Fingerprint Enhancement and Identification by Adaptive Directional Filtering

EE5359 MULTIMEDIA PROCESSING
SPRING 2015
Under the guidance of
Dr. K. R. Rao
Presented by
Vishwak R Tadisina
ID:1001051048
Acronyms

- **1D** - One Dimension
- **2D** - Two Dimension
- **AFIS** – Automatic Fingerprint Identification System
- **DC** - Direct Current
- **FBI** – Federal Bureau of Investigation
- **FFT** - Fast Fourier Transform
- **ICPR** – International Conference on Pattern Recognition
- **IEE** – Institution of Electrical Engineers
- **IEEE** - Institute of Electrical and Electronics Engineers
- **ISCV** – International symposium on Computer Vision
- **LPF** - Low Pass Filter
- **MATLAB** - Matrix Laboratory
- **MTF** – Monetary Transfer Function
- **WACV** - Winter Conference on Applications of Computer Vision
Introduction

• Identifying a person based on the biometrics has become important in current diverse businesses like law enforcement, information system security, finance physical access control etc. [4].

• Fingerprint recognition is one of the most important biometric technologies which has drawn a substantial amount of attention recently [4].

• The best aspect of fingerprint-based identification is that the fingerprints of a person are unique and does not alter with aging of an individual [1]

• A method to manually match fingerprint was developed by law enforcement agencies [4]. But this method is tedious and time taking.

• Automatic fingerprint identification system (AFIS)

• Input can be given by digitalizing the image take by ink or by using inkless scanners.
Stages in AFIS

Fig.1 Different stages involved in an Automatic fingerprint identification system [11].
Suitable features for representation of a fingerprint

• Keep back the uniqueness of each fingerprint in various levels of resolution.
• Distinct characteristics of a fingerprint can be estimated easily.
• Easy to apply automatic matching algorithms.
• Immune to noise distortions.
• Effective and simple representation [11].
Fingerprint Structure

- Fingerprint is the image of the surface of the skin of the fingertip.
- It consists of ridges and valleys as shown in Fig. 2.
- The ridge pattern in a fingerprint can be described as an oriented texture pattern with fixed dominant spatial frequency and orientation in a local neighbourhood [2].
- Orientation - flow pattern of the ridges [2].
- Frequency - inter-ridge spacing [2].
- The anomalies in a fingerprint are called minutiae (ex: ridge endings, bifurcations, crossovers, short ridges etc. as shown in Fig. 2

Fig. 2 Bifurcations and short ridges in a fingerprint structure [11].
Fingerprint Enhancement Algorithm

• An ideal algorithm must increase the contrast between the ridges and valleys of a fingerprint for visual examination or automatic feature extraction [2].

• In this algorithm [2] during the processing of each pixel a local neighbourhood of that pixel is considered and this can be explained using Fig. 3.

• As the ridges and valleys have well-defined frequency and orientation in the local area directional filters are used [2].

• The filtering process is adaptive as the parameters of these directional filters depend on the local ridge frequency and orientation [2].
Determining minutiae based on neighbouring pixels

Fig.3 In (a) the pixel with three neighbours is a ridge bifurcation and in (b), pixel with only one neighbour is a ridge ending [15].
Steps involved in the Fingerprint Enhancement Algorithm

- **Normalization**: To obtain a pre-specified mean and variance, an input fingerprint image is normalized [2].

- **Local orientation and Frequency estimation**: The normalized input fingerprint image is used for computing orientation and frequency images [2].

- **Region mask estimation**: Each block in the normalized input fingerprint image are sorted out into a recoverable or an unrecoverable block to find a region mask estimate [2].

- **Filtering**: A bank of Gabor filters or Butterworth filters that are tuned to local ridge orientation and ridge frequency are used [2].
Flowchart of a fingerprint enhancement algorithm

Fig. 4 Flowchart of a fingerprint enhancement algorithm [4].
Normalization

• Normalization reduces the variations in grey-level values along ridges and valleys [2]
• \( I(x, y) \) denote the grey-level value at pixel \((x, y)\)
• \( M_i \) and \( V_i \) denote the estimated mean and variance of \( I \)
• \( M_0 \) and \( V_0 \) are the desired mean and variance values
• \( N_i(x, y) \) denote the normalized grey-level value at pixel \((x, y)\)

\[
N_i(x,y) = \begin{cases} 
M_0 + \sqrt{\frac{V_0 (I(x,y) - M_i)^2}{V_i}}, & \text{if } I(x,y) > M_i \\
M_0 - \sqrt{\frac{V_0 (I(x,y) - M_i)^2}{V_i}}, & \text{otherwise} 
\end{cases} \quad (1)
\]
Fingerprint after normalization

Fig. 5 The result of normalization. (a) Input image. (b) Normalized image [4].
Local Ridge Orientation

- Local ridge orientation is usually specified blockwise rather than at every pixel [4].
- Least mean square orientation estimation based on gradient is used here [4].
- Each fingerprint image is divided into equal blocks and gradients are calculated for each pixel in a block and average squared gradient for the block is calculated from this [4].
- The average gradient $\varphi$ direction and dominant local orientation $O$ [1] for the block are given by:

$$\phi(i,j) = \frac{1}{2} \tan^{-1} \left( \frac{\sum_{u=-\frac{W}{2}}^{W/2} \sum_{v=-\frac{W}{2}}^{W/2} 2g_x(u,v)g_y(u,v)}{\sum_{u=-\frac{W}{2}}^{W/2} \sum_{v=-\frac{W}{2}}^{W/2} [g_x^2(u,v) - g_y^2(u,v)]} \right)$$

(2)

$$O(i,j) = \phi(i,j) + \frac{\pi}{2}$$

(3)

- Correction for 90 degrees is necessary since the angle of gradient is perpendicular to the ridge orientation [4]. Here blocks of size $W \times W = 8 \times 8$ for orientation estimation and gradients $g_x$ and $g_y$ are used and calculated using Sobel operator [2].
Local Ridge Orientation

- Additional smoothing (Low pass filtering) is required at distorted and noisy regions [4]. It is done by converting orientation image into a continuous vector field as shown in the Fig. 8, defined as follows

\[
\Psi_x(i, j) = \cos[2 \theta(i, j)] \tag{4}
\]

\[
\Psi_y(i, j) = \sin[2 \theta(i, j)] \tag{5}
\]

- Where \(\Psi_x(i, j)\) and \(\Psi_y(i, j)\) are the x and y components of the continuous vector field respectively.

Fig. 6 A continuous vector field formed by a local orientation image with a block of size \(W \times W\) and center \(O(i, j)\).
Local Ridge Orientation

• The filter implementation [1] is given by,

\[
\Psi_x'(i,j) = \sum_{u=-W\Psi/2}^{W\Psi/2} \sum_{v=-W\Psi/2}^{W\Psi/2} L(u,v) \Psi_x(i - uW, j - vW) \quad (6)
\]

\[
\Psi_y'(i,j) = \sum_{u=-W\Psi/2}^{W\Psi/2} \sum_{v=-W\Psi/2}^{W\Psi/2} L(u,v) \Psi_y(i - uW, j - vW) \quad (7)
\]

\[
\theta'(i,j) = \frac{1}{2} \tan^{-1} \left( \frac{\Psi_y'(i,j)}{\Psi_x'(i,j)} \right) \quad (8)
\]

• where L is a 2D LPF and \( W\Psi \times W\Psi \) specifies the size of the filter \( \Psi_x'(i,j) \) and \( \Psi_y'(i,j) \) are the x and y components of the continuous vector field respectively after smoothing.
Local Ridge Frequency

• Local ridge frequency is found by projecting the grey values of all the pixels located in each block along a direction orthogonal to the local orientation. 1D wave with the local extrema corresponding to the ridges and valleys of the fingerprint [4];

• Let $K(i, j)$ be the average number of pixels between two consecutive peaks in the 1D wave generated above. The frequency $\omega(i, j)$ [4] is computed as

$$\omega(i, j) = 1/K(i, j) \quad (4)$$

• In order to explain the above estimation a one dimensional (1D) modeled fingerprint image instead of the original raw fingerprint images can be used.

• A finite rectangular wave (as seen in Fig. 7) which is regarded as the simplification of the projection of all grey values of the pixels in a direction, normal to the local orientation of the block with local extrema corresponding to the ridges and valleys of the fingerprint.
Finite rectangular wave as a modeled fingerprint

Fig. 7 Finite rectangular wave as a modeled fingerprint [15].
Directional Filtering

• An ideal model of band pass directional filter [1] in Fourier domain can be expressed using polar coordinates $(\rho, \phi)$ as

\[ H(\rho, \phi) = H_r(\rho)H_a(\phi). \quad (9) \]

• $H_r(\rho)$ depends on local ridge spacing and $H_a(\phi)$ depends on local ridge orientation [3].

• Instead of applying appropriate filter for each pixel, we apply a finite number of predefined filters (regarding to finite number of discrete orientations, and fixed frequency) [3].

• The degradation of the filter image and number of filters must be small [3] and it can be obtained in following way:

1. Elimination of filter dependence of local ridge frequency; either an average ridge frequency is used, or a constant is set empirically for entire database set. By doing so the context of the filter is determined only by the orientation [3].

2. By discretization of orientation values to fix number (8 or 16) we can obtain a small number of directional filter components [3].
Directional Filter in Fourier Domain

Fig. 8 Filter in Fourier domain (a) band pass (radial) component, (b) directional (angular) component, (c) combination of previous two [1].
Filtering of input image

Filtering [3] an input fingerprint image \( q \) is performed as follow:

- The 2D-FFT \( F \) of input fingerprint image \( q \) is computed,

\[
F(u, v) = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} q(x, y) \exp[-j2\pi(ux + vy)] \, dx \, dy \quad (10)
\]

- Each directional filter \( P_i \) is point-by-point multiplied by \( F \), obtaining \( n \) filtered image transforms \( PF_i, i = 1, \ldots, n \).

- Inverse FFT is computed for each \( PF_i \) resulting in \( n \) filtered images \( pf_i, i = 1, \ldots, n \) (spatial domain) [3].

\[
pf_i(x, y) = \int_{u=-\infty}^{\infty} \int_{v=-\infty}^{\infty} PF_i(u, v) \exp[j2\pi(ux + vy)] \, du \, dv \quad (11)
\]

- The enhanced image is obtain in following manner: all pixels in one block of enhanced image take the value of pixels on the same position from the filtered image which emphasizes determined orientation for corresponding block [3].
Block Diagram of a Fingerprint Enhancement Algorithm

Fig. 9 Block diagram of a fingerprint enhancement algorithm [12].
Butterworth Filter

- The band pass Butterworth filter [3] for radial component $H_r(\rho)$ of order $k$ (usually $k = 2$), having centre frequency $\rho_0$ and bandwidth $\rho_{BW}$ [3]

$$H_r(\rho) = \sqrt{\frac{(\rho \rho_{BW})^{2k}}{(\rho \rho_{BW})^{2k} + (\rho^2 - \rho_0^2)^{2k}}}$$  \hspace{1cm} (12)

and the directional component is given by (7)

$$H_d(\phi) = \begin{cases} \cos^2 \frac{\pi(\phi - \phi_c)}{2\phi_{BW}} & \text{if } \phi < \phi_{BW} \\ 0 & \text{Otherwise} \end{cases}$$  \hspace{1cm} (13)

- Where $\phi_{BW}$ is the angular bandwidth, and $\phi_c$ is the orientation of the filter [3].
Frequency response of Butterworth Filter

Fig.10  Butterworth bandpass frequency response
Gabor Filter

• Gabor filters are very useful both in frequency and spatial domain, due to their frequency-selective and orientation-selective properties [4].

• By simple adjustment of mutually independent parameters, Gabor filters can be configured for different shapes, orientations, different width of band pass and different central frequencies [4, 6].

• An even Symmetric Gabor filter general form [4] in the spatial domain [1] is given by

\[
h(x, y, \phi, \xi) = \exp \left[ -0.5 \left( \frac{x}{\sigma_x} \right)^2 + \left( \frac{y}{\sigma_y} \right)^2 \right] \cos[2\pi \xi x_\phi]
\]  

(14)

\[x_\phi = x \cos \phi + y \sin \phi \] 

(15)

\[y_\phi = -x \sin \phi + y \cos \phi \] 

(16)
• $\phi$ is the orientation of the Gabor filter, $f$ is the frequency of the sinusoidal plane wave along the $x$-axis, $\sigma_x$ and $\sigma_y$ are the standard deviations of the Gaussian envelope along the $x$ and $y$ axes, respectively.

• The modulation transfer function (MTF) [4] of the Gabor filter can be represented as,

$$H(u, v, \phi, f) = 2\pi \sigma_x \sigma_y \exp \left\{ -0.5 \left( \frac{(u_\phi - u_0)^2}{\sigma_\phi^2} + \frac{(v_\phi - v_0)^2}{\sigma_v^2} \right) \right\} + 2\pi \sigma_x \sigma_y \exp \left\{ -0.5 \left( \frac{(u_\phi + u_0)^2}{\sigma_\phi^2} + \frac{(v_\phi + v_0)^2}{\sigma_v^2} \right) \right\}$$  \hspace{1cm} (17)

$$u_\phi = u \cos \phi + v \sin \phi$$  \hspace{1cm} (18)

$$v_\phi = -u \sin \phi + v \cos \phi$$  \hspace{1cm} (19)

$$u_0 = \frac{2\pi \cos \phi}{f}$$  \hspace{1cm} (20)

$$v_0 = \frac{2\pi \sin \phi}{f}$$  \hspace{1cm} (21)

the filter is more immune to noise, if $\sigma_x$ and $\sigma_y$ are significantly large, but is more likely to create unauthentic ridges and valleys. The filter is not effective in removing the noise, if standard deviations are too small.
An even symmetric Gabor filter and its MTF

Fig. 11 An even-symmetric Gabor filter. (a) The Gabor filter with $f = 10$ and $\varphi = 0$. (b) The corresponding MTF [4].
Scope of the Project

- The objective of this project is to apply the algorithm to smudges and corrupted fingerprints to obtain enhanced images. This is done by adaptive directional filtering in the frequency domain by using Butterworth and Gabor filter for fingerprint image enhancement and also removing noise.

- MATLAB is used to normalize the corrupted fingerprints. Then the frequency and ridge orientation are computed for each fingerprint image. After that the image is filtered using directional filters. Here Butterworth and Gabor filters are used to obtain an enhanced image. Then the quality of the images obtained from both filters are compared visually.

- Fingerprint identification is done based on the filtered enhanced fingerprint image by detecting corepoint and extracting minutiae from it. This extracted corepoint is compared with corepoints of the fingerprints in the database [21].
References


Project Proposal- Fingerprint Enhancement and identification by Adaptive Directional Filtering
References (contd..)

References (contd..)


Thank you