# Enhancement, and Minutias Detection II

Daniel Novák

16.10, 2012, Prague

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

## Cviceni

- Cviceni byla upresnena!!!!

# Estimation of Local Ridge Orientation

- Average orientation around indices i,j
- Unoriented directions
- Weighted ($r_{ij}$)
- Gradient, maximum pixel-intensity change, arctan gy/gx

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right).$$

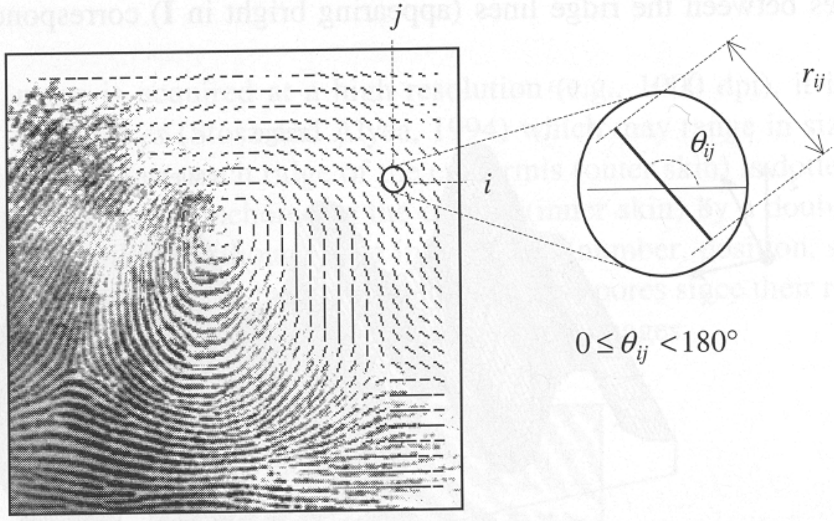


Figure 3.8. A fingerprint image faded into the corresponding orientation image computed over a square-meshed grid of size 16×16. Each element denotes the local orientation of the fingerprint ridges; the element length is proportional to its reliability.

# Estimation of Local Ridge Orientation

– Simple Approach

- Gradient with Sobel or Prewitt operators
- $\Theta_{ij}$ is orthogonal to the direction of the gradient

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Drawbacks:
- Non-linear and discontinuous around 90
- A single estimate is sensitive to noise
- Circularity of angles: Averaging is not possible
- Averaging is not well defined.

## Estimation of Local Ridge Orientation

– Averaging Gradient Estimates (Kass, Witkin 1987)

$$d_{ij} = [r_{ij}.\cos 2\theta_{ij}, r_{ij}\sin 2\,\theta_{ij}]$$

$$\bar{\mathbf{d}} = \left[\frac{1}{n^2}\sum_{i,j} r_{ij}\cdot\cos 2\theta_{ij}, \frac{1}{n^2}\sum_{i,j} r_{ij}\cdot\sin 2\theta_{ij}\right]. \qquad (2)$$

$$r = \nabla_x^{\,2} + \nabla_y^{\,2}$$

## Estimation of Local Ridge Orientation

– Effect of averaging

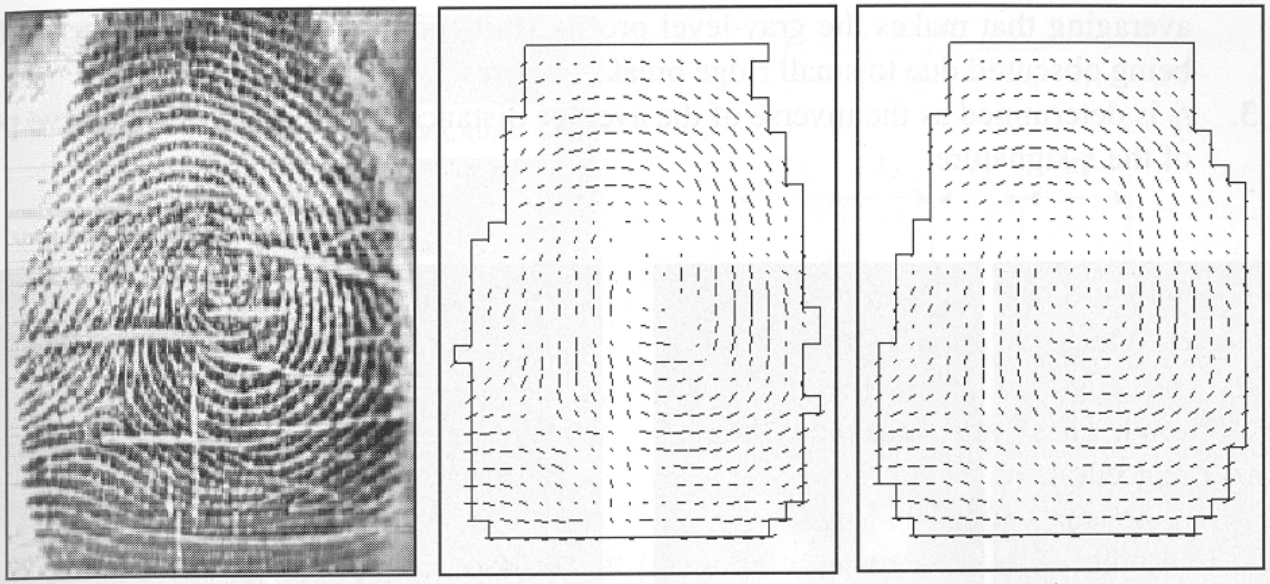


Figure 3.9. a) A poor quality fingerprint image; b) the orientation image of the fingerprint in a) is computed through the Donahue and Rokhlin (1993) method; the orientation of several elements is clearly inconsistent and a regularization step appears necessary; c) the orientation image is the result of the local averaging of each element in b) in its 3 × 3 window according to Equation (2).

# Orientation field

An orientation is calculated for each 16x16 block

- Compute the gradient of the smoothed block. $G_x(i,j)$ and $G_y(i,j)$ using 3x3 Sobel Masks

- Obtain the dominant direction in the block using the following equation:

$$\theta_d = \frac{1}{2} tan^{-1} \left( \frac{\sum_{i=1}^{16} \sum_{j=1}^{16} 2G_x(i,j)G_y(i,j)}{\sum_{i=1}^{16} \sum_{j=1}^{16} (G_x(i,j)^2 - G_y(i,j)^2)} \right), G_x \neq 0 \ and \ G_y \neq 0 \qquad (1)$$

$$G_{xy} = \sum_{h=-8}^{8} \sum_{k=-8}^{8} \nabla_x(x_i + h, y_j + k) \cdot \nabla_y(x_i + h, y_j + k),$$

$$G_{xx} = \sum_{h=-8}^{8} \sum_{k=-8}^{8} \nabla_x(x_i + h, y_j + k)^2,$$

$$G_{yy} = \sum_{h=-8}^{8} \sum_{k=-8}^{8} \nabla_y(x_i + h, y_j + k)^2,$$

$$\theta_{ij} = 90^\circ + \frac{1}{2} atan2(2G_{xy}, G_{xx} - G_{yy})$$

- **DEMO: Fp1 = computeorientationarray(Fp1);**
- Gradient is computed by (standard): [fx, fy] = gradient(double(im));
- Block 10x10

# Example: Orientation field

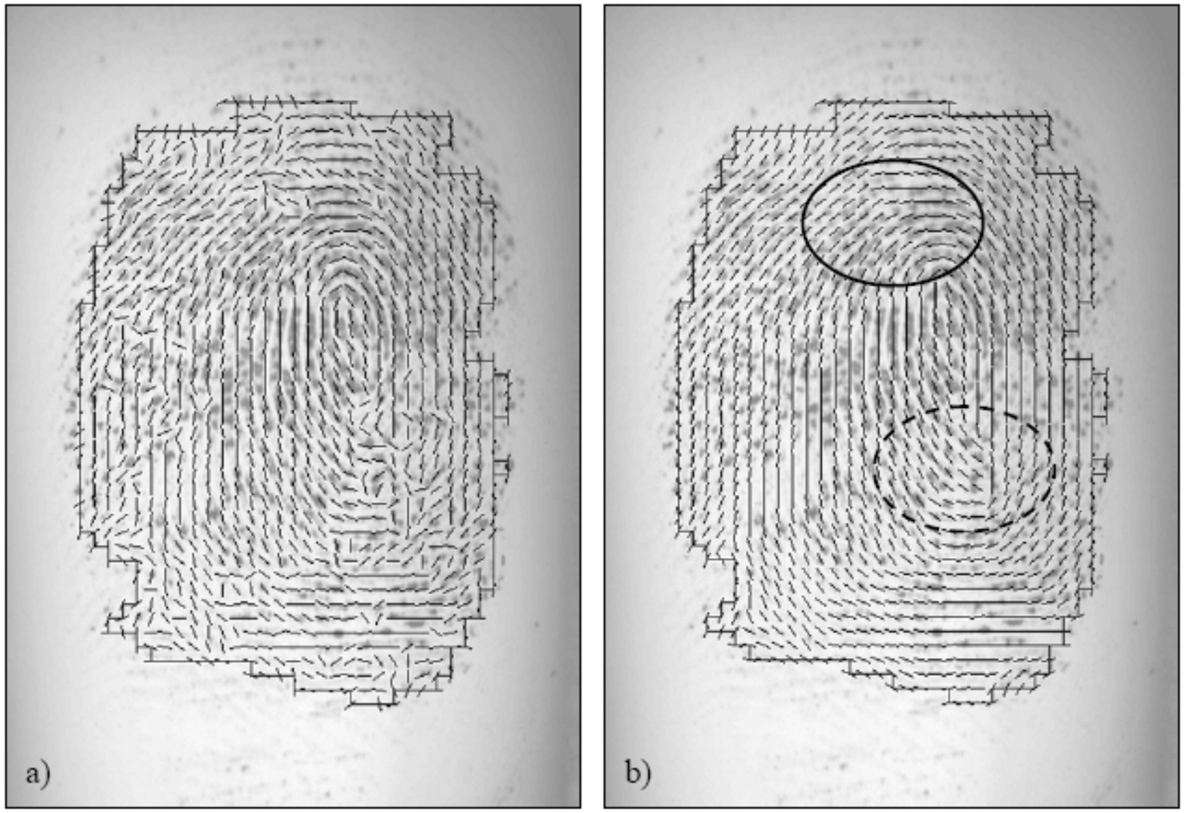
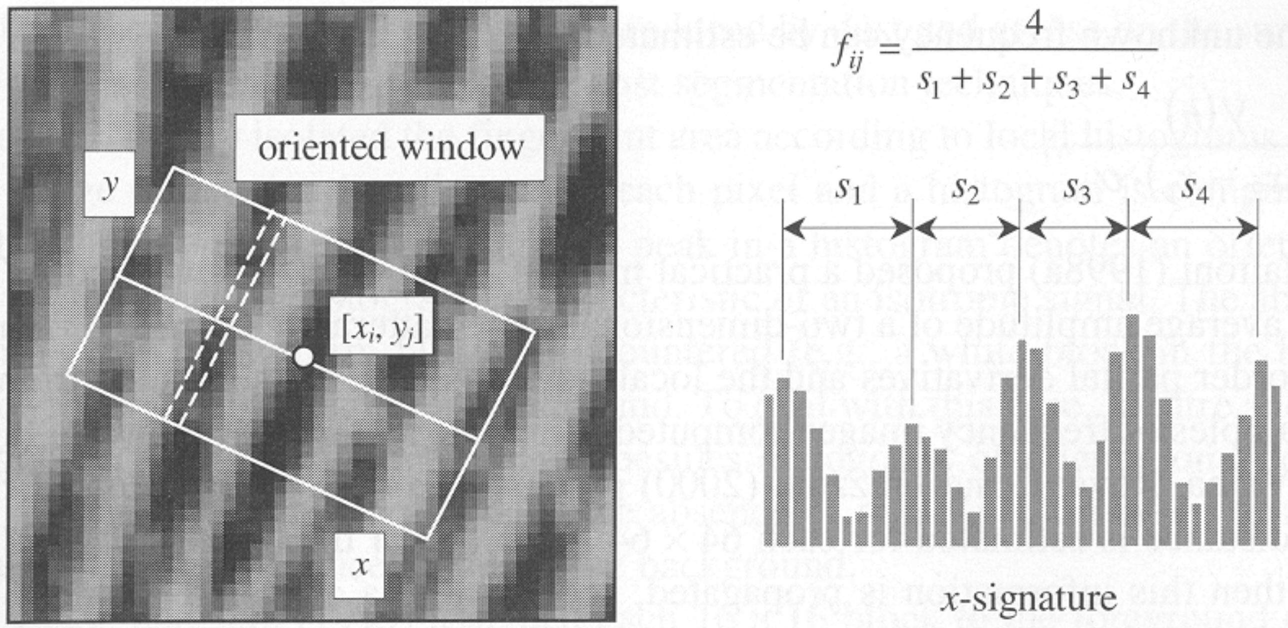


Figure 3.12. a) Estimation of local ridge orientation in a fingerprint through the gradient-based approach corresponding to Equation (3): in the noisy regions the estimation is unreliable; b) two iterations of local (3x3) smoothing are applied, resulting in a more consistent representation; it is worth noting that while the smoothing recovered the correct orientation at several places (e.g., inside the solid circle), it altered the average orientation inside the region denoted by the dashed circle where incorrect orientations were dominating the correct one.

# Estimation of Local Ridge Frequency



$$f_{ij} = \frac{4}{s_1 + s_2 + s_3 + s_4}$$

Figure 3.11. An oriented window centered at [$x_i$,$y_j$]; the dashed lines show the pixels whose gray-levels are accumulated for a given column of the *x*-signature (Hong, Wan, and Jain, 1998). The *x*-signature on the right clearly exhibits five peaks; the four distances between consecutive peaks are averaged to determine the local ridge frequency.

# Estimation of Local Ridge Frequency

Simple Algorithm

1) 32x16 oriented window centered at $[x_i, y_i]$

2) The x-signature of the grey levels is obtained

3) $f_{ij}$ is the inverse of the average distance

To handle noise interpolation and/or low pass filtering is applied.

**DEMO: Fp1 = computelocalfrequency(Fp1, Fp1.imOriginal);**

## Estimation of Local Ridge Frequency

– Examples of frequency maps

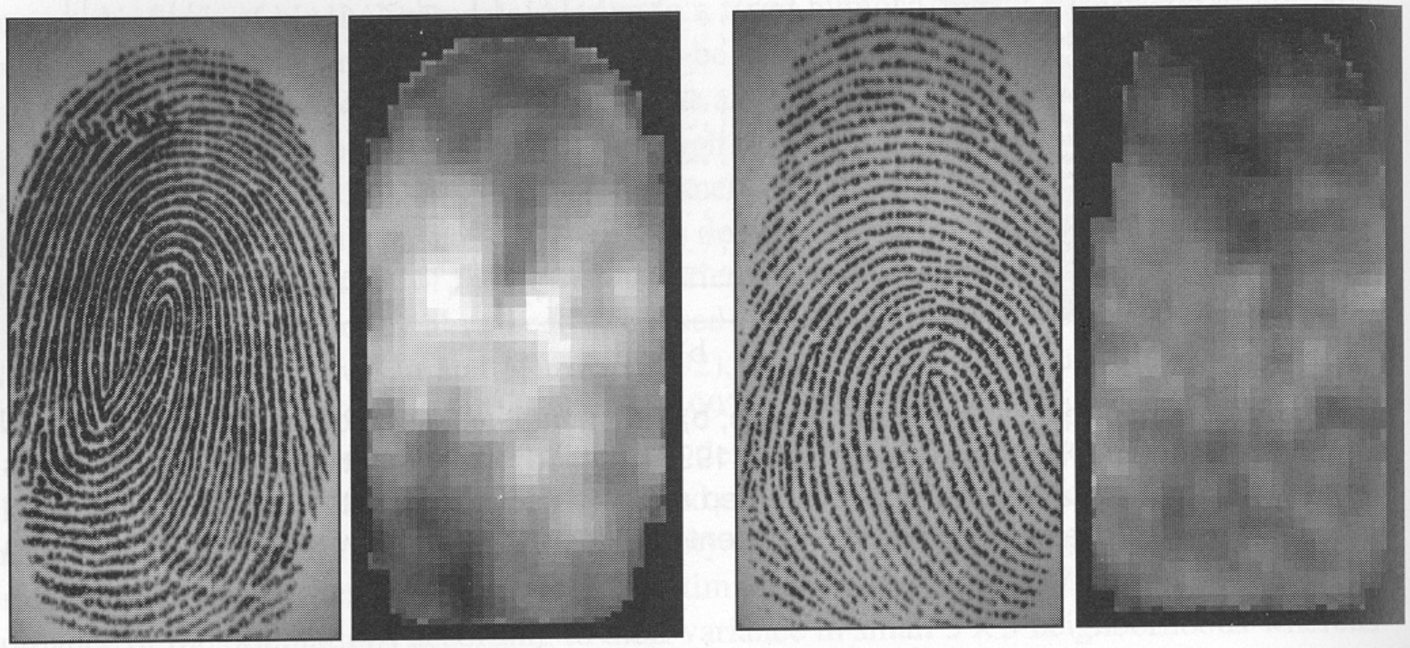


Figure 3.10. Two fingerprint images and the corresponding frequency image computed with the method proposed by Maio and Maltoni (1998a). A local 3 × 3 averaging is performed after frequency estimation to reduce noise. Light blocks denote higher frequencies. It is quite evident that significant changes may characterize different fingerprint regions and different average frequencies may result from different fingers.

## Contextual Filters

- The most widely used technique for fingerprint image enhancement
- Conventional image filtering – a single filter is used for convolution throughout
- Contextual filtering  - filter characteristics change according to local context
- Several types of contextual filters proposed
- Indented behavior –
  - 1)provide a low-pass [averaging] effect along the ridge direction. *Linking small gaps and filling impurities due to noise*
  - 2)perform a band pass [differentiating] in the direction orthogonal to the ridges
    *Increase discrimination between ridges and valleys*

## Method proposed by Hong, Wan, and Jain

- Based on Gabor filters
- Gabor filters have both frequency-selective and orientation-selective properties and have optimal joint resolution in spatial and frequency domains
- A Gabor filter is defined by a sinusoidal plane wave tapered by a Gaussian

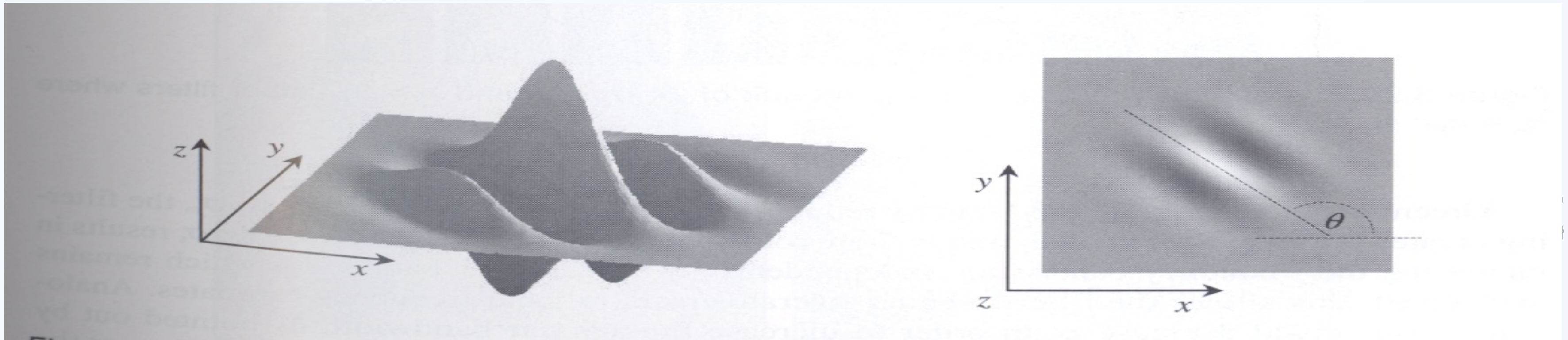
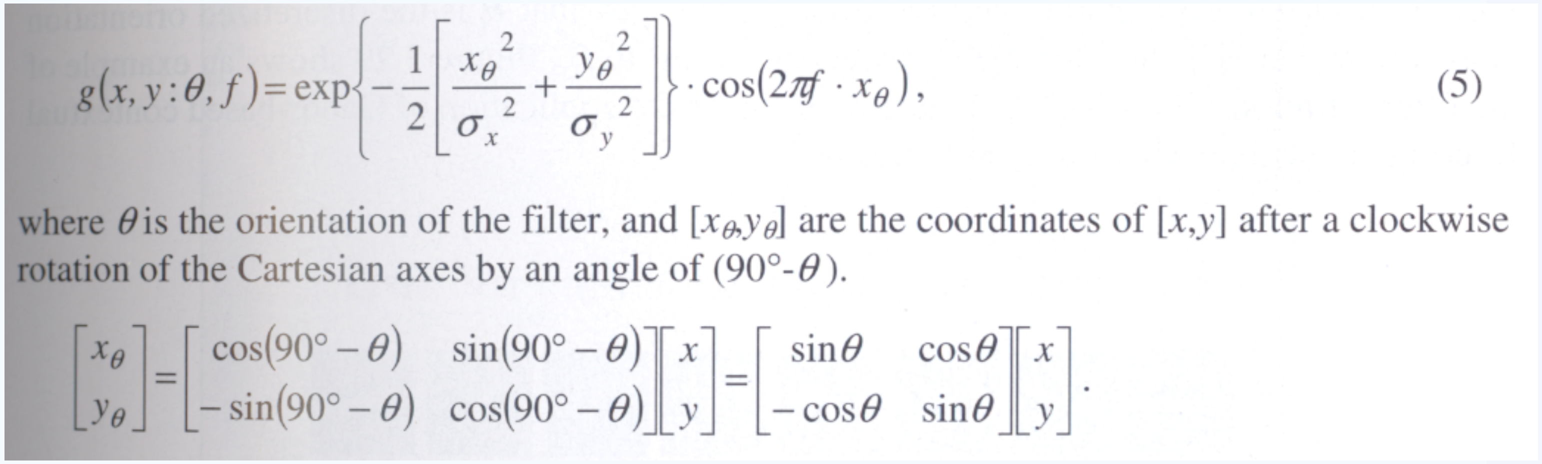


Figure 3.28. Graphical representation (lateral view and top view) of the Gabor filter defined by the parameters $\theta = 135°$, $f = 1/5$, and $\sigma_x = \sigma_y = 3$.

## Method proposed by Hong, Wan, and Jain (cont ..)

The even symmetric two-dimensional Gabor filter has the following form:



$$g(x,y:\theta,f)=\exp\left\{-\frac{1}{2}\left[\frac{x_\theta^{\,2}}{\sigma_x^{\,2}}+\frac{y_\theta^{\,2}}{\sigma_y^{\,2}}\right]\right\}\cdot\cos(2\pi f\cdot x_\theta), \qquad (5)$$

where $\theta$ is the orientation of the filter, and $[x_\theta, y_\theta]$ are the coordinates of $[x,y]$ after a clockwise rotation of the Cartesian axes by an angle of $(90°-\theta)$.

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}=\begin{bmatrix} \cos(90°-\theta) & \sin(90°-\theta) \\ -\sin(90°-\theta) & \cos(90°-\theta) \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}.$$

Here, f is the frequency of a sinusoidal plane wave and $\sigma_x$ and $\sigma_y$ are the standard deviations of the Gaussian envelope along the x and y axes

## Method proposed by Hong, Wan, and Jain (cont ..) Gabor Filter

- 4 parameters – $\theta, f, \sigma_x, \sigma_y$
- The selection of the values $\sigma_x$ and $\sigma_y$ involves a tradeoff
- A set $\{g_{ij}(x,y) \mid i=1\ldots n_0, 1..n_f\}$ of filters are priori created and stored , where $n_0$ is the number of discrete orientations $\{\theta_i \mid i=1,..n_0\}$ and $n_f$ the number of discrete frequencies $\{f_j \mid j=1,..n_f\}$
- Each pixel [x,y] is convolved, with filter $g_{ij}(x,y)$ such that $\theta_i$ is the discretized orientation closest to $\theta_{xy}$ and $f_j$ is the discretized orientation closest to $f_{xy}$
- **DEMO: Fp1 = enhance2ridgevalley(Fp1);**

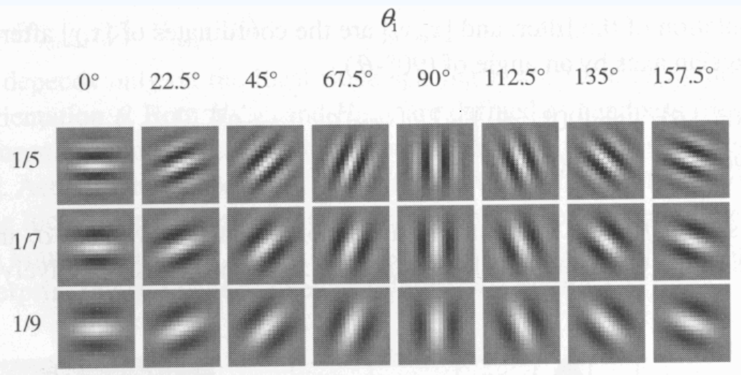


Figure 3.29. A graphical representation of a bank of 24 ($n_o = 8$ and $n_f = 3$) Gabor filters where $\sigma_x = \sigma_y = 4$.

## Method proposed by Hong, Wan, and Jain (cont ..) – Examples

- Shows the application of Gabor-based contextual filtering on medium and poor quality images

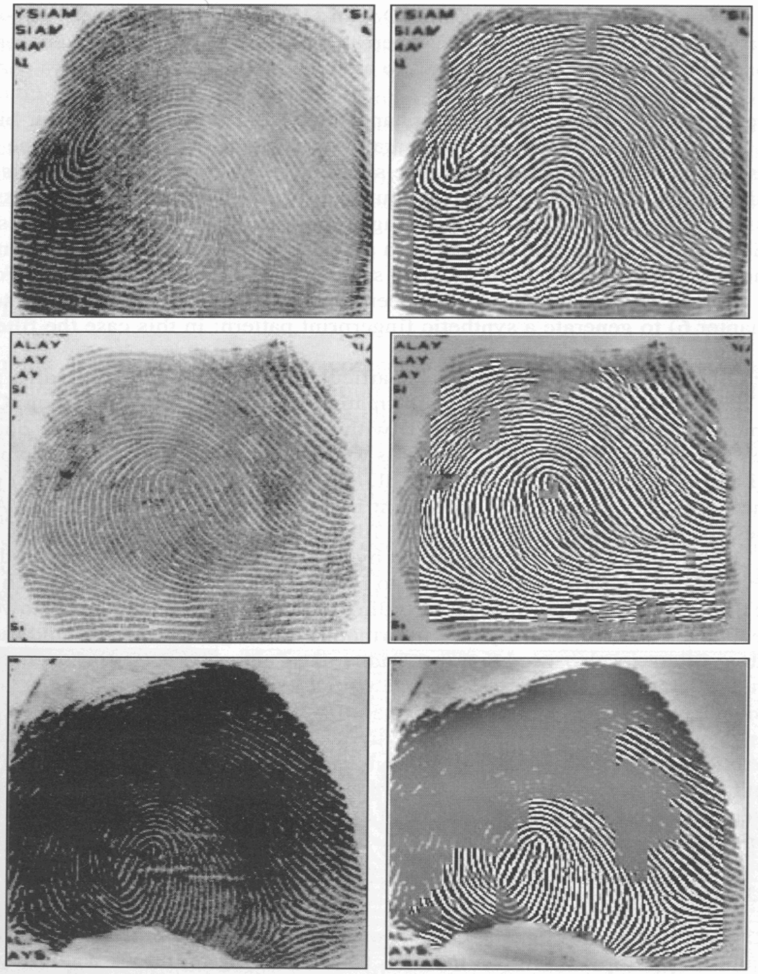


Figure 3.30. Examples of fingerprint enhancement with Gabor filtering as proposed by Hong, Wan, and Jain (1998). On the right, the enhanced recoverable regions are superimposed on the corresponding input images. ©IEEE.

## Minutiae Detection

- Reliable minutiae extraction is extremely important
  - Binarization
  - Thinning
  - Post processing – filling holes, linking breaks, removing bridges

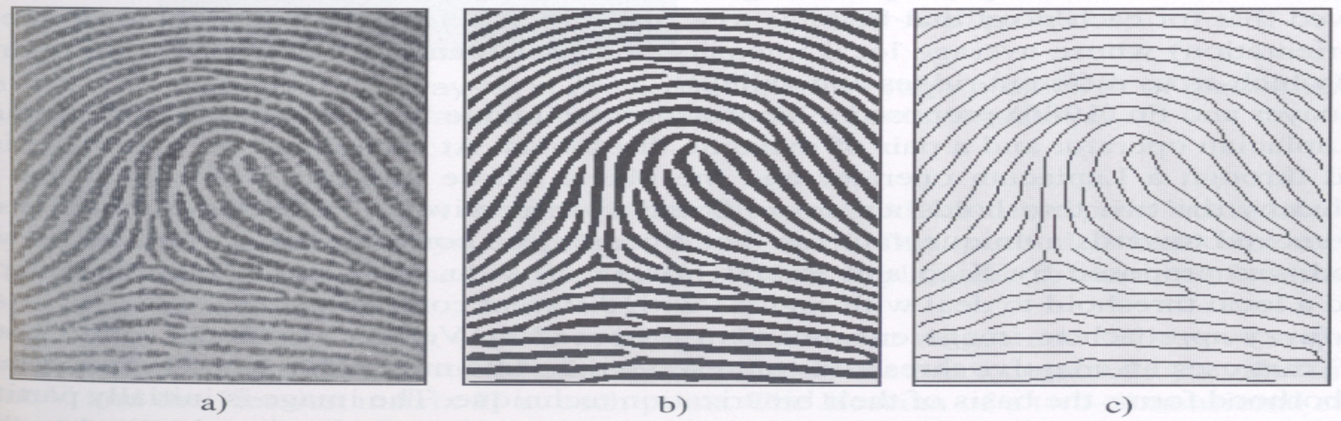


Figure 3.31. a) A fingerprint gray-scale image; b) the image obtained after a binarization of the image in a); c) the image obtained after a thinning of the image in b). Reprinted with permission from Maio and Maltoni (1997). ©IEEE.

## Binarization-based methods

- Simplest method - global threshold
- Local threshold technique
- Fingerprint specific solutions necessary

- Binarization is the output of Contextual Filtering: **enhance2ridgevalley.m**

```
binaryBlkSize = 20;
imReconstruct = blkproc(imReconstruct, [binaryBlkSize
   binaryBlkSize], @binarizeimage);

function  Iout = binarizeimage(Iin)
level = graythresh(Iin); %Otsu method
Iout = im2bw(Iin, level);
```

# Threshold computation: Otsu I

- Otsu's method: *N. Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9: 62–66*
  - *http://en.wikipedia.org/wiki/Otsu%27s_method*
  - *http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf*
- minimizes the intra-class variance – for each threshold T lot's of work

$$\sigma^2_{\text{Within}}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T)$$

$$\begin{aligned} n_B(T) &= \sum_{i=0}^{T-1} p(i) \\ n_O(T) &= \sum_{i=T}^{N-1} p(i) \\ \sigma^2_B(T) &= \text{the variance of the pixels in the background (below threshold)} \\ \sigma^2_O(T) &= \text{the variance of the pixels in the foreground (above threshold)} \end{aligned}$$

- between-class variance:

$$\begin{aligned} \sigma^2_{\text{Between}}(T) &= \sigma^2 - \sigma^2_{\text{Within}}(T) \\ &= n_B(T)\left[\mu_B(T) - \mu\right]^2 + n_O(T)\left[\mu_O(T) - \mu\right]^2 \end{aligned}$$

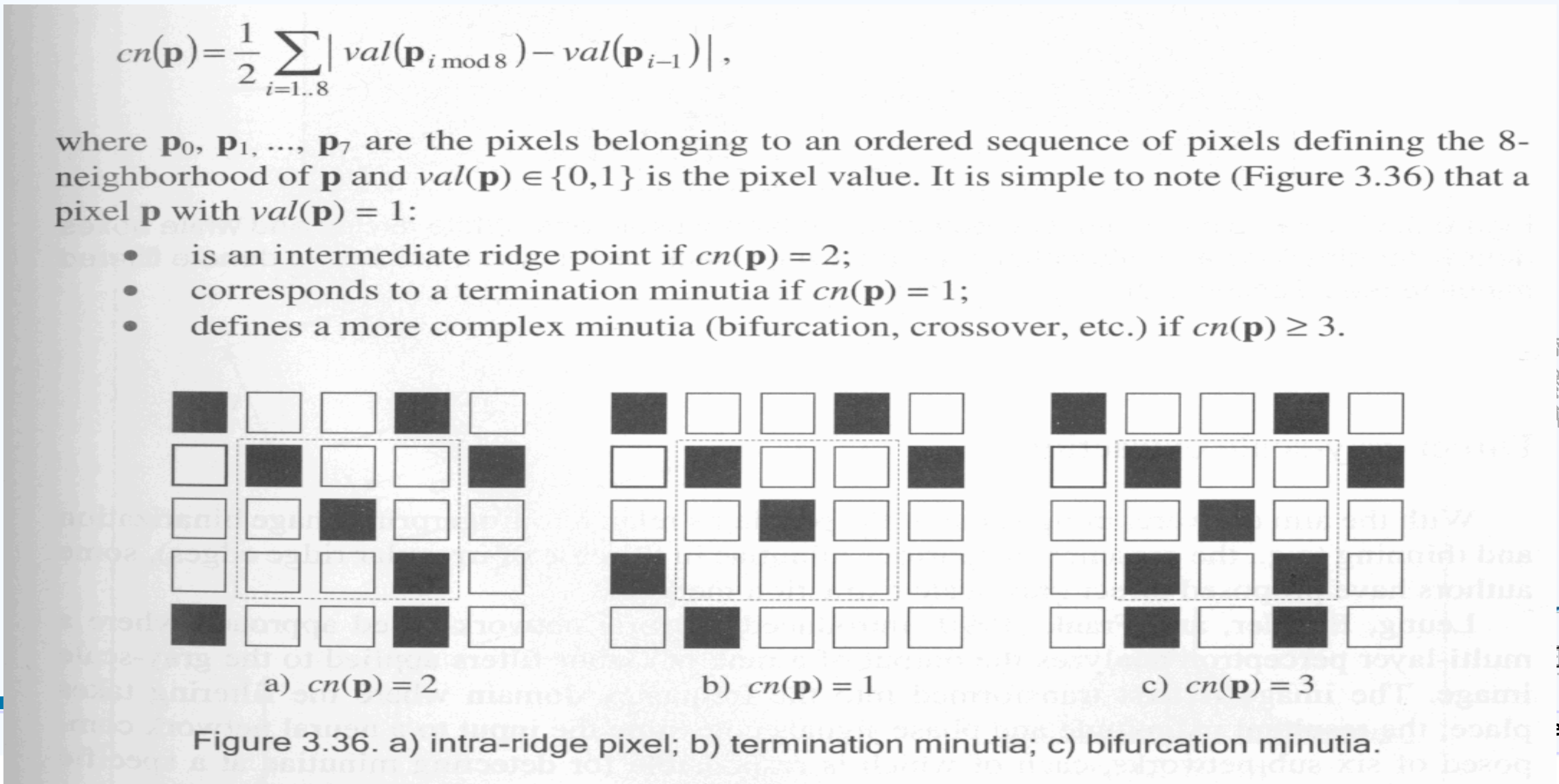
where $\sigma^2$ is the combined variance and $\mu$ is the combined mean

# Threshold computation: Otsu II

- Compute histogram and probabilities of each intensity level
- Set up initial $n_b(0)$ and $n_o(0)$ and $\mu_b(0)$, $\mu_o$ (0)
- Step through all possible thresholds T=1.... maximum intensity
  - Update $n_b(T)$,$n_o(T)$
  - Compute $\sigma_{between}(T)$
- Desired threshold corresponds to the maximum $\sigma_{between}(T)$

## Minutiae detection

- A simple image scan allows the pixel corresponding to minutiae to be detected
- *crossing number* of a pixel p
- **DEMO: Fp1 = cleanskeleton(Fp1);**



$$cn(\mathbf{p}) = \frac{1}{2} \sum_{i=1..8} \left| val(\mathbf{p}_{i \bmod 8}) - val(\mathbf{p}_{i-1}) \right|,$$

where $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_7$ are the pixels belonging to an ordered sequence of pixels defining the 8-neighborhood of $\mathbf{p}$ and $val(\mathbf{p}) \in \{0,1\}$ is the pixel value. It is simple to note (Figure 3.36) that a pixel $\mathbf{p}$ with $val(\mathbf{p}) = 1$:

- is an intermediate ridge point if $cn(\mathbf{p}) = 2$;
- corresponds to a termination minutia if $cn(\mathbf{p}) = 1$;
- defines a more complex minutia (bifurcation, crossover, etc.) if $cn(\mathbf{p}) \geq 3$.

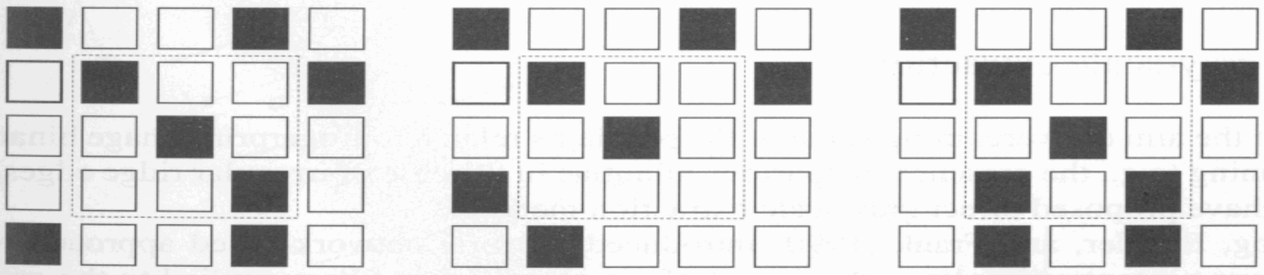


Figure 3.36. a) intra-ridge pixel; b) termination minutia; c) bifurcation minutia.

## Examples of minutiae extraction

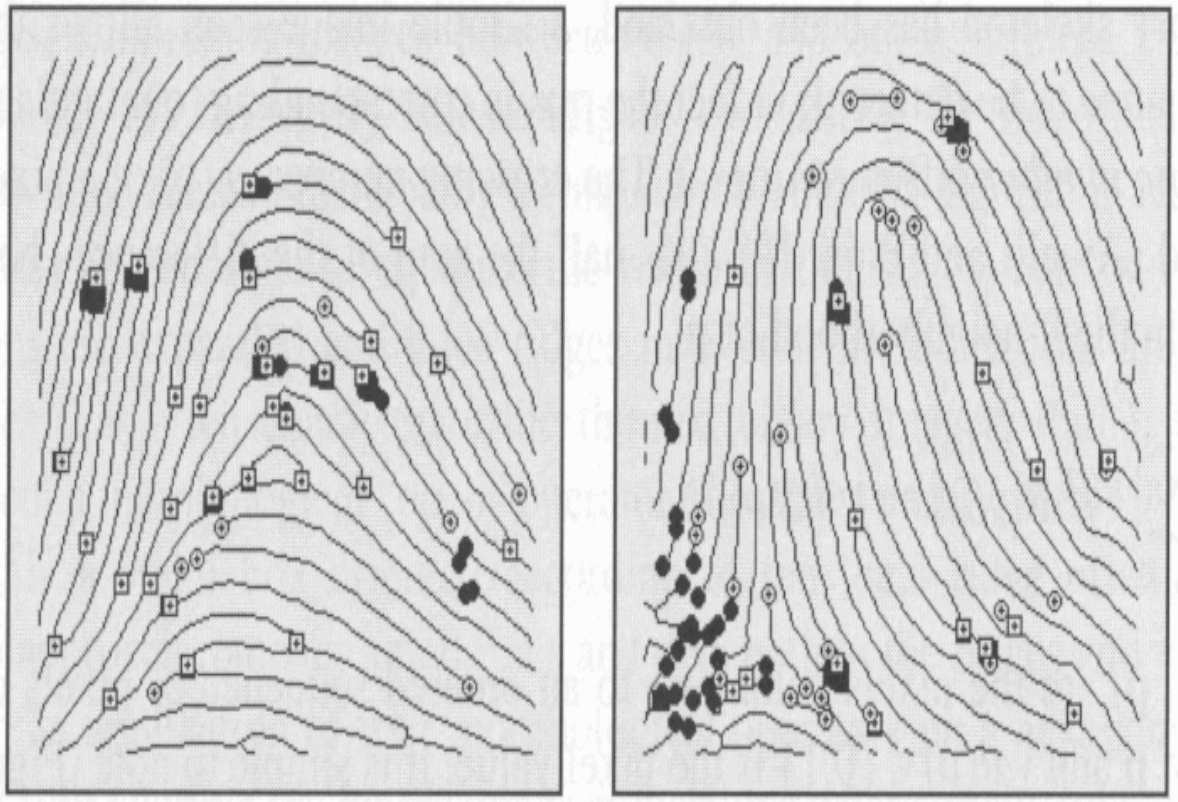
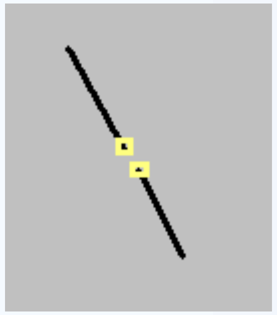


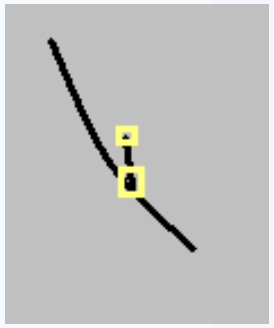
Figure 3.37. Examples of minutiae detection on binary skeletons. White circles and white boxes denote terminations and bifurcations, respectively; black circles and black boxes denote filtered minutiae (see Section 3.9).

# Minutiae Filtering

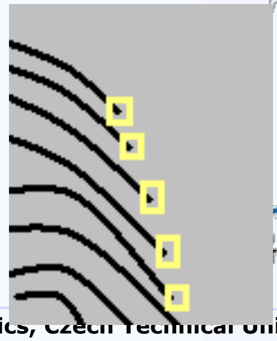
- Post-processing stage is useful for removing spurious minutiae [already present or introduced by previous steps]
- Two main post-processing types:
  - Structural post-processing
  - Minutiae filtering in the gray-scale domain



- Ridge breaks (insufficient ink or moist)



- Ridge cross-connections (over-ink, over-moist)



- Boundaries

## Example

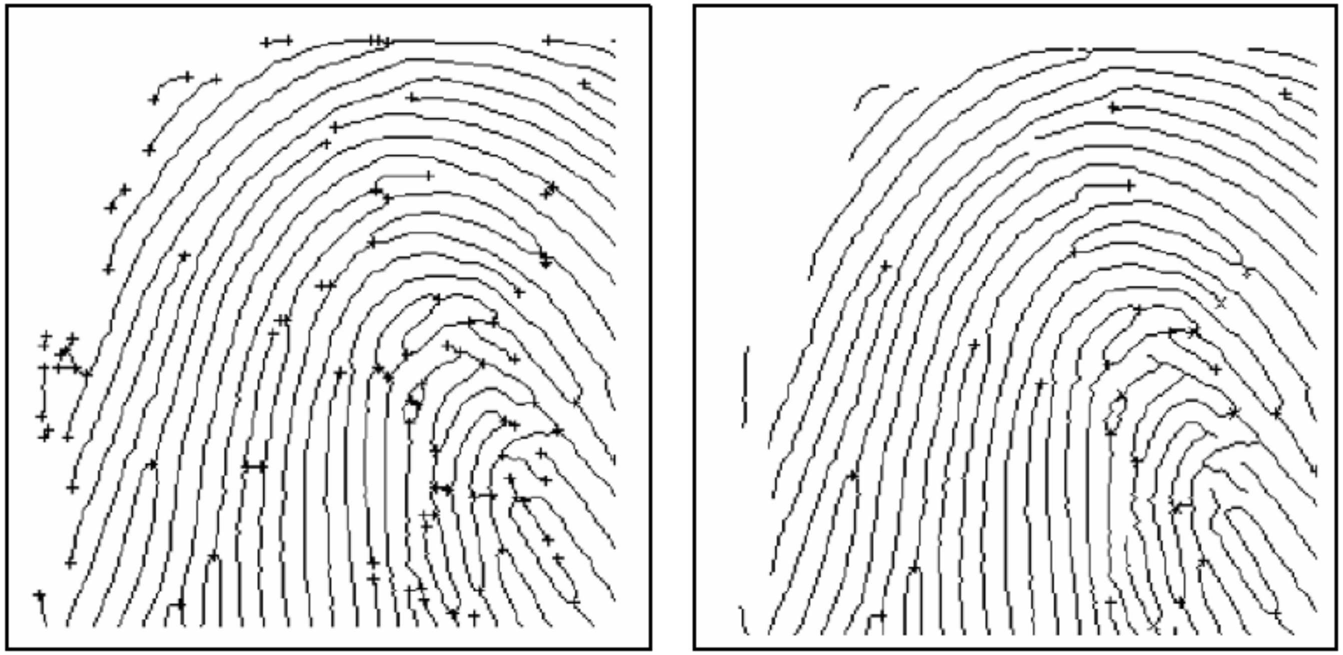


Figure 3.51. Minutiae post-processing according to Farina, Kovacs-Vajna, and Leone (1999). On the right, most of the false minutiae present in the thinned binary image (on the left) have been removed. © Elsevier.