

지문인식에서의 Gabor-filter를 사용한 Feature추출과 Matching 기법

박준범*, 송명철**, 김영구*, 고한석*

*고려대학교 전자공학과,

**고려대학교 영상정보처리학과

The fingerprint feature extraction and matching method using Gabor-filter in the fingerprint - recognition

Jun-bum Park, Myoungchul Song, Youngku Kim, HanSeok Ko

*Dept. of Electronical Engineering, Korea University

**Dept. of Visual Information Processing, Korea University

E-mail : jbpark@ispl.korea.ac.kr

Abstract

본 논문에서는, 지문인식에 있어서 특징 추출알고리즘과 추출된 특징을 가지고, matching하는 알고리즘을 제안하였다. 지문인식에 필요한 특징추출 알고리즘들은 Gabor-filter라는 알고리즘에 기반을 두었으며, minutiae와는 달리 특징추출에 있어서 전처리과정(smoothing, binarization, thinning, restoration)을 필요로 하지않는다. 또한, 지문의 matching에 있어서의 알고리즘은 fingerprint들 간의 유사성에 기반을 두었다. 이를 통한 실험결과로써, 인식의 정확성은 95.7(%), FAR(2.9%), FRR(1.4%)을 보여주었다.

1. 서론

복잡 다양한 현대사회에서 조직관리 및 기업에서 정보 및 주요 시설물의 보안체제를 구축하기 위해 지금까지는 기계적 또는 전자적인 장치 등으로 운영되어 왔으나 열쇠, ID Card, Password등을 이용한 보안 시스템은 분실, 도난, 망각, 위조, 변조 등의 위험이 상존하고 있어 근본적인 보안성에 있어서 안전하지 못하다. 이에 사람의 지문은 불변성과 유일성을 가지기 때문에, 지문은 인터넷 인증, 접근 제어, 범죄자 조사 등의 여러 분야에 널리 적용 되어왔다. 지문에서, 융선(ridge)과 골(valley)구

조는 global과 local정보를 제공한다. 그러므로, 지문인식에서 대부분의 연구자들은 지문의 local 융선 방향성을 이용하고, 융선의 끝점과 분기점을 특징으로하는 minutiae기반을 이용한다. 하지만, minutiae기반의 방법은 잡음과 변형에 매우 민감하다. 예를 들어, 외부의 환경 요인에 의해서 융선의 끝점과 분기점이 희미하게 나타날 수 있고, 겹쳐 보이기도 하다. 게다가, 지문을 찍을 때의 강도에 따라서도 융선의 끝점과 분기점이 사라질 수 있다. 즉, minutiae기반의 알고리즘은 입력 이미지의 질에 크게 의존하게 된다. 그러나, Gabor-filter기반은 gray-level 지문 이미지로부터 직접 Feature를 추출할 수 있다. Gabor-filter에 있어서의 가장 큰 특성은 주파수와 방향성을 갖는다는 것이다. 본 논문은 다음과 같이 구성되어 있다. 우선, 지문인식까지의 전체적인 흐름도를 보이고, 2장에서는 Gabor-filter를 이용하여 Feature추출하는 방법을, 3장에서는 추출된 Feature를 이용하여 하나의 fingerprint화하는 방법을 제시한다. 4장에서는 fingerprint를 가지고서 Matching하는 방법을 기존의 Matching의 문제점과 비교하면서 제시한다. 5장에서는 이를 통한 실험결과를 보이고, 6장에서는 결론을 맺고 향후 연구 방향에 대해서 논의한다.

- I. Original image의 reference point detection.
- II. 검출된 reference point의 a region of interest를 결정
- III. the reference point를 주위로 the region of interest를 sector로 분할
- IV. Gabor filtering으로 8개의 방향들의 feature extraction.
- V. 8개의 방향features의 FingerCode를 계산
- VI. 계산된 fingercode값으로 Matching.

<Fig1-1> 지문인식과정의 블록도

위의 표에서 첫 번째 부터 세 번째 까지의 과정은 <Fig1-2>에서 보여지고, 위의 네 번째 과정인 Gabor-filtering에 적용되는 Gabor-filter는 <Fig.1-3>에 나타내었다.

2. Gabor-filter를 이용한 Feature추출

Gabor-filter를 이용해서 Feature를 추출하기전에 지문 이미지의 reference-point를 찾았다고 가정하자. 이 reference-point를 중심으로 region of interest를 sector로 분할하고, 이를normalization의 과정을 거친 이미지를 가지고서 Feature추출을 하게된다. Gabor-filter는 다음과 같이 정의가 된다. f는 sinusoidal plane wave의 주파수를 의미하고, K는 용선과 용선사이의 픽셀를 나타낸다.

$$G(x, y, \theta_k, f, \sigma_x, \sigma_y) = \exp\left[-\frac{1}{2}\left(\frac{x_{\theta_k}^2}{\sigma_x^2} + \frac{y_{\theta_k}^2}{\sigma_y^2}\right)\right] \times \exp(i2\pi f x_{\theta_k})$$

$$x_{\theta_k} = x \sin \theta_k + y \cos \theta_k \quad \sigma_x^2: \text{Standard deviation (x axes)}$$

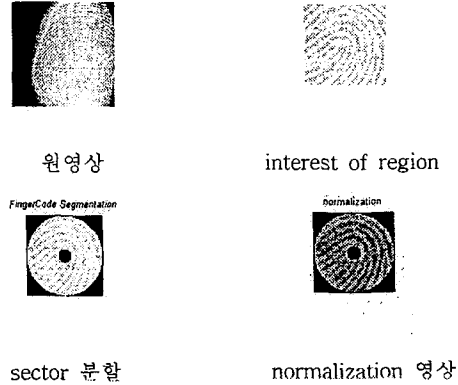
$$y_{\theta_k} = x \cos \theta_k - y \sin \theta_k \quad \sigma_y^2$$

f : Average ridge frequency (1/K)
 θ_k : Orientation of the gabor filter

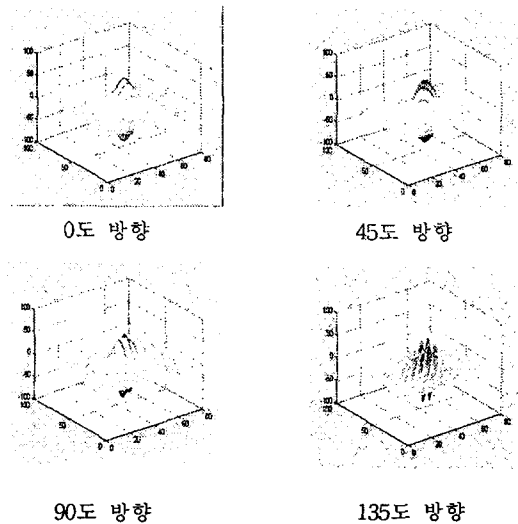
θ_k 는 Gabor-filter의 방향(즉, 여기서는 0~180도 사이의 22.5간격으로의 8개의방향성분을 나타낸다.)

$\sigma_x^2 \sigma_y^2$ 는 X,Y축으로의 각각의 Gabor-filter의 bandwidth를 나타내는 포락선(envelope)를 정의한다. 여기서, $\sigma_x \sigma_y$ 의 값은 filter설계의 사용자가 가장 적당한 값을 선택하되, 다음의 조건을 고려해야 한다. 만일, 위의 값이 임계값보다 너무크면, filter는 잡음에는 강인하지만 용선의 흐름이 smoothing되기 쉽다. 반면에, 임계값보다

너무 작으면, 잡음제거에 효율적이지 못한 단점을 가지고 있다. 그러므로, filter 설계를 할 때는 이점을 반드시 고려해서 적당한 값을 찾아야한다. 이 알고리즘에서는 각각 2와 4로 정의했다. 위의 Gabor-filter의 식을 적용해서 나온 filter의 그림과 <Fig1-3>과 같다



<Fig 1-2 지문인식 과정의 처리>



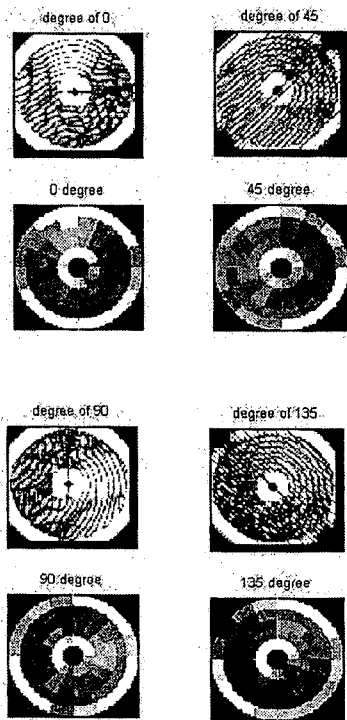
<Fig1-3 Gabor-filter>

3. Fingercode

위에서, filtering된 이미지로부터 8개 방향의 추출된 Feature를 가지고 각각의 방향의 code값을 계산한다. 각각의 방향으로의 filtering된 이미지의 finger-code를 구하는 식은 다음과 같이 정의한다.

$$V_{i\theta} = \frac{1}{n_i} (\sum_{x,y} |F_{i\theta}(x,y) - P_{i\theta}|)$$

여기서, $V_{i\theta}$ 는 각 방향의 i 번째 sector에서의 fingercode 값을 나타내고, $F_{i\theta}(x,y)$ 는 좌표(x,y)에서의 pixel의 값(즉, 각 pixel에서의 gray값)을 나타낸다. 또한, $P_{i\theta}$ 는 $F_{i\theta}(x,y)$ 에서의 pixel값들의 평균(mean)값을 나타낸다. 마지막으로 n_i 는 i 번째 sector에서의 pixel의 수를 나타낸다. 이렇게, 식에 의해 구해진 각각의 방향. 즉 8개의 각각의 방향으로부터 총 80개씩의 sector의 fingercode값을 얻게된다. 이로부터 얻은 fingercode값을 나타낸 그림은 < Fig.1-4 >과 같다. 이 그림에서 각각의 방향성분을 띠게되는 band에서의 sector부분은 밝은색(white)을 띠게되고, 그 외의 부분은 어두운색(black)을 띠게된다. 즉, 이는 각 방향성분이 추출됨을 의미한다.



< Fig.1-4 >

위의 그림에서 각각의 상위의 이미지는 filtering된 이미지를, 하위의 이미지는 fingercode된 이미지를 나타낸다. (총 8방향이지만 그림은 4방향만 표기했음)

4. Matching

이렇게, 얻은 fingercode값을 가지고서 등록되어있던 지문이미지와 인증을 요구하는 입력이미지의 Matching을 하게된다. 기존의 논문에서는 matching을 하기위해서 등록된 지문이미지와 입력되는 지문이미지의 같은 sector 사이에서의 Euclidean distance에 기반을 두고, 그 차이의 절대값의 합이 threshold보다 작으면 matching하는 알고리즘이었다. 하지만, 이 방법은, 주위환경(센서, 그 주위의 밝기)에 영향을 많이 받게된다. 그렇게되면, 등록된 이미지와 입력 이미지가 동일인의 지문이라고 해도 이미지의 gray값이 크게 작용하므로, 인식률이 떨어질 수 있다. 그래서 제시한 방법이 각 sector의 인접간의 유사성을 비교한다. 즉, 동일인의 지문이라면 등록된 이미지와 입력 이미지는 주위환경(이미지의 밝기)에 상관없이 방향성과, 인접 sector간의 유사한 흐름을 가질 것이다. 이런 성질을 이용해서 점수제(Score)를 도입한다. 즉, 유사한 정도에 따라 Score에 Weight을 두어서 차별을 둔다. 이렇게 합을 계산해서 어느 일정 임계값(threshold)보다 크면 유사도가 큰 것이므로, 동일인의 지문은 인증이 될 것이다. 이를 식으로 나타내면 다음과 같다.

$$T = \sum_{\theta=1}^8 \sum_{i=1}^{80} (|S_{r(i+1)\theta} - S_{r i \theta}| - |S_{i(i+1)\theta} - S_{i i \theta}|)$$

T : total score

θ : $0^\circ \sim 167.5^\circ$ 범위(22.5° 간격)

i : 1~80 sector

$S_{i,\theta}$: θ° 에서의 i 번째 sector

(S_r : reference image S_i : input image)

T는 각각의 8개의 방향의 80개씩의 sector에 대한 인접 sector들간의 유사성을 비교해서 reference image와 input image의 차이에 대해서 score의 weight를 두어 계산한다 이러한 Matching알고리즘 이용해서 translation뿐만아니라, rotation까지 확장을 하게된다. rotation이미지의 matching도 두 가지의 방법이 있다. 한가지는 입력 이미지가 들어왔을 때 reference이미지를 22.5도씩 회전시켜서 matching하는 방법과, 또 다른 방법은 등록된 사람의 이미지를 각각 7개의 template이미지(회전을 -67.5° 에서 67.5° 까지만 고려해서 22.5° 간격)의 rotation된 이미지를 저장시켜놓고서 입력이미지가 들어왔을 때, 그 입력이미지가 등록된 이미지와 각각을 비교하게 된다.(template matching방법) 그 비교된 값이 threshold를 넘으면 유사성을 인정하여, 동일한 지문으로 인증하게 된다. 참고로 Gabor-filter를 이용해서 얻은 fingercode는 matching에 있어서 약간의 rotation된 이미지는 그 자체로 matching이 된다.(그 rotation의 각은 대략 -11.25° 사이의 각이다.) 즉, 이사이의 각으로 rotation된 이미지는 등록된 0도의 이미지와 동일지문이라면 인증이 된다는 말이다. 여기서, 전자의 방법은 그만큼 적은 데이터를

갖는 장점을 가지고, 반면에, reference 이미지를 22.5도를 rotation할 때마다 feature를 추출해야하기 때문에 실시간 시스템에서는 최고 67.5도의 rotation된 이미지가 입력될 경우 인증 시간이 많이 걸린다는 단점을 가지고 있다. 후자의 경우는 데이터베이스에서는 많은 단점을 가지고 있지만, 실시간 시스템에서는 미리 각 사람들의 templates 이미지를 저장할 시켜놓기 때문에 인증시간은 빠르다는 장점을 지닌다. 이 논문에서는 후자의 방법을 택했다.

5. 실험결과

이 실험을 위해서, 15명으로부터 각각의 지문이미지 3장을 수집했다. 15장은 등록이미지로 나머지 30장은 테스트용 이미지로 분리했다. 등록된 15장의 이미지는 각각 7개의 template 이미지를 저장시켜놓았다. 그래서, 15명의 등록된 이미지는 총 105개의 이미지이다. 또한, 이미지의 사이즈는 256 by 256으로 한다. matching 방법은 일대일 matching에 기준으로 실험을 했다. 즉, 총 30장에서 1장을 등록하고 나머지 29장을 등록된 이미지와 비교해서 인증실험을 한다. 이러한 과정을 30번 반복한다. 덧붙여 threshold 값의 변화에 따른 인식률도 같이 실험을 했다. 하나의 threshold에서의 실험횟수는 841번이다. ($29 * 29 = 841$) 이 실험은 윈도우(OS)환경에서 MATLAB-programming을 가지고 실행되었다. 컴퓨터 simulation 결과는 다음과 같이 Table에 나타나있다. 여기서, 인식률 실험을 위해서 FAR(false acceptance rate), FRR(false reject rate), accuracy(total verification rate)로 나타낸다.

threshold	FAR(%)	FRR(%)	accuracy(%)
T=-20	5 %	1.3 %	93.7 %
T= 20	2.9 %	1.4 %	95.7 %

표 1 gabor-filter를 이용한 인식률 실험결과

	Gabor-filter기반	Minutiae기반
Feature (second)	2.10	5.35
Matching (second)	1.20	2.55
Total	3.30	7.90

표 2 gabor-filter와 minutiae 기반의 처리시간 (CPU-Time) 비교

6. 결론

이 논문에서, 지문인식 시스템은 Gabor-filter와 이를 이용해서 얻은 feature(주파수성분과 방향성분)를 기반으로 빠르고 강력한 시스템으로 개발되었다. 또한, Gabor-filter를 적용함으로써, 전처리과정(smoothing, binarization, thinning, restoration)을 피할 수 있었다.

7. 참고문헌

- [1] Anil K.Jain, Salil Prabhakar, Lin Hong " A multichannel approach to fingerprint classification" IEEE transactions on pattern analysis and machine intelligence, vol.21.no.4, april 1999
- [2] Anil K.Jain, Salil Prabhakar, Lin Hong "Filterbank-based fingerprint matching" IEEE transactions on image processing, vol.9, no.5, may 2000
- [3] Woo Kyu Lee, Jae Ho Chung, "Automatic real-time identification of fingerprint images using wavelets and gradient of gaussian" IEEE 1997
- [4] Chin-Jen Lee, Sheng-De Wang, "A gabor filter-based approach to fingerprint recognition", IEEE 1999