

마스크 블록을 이용한 지문영상의 개선된 중심점 검출

김성대*, 정순호**

*부경대학교 전자계산학과

**부경대학교 컴퓨터멀티미디어 공학부

e-mail: sdkim@aisol.pknu.ac.kr

Improved Core Point Detection of Fingerprint Using Mask Block

Sung-Dae Kim*, Soon-Ho Jung**

*Dept. of Computer Science, Pukyong National University

**Faculty. of Computer and Multimedia Engineering, Pukyong National University

요 약

본 논문은 지문인식물에 있어서 중요한 요소인 중심점(core point) 검출에 대하여 기존의 Poincare 지수를 이용하는 방법과 Sine을 취하는 방법의 결점을 해결하기 위해 마스크 블록을 이용하여 중심점을 검출하는 방법을 제안하였다. 이에 대한 실험결과는 기존의 방법보다 빠르면서 검출 일관성에서도 좀더 나은 결과를 나타내었고 Arch형 지문의 중심점 검출에 있어서도 기존 방법들의 오류를 줄일 수 있었다.

1. 서론

생체인식(biometric)을 이용한 기술 중에서 가장 보편화 되고 있는 지문인식은 21세기 정보화 시대에 있어서 전자 상거래와 도어 락(door lock) 등의 인증 수단으로 많이 상용화 되고 있다. 하지만 많은 지문 인식 시스템에서의 가장 큰 어려움 중의 하나는 여러 가지 지문 종류들에 있어서 중심점(core point)을 일관되게 찾기가 어렵다는 것이다. 중심점을 일관되게 찾는 것이 중요한 이유는 특징점(minutiae) 기반의 지문 인식 시스템의 경우엔 특징점 간의 상대적 거리 및 위치의 중심점이 필요하며, 최근 제안된 Gabor filter 기반의 지문 인식 시스템처럼 특징점 기반이 아닌 지문 인식 시스템에 있어서도 매칭시(matching) 지문의 중심점이 필요하기 때문이다[1].

일관된 중심점 검출이 되지 않았을 경우에는 지문 영상의 회전이나 상하 좌우 이동의 보정, 지문 매칭시 나타나는 중심점과 특징점, 특징점과 특징점 간의 정확한 거리 측정에 직접적인 영향을 미친다. 이러한 중심점 검출에서의 오류는 전체 지문 인식

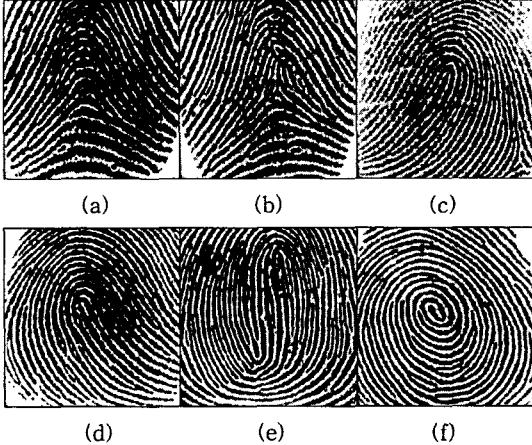
시스템의 성능저하에 큰 요인이 되므로 지문 영상의 중심점(core point)을 일관되게 찾는 것은 지문 인식 시스템에 있어서 중요한 과제중의 하나이다[2].

2. 관련 연구

2.1 지문의 종류

지문의 종류는 일반적으로 (그림 1)과 같이 6가지의 종류로 나누어 질 수 있는데 지문영상의 분류 기법은 크게 구문론적인 방법(syntactic), 구조적인 방법(structure-based), 특이점에 기반 한 방법(singularities) 그리고 주파수 영역에서의 접근방법(frequency-based) 등으로 나눌 수 있다[3]. 구문론적인 방법은 형식적인 문법(formal grammar)을 이용하여 지문을 분류하는 방법이고, 구조적인 방법은 일반적으로 추정된 방향 패턴을 이용하여 지문을 분류하는 방법으로 신경망을 이용한 방법[4]과 은닉 Markov모형을 이용한 방법[5] 등이 있다. 특이점에 기반한 방법은 지문 영상에서 중심점(core point)과 삼각주(delta point)를 찾아내고 그 특이점들의 개수

와 위치를 이용해 지문을 분류하는 방법이고, 주파수 기반 방법[6]은 지문 분류를 위해 영상의 주파수 스펙트럼을 이용하는 방법이다.



(그림 1) 지문의 종류
(a)Arch (b)Tented Arch (c)Left loop
(d)Right loop (e)Twin loop (f)Whorl

일반적으로 중심점의 개수는 (그림 1)과 같이 6개의 종류 중 arch형은 0개, tented arch형과 left loop형, right loop형은 1개, twin loop형과 whorl형이 2개의 중심점을 갖는다[3]. Arch형을 제외한 5가지 종류의 지문은 상위 중심점이 모두 존재하지만 하위(lower) 중심점의 경우는 tented arch형과 left loop, right loop형은 존재하지 않고, twin loop형과 whorl형은 하위(lower) 중심점이 존재 하지만 가장 자리에 나타나거나 누락될 가능성이 있기 때문에 본 논문에서는 상위(upper) 중심점을 찾는 알고리즘에 대하여 논할 것이다.

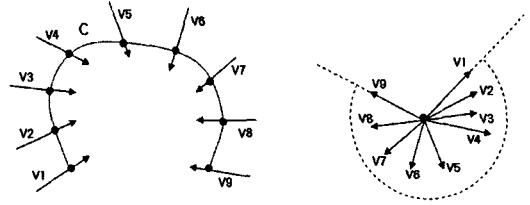
2.2 기존의 중심점(Core Point) 검출 방법

중심점을 찾는 기존의 방법 중 대표적인 두 가지 방법으로는 Poincare method index를 이용한 방법과 용선 방향에 Sine map을 취하는 방법이 있다[1].

2.2.1 Poincare method index

Poincare 지수는 (그림 2)의 곡선 C에 대한 벡터 V의 회전량으로 정의되며 지문의 방향 패턴에서 Poincare 지수를 구하려면, 한 점을 중심으로 원을 구성하여 각 영역 사이의 방향 차의 합을 구하면 되는데 계산된 Poincare 지수의 형태별 판단 기준은

180도이면 loop 형태의 중심점이고 360도이면 whorl 형태의 중심점, -180도 이면 삼각주의 위치를 나타내고 0도가 되면 특이점이 아니라고 판단 한다[7].



(그림 2) Poincare index의 정의

2.2.2 Sine map

용선 방향의 sine 값을 이용하는 방법의 순서로는 첫째, 지문을 x*y 부 블록으로 나누어 방향을 구한다. 둘째, 구해진 블록의 대표방향에 sine을 취한다. 셋째, 전체 영상에 sine값의 누적된 차가 최대인 위치를 찾는다. 넷째, 블록의 크기를 줄여가며 위의 방법을 반복하면서 개략적인 위치에서 최종 중심점을 찾는 과정을 수행한다[8].

3. 제안된 중심점(Core Point) 검출 방법

앞선 두 가지의 중심점 검출 방법 중 Poincare 지수를 이용하는 방법의 경우에는 Arch형 지문에서 중심점을 찾을 수 없고, Sine 값을 사용하는 방법의 경우에는 잡음에는 유리하나 블록의 크기를 줄여가며 탐색과정을 반복하므로 수행시간이 길어서 실시간 지문 인식에 적용하기는 부적합 하므로 본 논문에서는 시간 절약과 함께 중심점을 강건하게 찾을 수 있는 방법을 제안하고자 한다.

지문영상은 중심점 이외의 영역에서는 전체적으로 완만한 용선의 흐름을 보이지만 중심점 부근 영역에서는 급격한 변화를 가진다. 또한, 중심점 부근에서는 각도의 변화가 심한 반면 나머지 부분에서는 완만한 곡선이 나타나게 되는데 이러한 특성을 이용하여 상위 용선 에서부터 제안된 방법으로 한 라인씩 밀어서 스캔을 하다보면 특정한 지점에서 종료조건을 만족하게 되는데 종료부분에서 상위 라인의 가장 가까운 지점을 중심점으로 검출하게 된다.

3.1 방향성분 검출

3.2절의 탐색 마스크를 적용시키기 전에 먼저 방향 성분을 구해야 하는데 본 논문에서는 여러 가지

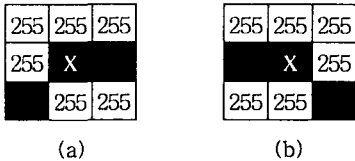
방향 성분을 구하는 방법들 중에서 기울기계산[9] 방법을 사용하였다. 방향성 검출 과정은 첫째, 원 영상에서 9*9 픽셀 크기의 부 블록으로 나눈다. 둘째, 각 픽셀 (x,y)에서의 3*3 소벨 연산자를 사용하여 기울기(gradient)를 구한다. 셋째, 9*9 픽셀에서의 중심픽셀 (x, y)의 대표 방향성을 식(1)로 구한다.

$$O(\theta, \theta) = \frac{1}{2} \frac{\sum_{\theta_0=\theta-\pi/2}^{\theta+\pi/2} \sum_{\theta_0=\theta-\pi/2}^{\theta+\pi/2} 2G_{\theta_0}(\theta, \theta)G_{\theta}(\theta, \theta)}{\sum_{\theta_0=\theta-\pi/2}^{\theta+\pi/2} \sum_{\theta_0=\theta-\pi/2}^{\theta+\pi/2} (G_{\theta_0}^2(\theta, \theta) - G_{\theta}^2(\theta, \theta))}$$

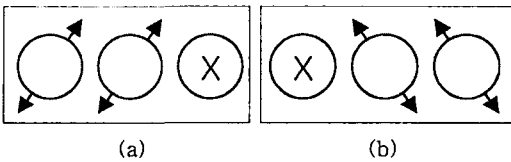
식(1)

3.2 중심점 후보 탐색 마스크

한 라인의 중심점 후보영역을 검출하기 위하여 (그림 3)과 같이 ■ 지점에 현재의 위치를 맞추어 놓고 3*3 마스크를 통과시키며 영상에서 후보영역을 검출한다.



(그림 3) 탐색 마스크
(a)왼쪽 검출 마스크 (b)오른쪽 검출 마스크

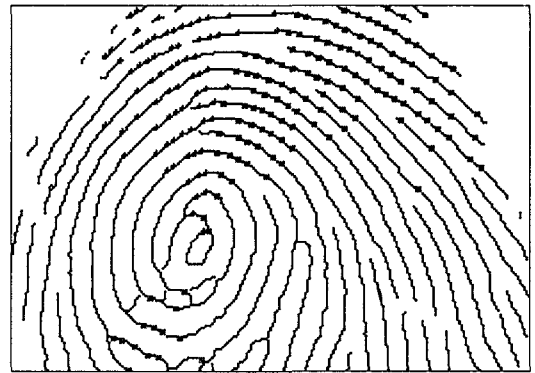


(그림 4) 후보지점의 주변 방향성분
(a)왼쪽 방향성분 (b)오른쪽 방향성분

탐색 마스크의 적용 순서는 첫째, 최 상단 지점에서 시작하되 (그림 3)(a) 왼쪽 마스크 후보 지점이 나타나면 그 좌측에 방향성분이 (그림 4)(a)의 원점 ⊗ 로부터 45도 방향 성분인지 확인하고 마킹한다. 둘째, (그림 3)(b) 오른쪽 후보 지점이 나타나면 그 우측에 방향 성분이 (그림 4)(b)의 원점 ⊗ 로부터 135도 방향 성분인지 확인하고 마킹한다. 셋째, 좌측의 가로방향 성분 중에서 가장 큰 후보지점은 우측의 가로방향 성분 중에서 가장 작은 후보지점보다 클 수 없으며 두 지점이 쌍으로 나왔는지 확인하여

쌍이 아니라면 삭제하고 쌍으로 나왔다면 두 지점을 후보 지점으로 마킹하고 다음 라인으로 넘어간다. 다음 라인부터는 위의 순서를 반복하되 모든 영역의 마스크 적용을 피하기 위해 다음과 같은 종료 조건을 만족하면 탐색 마스크 스캔을 정지한다.

첫째, 후보 쌍이 연속적으로 임계값(윤선 평균거리) 라인 이상 나오지 않을 경우 스캔을 정지 한다. 둘째, 상단의 후보지점 평균 위치보다 확연히 다른 곳으로 후보지점이 연속적으로 이동되어 나올 경우 스캔을 정지 하고 그 지점을 마킹한다. 셋째, 상위의 후보영역 넓이 보다 연속적으로 계속 넓어지면 스캔을 정지 하고 그 지점을 마킹한다.



(그림 5) 중심점 후보 탐색 결과

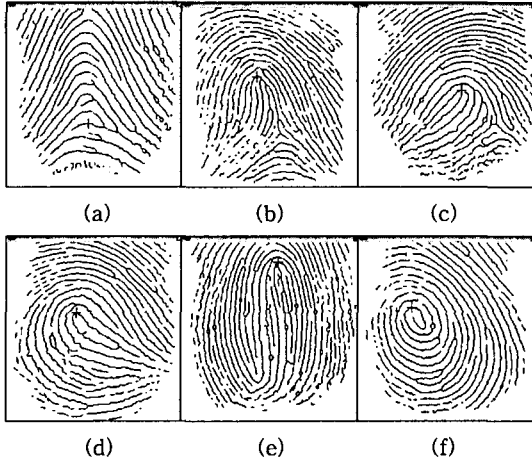
(그림 5)는 중심점 후보 탐색을 마친 결과로써 좌측상단의 '⊕'점이 (그림 3)(a)마스크와 좌측 대표 방향이 45도인 (그림 4)(a)의 조건에 맞는 것이 검출된 결과이고, 우측상단의 '⊕' 점이 (그림 3)(b)마스크와 우측 대표방향인 135도인 (그림 4)(b)의 조건에 맞는 것을 검출한 결과이다.

4. 실험 및 평가

본 논문의 실험환경은 P-IV 2.8G의 컴퓨터에서 Visual C++6.0으로 프로그램 하였고, 지문의 입력은 Safe-valley의 광학식 지문 입력기인 SV_USBF2000으로 6가지 종류별 지문을 각각 30개씩 총 180개의 지문을 입력받아 실험 하였다.

(그림 6)은 6가지 종류별 지문을 제안된 방법으로 중심점 검출을 실험한 것으로써 "⊕"부분이 중심점으로 판정되어 검출된 결과이고, <표 1>과 <표 2>는 지문 종류별로 기존의 방법과 제안된 방법의 일관성 있는 중심점 검출결과 중심점 검출 수행시간

을 나타내었다. 실험 결과 제안된 방법이 기존의 poicare index 방법보다 검출률에서 평균 9%, 검출 시간에 있어서도 평균 4[ms]가 개선되어 졌고, sine map 방법보다는 검출률에서 평균 3%, 검출 시간에 있어서도 평균 140[ms]가 개선되어 졌다.



(그림 6) 지문 종류별 중심점 검출 결과
(a)Arch (b)Tented Arch (c)Left loop
(d)Right loop (e)Twin loop (f)Whorl

<표 1> 지문 종류별 중심점 검출률 [%]

	poincare	sine	proposed
Arch	16.7	53.3	56.6
Tented Arch	100	100	100
Left loop	92.4	90.5	96.6
Right loop	92.4	93.3	96.6
Twin loop	98.8	98.8	96.6
Whorl	95.2	94.3	100
rate	82.58	88.36	91.06

<표 2> 지문 종류별 수행시간 [ms]

	poincare	sine	proposed
Arch	6.17	141.87	2.4
Tented Arch	6.6	146	2.4
Left loop	6.71	143.98	4.8
Right loop	11	154.35	5.6
Twin loop	10.4	140.23	4.8
Whorl	6.85	145.15	6.4
average	8.53	146	4.4

5. 결론

본 논문에서는 기존의 방법들보다 중심점을 빠르게 찾고, 중심점 검출에 있어서도 일관된 검출률을

얻을 수 있는 알고리즘을 제안하고 실험 하였다.

실험 결과 제안된 방법이 기존의 방법들보다 빠르고 높은 검출률을 보임을 알 수 있었다. 하지만 Arch형 지문의 경우에는 스캔을 끝까지 해도 중심점을 찾지 못하고 끝나버리는 경우가 종종 있었는데 이에 대한 보완작업이 앞으로 필요할 것으로 생각된다.

참고문헌

[1] Anil K.Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti, "Filterbank-based fingerprint matching," IEEE Trans. Image Processing, vol. 9, no. 5, pp. 849-859, May 2000.
 [2] 송영철, 박철현, 박길홍, "레이블링 방법을 이용한 지문 영상의 기준점 검출", 한국통신학회논문지, Vol.28, no. 9c, pp. 860-867, 2003.
 [3] A. K. Jain, S. Prabhakar, and L. Hond, "A Multichannel Approach to Fingerprint Classification," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 348-359, Apr. 1999.
 [4] H.V. Neto and D.L. Borges, "Fingerprint Classification with Neural Networks," Proc. IVth Brazilian Symposium on Neural Networks, pp. 66-72, 1997.
 [5] A. Senior, "A Hidden Marcov Model Fingerprint Classifier," Proc. 31st Asilomar Conf. Signals, Systems & Computers, pp. 306-310, 1997.
 [6] A. P. Fitz and R. J. Green, "Fingerprint Classification Using Hexagonal Fast Fourier Transform," Pattern Recognition, vol. 29, no. 10, pp. 1587-1597, 1996.
 [7] Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar. "Handbook of Fingerprint Recognition", 2003.
 [8] 김선주, 이동재, 김주섭, 김재희, "지문의 중심점 검출에 대한 연구", 대한전자공학회 논문지, 23권, 1호, 2000.
 [9] Lin Hong, Yifei Wan and A. K. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation", IEEE Pattern Analysis & Machine Intelligence, Vol. 20, No. 8, 1998.