

방향 패턴의 레이블링을 이용한 지문영상의 Core Point 검출

이경환, 박철현, 오상근, 박길흠
경북대학교 전자공학과

Core Point Detection using Orientation Pattern Labeling in Fingerprint

Kyung-hwan Lee, Chul-hyun Park, Sang-keun Oh, Kil-houm Park

Dep of Electronics, Kyungpook Univ.

onlooker44@palgong.knu.ac.kr

요 약

지문영상의 특이점(Singularities) 중의 하나인 Core Point는 대부분의 지문인증 시스템에서 기준점(Reference Point)으로 사용되고 있다. 또한 Core Point의 검출은 전체 지문인증 시스템의 가장 기본적인 단계로서 전체 시스템의 성능에 많은 영향을 준다. 본 논문에서는 지문 영상의 방향 패턴(Orientation Pattern)과 이의 리레이블링(Re-labeling)을 이용한 Core Point 검출 방법을 제안하고, 기존의 Poincaré Index를 이용하는 방법 및 Sine Map을 이용한 방법과 비교, 분석하였다.

1. 서 론

최근 비약적인 통신기술의 발달과 Off-Line에서 뿐 아니라 On-Line 상에서의 신용결제의 증가로 인해 보안 및 개인 인증 분야에 대한 관심이 증대되고 있다. 특히 생체인식 분야는 분실의 우려가 없고 신원 확인의 고유성으로 인해 많은 연구가 진행되고 있다. 그 중 지문은 그 고유한 특성으로 인해 오래 전부터 범죄 조사, 신원확인, 접근 제어 등 다양한 분야에서 응용되어 왔다. 최근 고성능 센서를 부착한 지문입력기의 개발로 인해 양질의 지문 영상을 얻을 수 있게 되어 이를 이용한 여러 응용분야의 연구가 활기를 띠고 있다.

일반적으로 지문 인식 시스템의 가장 큰 어려움 중 한가지는 지문 영상에 있어 기준점을 찾기가 어렵다는 것이다. 이로 인해 지문영상의 회전이나 상하좌우 이동의 보정, 매칭 시에 나타나는 특징점(Minutiae) 간의 정확한 거리를 추출하기가 어려우므로 전체 지문 인식 시스템

의 성능에 있어서 큰 저하 요인이 된다. 그러므로 지문 영상의 기준점으로 이용되는 가장 두드러진 특이점인 Core Point를 찾는 것은 지문 인식에 있어 무엇보다 중요한 과제 중 하나이다.

그러나 Core Point의 절대적 위치를 정의하기가 불가능하고 Arch 형 지문의 경우 Core Point가 존재하지 않을 수도 있다. 일반적인 지문 영상의 분류 방법에서 각 지문의 종류에 따라 Core의 개수는 Arch형이 0개, Tented Arch형과 Left Loop, Right Loop형이 1개, Twin Loop와 Whorl형이 2개의 Core를 갖는다[1][2].

본 논문에서는 매칭에서 일반적인 기준점으로 사용되는 상위(upper) Core의 검출만을 다루기로 한다. 상위 Core의 경우 Arch형을 제외한 모든 지문 유형에서 비교적 정확한 위치의 검출이 가능하지만 하위 Core의 경우에는 Twin Loop와 Whorl형을 제외한 지문 유형에서는 존재하지 않을 뿐 아니라 정확한 위치를 정의하기가 상당히 어렵다. 때문에 대부분의 지문 인증 시스템에서는 상위 Core를 기준점으로 삼는다.



그림 1. Core Point

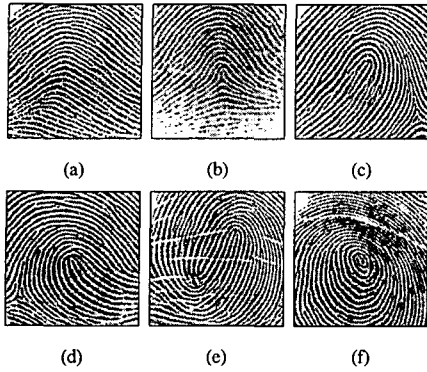


그림 2. 지문의 분류

(a)Arch (b)Tented arch (c)Left loop (d)Right loop (e)Twin loop (f)Whorl

2. 레이블링(Labeling)을 이용한 Core Point 검출 방법

일반적으로 이용되는 Core 검출방법으로는 Poincaré Index를 이용한 방법이 널리 사용되고 있으며 최근에는 저화질(poor quality)의 지문에 유용한 Sine Map을 이용한 방법이 제안되었다[2][3]. 그러나 Sine Map을 이용한 방법은 수행시간이 너무 길다는 단점이 있다. 본 논문에서는 수행시간과 정확도를 고려할 수 있는 새로운 Core 검출 방법으로 방향 패턴의 레이블링을 이용한 방법을 제안한다. 전체적인 수행과정은 그림3과 같다.

방향 패턴(Orientation Pattern)

방향 패턴은 “least mean square orientation estimation” 알고리즘을 사용하여 다음의 과정과 같이 구한다.

- 1) $M \times N$ 크기의 영상을 $w \times w$ 크기의 부 블록으로 나눈다. 본 논문에서는 5×5 픽셀 크기의 부 블록으로 나누었다.
- 2) 각 픽셀 (i, j) 에서의 그래디언트(gradient) G_x, G_y 를 계산한다. 실험에서는 3×3 Sobel operator를 사용하였다[4].
- 3) 각 블록의 중심 픽셀 (i, j) 에서의 대표 방향 $O(i, j)$ 를 계산한다[5].

$$\alpha(i, j) = \frac{1}{2} \tan^{-1} \left[\frac{\sum_{u=1-w/2}^{i+w/2} \sum_{v=1-w/2}^{j+w/2} 2G_x(i, j)G_y(i, j)}{\sum_{u=1-w/2}^{i+w/2} \sum_{v=1-w/2}^{j+w/2} (G_x^2(i, j) - G_y^2(i, j))} \right] \quad (1)$$

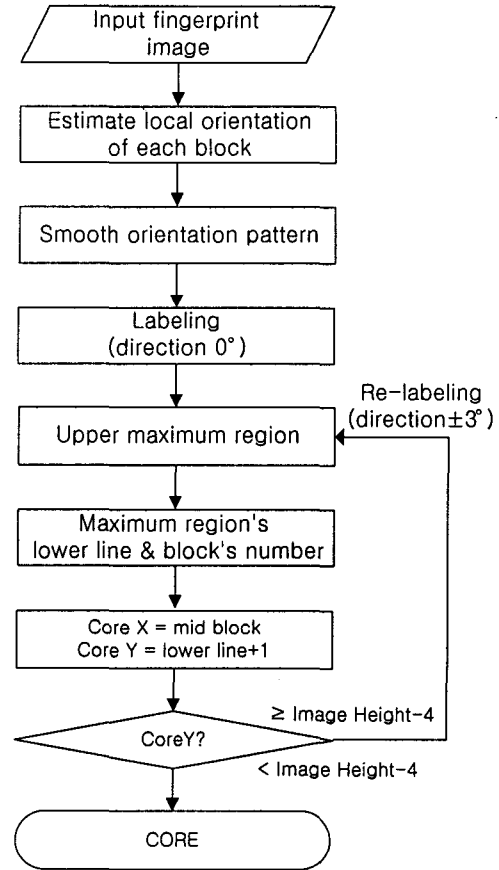


그림 3. Labeling을 이용한 Core detection 알고리즘

평활화(Smoothing)

구해진 방향 패턴은 Noise의 영향으로 연속적인 패턴의 흐름을 갖지 않는다. 이를 보정해 주기 위해서 방향 패턴을 다시 벡터 형태 즉 $\cos(2O(i, j)), \sin(2O(i, j))$ 로 변환한 뒤 각각의 성분 별로 Mean 필터를 이용하여 블록 단위로 평활화한 후 다시 방향 값으로 변환한다[3]. 실험에서는 5×5 픽셀 크기의 mean 필터를 사용하였다. 방향 패턴과 평활화 과정을 시각화한 것은 그림 4와 같다.

레이블링(Labeling)

대표 방향이 0° 인 블록들에 대해 레이블링을 수행한 후, 영상의 상반부에서 면적이 가장 큰 영역을 찾아내고 전체 영상에 대해서 해당 영역의 최하위 위치의 블록 열과 그 위치에 해당되는 블록들의 수를 계산하여 그 블록들의 중간 블록의 바로 아래 위치가 Core블록이 된다. 레이블링 과정을 시각화한 것은 그림 5와 같다.

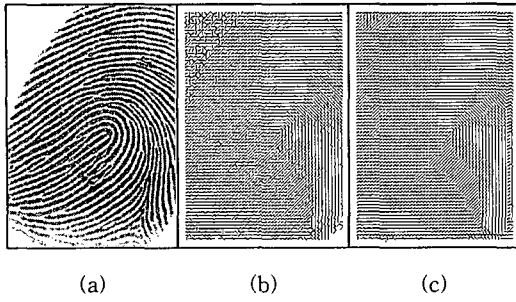


그림 4. 방향 패턴과 평활화
 (a)원영상 (b)Orientation field(8방향만 시각화) (c)
 Orientation smoothing(8방향만 시각화)

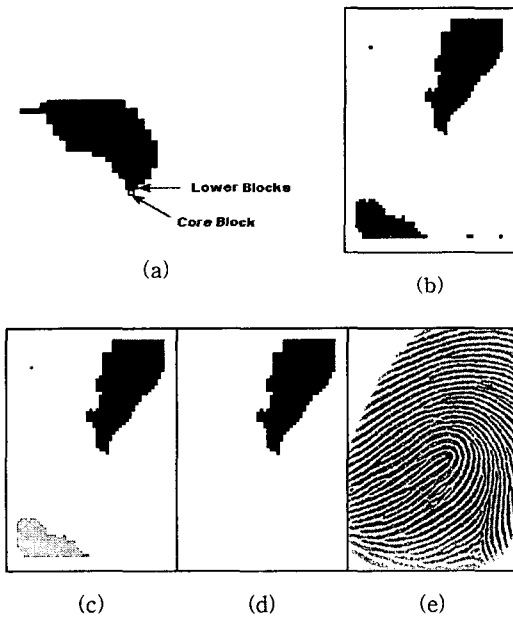


그림 5. Labeling
 (a)최하위 블록과 Core 블록의 위치(예시) (b)그림 4에
 서 대표 방향 0°인 블록 이진화 (c)이진화된 영상에
 대한 Labeling (d)상반부의 최대 Label region (e)원영
 상과 검출된 Core block의 위치

리레이블링(Re-labeling)

일부의 지문 영상의 경우 0°방향의 블록에 대해 레이블링했을 때, 영상의 상반부에서부터 하반부까지 레이블된 영역이 이어져 Core Point를 검출할 수가 없다. 이를 보정해주기 위해 블록의 대표 방향이 0°가 아닌 일정한 각도만큼 변경된 대표 방향을 기준으로 레이블링을 다시 수행한다. 실험에서는 $\pm 3^\circ$ 간격으로 리레이블링하여 Core Point를 검출하였다. 리레이블링 과정을 시각화한 것은 그림6과 같다.

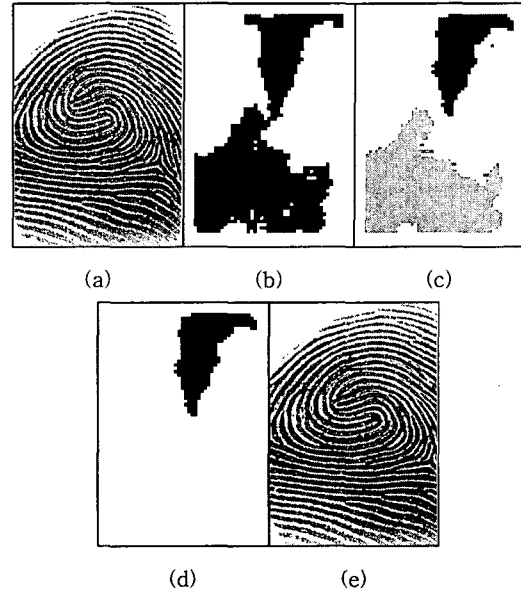


그림 6. Re-labeling
 (a)원 영상 (b)대표 방향 0°인 블록 이진화 (c)대표
 방향 -3°인 블록 Re-labeling (d)상반부 최대 Label
 Region(대표 방향 -3°인 블록) (e)원 영상과 검출된
 Core block의 위치

3. 실험결과 및 고찰

실험에서는 제안한 레이블링을 이용한 방법과 기존의 Poincaré Index를 이용하는 방법 및 Sine Map을 이용하는 방법을 적용하여 각각에 대한 수행시간과 검출률을 비교하였다. 본 실험은 Pentium III 700MHz의 CPU 속도를 가지는 IBM PC에서 수행하였으며 지문 영상의 획득은 휴노 테크놀로지사의 지문 입력기(MagicSecure 2000)를 이용하였으며 한 지문에 대해서 5장의 영상을 획득하였다. 5장의 영상은 각각 회전되지 않은 영상과 좌우로 회전 정도가 다른 2장씩의 영상으로 구성되었으며 364×256 픽셀 크기와 256 gray scale을 갖는 총 455장(91 finger \times 5 image/finger = 455 image)의 영상을 사용하였다. 실험에 사용된 지문의 분포는 표 1과 같다.

표 1. 실험 영상의 지문분포 단위(개)

	Arch	Tented arch	Left loop	Right loop	Twin loop	Whorl	Total
Image	30	5	105	105	105	105	455
Finger	6	1	21	21	21	21	91

실험결과 각 방법에 따른 수행시간, 검출률은 각각 표 2 및 표 3과 같다. 정확도의 측정은 실험자의 시각적 판단에 의한 것으로 Core Point를 찾을 수 없는 경우와 현저하게 Core Point의 위치가 잘못된 것으로 판명되는 경우에 대해 각각 오검출로 정의했으며, 또한 동일 지문에 대해 5장의 영상을 비교하여 검출된 Core의 위치 변동여부를 반영하였다. 즉, 5장의 동일지문에 대한 Core Point의 위치를 상호 비교하여 일정블록 이상 벗어난 영상에 대해서도 오검출로 정의했다. Arch형의 경우 엄밀한 의미에서의 Core Point는 존재하지 않지만 제안한 레이블링을 이용한 방법과 Poincaré Index를 이용하는 방법에서는 Core를 검출하는 방법과 동일한 방법에 의해 Core로 추정할 수 있는 블록을 검출할 수 있었다.

표 2. 수행 시간 단위(ms)

지문 형태 방법	Arch	Tented arch	Left loop	Right loop	Twin loop	Whorl	Total
Poincaré Index	6.17	6.6	6.71	11	10.4	6.85	8.53
Sine Map	141.87	146	143.98	154.35	140.23	145.15	146
Labeling	5.93	5.4	5.74	6.65	6.23	6.56	6.25

표 3. 검출률 단위(%)

지문 형태 방법	Arch	Tented arch	Left loop	Right loop	Twin loop	Whorl	Total
Poincaré Index	16.7	100	92.4	92.4	99	95.2	89.7
Sine Map	53.3	100	90.5	93.3	99	94.3	91.6
Labeling	46.7	100	95.2	96.2	99	95.2	93.2

실험결과 수행시간 및 검출률에서 제안한 방법이 세 방법들 중에서 상대적으로 우수한 것으로 나타났다. 수행시간의 경우 제안한 레이블링을 이용한 방법과 Poincaré Index를 이용하는 방법에 비해서 Sine Map을 이용한 방법이 매우 길었고 검출률에서는 Poincaré Index를 이용한 방법이 가장 낮다. Arch형을 제외한 경우 Poincaré Index를 이용한 방법이 Sine Map을 이용한 방법보다 오히려 더 우수한 것은 Poincaré Index

를 이용한 방법에서의 방향패턴 평활화의 반복횟수 및 영상의 화질과 관련이 있다.

4. 결론

본 논문에서는 기존의 Core 검출에 널리 사용되고 있는 방향 패턴에 대해 레이블링기법을 적용하여 비교적 간단하게 Core를 검출할 수 있는 새로운 방법을 제안하였다. 제안한 레이블링을 이용한 Core 검출방법은 수행시간과 검출률에서 기존의 방법보다 우수함을 확인하였다. 또한 Sine Map을 이용한 방법의 단점인 수행시간과 Arch형에서의 중심점으로 사용할 Core를 검출할 수 없는 Poincaré Index를 이용한 방법의 단점을 보완할 수 있음을 확인하였다.

향후 잡음이 있는 지문 영상에 대한 Sine Map을 이용한 방법과의 비교 연구와 실제 지문 인증 시스템에서의 적용에 대한 연구가 필요하다.

참고문헌

- [1] K. Karu and A. K. Jain, "Fingerprint Classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.
- [2] A. K. Jain, S. Prabhakar, and L. Hong, "A Multichannel Approach to Fingerprint Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348-359, Apr. 1999.
- [3] Anil K. Jain, fellow, IEEE, Salil Prabhakar, Lin Hong, and Sharath Pankanti, "Filterbank-Based Fingerprint Matching" *IEEE Trans. Image Processing*, vol. 9, NO. 5, pp. 849-859, May. 2000.
- [4] Rafael C. Gonzalez, Richard E. Woods "Digital Image Processing", Addison-Wesley Publishing Company, pp. 416-429, 1992.
- [5] A. R. Rao, A Taxonomy for Texture Description and Identification, New York, Springer-verlag, 1990.