

방향정보와 특징점을 사용한 지문영상의 형상 분류에 관한 연구

권용재, 박영태

경희대학교 전자공학과 컴퓨터비전 연구실

경기도 용인시 기흥읍 서천리 1번지

Tel: 0331-201-2582, Fax: 0331-204-4968, E-mail: ytpark@nms.kyunghee.ac.kr

A Study on Fingerprint Classification Using Directional Information and Singular Points

Yongjae Kweon, Youngtae Park

Dept. of Electronic Eng. Kyunghee Univ

Abstract

In this paper, a fingerprint classification algorithm is presented. Fingerprint types are classified into five categories: arch, tented arch, left loop, right loop and whorl. Singular points (cores and deltas) are detected using Poincare index on the directional image smoothed by adaptive window size. The method is shown to be robust to the variation of fingerprint image quality.

1. 서론

모든 사람은 고유한 지문을 가지고 있기 때문에, 지문 인식은 신원 확인에서 가장 믿을 만한 방법중의 하나이다. 지문인식은 보통 두 개의 다른 단계에서 수행된다. 개략 단계에서 지문은 형태에 따라 5개의 주부류 (arch, tented arch, right roop, left roop, whorl)로 분류된다. 세부 단계에서는 지문으로부터 ridge endings와 branching points(minutiae)를 추출하여 세선점의 비교에 의해 두 지문 사이의 유사성을 결정한다. 이와 같은 2단계에 의한 지문 분류기법은 범패수사 목적의 대규모 조회 작업을 효율적으로 수행할 수 있다. 개략 분류에서는 지문을 유일하게 확인할 수는

없어도, 두 지문이 일치하지 않을 때 결정하는데 도움을 주고, 지문10개를 모두 사용할 경우, 10개 지문의 개략 단계에서의 분류는 데이터베이스를 대폭 감소시켜 세부단계의 정합 과정을 신속히 구현할 수 있다.

지문분류의 기법으로서 Srinivasan 기법[3]은 특징점을 위치시키는 방법을 제외하면 본 논문에서 구현한 기법과 유사하며, Blue와 Candela[2]는 지문영상을 분류하는데 신경망을 사용했다. 지문10개를 모두 사용하여 대규모의 데이터베이스 검색 시스템을 실용화하기 위해선 다양한 지문영상의 형태에 매우 강건한 형상 분류 알고리즘을 필요로 한다. 따라서 본 논문에서는 기존의 알고리즘 중 강건성이 높은 것으로 알려진 Karu 기법을 구현하고 대규모의 데이터베이스에 적용하여 문제점을 파악하고 보완하는 데 있다.

본 논문의 지문영상 분류 알고리즘은 1) 용선의 방향 정보 추출 및 평활화 과정, 2) Poincare 지수에 의한 Singular point 검출, 3) Singular point의 개수와 기하학적 위치 정보에 의한 지문 형상분류 등과 같은 세단계로 이루어진다. 2장에서는 방향영상을 구하는 세부 알고리즘을 기술하고 3장에서는 특이점 검출기법과 이를 이용한 지문의 형상 분류 알고리즘을 기술한다. 실험 결과에 대한 고찰과 개선 방향을 4장에서 제시한다.

2. 방향영상 생성기법

입력 영상의 각 화소에서의 초기 용선 방향은 그림 2의 마스크를 사용하여 결정한다. 마스크의 8방향에서의 화소의 slit sum은 각 방향에 존재하는 화소들의 합으로 계산된다. 예를 들어 1 방향의 slit sum s_1 은 다음과 같이 계산된다.

$$s_1 = I(i-2, j-4) + I(i-1, j-2) + I(i+1, j+2) + I(i+2, j+4)$$

slit sum의 최소값을 S_p , 최대값을 S_q 라 했을 때 중심화소 C의 방향은 다음과 같이 결정된다.

$$d = \begin{cases} p & \text{if } \left| c - \frac{s_p}{4} \right| < \left| c - \frac{s_q}{4} \right| \\ q & \text{otherwise} \end{cases}$$

이러한 방향 정보는 영상의 국부 정보만을 사용하기 때문에 많은 잡음을 포함하고 있으므로 평활화 과정을 필요로 한다. 각 화소의 방향정보로부터 방향각 α ($\alpha \in [0, 180)$)를 계산하고, 이 방향에서의 단위벡터 v 는 다음과 같이 계산된다.

$$v = (\cos 2\alpha, \sin 2\alpha)$$

벡터의 두 개 성분 값을 주어진 평활화 윈도우 내의 모든 벡터에 대한 평균값으로 대체하면 평활화된 방향각도는

$$\frac{1}{2} \arctan\left(\frac{y}{x}\right)$$

로 주어진다. 또한 평균 벡터의 길이는 계산된 방향각도의 신뢰도의 척도로 사용될 수 있다.

여기서는 128×128 의 벡터 영상을 사용했고, 영상의 특성에 따라 적절한 평활화 윈도우의 크기가 달라질 수 있으므로 특이점의 수가 변하지 않을 때까지 윈도우의 크기를 증가시키면서 자동으로 특이점을 검출한다. 즉, 초기에 5×5 의 윈도우로 평활화 하여 특이점을 검출하고 실패하면 윈도우 크기를 증가시켜 위의 과정을 되풀이한다. 윈도우 크기를 5×5 , 7×7 , 9×9 등으로 증가시킬 경우의 방향영상의 예를 그림 3에 나타냈다.

3. 지문 분류기법

방향 벡터의 128×128 영상을 분류하기 위해서 먼저 특이점(core, delta)을 찾고 그 수와 위치에 의해 영상을 분류한다. 특이점은 방향 영상에서 점 주위의 작은 폐쇄 곡선을 따라 Poincare 지수를 계산함으로써 보통점, core, delta 점으로 분류된다. 128×128 영상의 (i, j) 화소에서의 Poincare 지수는 다음과 같은 방향으로 계산된다.

$$(i, j) \rightarrow (i+1, j) \rightarrow (i+1, j+1) \rightarrow (i, j+1) \rightarrow (i, j)$$

방향 영상의 곡선주위에 시계반대 방향으로 회전할 때, 각의 변화가 0° 이면 보통점, 각의 변화가

180° 이면 core, 각의 변화가 -180° 이면 delta로 분류한다.(그림 4 참조) 모든 core 와 delta 점을 검출한 후, 이 점들의 수와 위치에 따라 지문영상을 분류한다. arch 지문영상에서는 core 와 delta 점이 없으며 loop와 tented arch 지문영상에서는 1개의 core 와 1개의 delta를 포함하고, whorl 지문영상은 두 개의 core 와 두 개의 delta를 포함한다. 여기서 tented arch와 loop는 core 와 delta 점을 연결한 직선 부근의 방향정보를 사용하여 분류할 수 있다. tented arch 영상에서는 이 직선의 방향은 지역 방향 벡터를 따라가는 반면, loop 영상에서는 직선이 지역 방향선을 가로질러 교차한다. β 를 core와 delta를 연결하는 직선의 기울기로 정하고, $\alpha_1, \alpha_2, \dots, \alpha_n$ 을 이 직선 부분의 지역 방향각으로 놓자. 각 지역 방향각이 β 와 유사할 경우 그 영상은 tented arch이고, 그렇지 않으면 loop 영상이다.

left loop와 right loop는 다음과 같이 구별할 수 있다. core 점을 출발하여 방향벡터를 따라 움직였을 때, delta 점은 left loop 영상에서는 오른쪽에 right loop 영상에서는 왼쪽에 위치한다. 즉, C를 core, D를 delta 점으로 놓고, C를 출발하여 방향벡터 방향의 경계점을 B라 했을 때

$$(B_r - C_r)(D_c - C_c) - (B_c - C_c)(D_r - C_r)$$

의 값이 0보다 크면 right roop로 분류되고 그렇지 않으면, left roop로 분류한다.(첨자 r은 행 좌표이고, c는 열 좌표이다.)(그림 5 참조)

4. 실험 결과 및 고찰

본 논문에서는 지문을 다섯 가지 형태, 즉 arch, tented arch, left loop, right loop, whorl로 나누고, 이를 분류하는 실험을 하였다. 데이터베이스로 사용한 NIST-4 는 4000개의 지문영상으로 구성되어 있으며 전체 데이터베이스에 대한 실험은 현재 진행 중이므로 정량적인 결과보다는 부분적인 실험 결과와 알고리즘의 문제점을 다음과 같이 제시한다. 심하게 회전하거나 압착하여 날인 된 지문영상, 혹은 매우 조악한 지문영상을 제외하면 대부분의 영상에 대해 매우 강건한 분류 특성을 보이고 있다. 그림 6은 5가지 형태의 대표적인 지문영상의 분류 결과를 보인다. 영상의 특성에 따라 평활화 윈도우의 크기가 자동으로 조정되는 과정을 그림 7에 보인다. 윈도우의 크기가 증가할수록 특이점의 수와 위치가 빠르게 수

렴함을 알 수 있다. 지문형상 분류의 오류는 그림 8과 같이 주로 압착 날인, 혹은 주름이 있는 부분에서 검출된 특이점으로 인한 오류이며 이를 개선하기 위해선 용선방향의 국부정보 외에 용선의 Global 정보를 활용해야 한다. 지문 영상 분류의 속도는 200MHz PC에서 평균 2.5초로서 대부분의 시간이 입력영상의 각 화소에서의 방향을 계산하고 평활화하는 데 소요되었다.

본 연구에서 사용한 지문형상 분류 알고리즘은 알고리즘의 강건성의 측면에서 대규모의 지문영상 데이터베이스의 검색 기법으로 활용이 가능하지만 조악한 지문영상에 대한 분류 능력의 향상을 위해서는 1) 용선의 Global 정보의 활용 혹은 방향정보의 신뢰도를 사용한 오류 특이점의 제거 기법, 2) 방향성 Fourier 필터를 사용한 지문형상의 보완 기법, 3) 초기 방향정보 추출에 있어 윈도우 크기를 확대함으로써 방향정보의 신뢰도 향상 기법 등의 도입에 의한 알고리즘의 보완을 필요로 한다.

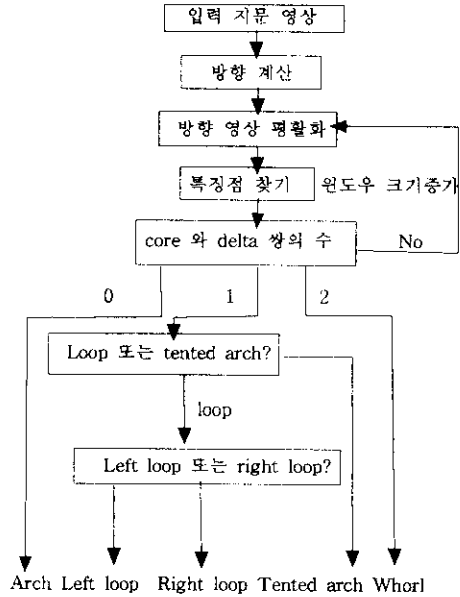


그림 1 .지문 분류 알고리즘의 블록 선도.

5. 참고 문헌

[1] M. Kawagoe and A. Tojo, "Fingerprint Pattern Classification", Pattern Recognition 17(3), 295-303, (1984)

[2] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellapa and C. L. Wilson, "Evaluation of pattern classifiers for fingerprint and OCR applications", Pattern Recognition 27(4), 485-501, (1994).

[3] V. S. Srinivasan and N. N. Murthy, "Detection of singular points in fingerprint images", Pattern Recognition 25(4), 139-153, (1992)

[4] K. Karu and A. K. Jain, "Fingerprint Classification", Pattern Recognition 29(1), 389-404, (1996)

[5] R. M. Stock and C. W. Swonger, "Development and evaluation of a reader of fingerprint minutiae", Cornell Aeronautical Laboratory, Technical Report CAL, No. XM-2478-X-1:13-17, (1969)

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | 5 | 4 | 3 | 2 | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 0 | 0 | | | | 1 | 0 |
| | 1 | | | | 7 | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 3 | 4 | 5 | 6 | | |

그림 2. slit sum을 계산하기 위한 9×9 마스크

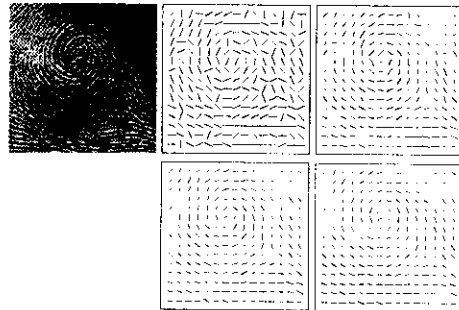


그림 3. 윈도우 크기에 따른 방향영상

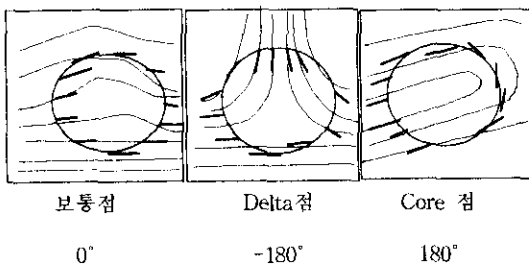
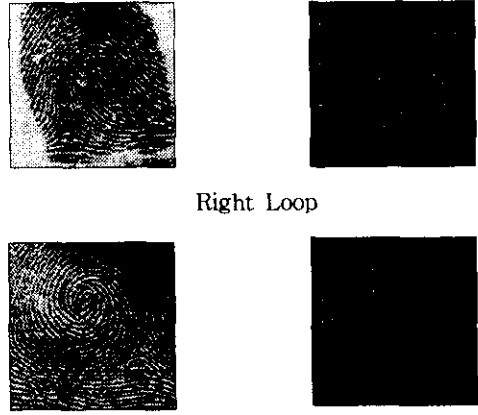


그림 4. Poincaré 지수의 계산과 보통,core,delta 점의 정의.



Right Loop

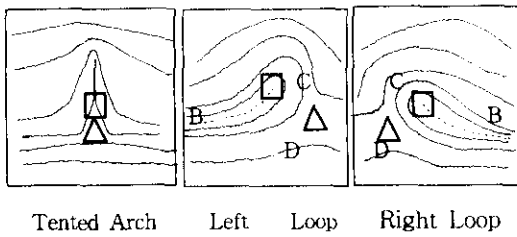
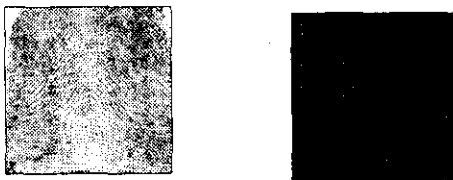


그림 5. 다른 부류에 속하는 지문 영상에서의 core(□)와 delta(△)



Arch



Tented Arch



Left Loop

그림 6. 검출된 core(□)와 delta(△)의 예

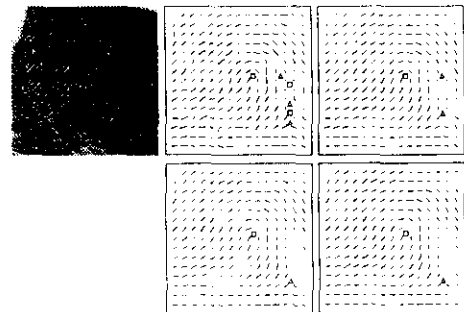


그림 7. 윈도우 크기에 따른 특징점 검출

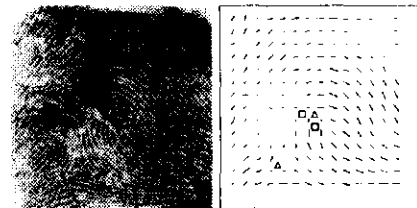


그림 8. 특이점 검출의 오류