

다해상도 방향성 정보를 이용한 지문영상의 특이점 추출

正會員 이 준 재*, 심 재 창**, 황 석 윤***, 남 재 열****, 이 주 형*****

Extraction of Singular Points of Fingerprint Image Using Multiresolution Directional Information

J. J. Lee*, J. C. Shim**, S. Y. Hwang***, J. Y. Nam****,
J. H. Lee***** *Regular Members*

※본 연구는 (주)한국정보 시스템의 연구비 지원에 의해 수행되었음.

요 약

본 논문에서는 방향성 정보를 이용한 지문영상 특이점 추출 알고리즘을 제안한다. 이 방법은 방향성 영상을 두 해상도에서 구하고 Poincare 지수를 이용해 여러 후보 특이점을 추출한 다음, 하위 해상도에서 방향성 평활화에 의해 거짓 특이점을 제거하고 이에 일치하는 상위 해상도의 특이점을 선택하는 것이다. 특히, 제안한 폐곡선을 따라 Poincare지수를 적용함으로써 하위 해상도에서 잃어버릴 가능성이 있는 특이점이 발견되기 때문에 이를 다시 해석하여 확실한 특이점을 선택한다. 또한 이의 전처리과정으로서 수행속도와 시스템의 성능을 높이기 위해 지문 영상을 분할하는 효율적인 방법이 제시된다.

ABSTRACT

We propose an algorithm for extracting singular points of fingerprint image using directional information. First, we extract the candidates of singular points using Poincare index in two(lower and higher) resolutional directional images. Then we remove the false singular points using smoothing technique from lower resolutional directional image. And finally we select the singular points in higher resolution corresponding to those in lower resolution. The

*동서대학교 컴퓨터공학과
**안동대학교 컴퓨터공학과
***대구대학교 수학과
****계명대학교 전자계산학과
***** (주)한국정보시스템
論文番號: 97100-0314
接受日字: 1997年 3月 14日

possible missing points in lower resolution are found by computing Poincare index along the proposed small curve. And the reliable points are selected from analysis around them. We also propose a method for segmentation of fingerprint as preprocessing step to enhance the computational speed and the performance of system.

I. 서 론

최근에 시스템의 안전성 확보가 대단히 중요한 문제로 부각되고 있다. 이러한 안전성을 확보하기 위한 방법들 중 하나가 지문인식을 이용한 신분확인 기법으로 인식결과에 대한 신뢰도와 안정도가 다른 수단보다 높은 것으로 평가되어 가장 효율적인 개인인증 수단으로 인정받고 있다[1]. 지문을 이용한 개인 인증 과정은 크게 지문의 유형을 구분하는 분류(classification)와 본인임을 확인하는 정합(matching)의 과정을 포함한다. 이러한 지문의 개인 인증 시스템은 응용분야에 따라 입력된 지문을 데이터베이스에 등록되어 있는 다수의 지문중에서 가장 유사한 패턴을 구별해 내는 1대 다수의 식별(identification)과 입력된 지문과 등록된 지문이 동일한지를 1대 1로 판별해내는 인증(verification)으로 나누어진다. 전자는 실종자의 검색, 범인수사등에 이용되고 후자는 출입을 제한하는 보안시스템이나 개인 금고, 자동차문의 개폐, 전자결제 시스템, 패스워드와 병행하여 컴퓨터의 접근 허용 등에 응용될 수 있다[1].

본 논문에서는 지문인증 기술 단계중에서 지문영상에 존재하는 특이점(singular point) 추출의 효율적인 알고리즘에 대해 연구하였다. 이런 특이점은 지문의 유형을 분류하는데 이용되거나 정합을 위해서도 필요하다. 지문의 유형은 크게 궁상문(arch), 초궁상문(tented arch), 좌제상문(right loop), 우제상문(left loop), 와상문(whorl)의 다섯가지로 분류하거나 이중루프(double loop)를 와상문에 포함시키지 않고 분리시켜 여섯가지로 분류한다[1-4]. 이를 분류하는 척도는 지문영상에 존재하는 특이점의 수와 방향성 흐름 패턴(directional flow pattern)에 의존한다. 특이점은 코어(core), 델타(delta) 점을 말하며 이들의 수에 의해 지문유형이 분류될 수 있다[4]. 또한 이런 점들이 정합을 위한 등록점(registration point)의 역할을 하므로 이들을 정확히 잘 찾는 것이 매우 중요하다[1-9]. Rao 등[5]은 구문론적 트리 문법(syntactic tree grammer)에 의해 이를 찾았으며, Srinivasan 등[7]은 주변의 방향성

히스토그램을 이용하여 구조적으로 특이점을 추출하는 방법을 제안하였다. Kawagoe 등[9]은 Poincare지수를 이용하여 특이점을 찾아 이들에 의해 지문을 분류하였고 그후에 Karu 등[4]이 Poincare지수에 의해 검출된 특이점 중에서 거짓 특이점을 제거하기 위하여 방향성 영상의 평활화를 반복적으로 수행하고 이를 기반으로 6가지의 지문 패턴을 분류하는데 적용하였다. 그러나 방향성 영상을 무한히 반복처리하는 것은 시간이 많이 걸리고 또한 찾은 특이점이 평활화에 의해 손실될 우려가 있다. 특이점들은 다 찾으면서 정확한 위치에서 얻어져야 한다. 현재 사용되고 있는 시스템의 대부분이 이 특이점의 위치가 데이터 구조에 포함되어 있어 시스템의 성능에 영향을 미친다. Karu 등은 방향성 영상의 블락 크기를 500DPI영상에 대해서 8×8크기의 화소로 사용하였다. 이럴 경우 특이점의 위치는 정확성을 기할 수 있으나 잡음이 있는 영상에 대해 초기에 너무 많은 특이점이 추출되어 이를 제거하기가 어렵고 시간이 많이 걸린다. 한 블락의 크기를 16×16으로 할 경우 거짓 특이점은 적으나 위치에 대한 정확성이 떨어진다. 또한 특이점을 추출 못하는 경우도 있다. 블락의 크기를 8×8로 할 경우 방향성 영상(블락방향성 영상)의 크기는 512×480영상에 대해서 64×60이되고 블락의 크기를 16×16으로 할 경우 32×30이 된다. 여기서 64×60의 크기를 고해상도 방향성 영상이라 하고 32×30을 저해상도 방향성 영상이라 부르기로 한다.

본 논문에서는 이 두해상도로 방향성 영상을 구하고 저 해상도에서 특이점을 기반으로 고 해상도에서 특이점을 선택하는 효과적인 방법을 제시한다. 중요한 것은 특이점 후보의 수를 줄이면서 잃어버리는 특이점(missing points)이 발생되지 않도록 하는 것이다. 저해상도 영상의 경우 거짓 후보점은 매우 적으나 잃어버리는 특이점이 발생할 경우가 있고 고해상도의 경우 거짓 후보점이 너무 많이 생겨 이를 제거하는데 시간이 많이 걸리며 잘 못제거될 확률이 있다. 따라서 저해상도에서 후보 특이점을 추출하여 거짓 특이점을 제거하고 이에 일치하는 점을 상위해상도에서

선택하면 된다. 그러나 문제는 잃어버리는 점이 존재하여 이에 대한 추출방법이 필요하다. 이런 점들은 Karu의 폐곡선 추적 대신 제한한 다른 폐곡선의 추적에 의해 이를 발견한다. 마지막으로 고해상도에서 특이점을 추출하고 저해상도에 일치하는 특이점을 선택하므로써 위치에 대한 정확성을 보완한다. 코아점은 델타점보다 중앙에 위치하기 때문에 지문의 화질이 좋아 더 잘 추출된다. 따라서 Karu의 방법에서 이들을 함께 추출하는 것과 달리 거짓 델타점의 제거는 코아점을 먼저 추출한 후 방향성 평활화와 기하학적인 관계를 함께 고려해 수행한다. 기하학적인 관계는 지문영상이 회전이 심하게 일어나지 않았다는 가정하에 델타점이 코아점보다는 아래에 있으며 델타점이 2개인 경우 코아를 중심으로 같은편에 존재하지 않는다는 것만 고려한다. 실제 16×16크기의 블락영상은 8×8을 만들 때 같이 같이 구해지므로 시간 소모가 거의없다. 또한 영상의 평활화로 인해 손실될 수 있는 특이점들이 폐곡선을 다르게 추적함으로써 신뢰도가 높은 특이점을 추출한다.

본 논문의 알고리즘은 다음의 단계로 구성되어 있다. 첫째, 영상의 각 블락에서 융선의 대표 방향을 구한다. 블락내의 방향정보를 구하는 과정에서 유도된 결과를 이용하여 지문영상의 전경과 배경을 분리한다. 이것은 영상처리 및 패턴인식 분야에서도 마찬가지로 초기단계에서 주로 행해지며 그 이후의 처리가 전경에서만 수행될 수 있으므로 시간절약 뿐만 아니라 시스템의 성능에도 중요한 영향을 끼친다[10-14]. 특히, 배경부분에서 무작위한 방향성분 때문에 특이점이 많이 추출된다. 이의 처리는 Sobel연산자의 크기와 방향성 누적값에 의해 빠르게 수행된다. 둘째, 상하위 해상도의 방향성 영상에서 Poincare지수를 이용하여 특이점을 추출한다. 상위 해상도는 특이점 후보의 수가 너무 많고, 하위 해상도는 특이점을 발견하지 못하는 경우가 있다. 셋째, 하위 해상도에서 제한한 폐곡선을 따라 특이점을 추출하고 후보 특이점에 대해 다시 해석하여 거짓 특이점을 제거하고 상위 해상도에서 그에 일치하는 특이점을 추출한다. 이것은 잃어버린 특이점의 발견뿐 아니라 거짓 특이점을 제거하는 과정을 포함한다.

II. 다해상도 방향성 영상에 기반한 특이점 추출

지문영상은 융선(ridge)과 계곡(valley)으로 구성되어 있다. 지문의 정합(matching)을 위해 사용되는 점을 지문분야의 용어로 특징(minutiae)이라 부르며 일반적으로 융선이 끝나는 점(ridge endings)와 융선이 갈라지는 점(ridge bifurcations)이 사용되고 있다. 지문영상으로부터 이런 특징을 추출하여 데이터 베이스에 저장시에 탐색속도를 줄이기 위해 먼저 지문의 유형을 분류한다. 지문은 크게 궁상분(arch), 좌제상분(right loop), 우제상분(left loop), 와상분(whorl), 초궁상분(tented arch)의 다섯가지로 분류하거나 이중루프(double loop)를 와상분에 포함시키지 않고 분리시켜 여섯가지로 분류한다[1-4]. 이를 분류하는 척도는 지문영상에 존재하는 특이점의 수와 방향성 흐름 패턴(directional flow pattern)에 의존한다. 특이점은 그림 1에서 보듯이 코아(core), 델타(delta) 점을 말하며 이들의 수에 의해 지문유형이 분류될 수 있다[4, 9]. 또한 이런 점들이 정합을 위한 등록점(registration point)을 구하는 중요한 역할을 하므로 이들을 정확히 잘 찾는 것이 매우 중요하다. 본 논문에서는 지문영상에 존재하는 특이점의 추출을 목표로 하고 있으며, 지문유형의 분류나 등록점에 대한 적용은 그 이후의 단계로서 여기서는 다루지 않는다.



그림 1. 와상분 패턴의 특이점들: ⊠는 상위 코아점, ⊡는 하위 코아점, △는 델타점을 나타낸다.

Fig. 1 The set of singular points for a whorl patterns: ⊠, ⊡, upper and lower core points, respectively, D, delta points

특이점 추출을 위한 본논문의 알고리즘은 다음의 단계로 구성되어 있다. 첫째, 영상의 각 블락에서 용선의 대표 방향을 구한다. 블락내의 방향정보를 구하는 과정에서 유도된 결과를 이용하여 지문영상의 전경과 배경을 분리한다. 둘째, 초기에 구한 방향은 잡음이 많이 존재하므로 주변의 방향과 평균을 구하여 평활화(smoothing)를 한다. 셋째, 상 하위 해상도의 방향성 영상에서 Poincare지수를 이용하여 특이점을 추출한다. 상위 해상도는 특이점 후보의 수가 너무 많고, 하위 해상도는 특이점을 발견하지 못하는 경우가 있다. 넷째, 하위 해상도에서 제안한 폐곡선을 따라 특이점을 추출하고 후보 특이점에 대해 다시 해석하여 거짓 특이점을 제거하고 상위 해상도에서 그에 일치하는 특이점을 추출한다. 이것은 잃어버린 특이점의 발견뿐 아니라 거짓 특이점을 제거하는 과정을 포함한다.

2.1 Sobel연산자에 의한 지문영상의 분할 및 방향성 영상 추출

지문영상을 처리하는데 적절한 정보를 포함하지 않는 영역은 제거하는 것이 필요하다. 이는 영상처리 및 패턴인식 분야에서도 마찬가지로 초기단계에서 주로 행해지며 그 이후의 처리가 전경에서만 수행될 수 있으므로 시간절약 뿐만아니라 시스템의 성능에도 중요한 영향을 끼친다. 좋은 분할 방법은 입력영상의 향상되었든 아니었든에 관계없이 원 영상의 대비에 민감하지 않아야 한다. 방향성(direction) 통계치와 분산(variance)에 기반을 둔 복합적 방법(composite) 방법[12]이 왜곡 영역을 결정하는데 효과적이다.

지문은 여러 곡선의 세그먼트로 구성되어 있으며 적절한 알고리즘을 사용하여 용선흐름(ridge flow)를 나타내는 2차원 방향성 영상(directional image)을 구성할 수 있다. 이런 용선 흐름이 지문의 유형 분류를 위한 특징 벡터로도 사용되며[2, 3], 이진화에 사용되기도 한다[13]. 방향성 영상을 계산하는 여러 알고리즘이 있으며 초기에 지문의 전경과 배경을 분할하기 위한 방법[11, 12]과 지문의 이진화를 목적으로 제안된 방법[2]이 널리 사용되고 있다.

방향성은 [2]의 방법으로 하고 지문 영상의 분할은 복합적인 방법[12]로 하는 것이 가장 좋은 성능을 나타내나 이렇게 할 경우 처리속도가 너무 많이 걸리는

단점이 있다. 한편 Ratha 등[13]은 Sobel연산자를 사용하여 방향성 영상을 구하였다. 이 방법은 200DPI이하의 영상에 대해서는 16×16블락영역에 대해 5~8개의 용선이 나타나므로 방향성정보 추출이 어려울 수 있으나 480DPI내지 500DPI의 영상에 대해서는 용선과 골이 두껍게 나타나기 때문에 잘 적용된다. 또한 지문의 유형분류를 위해 Halici 등[3]은 블락의 방향성에 대한 확실성(certainty)을 도입하였다. 확실성은 각 화소에서의 방향을 단위 벡터로 나타내고 블락내에서 모든 벡터를 누적하여 평균한 크기를 가지고 신경회로망의 연결강도(connection weight)를 갱신(update)하는데 가중치로 사용하였다.

본 논문에서는 Sobel연산자를 이용하여 방향성을 추출하고 동시에 지문영상을 분할하는 효율적인 알고리즘을 제시한다. 방향성 영상은 Sobel연산자를 이용하였고, 지문 영상의 분할은 방향성영상을 구하면서 얻어지는 Sobel연산의 크기(이것은 분산과 같은 역할을 함)와 블락내의 평균 벡터의 길이(방향성 통계치 방법과 같은 역할을 함)를 사용하였다. Sobel연산의 크기는 균일한 영역을 분리하는데 분산방법과 같으나 처리시간이 적게 걸린다. 그러나 잡음이 있는 영상이나 방향성을 가지고 있지 않은 영역에서 잘 동작하지 않는다. 따라서 위의 복합적인 방법에서 분산 대신 Sobel연산의 크기를 사용하고 방향성 통계치 대신 [11]의 평균벡터의 길이를 사용하여 지문영상을 분할한다. 따라서 이의 처리는 Sobel연산 한번으로 방향성 및 전경 배경을 동시에 수행하기 때문에 처리속도가 빠르고 안정적이다. 방향성 추출 및 분할 방법을 요약하면 다음과 같다.

1) 8×8화소 혹은 16×16을 하나의 블락으로 나눈 후 각 블락에 대해 영상의 각 화소에서 x, y 방향으로 3×3의 Sobel 연산자를 통과시킨 값 $G_x(i, j), G_y(i, j)$ 를 구한다.

2) 각 화소에서 크기(magnitude)를 구하고 블락내에서 누적하여 블락의 평균 크기를 구한다.

$$A(k, l) = \frac{1}{XY} \sum_{j=0}^Y \sum_{i=0}^X \sqrt{G_x^2(i, j) + G_y^2(i, j)} \quad (3)$$

여기서 X 는 블락의 넓이고 Y 는 높이이다.

3) 다음 식을 사용하여 각 화소에서 방향을 구한다.

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \frac{2G_x(i, j)G_y(i, j)}{G_x(i, j)^2 - G_y(i, j)^2}$$

$G_x \neq 0, G_y \neq 0$ (4)

G_x 나 G_y 가 0이면 방향은 0이거나 90이된다.

4) 각 화소에서 방향을 8방향으로 양자화하여 누적하지 않고 이 방향 θ_d 에 2를 곱하여 단위벡터로 만들어 주면서 각 성분으로 분리하여 표기한다.

$$v(i, j) = (\cos 2\theta_d, \sin 2\theta_d) \quad (5)$$

5) 이 벡터의 각 요소를 분리하여 한 블락내에서 모두 더한다.

$$V(k, l) = \left(\sum_{j=0}^y \sum_{i=0}^x \cos 2\theta_d, \sum_{j=0}^y \sum_{i=0}^x \sin 2\theta_d \right) \quad (6)$$

6) 각 성분을 x, y 로 하여 주방향을 구하고 그 크기를 계산한다.

$$\Theta(k, l) = \frac{1}{2} \tan^{-1} \frac{\sum_{j=0}^y \sum_{i=0}^x \sin 2\theta_d}{\sum_{j=0}^y \sum_{i=0}^x \cos 2\theta_d} \quad (7)$$

$$L(k, l) = \frac{\sqrt{\left(\sum_{j=0}^y \sum_{i=0}^x \cos 2\theta_d, \sum_{j=0}^y \sum_{i=0}^x \sin 2\theta_d \right)^2}}{XY} \quad (8)$$

여기서 Θ 는 그 블락의 주방향이고 L 는 확실성으로 사용했던 블락내의 평균 벡터의 길이이다. 잡음이 있는 영역이나 배경부분은 방향이 랜덤하기 때문에 상쇄되어 주 방향의 크기가 매우 작게 나타난다. 이 방향의 크기가 방향성 방법에서 주 방향을 누직한 최빈치의 수와 유사하다. 여러 영상에 대해 실험한 결과 방향성 방법과 유사한 결과를 나타낸다. 그러나 이 방법은 8방향으로 양자화하는 과정에서 오차를 제거한다. 주 방향은 2 가지 방법으로 구할 수 있다. 즉 각 화소에서 방향을 구하여 단위벡터로 만들고 각각 누적하는 방법과 하나의 블락에서 전체의 방향을 구하는 다음의 방법이 있다.

$$\Theta = \frac{1}{2} \tan^{-1} \frac{\sum_{j=0}^y \sum_{i=0}^x 2(G_x(i, j)G_y(i, j))}{\sum_{j=0}^y \sum_{i=0}^x (G_x(i, j)^2 - G_y(i, j)^2)}$$

$G_x \neq 0, G_y \neq 0$ (9)

전자는 화소별 단위 벡터를 만들기 때문에 각 화소의 기울기 크기는 누적이 되지 않는다. 즉 낮은 대비 영상이나 높은 대비 영상이나 같은 값이 나온다. 반면 후자는 각 화소에서 방향과 크기를 가지고 누적되어 최종에 방향을 구한다. 그리고 각 화소에서 크기를 누적하면 한 블락의 기울기 크기가 되는데 이 크기가 영역을 분리하는 척도로 분산과 비슷하다.

이렇게 구한 주 방향의 크기가 영역 분할을 위한 척도로 사용된다. 제안한 알고리즘은 이 둘을 복합적으로 사용한다. Sobel의 크기로 전경과 배경을 분리하고 전경부분에 대해서 주방향의 크기에 의해 배경부분을 제거한다. 즉, 만일 미분 크기의 합 $A(k, l) > T_A$ 보다 크고 주 방향의 크기 $L(k, l) > T_C$ 이면 전경으로 간주한다. 그렇지 않으면 배경이다. 그림 2의 (a)는 Sobel 연산 크기를, (b)는 방향성 평균 누적벡터의 크기를 각 블락별로 나타내었다. 여기서 흰 영역이 크기가 크다. 그림 3은 이 두 방법에 의해 분할된 전경 영상이고 간단한 평활화를 수행하여 둥성한 구멍을 채우주고 난후 전경영역만을 나타낸 것이다. 실제 왼쪽 하단의 텔타점 좌하 부분은 정합을 위해 특징점을 추출할때는 배제되어 배경으로 간주된다. 이런 영상에서의 특징(minutiae) 추출은 오류를 많이 수반하기 때문이다. 그러나 특이점을 추출하때 이것을 배제하면 화질이 안좋은 영상에서 오류를 많이 일으키므로 본 논문에서는 평균벡터의 크기에 대한 문턱치를 다르게 하여 이것을 조절한다.

이 방법은 Sobel 연산만으로 방향성 정보와 평균 크기 및 주방향의 벡터 크기를 동시에 구할 수 있으므로 이 성능은 방향성방법과 분산방법을 동시에 이용하는 복합방법[12]과 유사하나 알고리즘의 간편성과 빠른 수행속도 그리고 방향성누적분포의 양자화 오차를 줄일 수 있는 장점이 있다. 또한 이렇게 구한 방향성정보는 후에 이진화과정 및 지문 분류시 매우 중요한 정보로 사용된다.

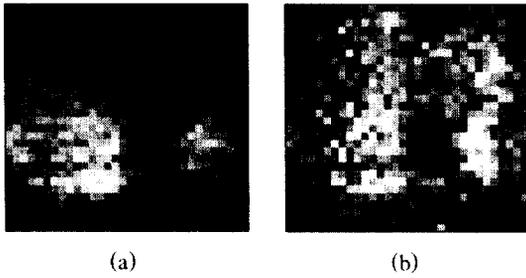


그림 2. Sobel 연산 결과 (a)크기 (b)평균 방향 벡터
Fig. 2 Results of Sobel operation, (a)magnitude, (b)average direction vector



그림 3. 분할된 영상
Fig. 3 Segmented image.

2.2 방향성 영상의 평활화(smoothing)

각 블록에서 방향을 식 (9)에 의해 구한 다음 8방향으로 양자화한다. 512×480의 입력 지문영상에서 한 블록의 크기를 8×8화소로 할 경우, 방향성영상의 크기는 64×60이 되고 16×16화소로 할 경우 32×30이 된다. 초기에 구한 방향은 잡음이 많이 존재하므로 주변의 방향과 평균을 구하여 평활화(smoothing)를 한다. 방향을 평활화하는데 간단히 더해서 modulo 8처리 하는 것은 정확한 결과를 내지 못한다. 그러나 방향을 벡터로 나타내서 더하는 좋은 방법이 있다[4]. 즉, 각 화소에서 방향을 $\alpha \in [0, 180)$ 으로 구한후 2를 곱해서 $v = (\cos 2\alpha, \sin 2\alpha)$ 의 단위벡터로 나타낸다. 평

활화는 x, y 각 벡터요소를 분리해서 처리한 후 역으로 방향을 구할때는 $\theta = 1/2 * \arctan(y/x)$ 에 의해 구하며 3×3의 창을 사용한다. 16×16화소의 블록에 대한 방향성 영상은 8×8화소의 블록 방향성 영상에서 평활화하는 방법과 마찬가지로 단위 벡터로 나누어 블록 4개를 평균하면 쉽게 구해진다. 그림 4의 (a)는 그림 2에 대해 16×16화소를 한 블록으로 하여 구한 방향성 영상이고 (b)는 평활화한 후에 방향성 영상이다.

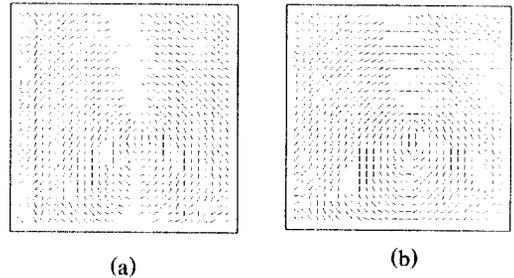


그림 4. 방향성 영상. (a)평활화 전 (b)평활화 후
Fig. 4 Directional image. (a)before smoothing, (b)after smoothing.

2.3 Poincare지수에 의한 특이점 추출

평활화한 두 해상도의 방향성 영상에서 특이점은 주위의 작은 폐곡선을 따라 Poincare 지수를 계산함으로써 보통점, 코아점, 델타점을 추출한다. Poincare 지수는 폐곡선을 따라 방향성 각도의 변화를 더함으로써 계산된다. 방향성 영상에서 폐곡선을 따라 시계반대 방향으로 돌 때 방향각이 0, ±180, ±360가 된다. 각이 0라면 보통점, 180라면 코아점, -180라면 델타점이다[4]. 이 영상에서 특이점은 Poincare 지수를 모든 점에서 주위 영역을 시계반대방향으로 돌아가면서 계산한다. Karu 등[4]은 8×8화소를 하나의 블록하는 방향성 영상에서 2×2창을 이용하여 즉, $(i, j) \rightarrow (i+1, j) \rightarrow (i+1, j+1) \rightarrow (i, j+1) \rightarrow (i, j)$ 을 돌면서 계산한다. 수행영역은 지문의 전경영역에서만 수행된다. 한번 평활화한 방향성 영상에서 특이점의 추출결과는 지문의 전경부분 분할성능에 크게 의존한다. 배경부분이 많이 포함되면 너무 많은 특이점이 추출되고, 전경영역이 훼손되어 분할되면 외곽부분의 델타점이 추출 안될 수 있다. 따라서 적절한 전경배경 분할 방

법이 선행되어야 한다. 전 단원에서 제시한 간단한 분할 방법은 이를 수행하는데 매우 효율적이다. 지문 영상에서 코아점과 델타점은 많아야 각각 2개이다. 초기에 추출한 특이점은 잡음의 영향에 의해 2개 이상일 수 있다. 특히, 8×8화소를 한블락으로 한 경우에는 너무 많은 특이점이 추출되고 이를 제거하는데

어려울뿐 아니라 시간 소모가 많다[4]. 또한 이럴 경우 진짜 특이점이 없어질 경우도 있다. 16×16화소 크기의 블락 영상인 경우 특이점은 그리 많지 않으나 특이점을 찾지 못하는 경우가 발생한다. 그림 5는 두 해상도의 방향성 영상에서 Poincare지수를 이용해 각각 코아점 및 델타점을 추출해 표시한 것이다. 여기서 사각형은 코아점이고 삼각형은 델타점이다. 그림에서 보듯이 8×8 해상도의 경우는 너무 많은 특이점이 추출되었고, 16×16 해상도는 왼쪽에 델타점이 없어지는 점이 발생되었다.



(a)



(b)

그림 5. Poincare 지수에 의해 검출된 특이점들, 코아(□), 델타점(△).

(a)8×8화소인 경우 (b)16×16화소인 경우

Fig. 5 Detected singular points using Poincare index, Core (□), delta(△).

(a)8×8 pixels, (b)16×16 pixels.



그림 6. Karu의 방법에 의해 검출된 특이점들. ⊠, ⊞는 각각 상위 및 하위 코아점, △는 델타점.

Fig. 6 Detected singular points by Karu's method. upper core(⊠), lower core(⊞), delta(△) points

2.5 다해상도 방향성 영상을 이용한 특이점 추출

이의 수행속도를 높이고 정확한 특이점의 위치를 검출하기 위한 하나의 방법은 다해상도에 의해 문제를 해결하는 것이다. 즉, 두 해상도로 방향성 영상을 구하고 저 해상도(16×16)에서 특이점을 기반으로 고 해상도(8×8)에서 특이점을 선택하는 효과적인 방법을 제시한다. 그림 7에 제안한 방법의 흐름도를 나타낸다. 16×16해상도는 8×8해상도로부터 4개블락을 평균하여 발생된다. 중요한 것은 특이점 후보의 수를 줄이면서 잃어버리는 특이점이 발생되지 않도록 하는 것이다. 그러나 저해상도 영상의 경우 거짓 후보점은 매우 적으나 잃어버리는 특이점이 발생할 경우가 있고 고해상도의 경우 거짓 후보점이 너무 많이 생겨 이를 제거하는데 시간이 많이 걸리며 잘 못 제거될 확률이 있다. 따라서 고해상도에서 후보 특이점을 추출하고 거짓 특이점의 제거는 방향성 영상 평활화 방법으로 수행한다. 그러나 검출하지 못하는 점(missing points)이 발생되기 때문에 이를 발견하기 위해 Karu가 사용한 4 화소 폐곡선 대신 8화소 폐곡선을 따라 이를 해결한다. 그림 8은 4 화소 폐곡선의 자취와 8화소 폐곡선의 자취를 표시한 것이고 그림 9는 8 화소 폐곡선을 따라 Poincare지수를 계산하여 특이점을 추출한 영상이다. 하나의 코아점이 4군데 나타난다. 4개가 발생한 지역은 확실한 코아점으로 간주한다. 따라서 반복처리 없이 2개로 끝낸다. 그러나 델타점은 4화소 폐곡선에서 찾지 못했던 왼쪽 아래 부분에 2개가 검출되었다. 이런 경우 50%가 있을 확률이 존재한다. 이것은 블락이 잘못 나뉘어져 발생된 경우일 수 있으므로 그 점 주변영역을 16×16화소의 블락으로 오버랩 하여 다시 블락을 만들고 바로 이웃화소에만 Poincare지수를 적용한다. 즉, 0~15, 16~31 등으로 블락을 만드는 대신 8~23, 24~39 등으로 블락을 다시 구성하면 화질이 좋지 않은 영상에서 블락의 구성으로 인해 없어진 특이점들이 다시 발견될 수 있다. 이를 구성하는 것도 8×8화소 블락의 해상도에서 바로 발생되기 때문에 시간소모가 거의 없다. 재해석시 특이점이 3개 이상 나오면 즉, 75%이상이면 그 지점은 특이점이 있는 영역으로 선택한다. 그림 10에 보듯이 3개가 나타나 특이점으로 선택되었다. 이마지막으로 16×16해상도는 위치에 대한 정확성이 없다. 따라서 그점에 대한 가장 가까운 점을 8×8영역에서

선택하면 그림 1과 같다.

본 논문에서 코아점과 델타점을 동시에 추출하는 Karu의 방법과는 달리 코아점을 먼저 추출한다. 여러 영상에 대해 실험한 결과 코아점 델타점보다 중앙에 위치하기 때문에 지문의 화질이 좋아 더 잘 추출될 뿐만아니라 반복 처리 횟수가 틀려 찾은 특이점이 잃어버리는 경우가 발생한다. 또한 거짓 델타점의 제거는 코아점을 먼저 추출한 후 방향성 평활화와 기하학적인 관계를 함께 고려해 수행한다. 실제 기하학적인 관계는 지문영상이 회전이 심하게 일어나지 않았다는 가정하에 델타점이 코아점보다는 아래에 있으며 델타점이 2개인 경우 코아를 중심으로 같은편에 존재하지 않는다는 것만 고려한다. 16×16크기의 블락영상은 8×8을 만들 때 같이 같이 구해지므로 시간 소모가 거의없다. 또한 영상의 평활화로 인해 손실될

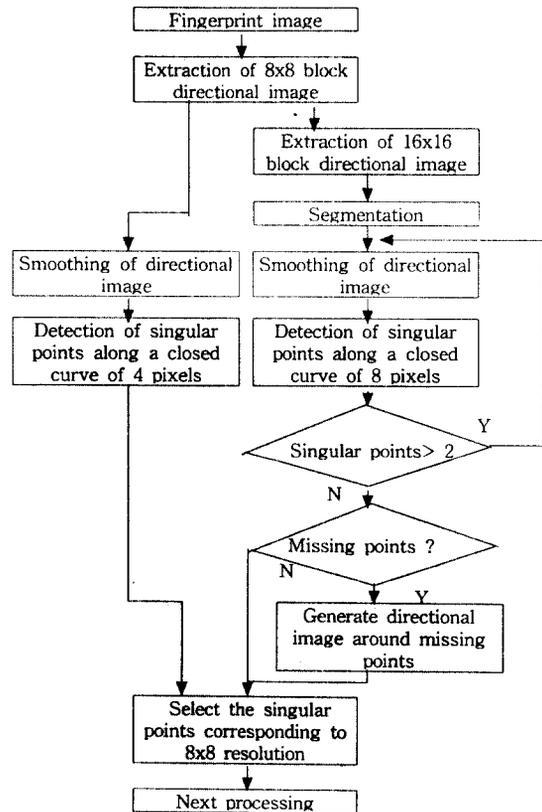


그림 7. 특이점 추출의 흐름도
Fig. 7 Flow chart for detection of singular points

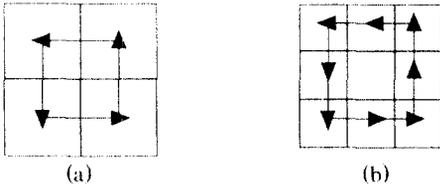


그림 8. Poincare지수를 계산하기 위한 폐곡선 (a)4화소 (b) 8화소

Fig. 8 Closed curves for computing Poincare index, (a)four pixels, (b)eight pixels.



그림 9. 8 화소 폐곡선을 따라 검출된 특이점들

Fig. 9 Detected singular points along a closed curve of 8 pixels.



그림 10. 델타 후보점 주위에 방향성 영상을 다시 구성하여 추출한 특이점

Fig. 10 Detected delta points from directional image reconstructed around probable delta points

수 있는 특이점들이 폐곡선을 다르게 추적함으로써 신뢰도가 높은 특이점을 추출한다.

Ⅲ. 실험 및 고찰

제안한 방법을 여러 지문영상에 대해 테스트하였다. 지문영상은 지문을 잉크에 묻혀 종이에 찍은 후 이를 카메라나 스캐너로 읽어들이는 방법이 있고, 직접 프리즘에 지문을 대고 CCD카메라로 읽어들이는 방법이 있다. 본 논문에서는 전자주민 카드 제작시 이 둘을 수용하기 위하여 전자주민카드용으로 개발된 화상입력장비로 지문을 입력받아 테스트 하였다. 지문의 화질은 종이에 찍어 다시 받는 간접입력 방식보다 프리즘을 통해 CCD카메라로 직접 획득한 영상이 훨씬 좋다. 지문의 해상도는 480DPI에 512×480영상이며 화상입력보드는 두인전자의 미디어 캡7을 사용하였다. 여러 다른 지문의 유형에 대해 실험하기 위해 각 연령층에 걸쳐 남녀 50명으로부터 채취하였다. 획득한 영상에서 이동은 있으나 화전은 가능한 발생되지 않도록 하였으며, 각 사람마다 8장씩 총 400장의 오른손 엄지손가락으로 구성하였다. 표 1에 보듯이 실험결과로는 15개의 특이점을 잘 못찾아낼 뿐 거의 모든 특이점을 찾아내는 좋은 결과를 나타내었다. 굵은 글씨로 밑줄친 부분이 맞게 찾은 숫자이다. 예를 들리 와상문(whorl)에서 코아가 2인데 1개밖에 못찾은 경우가 1이고, 델타점이 2개인데 1개로 찾은 경우가 2. 우제상문(left loop)에서 델타점이 1개인데 2개로 찾은 경우가 2 존재한다. 그러나 코아가 2인 경우 반드시 델타가 2개이므로 찾지 못한 다른 하나를 발견할 수 있고, 코아가 1인 경우에 델타가 2인 경우는 흐름패턴에 의해 다른 한쪽을 제거할 수 있다. 이는 지문의 유형분류에 연관된 문제이므로 여기서는 생략한다. 개발환경은 120MHz 펜티엄 CPU, 16MB의 메모리를 가진 PC의 Window 95환경하에서 개발하였으며 개발도구는 MSVC 4.0에서 C++로 구현하였다. 그림 11은 개발된 시스템이 Window95환경하에서 실행된 모습을 보여준다. 표 2는 각 단계별 실행시간을 나타낸다. 실행시간은 방향성 영상을 구하는데 거의 모든 시간이 소모하나 분할을 위한 모든 정보를 포함하여 계산하므로 실제 분할 시간은 매우 짧다. 16×16의 방향성 영상은 8×8방향성영상으로부터 4

표 1. 실험결과

Table 1. Experimental results

Assigned class	Number of samples	Number of detected core points			Number of detected delta points			Error
		2	1	0	2	1	0	
Whorl	12×8=96	95	1	0	94	2	0	3
Double Loop	13×8=104	103	1	0	101	3	0	4
Left loop	12×8=96	1	95	0	2	94	0	3
Right loop	11×8=88	0	88	0	2	85	1	3
Arch	1×8=8	0	0	7	0	1	7	1
Tented arch	1×8=8	0	7	1	0	7	1	1
	400							15

표 2. 알고리즘의 각 단계별 수행시간

Table 2. Execution times for steps in the algorithm

Step	Time(s)
Directional image	1.63
Directional smoothing	0.01
Segmentation	0.02
Finding core points	0.03
Finding delta points	0.1
Total	1.80



그림 11. 윈도우95환경에서 구현된 시스템의 실행화면
Fig. 11 Implemented system running under Window95

블락을 평균하여 얻어지므로 시간 소모가 거의 없다. 코아점 및 델타점의 추출은 16×16영상에 기반을 두어 수행하므로 초기에 거짓 후보점이 별로 없어 매우 빠르게 추출된다. 델타점 추출은 가장자리에 존재하기 때문에 이를 해결하기 위해 약간의 시간이 더 소

비된다. 이 시간은 여러번 실험을 하여 평균값을 취한 것이나 거의 2초내에 전과정이 이루어진다.

IV. 결 론

본 논문에서는 지문영상의 특이점을 추출하기 위해 다해상도 방향성 영상을 구성하고 Poincare지수에 기반한 효율적인 알고리즘을 제안하였다. 또한 전처리 과정으로서 Sobel 연산자를 이용한 지문영상의 분할 방법을 제시하였다. 이방법은 기존의 방향성 방법과 분산방법을 모두 포함하는 하나의 방법으로서 매우 간단하고 수행속도가 빠르며, 특이점을 정확히 잘 추출해 낸다. 제안한 알고리즘의 요점은 후보 특이점의 수를 줄이면서 잃어버릴 가능성이 있는 특이점을 추출하여 다시 해석함으로써 확실한 특이점을 선택하도록 하는데 있다. 또한 코아점을 먼저 추출하여 델타점 추출시 기하학적인 정보로 사용함으로써 시스템의 성능을 향상시킨다. 이 알고리즘은 전자주민 카드용 화상입력장비 개발시 소프트웨어부문에서 지문 입력모듈의 일 부분이다. 전자주민카드에는 사진, 인감, 지문의 영상정보가 저장되며, 이중에 지문은 500바이트의 특징점으로 저장된다. 이 특징구조에 코아나 델타점이 6바이트 공간을 차지하고 있으며 이 정보가 정합시 등록점의 역할을 하여 시스템의 성능에 영향을 미친다. 나아가 이 알고리즘에 의해 추출된 특이점의 수와 위치정보, 그리고 특이점에서의 방향패턴에 의해 지문의 유형을 분류하는데 확장될 수 있다.

참 고 문 헌

1. H. C. Lee and R. E. Gaensslen (Eds.), *Advances in fingerprint technology*, Elsevier, 1991.
2. C. L. Wilson, G. T. Candela, C. I. Watson, "Neural-network fingerprint classification," *J. of Artificial Neural Networks*, 1(2), 203-228. 1994.
3. U. Halici and G. Ongun, "Fingerprint classification through self-organizing feature Maps modified to treat uncertainties," *Proc. of The IEEE*, Vol. 84, No. 10, October 1996.
4. K. Karu and A. K. Jain, "Fingerprint classification," *Pattern Recognition*, Vol. 29, No. 3, pp.

389-404, 1996.

5. C. V. Kameshwara Rao and K. Black, "Finding the core point in a fingerprint," *IEEE Trans. Comput.*, Jan. 1978.
6. C. V. Kameshwara Rao and K. Black, "Type classification of fingerprints-a synthetic approach," *Proc. 3rd Int. Conf. on Pattern Recognition*, San Diego, pp. 778-782, Nov. 1976.
7. V. S. Srinivasan and N. N. Murthy, "Detection of singular points in fingerprint images," *Pattern Recognition*, Vol. 25, No. 2, pp. 139-153, 1992.
8. 안도성, 김학일, "블록 FFT를 이용한 실시간 지문 인식 알고리즘," *대한 전자공학회 논문지* 제 32권 B편 제6호 pp. 89-101, 1995년 6월.
9. M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognition*, Vol. 17, No. 3, pp. 295-303, 1984.
10. B. M. Mehtre and N. N. Murthy, "A minutiae based fingerprint identification system," *2nd Int. Conf. Advances in Pattern Recognition and Digital Techniques*, Calcutta, (1986)
11. B. M. Mehtre et al. "Segmentation of fingerprint images using the directional image," *Pattern Recognition* Vol. 20, No. 4, pp. 429-435, 1987.
12. B. M. Mehtre et al. "Segmentation of fingerprint images-a composite method," *Pattern Recognition*, Vol. 22, No. 4, pp. 381-385, 1989.
13. N. K. Ratha, S. Chen and A. K. Jain, "Adaptive orientation-based feature extraction in fingerprint images," *Pattern Recognition*, Vol. 28, No. 11, pp. 1657-1672, 1995.
14. A. Rosenfeld and A. C. Kak, *Digital Image Processing*, Academic Press, N.Y., 1976.

이 준 재(Jun Jae Lee)

정회원

한국 통신학회지 제 21권 12호 참조
현재: 동서대학교 컴퓨터공학과 전임강사

심 재 창(Jae Chang Shim)

정회원

한국 통신학회지 제 21권 12호 참조
현재: 안동대학교 컴퓨터공학과 조교수



황 석 윤(Seok Yoon Hwang) 정회원

1978년 2월: 경북대학교 수학과 학사

1983년 2월: 대구대학교 수학과 석사

1989년 2월: 계명대학교 수학과 박사

1991년~현재: 대구대학교 수학과 부교수

※주관심분야: 퍼지 및 신경회로망, 패턴인식, 영상처리



남 재 열(Jae Yeal Nam) 정회원

1983년 2월: 경북대학교 전자공학과 학사

1985년 2월: 경북대학교 전자공학과 석사

1991년 8월: University of Texas at Arlington 전자공학과 박사

1991년 9월~1995년 2월: 한국전자통신 연구소 선임연구원

1995년 3월~현재: 계명대학교 전자계산학과 조교수
※주관심분야: 영상통신, 영상압축 및 관련 하드웨어 설계



이 주 형(Joo Hyung Lee) 정회원

1982년 2월: 강원대학교 통계학과 학사

1996년 2월: 경북대학교 컴퓨터공학과 석사

1983년 4월~현재: (주) 한국정보시스템 SI사업팀장

※주관심분야: MIS, 지문인식, 시스템 보안