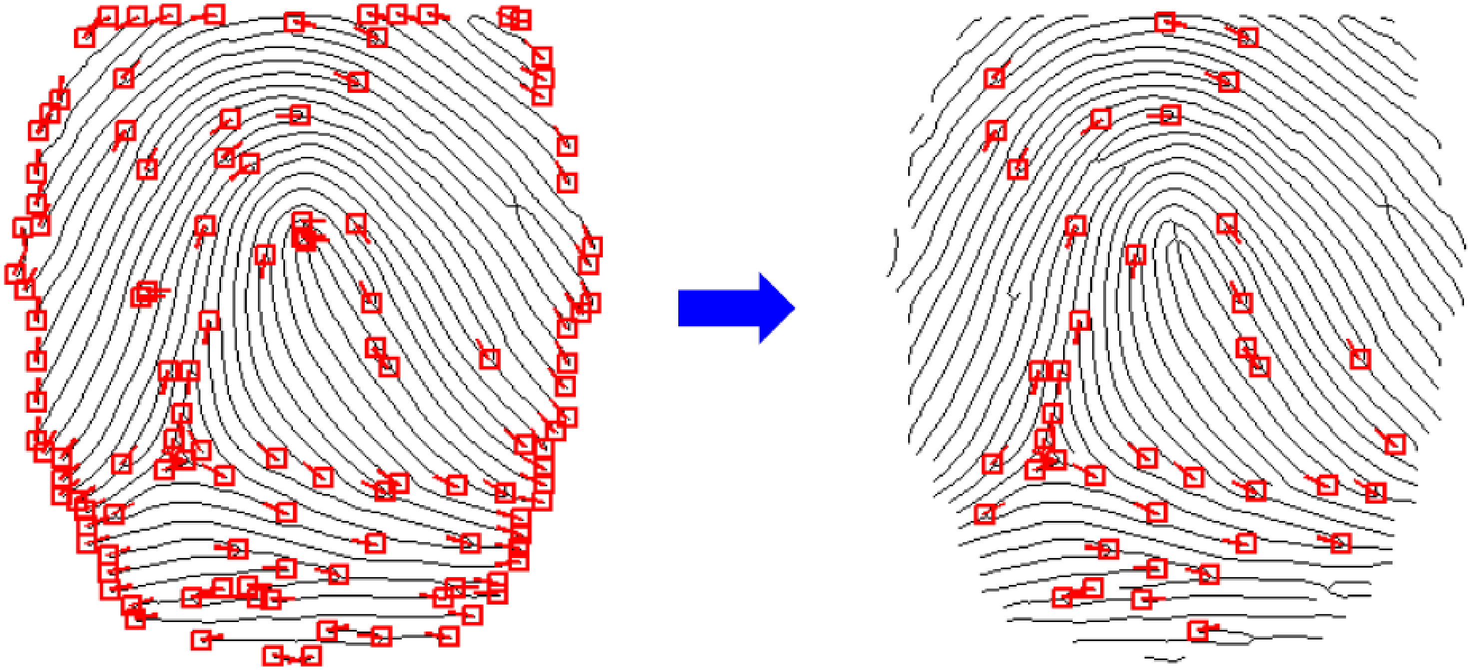


- **Direction of a bifurcation:**
  - **Trace the ridges to get three directions.**
  - **The direction is the mean of the two smallest different directions.**



- **Some spurious minutiae:**
  - **artifacts in image processing;**
  - **noise in a fingerprint.**
- **They satisfy any of the following conditions:**
  - **have no adjacent ridge on either side;**
  - **be close in location and almost opposite in direction;**
  - **too many minutiae in a small neighborhood.**
- **They should be discarded.**

- **An example of removing spurious minutiae.**



Recap:

We extracted level 1 features (ridge orientation map). We can use this map to find singularities, and then use singularities as an abstraction of the ridge orientation map. But if we want finer features, we can move to

level 2 and find fingerprint minutiae. For this, we need thin and cleanly defined ridges. We accomplished this through contextual filtering. Then we simply count neighboring pixels for obtaining level 2 features.

Matching Minutiae

- We need to determine the identity of the individual given a query set of minutiae by matching against a stored, or template, set

Three Steps

- **Matching algorithm:**
  - **Alignment**: Determine the geometric transformation between the two minutiae sets so that they can be brought into the same coordinate system.
  - **Correspondence**: Form pairs of corresponding minutiae.
  - **Score generation**: Compute the match score based on the corresponding minutiae points.

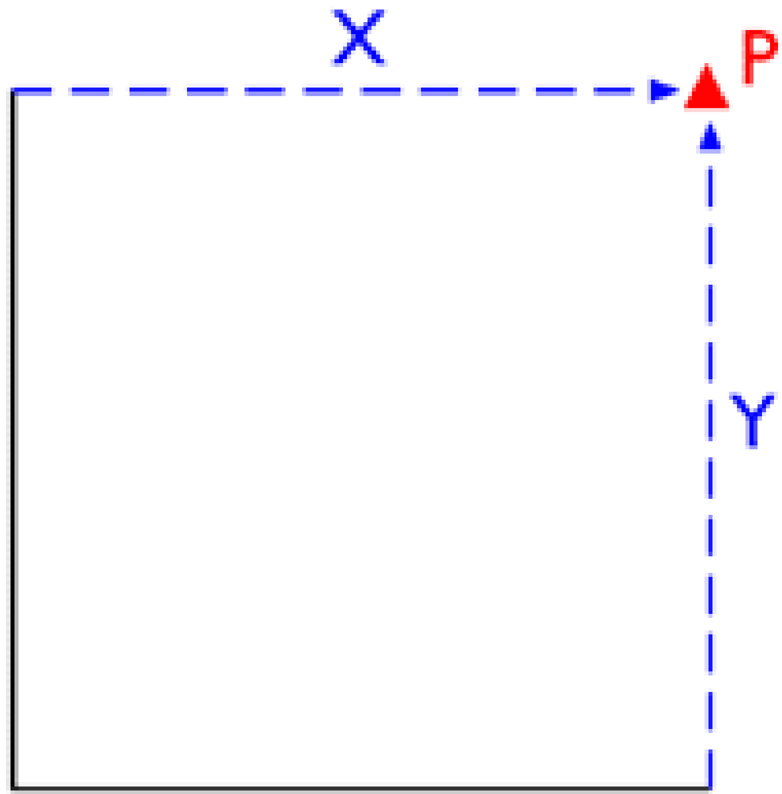
Why Alignment?

- Two impressions of the same finger could differ due to variations in placement
- Also referred to as **registration – geometric alignment of two images**

OPTION 1: Use a **rigid transformation such as generalized Hough transform** for estimating a **spatial transformation**

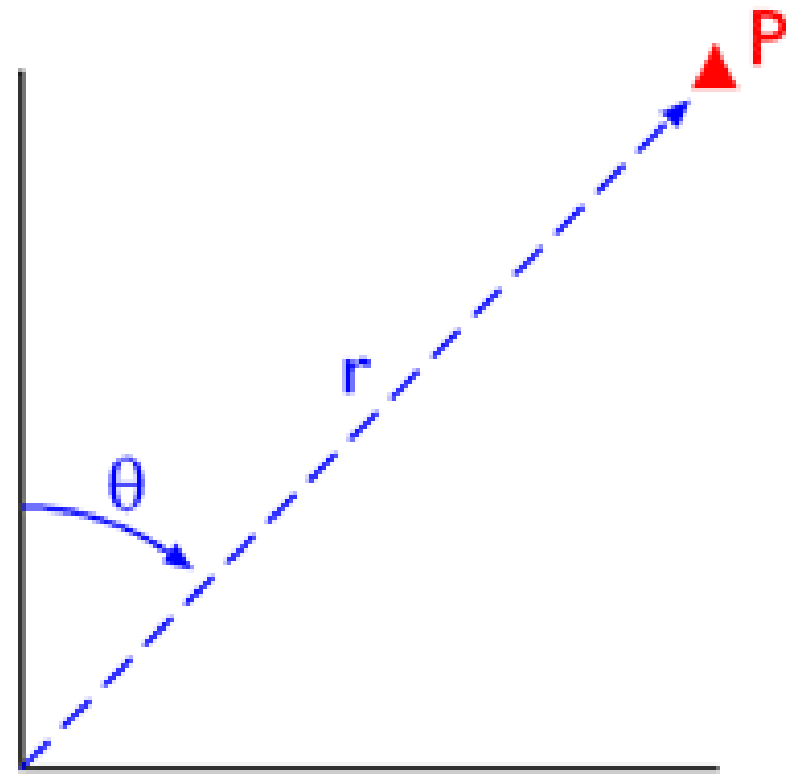
input : Two minutiae sets $\{x_i^T, y_i^T, \theta_i^T\}_{i=1}^{M}$ and $\{x_j^Q, y_j^Q, \theta_j^Q\}_{j=1}^{N}$
output: Transformation parameters
Initialize accumulator array A to 0
for $i = 1, 2, \cdots, M$ **do**
    for $j = 1, 2, \cdots, N$ **do**
        $\Delta\theta = \theta_i^T - \theta_j^Q$
        $\Delta x = x_i^T - x_j^Q cos(\Delta\theta) - y_j^Q sin(\Delta\theta)$
        $\Delta y = y_i^T + x_j^Q sin(\Delta\theta) - y_j^Q cos(\Delta\theta)$
        $A[\Delta\theta][\Delta x][\Delta y] = A[\Delta\theta][\Delta x][\Delta y] + 1$
    end
end
return location of peak in A

Cartesian coordinates



$P = (X, Y)$

Polar coordinates

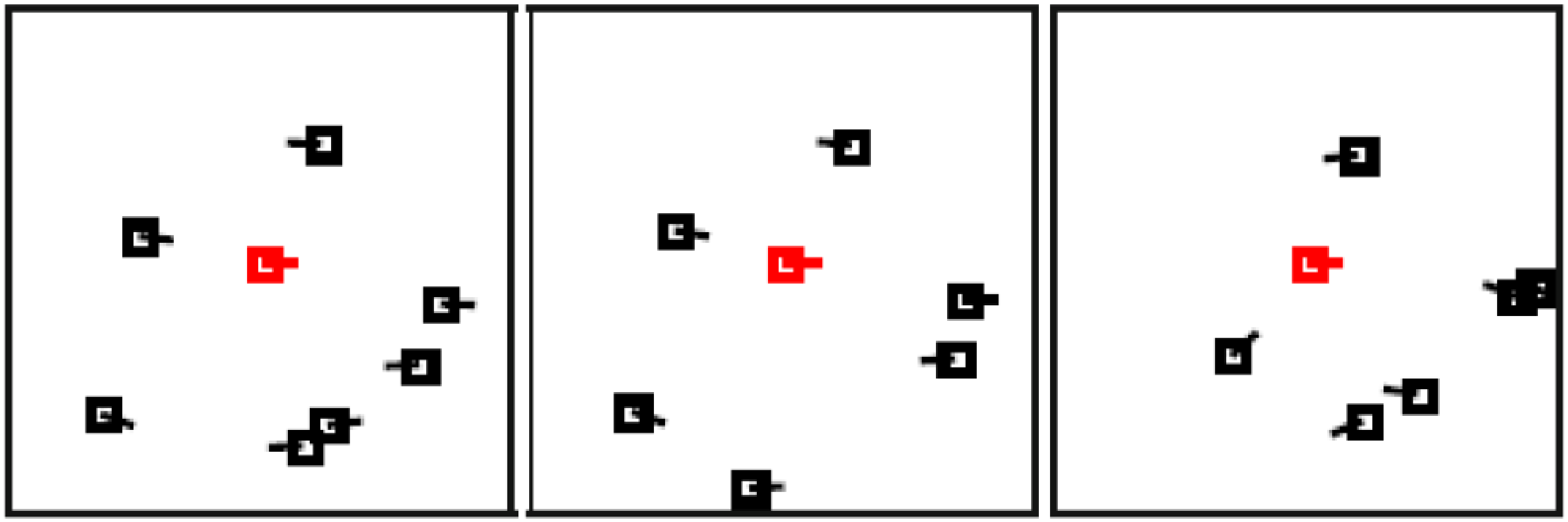


$P = (\theta, r)$

OPTION 2: First find a pair of matched minutiae and then compute the rotation and translation parameters.

Can we accomplish this knowing what we know about minutiae points, i.e., location, direction, and type?

- A popular descriptor is based on the relationship between the neighboring minutiae and the center minutia



descriptor of a minutia

descriptor of the mated minutia (not exactly the same due to noise)

descriptor of a non-mated minutia

Pairing Remaining Minutiae

- **When two minutiae sets are aligned, the corresponding minutiae are paired.**
- **A minutia a in T is correspondence with minutia b in Q if and only if**
  - **their distance is within a predefined distance threshold;**
  - **the angle between their directions is within another predefined angle threshold.**

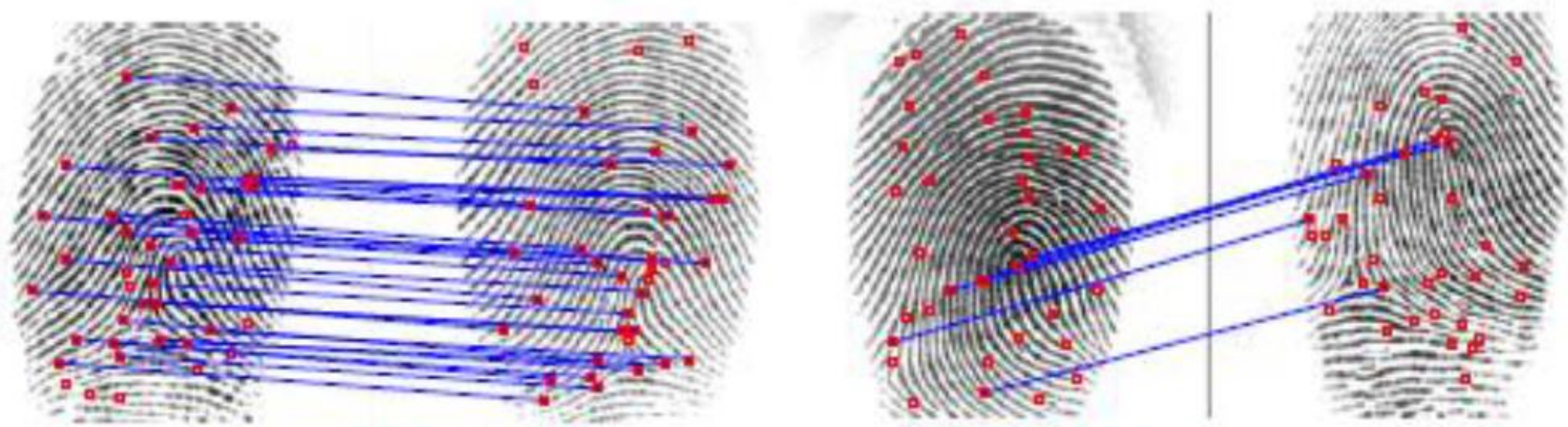
Pseudocode is in the slides!

Matching between two feature sets yields what?

Match Score Generation:

- According the match score, two fingerprints can be classified into
  - a genuine match (they come from the same finger);
  - an impostor match (they come from two different fingers).
- Distinguishing features:
  - the number of paired minutiae;
  - the percentage of matched minutiae in the overlapped area.

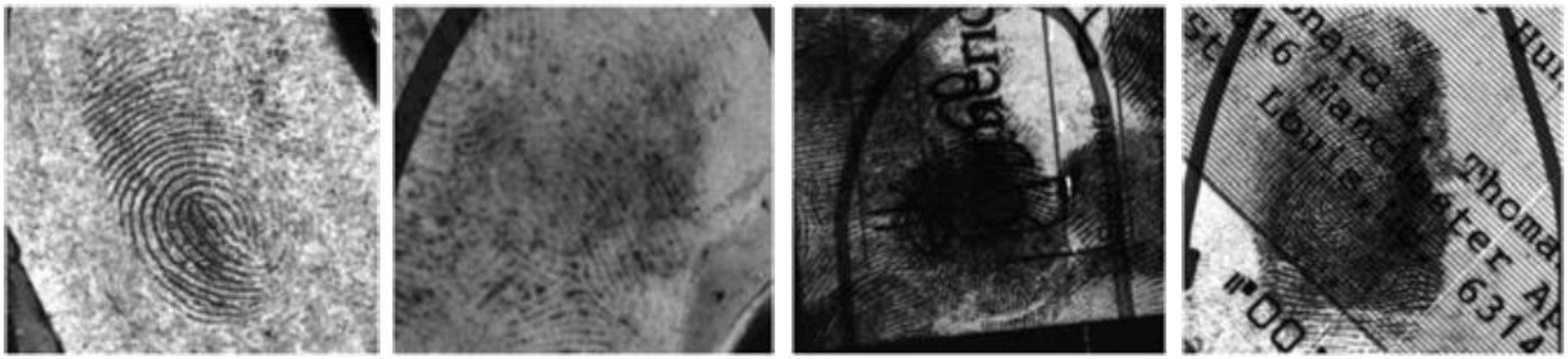
- Genuine matches should have **more** matched minutiae
- **Percentage** of matched minutiae in the overlapped area should be larger for genuine matches



(a) Match score = 614 (b) Match score = 7

What are your thoughts on finding and matching minutiae in latent fingerprints?

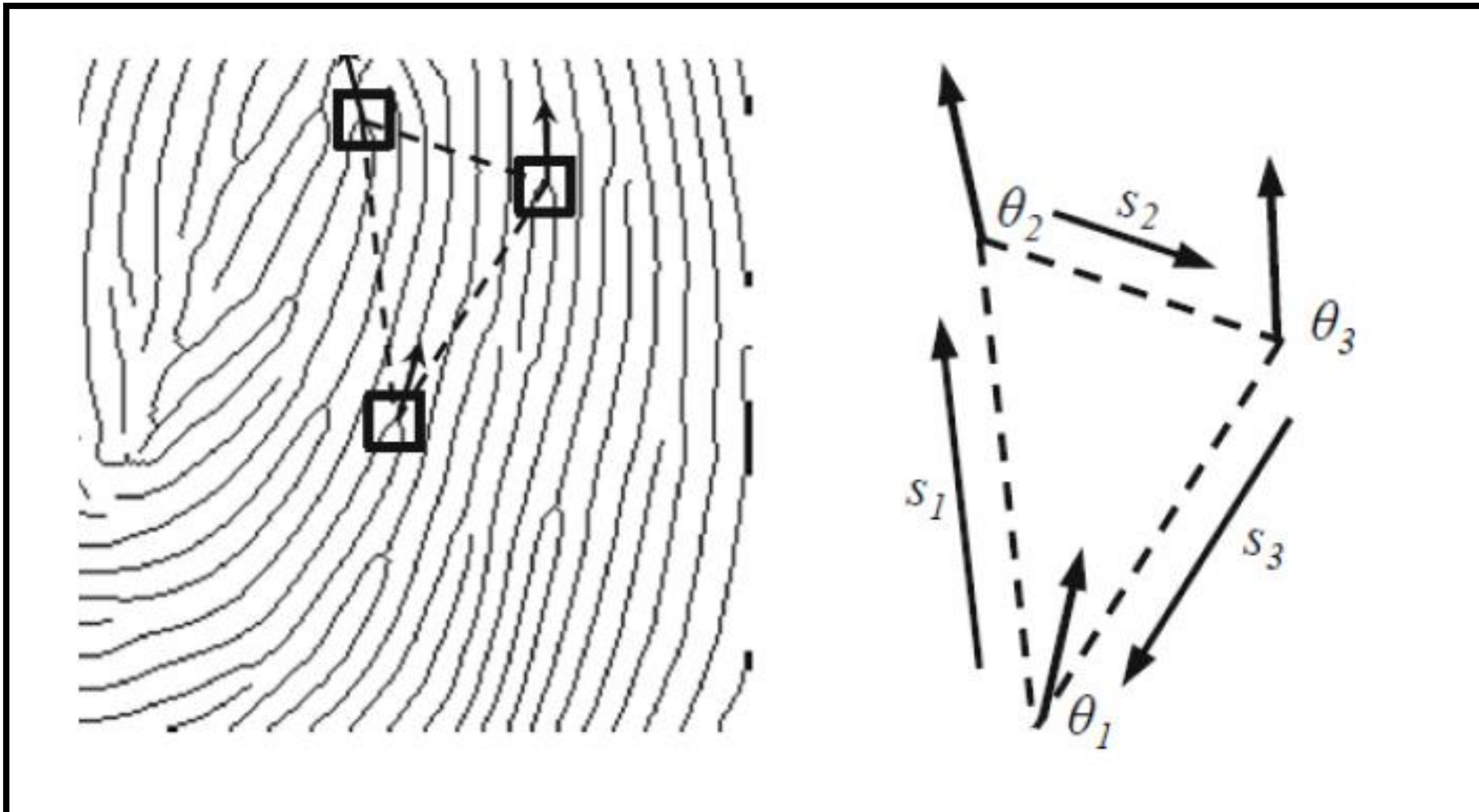
- **Latent fingerprint matching is much more challenging than rolled or plain fingerprint matching due to**
  - **poor image quality;**
  - **small finger area;**
  - **large nonlinear distortion.**
- **A main challenge is how to reliably encode and match Level 3 features in poor quality latents.**



- Minutiae matchers have limited accuracy.
- Using extended features (orientation field, quality map, ridge skeleton, pores, etc.) to improve accuracy.

Fingerprint Indexing:

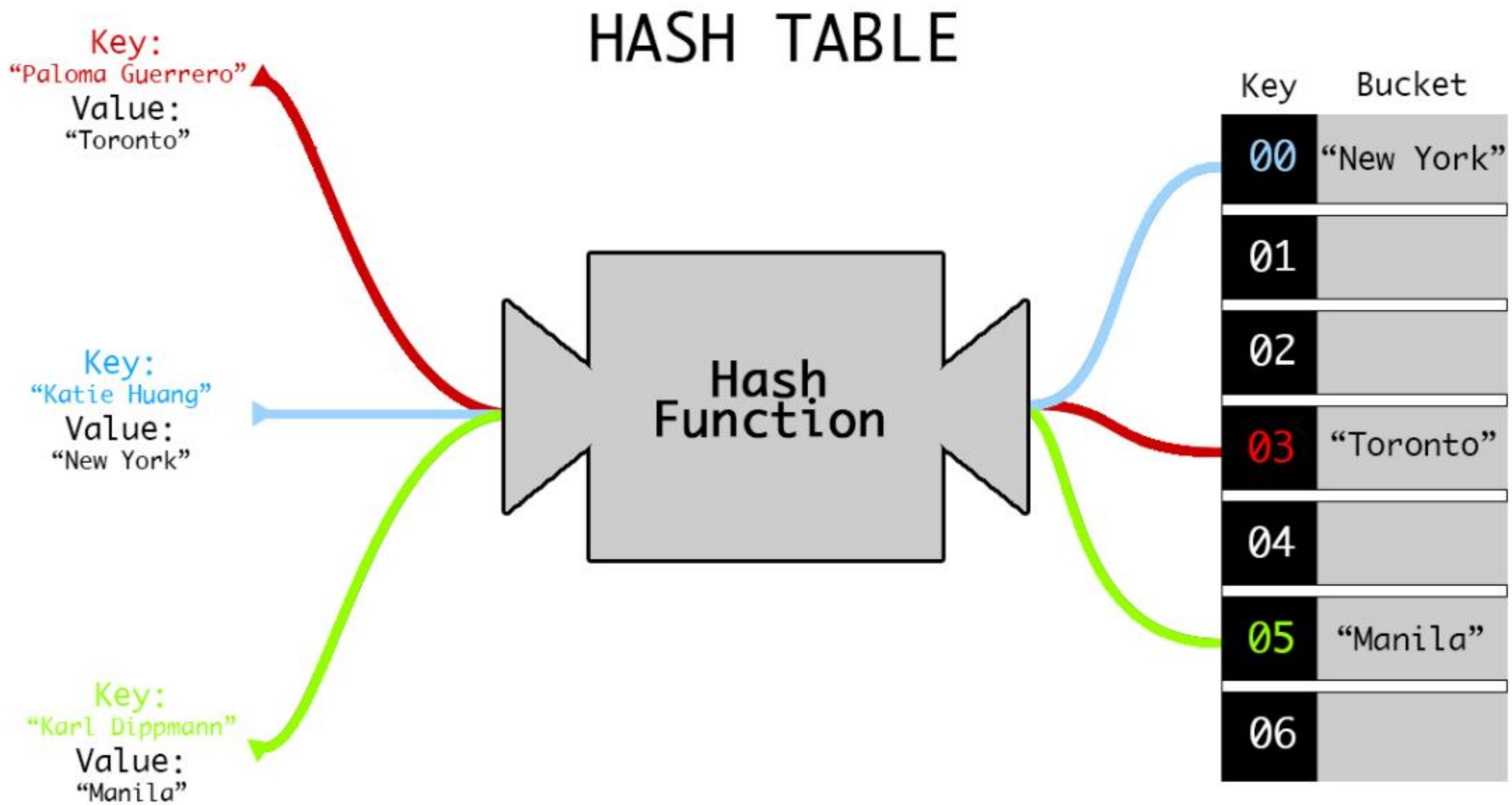
- Indexing – Assign a numerical value to a fingerprint image based on its features
- Used to speed up identification by comparing images of comparable index values
- Useful for partial fingerprints as singular points are not needed
- **Popular technique for indexing: triplets of minutiae**
    - Minutiae triplet described by a 9D feature vector consisting of geometric properties
        - Lengths of the sides
        - Ridge count between pairs of vertices
        - Orientation at vertices w.r.t. length of longest side


θ_2
s_2
θ_3
s_1
s_3
θ_1

- **These features are invariant to rotation and translation.**
- **But, the lengths of the sides are highly sensitive to non-rigid distortions.**
- **The correct ordering of the features and the three angles depend on correctly identifying the longest side of the triangle.**

Geometric Hashing

1. All triangles from a fingerprint are placed in a hash table
   a. Feature vector defines the key
   b. ID defines the data value



2. During retrieval, each key of the **input** is used to extract all IDs stored under the same key
3. The number of times an ID is retrieved is used as a **similarity score** between the ID and the **input**

Can improve this by using a better set of features

- Length of longest side
- Two smaller internal angles
- Handedness
- Type
- Direction
- Additional geometric constraints

Fingerprint Synthesis – Generating artificial, but realistic, fingerprint images

Why would we want to do this?

Why would we want to do this?

- Testing algorithms since collecting many fingerprint samples is expensive in time and money
- Modeling and identification of an appropriate set of characterizing parameters
- Level 1 and Level 2 Feature Synthesis
    - What do you think we’re generating in these levels?

Level 1 Feature Synthesis: Ridge Orientation Map

- Use a fixed ridge frequency – 0.1 ridges per pixel
- Desire a parametric model for the orientation field
    - Why?

Level 1 Feature Synthesis: Ridge Orientation Map

- Use a fixed ridge frequency – 0.1 ridges per pixel
- Desire a parametric model for the orientation field
  - Can adjust the parameters to simulate various directions
  - Example: **Zero-Pole Model**
    - Location and type of singular points
      - **zero-pole model:**

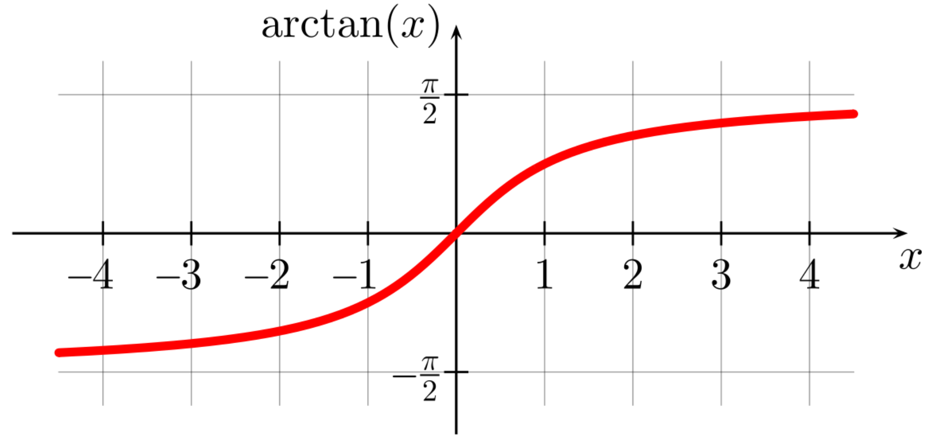
$$RO(x,y) = \frac{1}{2}\sum_{i=1}^{M} t_i \arctan\left(\frac{y-y_i}{x-x_i}\right)$$

- M: **the number of singular points;**
- (x_i, y_i) **: the coordinates of the** i**th singularity;**
- $t_i \in \{1, -1\}$**: the type (1 for loop and −1 for delta).**

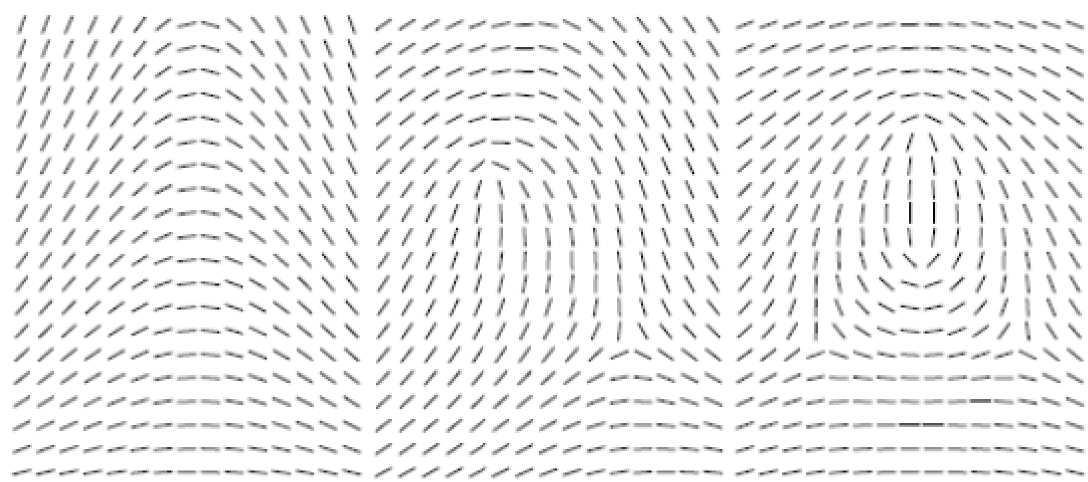
- **Zero-pole model cannot correctly model the arch-type fingerprints.**
- **An orientation field model for arch:**

$$RO_{arch}(x,y) = \arctan(\max\{0, (k - 3\frac{y}{H})\} \cdot \cos(\frac{x}{W}\pi))$$

  - H **and** W **denote the height and width of the image;**
  - k $(2 < k < 5)$ **controls the curvature of arch.**

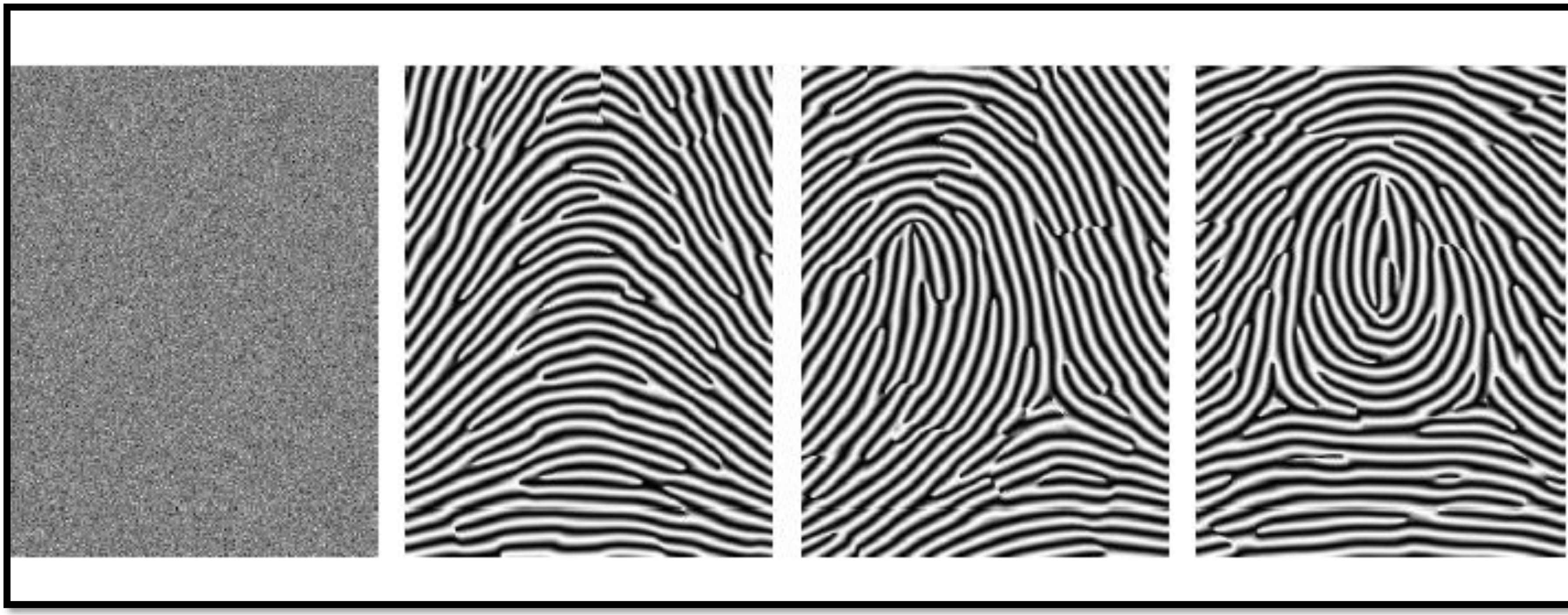


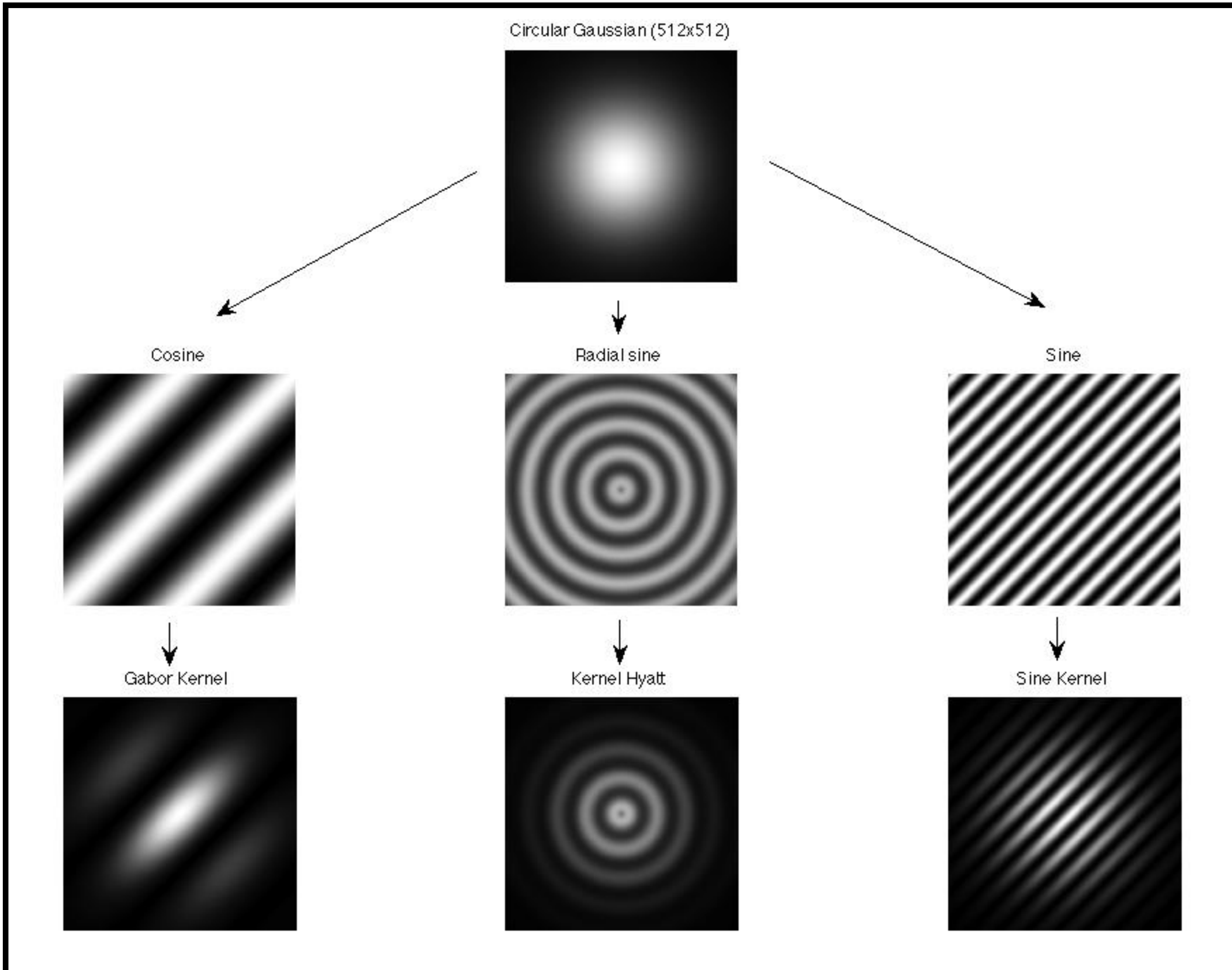
Examples of simulated fingerprint orientation fields (arch, left loop, and whorl).



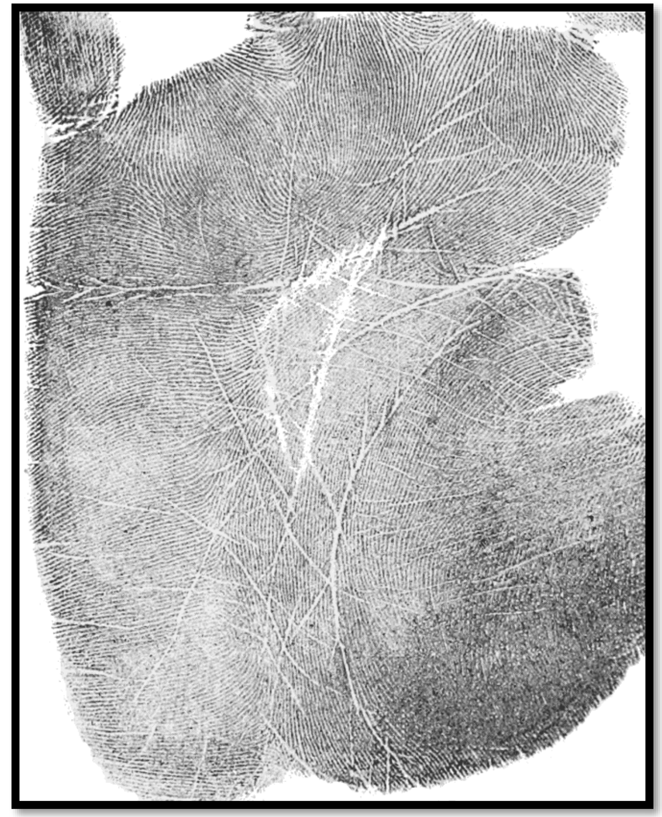
Level 2 Features: Ridge Pattern

- Use Gabor filtering on an image of random noise
  - Uniform distribution of pixels from 0-255
- Parameters are orientation field and ridge frequency



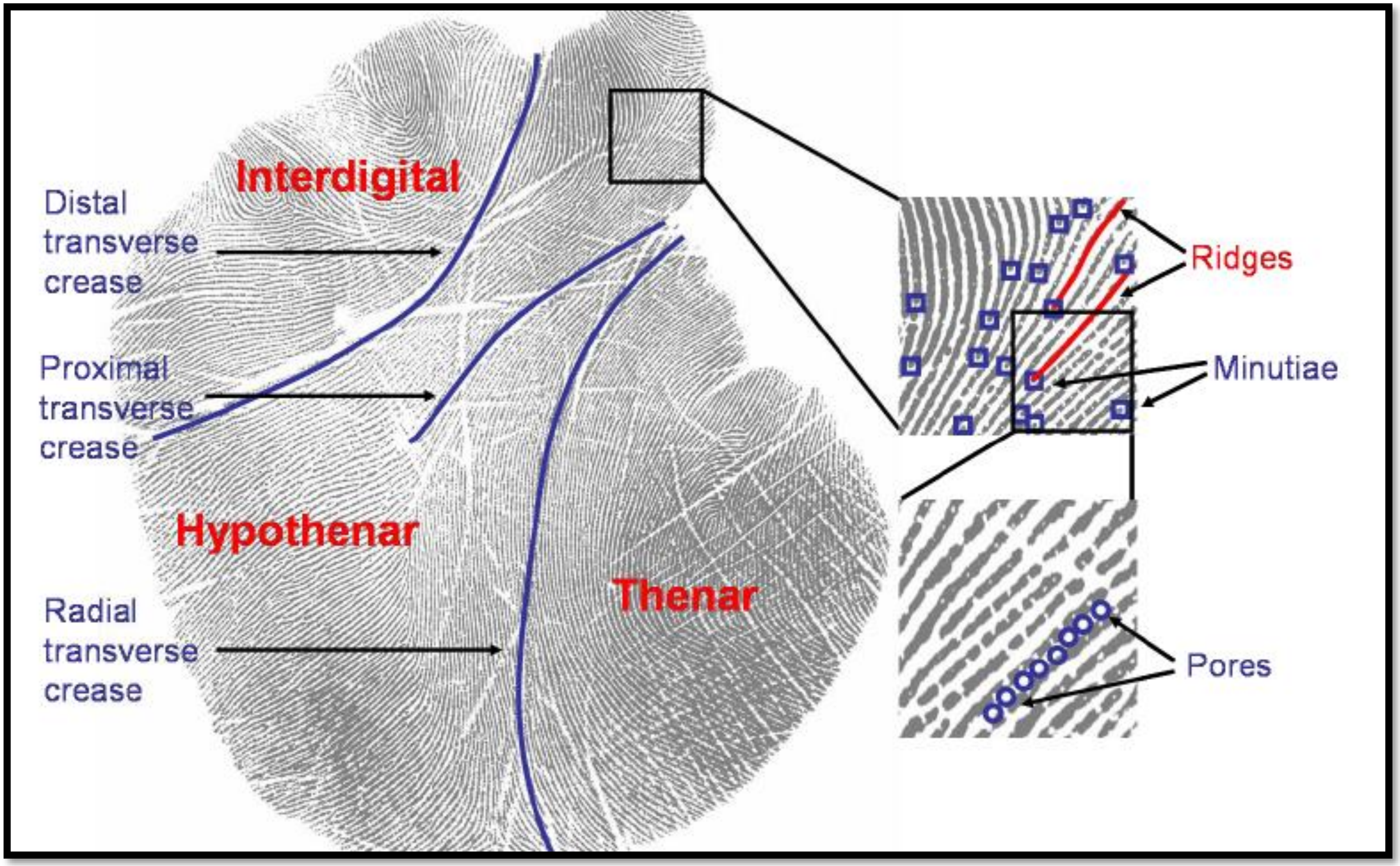

Circular Gaussian (512x512)
Cosine
Radial sine
Sine
Gabor Kernel
Kernel Hyatt
Sine Kernel

- **Different from real fingerprint images:**
  - **no sweat pores;**
  - **the ridge contours are too straight.**
- **It is also necessary to simulate various intra-class variations:**
  - **ridge thickness and image contrast;**
  - **finger placement on the sensor;**
  - **skin distortion.**

Palmprint

- Palm and sole prints are also unique
- Fewer applications….why?
- Main benefit is in latent palmprint matching
    - 30% of latents found at crime scenes are of palms
- Harder to capture compared to fingerprints
    - Larger sensors
    - Expensive
    - User cooperation
- 800 minutiae in palmprint vs. 80 minutiae in fingerprint
- **Two unique features: Palmar Friction Ridges and Palmar Flexion Creases**



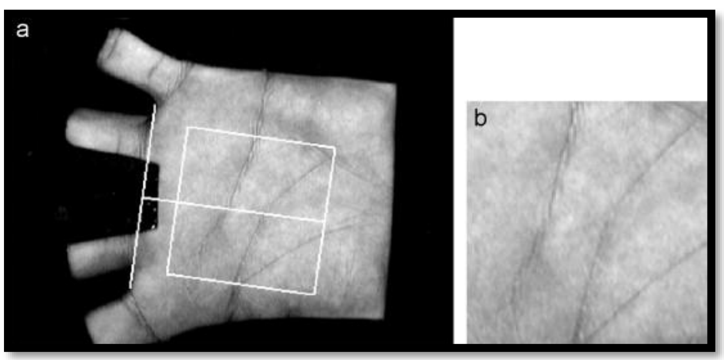
Main Difference Between Palm and Fingerprint

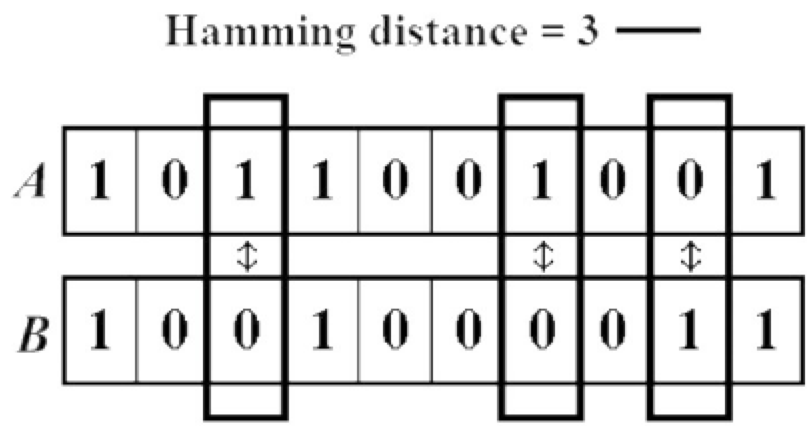
Handling Creases
Modified Orientation Field Estimation

1. Detect a set of 6 strongest sinusoid waves via Fourier transform of each 16x16 pixel blocks
2. Cluster strongest waves based on compatibility, i.e., similar orientation and frequency
3. Grow each seed orientation field by including adjacent and compatible waves
4. Select largest seed as final orientation field
5. Proceed with typical fingerprint feature extraction steps
    a. Gabor Filtering -> contextual filtering
    b. Binarization
    c. Thinning
    d. Minutiae Extraction

Use in Forensics

- Control for costs by using low resolution cameras that are not capable of getting well-defined ridges
- Rely on creases
  - Cropping and normalization based on distances between fingers
  - Gabor filtering with predefined parameters
  - Binarization
  - Match score using Hamming Distance





9. Summary

- Fingerprint recognition is one of the most mature biometric technologies.
- Some challenges still exist:
  - 3D & touchless fingerprint sensing
  - assure the integrity of fingerprints
  - Automated latent processing and matching
  - Privacy and security technology