Fingerprint Modeling + Orientation Field Estimation

Prof. Arun Ross

Michigan State University

rossarun@cse.msu.edu

http://iprobe.cse.msu.edu

Thanks to:

Dr. Jianjiang Feng

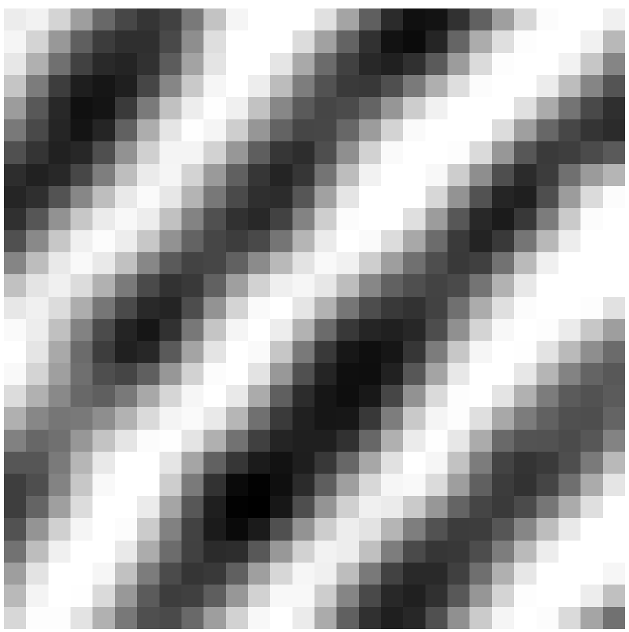
Tsinghua University

Fingerprint Modeling

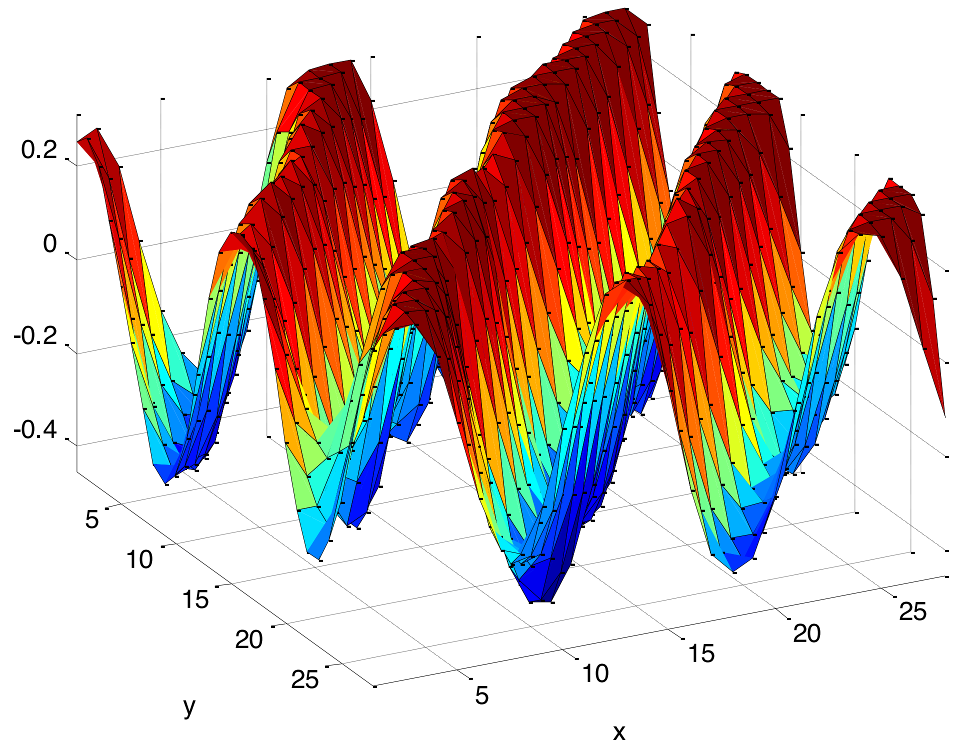
Ridge pattern in a local area of a fingerprint can be approximated by a cosine wave

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

pixel value amplitude frequency orientation



Local fingerprint region



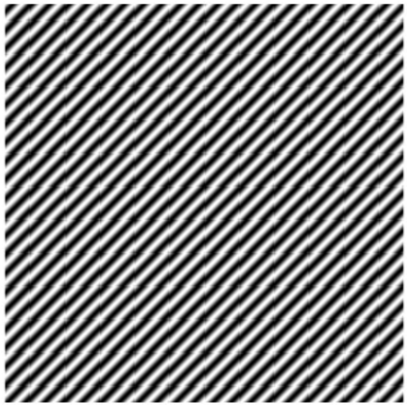
Shown as surface

Fingerprint Model

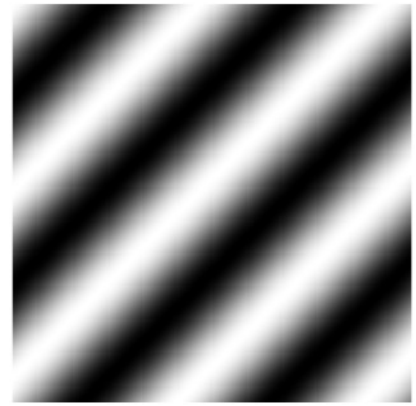
- $w(x, y) = A\cos\left(2\pi f(x\cos\theta + y\sin\theta)\right)$
- The local fingerprint region can be modeled using this equation
- Consider:
  - A 300 x 300 blank image (as an example)
  - Some values for parameters f, θ, A
- By varying (x, y) in the range 1:300, the pixel intensity at (x, y) can be computed

Fingerprint Model

Examples



$A = 50; f = 10; \ \theta = \pi/4;$



$A = 80; f = 0.01; \ \theta = \pi/4;$

$A = 50; f = 0.1; \ \theta = \pi;$

$A = 50; f = 0.01; \ \theta = \pi/2;$

2 methods to compute orientation field

- The first one is based on the Fourier Transform applied to the local region
- The second one is based on filtering the image using edge detection filters

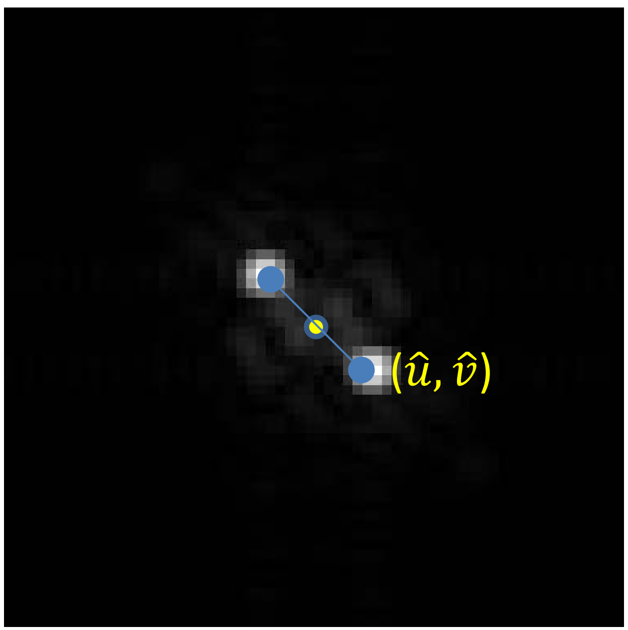
Ridge orientation & frequency estimation

2D Fourier transform of cosine wave

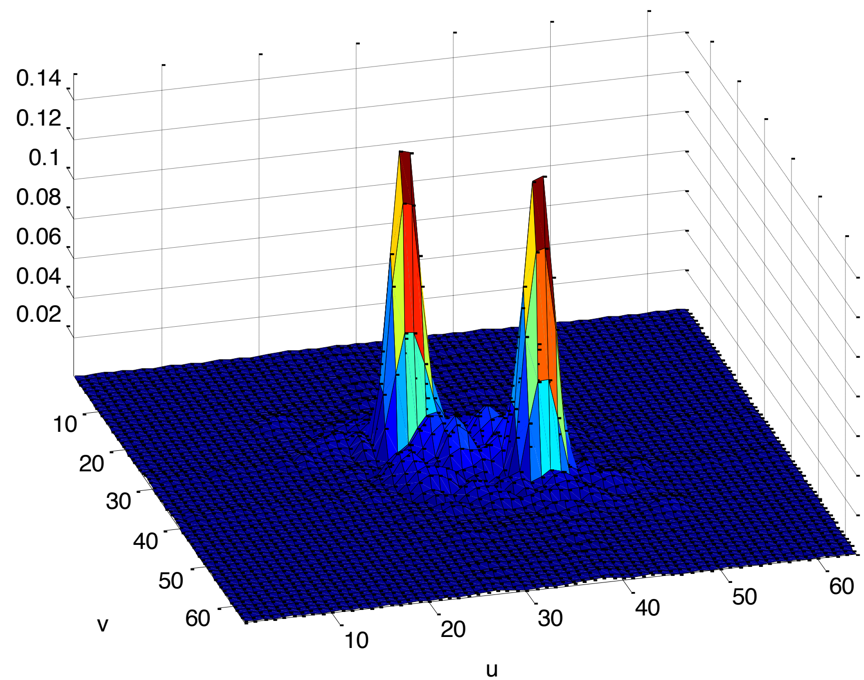
$$W(u,v) = \frac{A}{2}[\delta(u - f\cos\theta, v - f\sin\theta) + \delta(u + f\cos\theta, v + f\sin\theta)]$$

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

$$\hat{\theta} = \arctan\left(\frac{\hat{v}}{u}\right), \hat{f} = \sqrt{\hat{u}^2 + \hat{v}^2}$$



Magnitude spectrum



Magnitude spectrum shown as surface

Orientation field smoothing

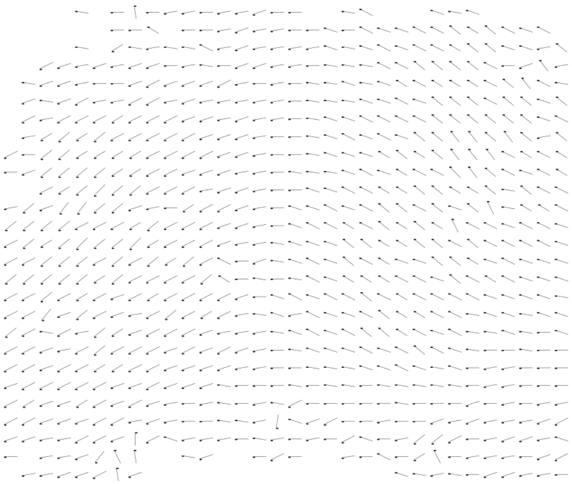
- Orientation field computed using the above method is vulnerable to noise.
- Orientation field is particularly important for extracting minutiae. To deal with noise, we should smooth the orientation field.
- Special consideration on ridge orientation:
  - defined in the range $[0, \pi)$
  - θ and $(\theta + \pi)$ are the same orientation
  - the average value between θ and $(\theta + \pi)$ should be θ rather than $\frac{2\theta+\pi}{2}$

Orientation field smoothing

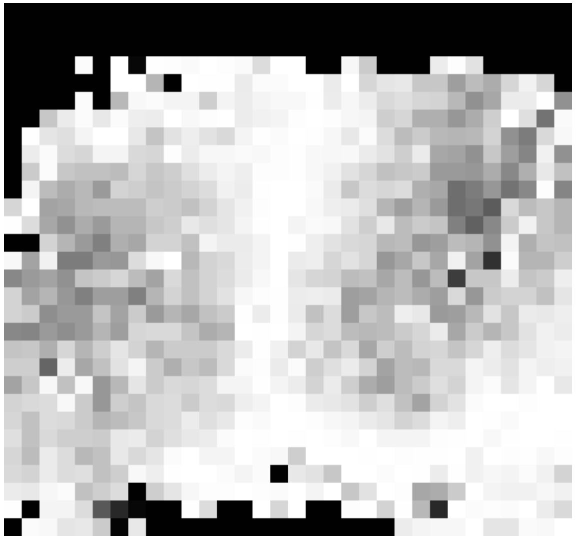
3 steps to smooth orientation field:

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

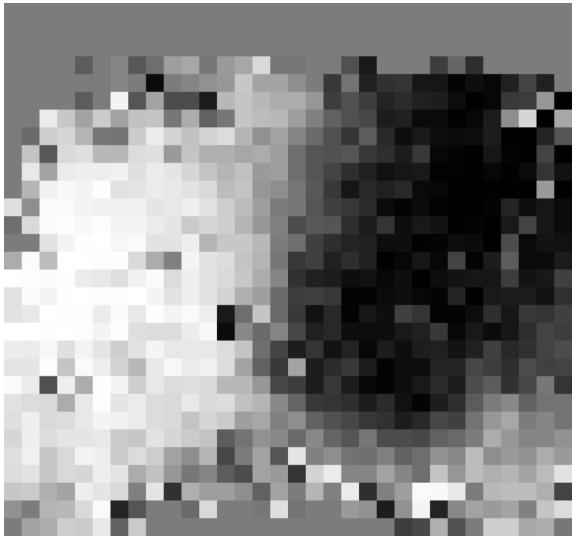
Orientation field smoothing



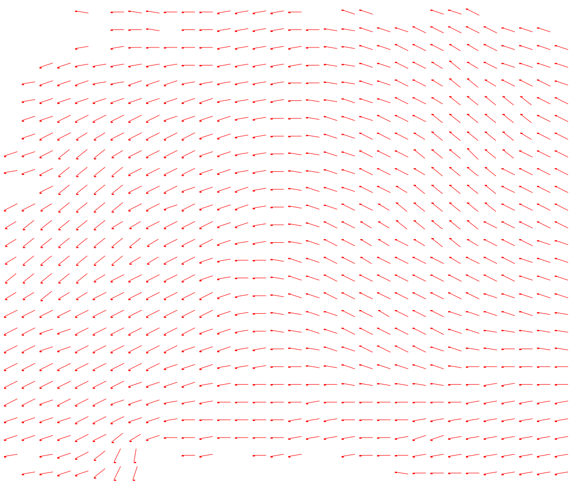
Initial OF, θ



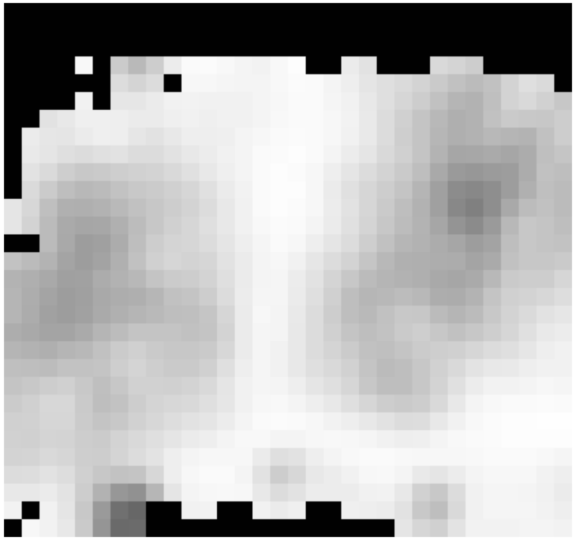
$V_x = \cos(2\theta)$



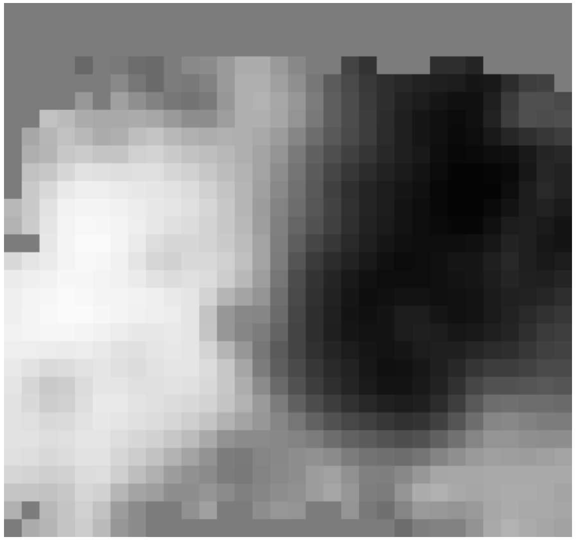
$V_y = \sin(2\theta)$



Smoothed OF, θ'



$V_x{}'$



$V_y{}'$

Edge Detection

- Image filters can be used to detect edges in an image
- An edge corresponds to a collection of pixels
- The pixel intensity changes sharply across the edge
  - An edge may correspond to the boundary of an object
  - In the case of fingerprints, the contours of the ridges may be viewed as "boundaries"
- The magnitude of the gradient or difference in pixel intensities in the neighborhood of a pixel can help in detecting edge pixels

Edge Filters (Kernels)

- Gradient can be computed in the horizontal and vertical directions
- **Prewitt Filters**:

$$P_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

This will highlight horizontal edges

$$P_y = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

This will highlight vertical edges

Edge Filters (Kernels)

- Gradient can be computed in the horizontal and vertical directions
- **Sobel Filters**:

$$S_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

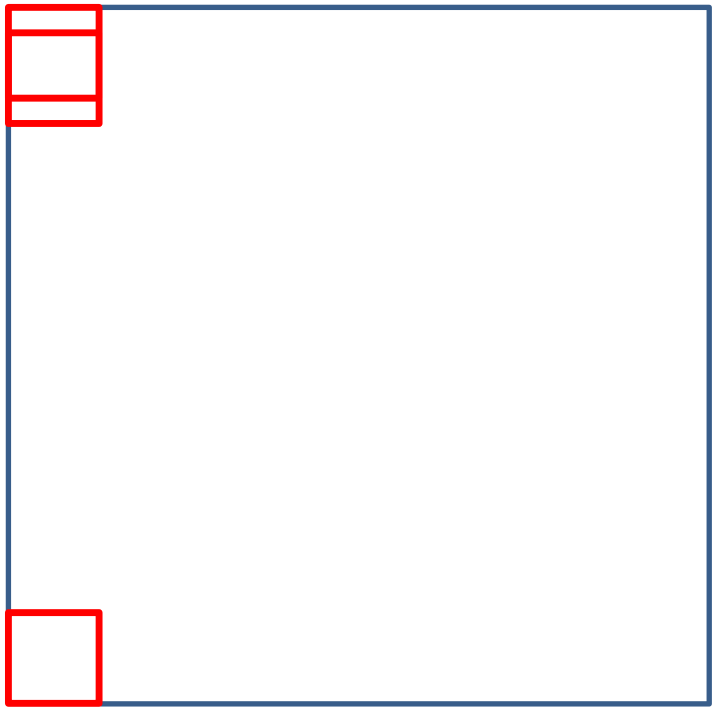
This will highlight horizontal edges

$$S_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

This will highlight vertical edges

Convolution: Gradient Images

- The image, I, is convolved with the edge filters to obtain two gradient images, G_x and G_y
  - $G_x = I \otimes S_x$
  - $G_y = I \otimes S_y$



Convolution: Gradient Images

- The image, I, is convolved with the edge filters to obtain two gradient images, G_x *and* G_y
  - $G_x = I \otimes S_x$
  - $G_y = I \otimes S_y$

- Magnitude of Gradient at a pixel (a,b) is $[G_x(a,b)]^2 + [G_y(a,b)]^2$

- Edge direction at the pixel can also be computed

Computing Orientation Field

- O(x,y) =

$$\frac{1}{2}\tan^{-1}\left[\frac{\sum_{i=-k}^{k}\ \ \sum_{j=-k}^{k} 2Gx(x+i,y+j)G_y(x+i,y+j)}{\sum_{i=-k}^{k}\ \ \sum_{j=-k}^{k} {G_x}^2(x+i,y+j)-{G_y}^2(x+i,y+j)}\right]$$

- So first apply the two filters independently on the image
- Based on the two filtered outputs, compute the orientation at a pixel

Singularity extraction

Let $O[i], i = 0, \cdots, 7$,denote the orientations at 8 neighbors of point x.

3	2	1
4	x	0
5	6	7

Poincaré index (PI) at x is:

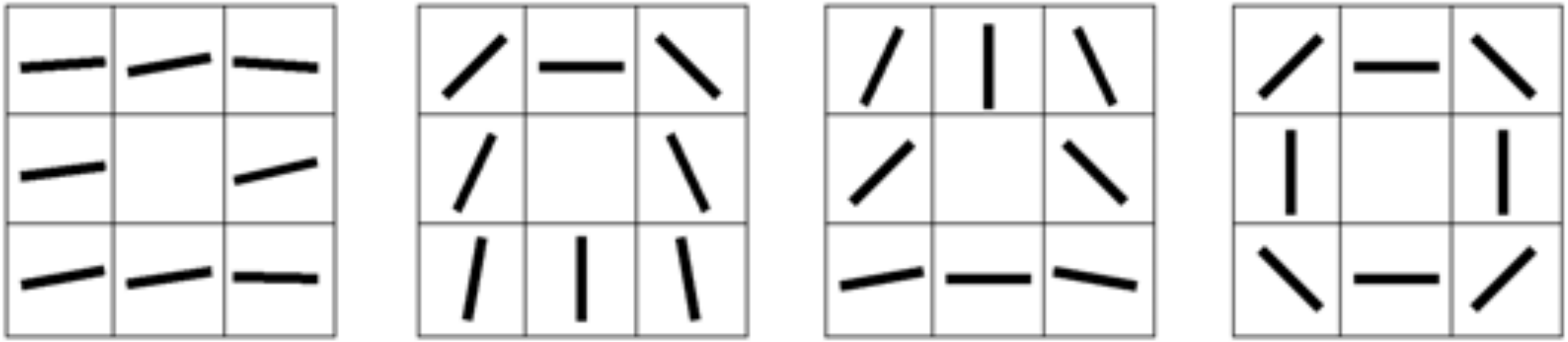
$$PI = \frac{1}{\pi} \sum_{i=0}^{7} \delta(O[(i+1)_{\mathrm{mod}\ 8}] - O[i]),$$

where

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

This function is used to make sure the orientation change is within $\pm$90 degrees
Note that above equations assume “radians”.

Poincaré index in fingerprint



5	10	-5
7		13
10	8	-1

Non-singularity

0

45	0	-45
65		-65
80	90	-80

Loop

1

65	90	-65
45		-45
10	0	-10

Delta

-1

45	0	-45
90		90
-45	0	45

Whorl

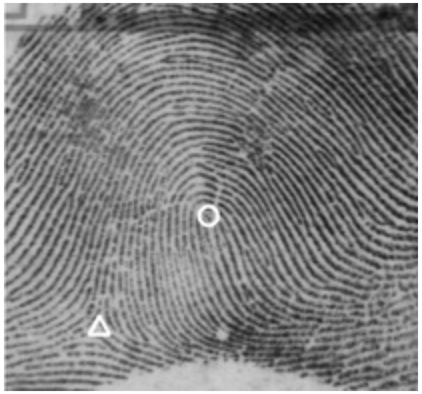
2

Here, angle is specified in “degrees”. In some cases, it will be specified in “radians”

Pattern classification

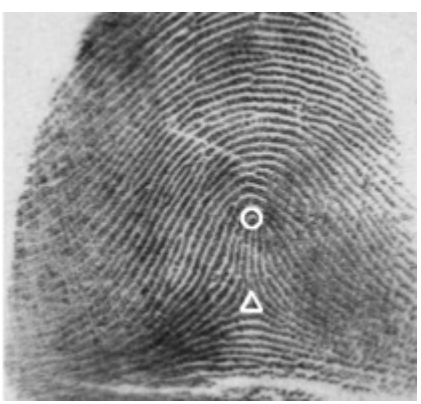
We can classify a fingerprint into one of 6 major pattern types based on singular points (SP):

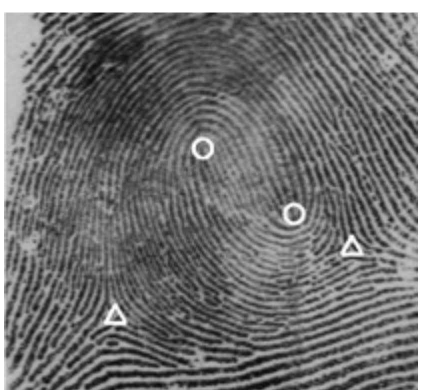
- **plain arch**: contains no SP.
- **left loop**: contains 1 delta and 1 loop whose direction points to the left side of the delta.
- **right loop**: contains 1 delta and 1 loop whose direction points to the right side of the delta.



Pattern classification

- **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 do not form a circular orbit.





Ridge extraction

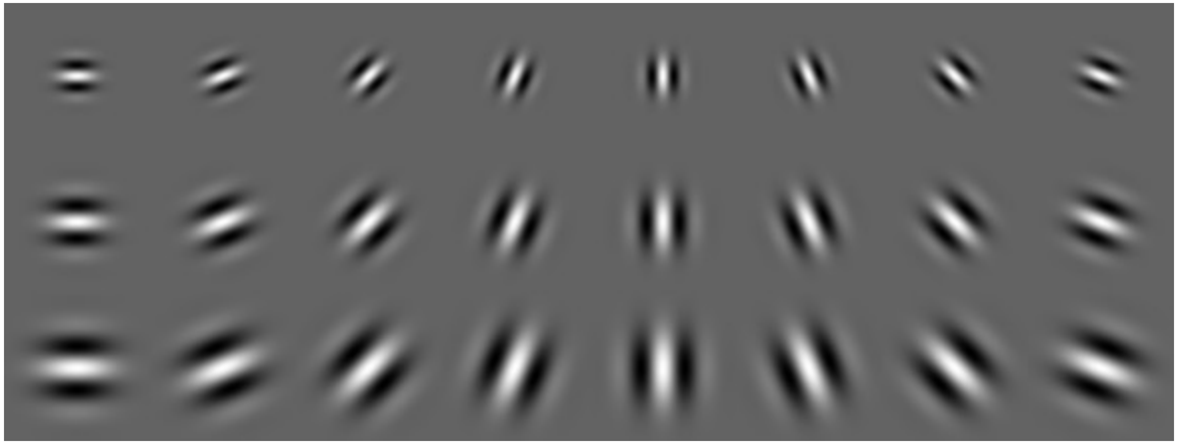
- A straightforward method is binarization.
- Problems:
  - Sweat pores on ridges are brighter than the surrounding pixels;
  - ridges can be broken due to cuts or creases;
  - adjacent ridges may appear to be joined due to wet skin or large pressure.
- Countermeasure: fingerprint enhancement.
- General purpose image enhancement is not effective for fingerprint.
- A successful fingerprint enhancement method is contextual filtering, such as Gabor filtering.

2D Gabor filters

2D Gabor wavelet:

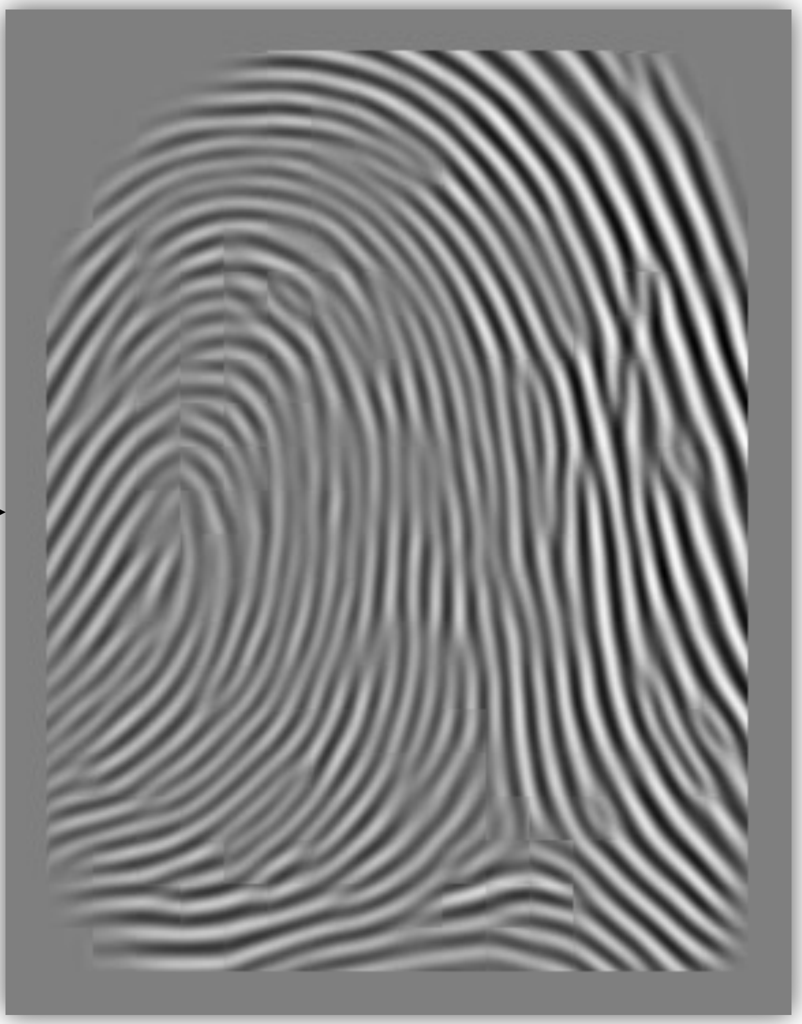
$$G(x,y) = e^{-\pi\left[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2\right]}e^{-2\pi i[u_0(x-x_0)+v_0(y-y_0)]}$$

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 $\omega_0 = \sqrt{{u_0}^2 + {v_0}^2}$.



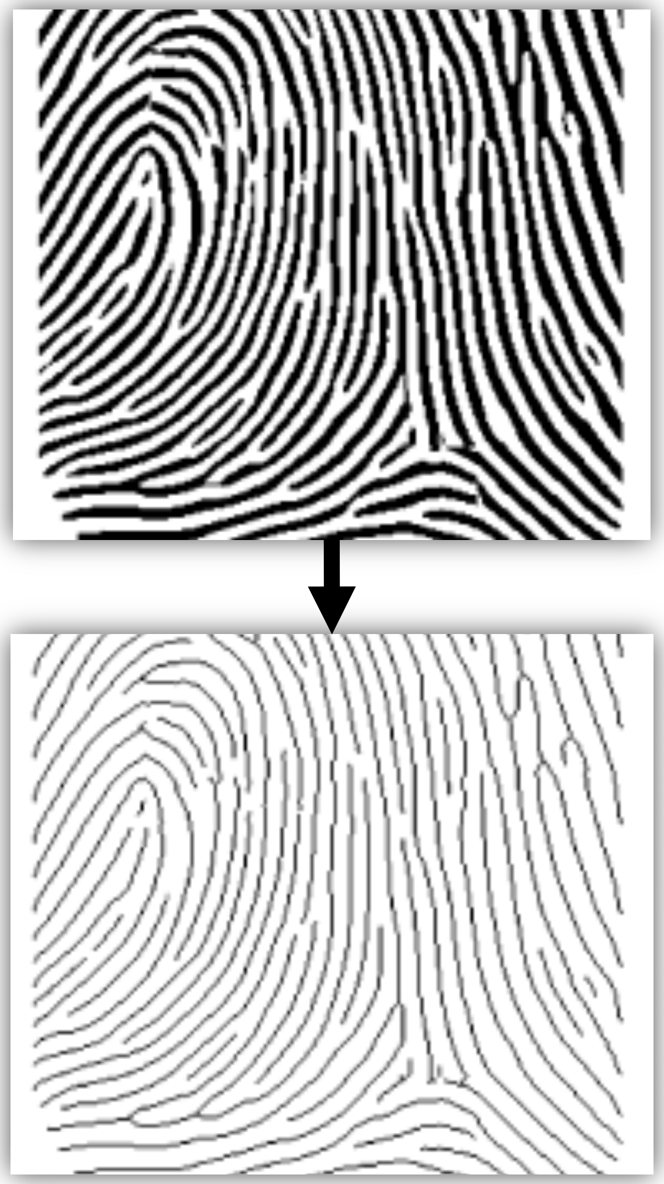
Real parts of Gabor filters (8 orientations and 3 scales)

Effect of Gabor filtering



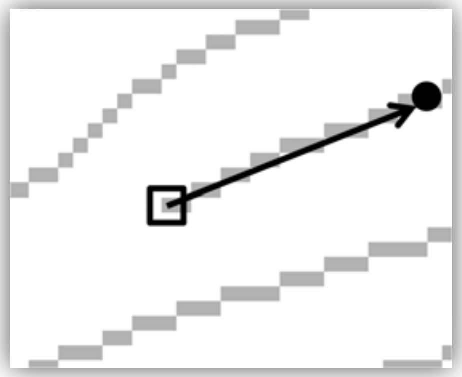
Ridge extraction

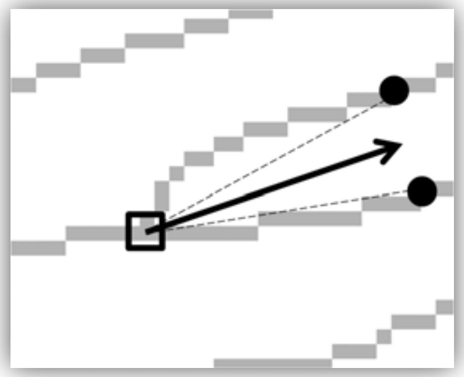
- Enhanced image can be converted into a binary image by comparing with thresholds (e.g. local mean).
- A morphological operation, thinning, is used to obtain the skeleton image.
- Thinning is a common technique in image processing, which involves iteratively removing outer ridge pixels.



Minutiae extraction

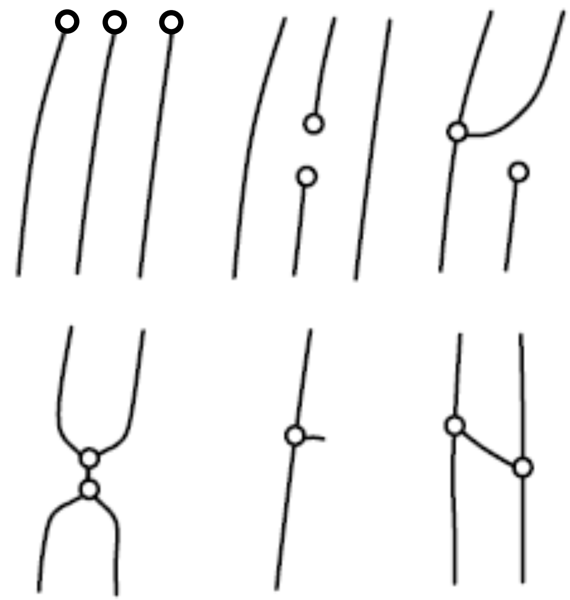
- Minutiae are special points on ridges:
  - ridge bifurcation (3 neighbors are black)
  - ridge ending (1 neighbor is black)
- Direction of a ridge ending:
  - Trace the associated ridge with a fixed distance (say 10 pixels) from $\boldsymbol{x}$ to $\boldsymbol{a}$. The direction $\boldsymbol{xa}$ 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.





Minutiae verification

- Previous method considers only 3×3 window, producing false minutiae due to:
  - artifacts in image processing
  - noise in a fingerprint
- A minutia is classified as false if it meets any of the following conditions:
  - has no adjacent ridge on either side
  - be close in location and opposite in direction
  - too many minutiae in a small neighborhood



Handbook of fingerprint recognition