
런길이 및 체인코드를 이용한 지문 융선의 방향 검출

이정환* · 박세현*

Detection of Fingerprint Ridge Direction Based on the Run-Length and Chain Codes

Jeong-Hwan Lee* · Se-Hyun Park*

이 논문은 2003학년도 안동대학교 학술연구비의 지원을 받았음.

요 약

본 논문에서는 지문 융선의 방향정보를 검출하는 효과적인 방법을 제안하였다. 제안방법은 먼저 지문영상을 정규화하고 융선이 있는 전경영역과 융선이 없는 배경영역으로 분할하여 문턱값으로 이진영상으로 변환한다. 전경영역은 융선(ridge)과 골(valley)부분으로 구성되는데 융선의 경계를 런길이 부호를 이용하여 체인코드로 표현한다. 지문 융선의 각 화소에서 방향정보를 검출하기 위해서 체인코드로 표현된 융선 경계를 추적하면서 방향정보를 구한다. 그리고 일정한 블록내의 융선방향은 급격하게 변하지 않으므로 블록별로 평활화하여 각 화소의 방향정보를 구한다. 제안방법의 성능평가를 위해 NIST 및 FVC2002 지문데이터베이스를 이용하여 컴퓨터시뮬레이션을 수행하였다. 실험 결과 제안방법을 지문 융선의 방향정보를 효과적으로 검출하는데 이용할 수 있음을 보였다.

Abstract

In this paper, we proposed an effective method for detecting fingerprint ridge direction based on the run-length and chain codes. First, a fingerprint image is normalized, and it is thresholded to obtain binary image with foreground and background regions. The foreground regions is composed of fingerprint ridges, and the ridges is encoded with the run-length and chain codes. To detect directional information, the boundary of ridge codes is traced, and curvature is calculated at each point of boundary. And the detected direction value is smoothed with appropriate window locally. The proposed method is applied to NIST and FVC2002 fingerprint database to evaluate performance. By the experimental results, the proposed method can be used to obtain ridge direction value in fingerprint image.

키워드

지문영상, 방향검출, 런길이부호화, 영상처리

1. 서 론

지문인식은 자동으로 개인의 신원을 확인하는 가장 대표적인 생체인식시스템으로 현재 활발한

연구가 진행되고 있다. 생체인식시스템에서 지문 인식을 많이 사용하는 이유는 지문은 사람마다 모두 다르고, 평생 동안 변하지 않는 고유한 신체적인 특징을 포함하고 있기 때문이다. 특히 근래 들어 보안시스템 등 컴퓨터를 이용한 지문자동인식에 대한 많은 연구가 진행되고 있는데 지문인식에 필요한 데이터가 급격히 증가하고 또한 고속처리가 요구되고 있기 때문에 이에 대한 연구가 활발하게 진행되고 있다[1,4,5,8,14].

지문영상은 융선과 골 영역이라는 두 가지 성분으로 구성된 무늬(texture) 영상으로 생각할 수 있다. 지문인식시스템에서 사용되는 특징은 지문 융선의 단점 및 분기점의 위치 및 방향정보 등이다. 단점은 융선이 끝나는 점이고 분기점은 하나의 융선이 두개 이상의 융선으로 나뉘어 지는 점이다 [2,3,7]. 즉 단점의 위치 및 방향정보, 분기점의 위치 및 방향정보, 그리고 분기되는 두 융선이 이루는 각도 등이 지문인식을 위한 중요한 특징점(minutia point)으로 사용된다. 지문의 특징점을 검출하는 여러 방법이 제안되었는데[1,8-11,13], 이들 연구에서는 검출된 특징점에서 융선의 구조적, 형태적 정보보다는 단점 및 분기점의 상대적인 위치 정보만을 포함하는 경우가 대부분이다[4,5,7].

현재 까지 발표된 대부분의 지문인식방법에서 특징점을 찾기 위해서는 입력된 지문영상의 밝기값의 불균일성을 제거하기 위해 감마교정, 히스토그램 균일화, 모폴로지 필터링 등과 같은 전처리과정을 거친다. 전처리된 영상에 대하여 밝기값에 대한 편차를 보정하기 위해 정규화 과정을 수행한다. 그리고 지문 융선에 대한 방향값을 블록별로 계산하여 융선이 없는 배경부분과 융선이 있는 전경영역으로 분할한다. 전경영역에 대하여 Gabor필터, 가우시안필터 등을 사용하여 영상을 개선(enhancement)한 후 융선과 골을 구분하여 이진영상으로 만들고, 세선화 과정을 적용하여 한 화소로 구성되는 가는 융선을 얻는다. 마지막으로 세선화 과정을 거친 가는 융선을 추적하여 단점, 분기점등 특징점을 검출하여 지문영상을 인식한다[7,8,10].

지문인식시스템에서 융선의 방향정보는 매우 중요한 특징이 되며, 특징점의 방향정보의 정확성은 전체 인식시스템의 성능에 큰 영향을 미치게 된다. 특히 지문영상개선을 위한 필터를 융선에 적용적으로 적용하기 위해서는 융선 방향정보를 보다 정확하게 추정하여야 필터설계를 위한 변수를 정확하게 추정할 수 있다[3]. 따라서 융선상의 각 화소에서 보다 정확한 방향 정보를 검출할 필요가 있다. 일반적인 방향정보 검출방법은 지문영상을 먼저 중첩되지 않게 블록별로 나누고, 각 블록에 대

하여 기울기 연산자로 기울기 값을 구하고 이를 이용하여 블록의 방향정보를 계산한다. 그러나 이 방법은 융선상의 각 화소에서의 방향값이 아니라 블록의 대표 값이므로 융선상의 각 화소의 방향정보는 오차를 포함하게 된다[1,3]. 또한 기존방법은 지문영상에서 국부적으로 융선 간격이 서로 다른 경우 방향정보가 부정확하게 된다.

본 논문에서는 이러한 문제점을 해결하기 위해서 보다 정확한 방향값을 검출할 수 있는 방법을 연구하였다. 제안된 방법은 이진영상을 표현하는데 많이 사용하는 런길이(run length) 및 체인코드를 이용하여 각 융선상의 화소에서 방향값을 계산하는 방법이다. 제안된 방법은 먼저 입력된 지문영상을 정규화하고, 정규화된 지문영상에 대하여 블록별 분산을 이용하여 전경과 배경으로 영상을 분할한다. 분할된 이진영상에서 전경영역에 런길이 부호화를 적용하여 융선의 경계를 체인코드로 표현한다. 그리고 융선의 경계를 추적하면서 각 화소의 방향정보를 검출하는 방법이다.

II. 제안된 지문영상의 방향검출 방법

1. 지문영상 정규화 및 분할

입력된 지문영상을 $F(x, y)$ 라고 하면 먼저 융선이 있는 전경영역과 융선이 없는 배경영역으로 지문영상을 구분하여야 한다. 지문영상의 특징점은 융선의 양 끝점 및 갈라지는 분기점이므로 융선이 분포된 전경영역만 분리하여 이 영역만 처리하는 것이 시간단축 등 보다 효과적이다. 따라서 지문영상을 전경영역과 배경영역으로 분할하기 위해 여기서는 블록별로 분산을 계산하여 문턱값 보다 작을 경우 해당블록은 배경영역으로 구분하고, 반대로 문턱값 보다 큰 경우 전경영역에 속하는 것으로 분할한다[1].

즉 지문영상 $F(x, y)$ 를 블록크기가 $W \times W$ 인 블록으로 겹치지 않게 나눈다. i 번째 블록의 분산 σ_i^2 를 아래 식(1)를 이용하여 계산한다.

$$\sigma_i^2 = W^{-2} \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} (F_i(x, y) - M_i)^2, \quad (1)$$

for $i = 0, 1, 2, 3, \dots$

여기서 M_i 는 i 번째 블록의 평균값이고 분할된 영상 $F_S(x, y)$ 는 i 번째 블록의 분산 σ_i^2 가 설정된

문턱값 보다 작으면 해당 블록은 융선이 없는 배경 영역에 속하고, 반대로 문턱값 보다 크면 전경영역에 속하는 것으로 구분하여 얻는다.

위와 같이 분할된 지문영상은 융선이 있는 부분만 처리하면 되므로 밝기값의 변화에 대한 영향을 줄이기 위해 다음 식(2)를 이용하여 전경영역을 정규화 한다.

$$F_N(x, y) = \begin{cases} M_N + \sqrt{\frac{\sigma_N^2(F_s(x, y) - M)^2}{\sigma^2}} \\ M_N - \sqrt{\frac{\sigma_N^2(F_s(x, y) - M)^2}{\sigma^2}} \end{cases} \quad (2)$$

여기서 $F_s(x, y)$ 는 분할된 전경영역의 영상이고, $F_N(x, y)$ 는 정규화된 영상, M_N, σ_N 은 각각 $F_N(x, y)$ 의 평균과 분산 M 과 σ 는 각각 분할된 영상 $F_s(x, y)$ 의 평균과 분산이다. 전경영역을 이진영상으로 만들기 위해 본 논문에서는 위의 정규화 과정에서 구한 평균값 M_N 를 사용한다. 즉 분할된 전경영역의 각 화소값을 M_N 과 비교하여 M_N 보다 크면 1(혹은 255)로 그렇지 않으면 0으로 하여 이진영상을 얻는다. 그림 1에 지문영상을 분할한 결과와 이진영상을 나타내었다.



(a) 분할된 지문영상 (b) 이진영상
 그림 1. 분할된 지문영상 및 이진영상
 Fig. 1 Segmented fingerprint and binary image

2. 융선의 경계검출

(1) 런길이 부호화

본 논문에서는 입력된 지문영상에서 전경영역만을 이진영상으로 변환한다. 그리고 융선의 방향정보를 검출하기 위해서 이진영상을 런길이 부호화 및 체인코드로 표현한다. 이진영상을 부호화 하는 방법은 크게 런길이 부호화 및 체인코드로 나눌 수 있는데, 런길이 부호화는 하드웨어 구현이 용이하

여 많은 영상처리에 사용되지만 물체들 사이의 계층적인 관계를 표현하기가 쉽지 않다. 반면 체인코드는 물체의 경계로부터 여러 가지 특징을 쉽게 추출할 수 있으며 계층적인 표현도 쉬우나 하드웨어 구현이 비교적 어렵다. 여기서는 체인코드 및 런길이 부호화의 장점을 살려서 먼저 하드웨어 구현이 가능한 런길이 부호화를 구한 뒤 이를 이용하여 지문 융선의 경계를 체인코드로 표현한다[12].

런길이 부호화는 이진영상을 런(run)들의 집합으로 표시하는데, 런은 라스터 주사방식의 영상처리장치에서 임의의 수평 주사선에서 물체 혹은 배경에 속하는 연속된 화소들의 모임이다. 런길이 부호화는 각 런의 시점과 종점의 (x,y)좌표로 부호화하는 방법이다. 영상을 n번째 라스터 주사선의 전경에 속하는 런의 시점 및 종점을 각각 $(X_L(n), Y(n)), (X_R(n), Y(n))$ 이라 하면, 이진영상 Z는

$$Z = \bigcup_{n=1}^N R(n) = \bigcup_{n=1}^N (X_L(n), X_R(n); Y(n)) \quad (3)$$

where

$$(X(n), Y(n)) \in (X_L(n), X_R(n); Y(n))$$

$$X_L(n) \leq \forall X(n) \leq X_R(n), X(n): \text{integer}$$

와 같이 표현되는데, 여기서 $R(n)$ 은 n번째 런이며 N은 이진영상의 런수이다. $X(n)$ 은 n번째 런의 x축 좌표이다. 그리고 각 영역을 표현하기 위해 런들의 연결성을 조사할 필요가 있는데 이를 위해 식(4)와 같은 "이웃함수 $ADJ(\cdot)$ "를 정의한다.

$$ADJ(R(m), R(n)) = \begin{cases} 1 & \text{if } R(m) \text{ and } R(n) \\ & \text{are connected} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where

$$R(m) = (X_L(m), X_R(m); Y(m))$$

$$R(n) = (X_L(n), X_R(n); Y(n))$$

여기서 $R(m), R(n)$ 은 각각 m 및 n번째 런이다. 런길이 부호로부터 체인부호를 검출하기 위해서는 런들 사이의 연결 관계를 위해 서로 독립적인 성질을 갖는 5개의 특수런인 D, F, R, S, X-run을 다음과 같이 정의한다.

D-run: n보다 작은 모든 m에 대하여 $ADJ(R(m), R(n))=0$ 일 때 R(n)런.

F-run: n보다 큰 모든 m에 대하여 $ADJ(R(m), R(n))=0$ 일 때 R(n)런.

S-run: n보다 작은 임의의 p에 대하여 $ADJ(R(n+1),R(n))=0, ADJ(R(n),R(p))=1, ADJ(R(n+1),R(p))=1$ 인 조건을 만족하는 R(p)가 존재할 때의 R(n)런.

X-run: 위의 D, F, R, S-run을 제외한 모든 런.

영상을 라스터 주사하면서 앞에서 정의한 런을 구하기 위해서는 3x2 창틀을 사용하여 각 런의 시점과 종점의 (x,y)좌표값과 특수런의 종류를 결정하면 된다. 여기서 3x2창틀을 사용한 것은 런의 종류를 결정하기 위해서는 현재 주사선과 바로 전의 주사선 그리고 바로 다음 주사선에 위치한 런들의 연결상태가 필요하기 때문이다. 그림 2는 런길이 부호화의 예이다.

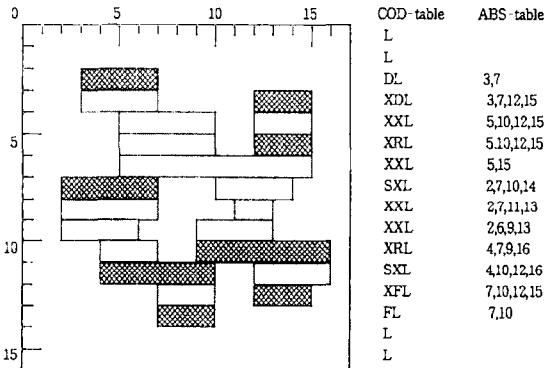


그림 2. 런길이 부호화의 예
Fig.2 An example of run-length coding

런길이 부호화는 라스터 주사하면서 각 런의 시점과 종점에 대하여 발생순서에 따라 그 점의 x좌표를 "ABS(abscissa) 테이블"에 저장하고 "COD(code) 테이블"에는 검출된 특수런을 저장한다. 또 주사선이 끝날 때 마다 주사선의 동기를 위해 "L" 코드를 COD 테이블에 저장하여 주사선의 y좌표를 알 수 있도록 하였다.

(2) 경계검출 방법

경계추적을 위해 이진영상이 저장된 메모리를 직접 주사하지 않고 런길이 부호화에서 구한 COD 및 ABS테이블을 검색하여 서로 연결된 두 런의 좌표를 찾아서 체인 부호를 구하는데, 이 방법은 아래와 같이 3부분으로 구분된다.

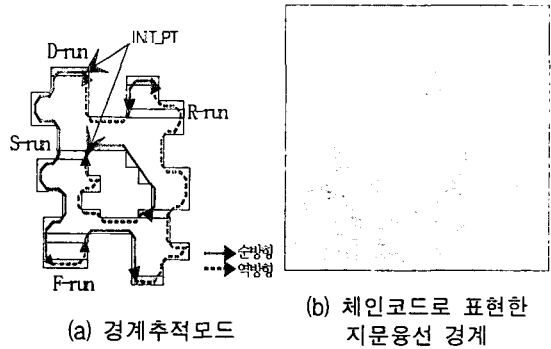
Step 1. Search INIT_RUN and INIT_PT where a boundary begins.

Step 2. Search NEXT_RUN and NEXT_PT

from CURR_RUN and CURR_PT.

Step 3. Generate chain codes from NEXT_PT and CURR_PT.

여기서 INIT_RUN은 라스터 주사하면서 영역 경계에 속하는 첫 번째 런으로 외부경계인 경우는 항상 D_run이 되고 내부경계인 경우는 항상 S_run이 된다. INIT_PT는 INIT_RUN의 시점 혹은 종점이다. 한 경계를 추적할 때 현재 경계점을 CURR_PT라고 하고, CURR_PT를 포함하는 런을 CURR_RUN이라고 한다. 그리고 경계화소 중에서 CURR_PT 다음에 추적할 경계점을 NEXT_PT라고 하고 NEXT_PT를 포함하는 런을 NEXT_RUN이라고 하면, CURR_RUN과 NEXT_RUN은 서로 연결된 런이 된다. 경계를 추적하는 모드는 CURR_PT가 CURR_RUN의 시점이면 순방향추적모드, 반대로 종점이면 역방향추적모드이다. 순방향추적모드에서 CURR_PT의 y축 좌표는 NEXT_PT의 y 좌표보다 항상 같거나 작다. 또한 역방향추적모드는 CURR_PT의 y축 좌표는 NEXT_PT의 y축 좌표보다 크거나 같다. 그림 3에 순방향 및 역방향추적모드와, 지문융선의 경계를 체인코드로 표현한 결과를 나타내었다. 그림 3에서 흰색선은 융선의 외부 경계이고, 회색선은 내부경계를 표시한 것이다.



(a) 경계추적모드 (b) 체인코드로 표현한 지문융선 경계
그림 3. 경계추적모드 및 체인코드
Fig. 3. Boundary tracking mode and ridge boundary represented by chain codes

3. 지문 융선의 방향검출

본 논문에서는 융선 경계상의 각 화소에서의 방향정보를 검출하고 이로부터 융선에 속한 런들의 방향값을 얻는다. 그리고 융선의 방향값은 일정한 블록내에서는 급격하게 변하지 않으므로 융선에 속하지 않는 화소의 방향값을 블록내의 이웃화소들의 방향값을 평균하여 얻는다. 기존의 방향정보 검출방법[1]은 먼저 지문영상을 WxW크기로 블록

으로 나누고, 블록내의 각 화소 (i, j) 에서 기울기 연산자를 사용하여 x 축 및 y 축의 기울기성분 $\delta_x(i, j), \delta_y(i, j)$ 를 각각 계산하고, 블록내의 중심점 (x, y) 에서의 방향정보 $\theta(x, y)$ 를 아래 식으로 계산한다.

$$\theta_x(x, y) = \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} 2\delta_x(i, j)\delta_y(i, j) \quad (5)$$

$$\theta_y(x, y) = \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} (\delta_x^2(i, j) - \delta_y^2(i, j)) \quad (6)$$

$$\theta(x, y) = \frac{1}{2} \tan^{-1} \frac{\theta_y(x, y)}{\theta_x(x, y)} \quad (7)$$

여기서 $\theta_x(x, y), \theta_y(x, y)$ 는 각각 블록의 중심점 (x, y) 의 x 축 및 y 축의 방향성분이다. 위 식(7)로 계산된 방향정보는 지문영상의 잡음, 융선 구조 등에 따라 방향값에 오차가 포함되어 부정확하므로 이웃블록의 방향정보를 이용하여 오차를 보정할 필요가 있다. 이를 위해 먼저 블록의 중심화소의 방향정보를 다음 식(8),(9)와 같이 벡터성분으로 분해한다.

$$\phi_x(x, y) = \cos(2\theta(x, y)) \quad (8)$$

$$\phi_y(x, y) = \sin(2\theta(x, y)) \quad (9)$$

그리고 위와 같이 분해된 벡터성분을 평활화한 후 식(10)과 같이 블록 중심점의 방향정보 $\phi(x, y)$ 를 구한다.

$$\phi(x, y) = \frac{1}{2} \tan^{-1} \frac{\phi_y(x, y)}{\phi_x(x, y)} \quad (10)$$

제안방법에서는 융선 경계를 추적하면서 경계상의 각 점에서의 곡률값을 계산하여 방향정보를 구한다. 지문 융선의 경계를 체인코드로 표현하면 다음 식(11)과 같이 표현할 수 있다.

$$C = \{p_i = (x_i, y_i), i = 1, \dots, n, 0 \leq p_i \leq 7\} \quad (11)$$

여기서 n 은 융선경계점(체인코드)의 개수이고 p_i 는 지문 융선 경계상의 i 번째 체인코드이고, p_i 의 좌표는 (x_i, y_i) 가 된다. 융선의 경계점을 체인코드로 추적하면서 경계상의 좌표로부터 곡률

값을 계산하는데, i 번째 체인코드 p_i 가 경계상의 한 점을 표현하므로 p_i 로부터 전후로 k 만큼 떨어진 경계점을 p_{i+k} 와 p_{i-k} 라고 하면, 경계점 p_i 에서의 곡률값 c_{ik} 는

$$c_{ik} = \frac{\vec{a}_{ik} \cdot \vec{b}_{ik}}{|\vec{a}_{ik}| |\vec{b}_{ik}|} \quad (12)$$

여기서 $\vec{a}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k})$ 이고, $\vec{b}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k})$ 인데 각각 경계점 p_i 와 p_{i+k} 및 p_{i-k} 와 이루는 벡터이다. 그리고 \cdot 는 벡터내적이며 $||$ 는 벡터 크기이다. 지문 융선상의 각 화소에서 방향정보는 곡률값 c_{ik} 를 계산하면 구할 수 있다. 융선을 구성하는 런 내부 화소의 방향정보는 융선 방향이 급격하게 변하지 않으므로 융선 경계의 방향정보를 이용하여 구할 수 있다. 본 논문에서는 융선의 경계점은 런의 시점과 종점으로 구성되므로 런 내부의 방향정보는 런의 시점 및 종점의 방향정보를 사용하여 구한다. 그림 4는 융선상의 경계 p_i 점에서 곡률값 c_{ik} 를 표시한 것이다.

III. 실험 결과 및 검토

제안방법의 성능을 평가하기 위해서 NIST 및 FVC2002 지문데이터베이스[6]의 지문영상을 사용하여 Visual C++6.0으로 컴퓨터 시뮬레이션을 수행하였다.

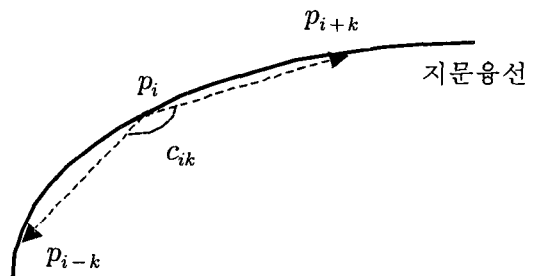
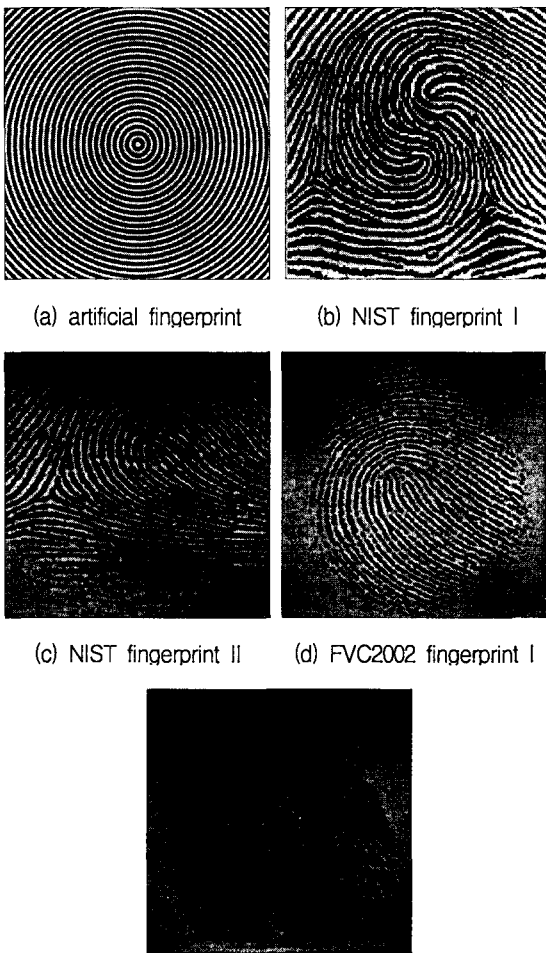


그림 4. 융선의 곡률값 c_{ik}
Fig. 4 Ridge curvature c_{ik}

실험에 사용된 영상은 크기는 256x256이고 256 단계의 밝기값을 갖으며 500dpi로 스캔된 데이터이다. 그림 5(a)는 인공적으로 만든 영상으로 제안 방법이 방향정보를 정확하게 검출하는지를 알아보기 위한 것이다. 그림 5의 (b) 및 (c)는 영상 전체에 융선이 분포된 NIST 데이터베이스의 지문영상으로 전경영역만 존재하는 지문영상이다. 그림 5의 (d) 및 (e)는 FVC2002 데이터베이스(DB4)의 영상으로 전경영역과 배경영역이 같이 있는 지문영상이다.



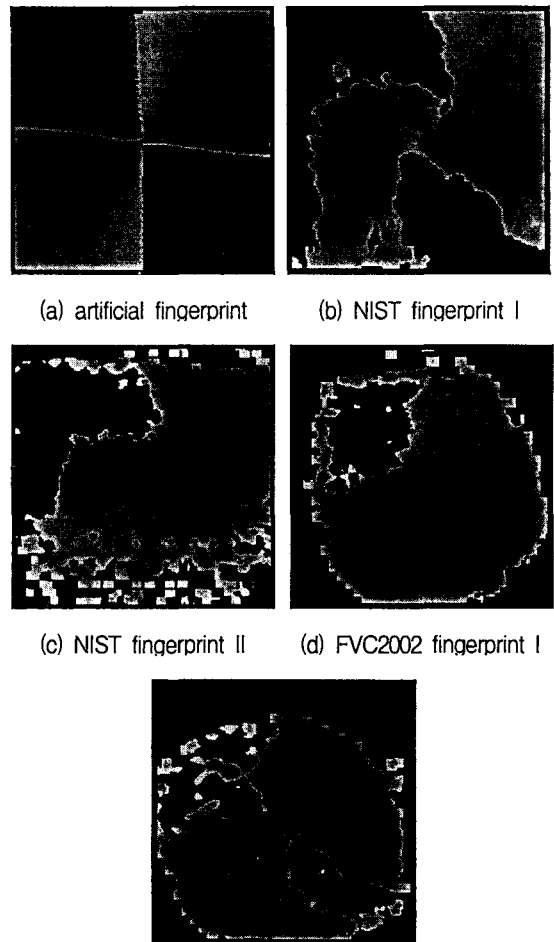
(a) artificial fingerprint (b) NIST fingerprint I
(c) NIST fingerprint II (d) FVC2002 fingerprint I
(e) FVC2002 fingerprint II

그림 5. 실험에 사용된 지문영상

Fig. 5 Fingerprint images used in experiments

그림 6은 전처리 과정을 거쳐 전경과 배경을 분할하고, 전경영역에 대하여 제안방법으로 방향정보를 구하여 0°도에서 180°사이 값으로 표시한 것이다. 그림 6에서는 전경영역의 각 화소에서의 방

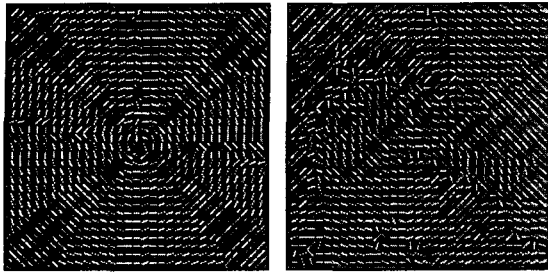
향정보를 표시한 것으로 밝은 부분은 각도가 큰 화소이고, 어두운 부분은 각도가 작은 화소들이다. 방향정보는 융선상의 경계점에서 곡률값을 계산하고, 융선의 곡률값은 일정한 블록내에서는 급격히 변하지 않으므로 이를 평활화하여 융선에 속하지 않는 화소의 방향정보를 얻었다. 본 논문에서는 평활화를 위한 블록의 크기를 8x8로 하였다. 그림 7은 제안방법을 적용하여 얻은 그림 6의 방향정보를 16방향 Poincare 인덱스[14]로 표시한 것이다. 그림 8은 기존방법[1]을 적용하여 방향정보를 구한 것이다. 실험결과 제안방법이 지문영상의 방향정보를 효과적으로 검출하는데 사용될 수 있음을 알 수 있었다.



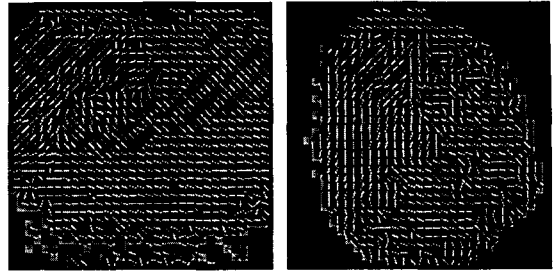
(a) artificial fingerprint (b) NIST fingerprint I
(c) NIST fingerprint II (d) FVC2002 fingerprint I
(e) FVC2002 fingerprint II

그림 6. 제안방법으로 검출한 방향정보

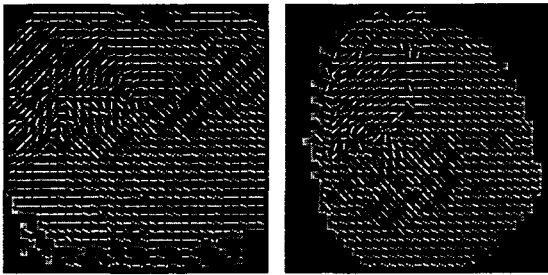
Fig. 6. Ridge direction value detected by the proposed method



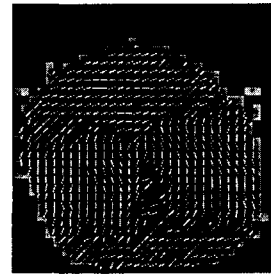
(a) artificial fingerprint (b) NIST fingerprint I



(c) NIST fingerprint II (d) FVC2002 fingerprint I



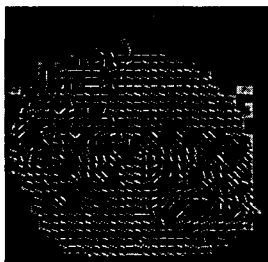
(c) NIST fingerprint II (d) FVC2002 fingerprint I



(e) FVC2002 fingerprint

그림 8. 기존방법[1]의결과

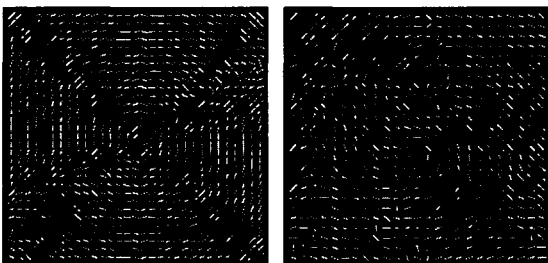
Fig. 8. The results ofconventional method[1]



(e) FVC2002 fingerprint II

그림 7. 제안방법의 결과

Fig. 7. The results of proposed method



(a) artificial fingerprint (b) NIST fingerprint I

IV. 결 론

본 논문에서는 런길이 부호화 및 체인코드를 이용하여 지문 윤선의 방향정보를 효과적으로 구하는 방법을 제안하였다. 지문영상의 방향정보 검출 과정은 지문영상의 개선, 윤선의 양 끝점 및 분기점에 해당하는 특징점 추출을 위해 꼭 필요하며 지문인식과정의 성능에 큰 영향을 미치는 중요한 과정이다. 제안방법은 먼저 지문영상을 정규화하고 블록별로 분산을 계산하여 지문영상을 윤선이 있는 전경영역과 윤선이 존재하지 않는 배경영역으로 분할하고 전경영역만을 이진영상으로 변환한다. 그리고 이진영상에 대하여 런길이 부호화를 수행하고 이로부터 윤선의 경계선을 체인코드로 표현한다. 윤선의 경계선을 추적하면서 각 경계점에서의 방향값을 계산한다.

또한 동일한 런의 양 끝점에서의 방향값을 비교하여 런내부의 방향값을 계산한다. 이렇게 얻은 방향값을 블록별로 평균하여 윤선이 있는 전경영역에 속한 모든 화소의 방향값을 구한다.

제안된 방법의 성능 평가를 위해 인공적으로 만든 지문영상, 전경영역으로만 구성된 지문영상, 전경과 배경영역으로 나누어지는 지문영상 등으로

지문영상의 각 화소의 방향값을 계산하여 컴퓨터 시뮬레이션을 수행하였다. 실험결과 제안방법이 지문영상의 방향값을 보다 정확하게 계산할 수 있음을 알 수 있었다. 앞으로 제안방법을 확장하여 지문영상의 개선 및 인식방법에 대한 연구가 수행되어야 한다.

참고문헌

[1] Lin Hong, Yifei Wan, and Anil K. Jain,, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.20, no.8, pp.777-789, August, 1998.

[2] Anil K. Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti, "Filterbank_based Fingerprint Matching", IEEE Trans. on Image Processing, vol.9, no.5, pp.846-859, May. 2000

[3] D. Sherlock, D. Monor, and K. Millard, "Fingerprint Enhancement by Directional Fourier Filtering", IEE Proc. Vision, Image and Signal Processing, vol.141, no.2, pp.87-94, 1994.

[4] Anil K. Jain, Lin Hong, and R. Bolle, "On-Line Fingerprint Verification", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.19, no.4, pp.302-313, 1997.

[5] D.Miao and D. Maltoni, "Direct Gray-Scale Minutiae Detection in Fingerprints", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.19, no.1, 1997.

[6] D. Maio, D. Maltoni, R. Cappelli, J.L Wayman, and Anil K. Jain, "FVC2000 : Fingerprint Verification Competition", ICPR, Barcelona, Sept. 2000, <http://www.bias.csr.unibo.it/fvc2000>.

[7] 양지성, 안도성, 김학일, "지문의 의사특징점 제거알고리즘 및 성능분석", 전자공학회 논문지, 제37권, C편, 제3호, pp.130-144, 2000.

[8] 김현, 김학일, "RSTI 불변 지문인식 알고리즘", 전자공학회 논문지, 제35권 S편, 제6호, pp.838-850, 1998.

[9] C. J. Lee and S. D. Wang, "Fingerprint Feature Extraction using Gabor Filters", Electronics Letters, vol.35, no.4, pp.288-290, 1999.

[10] N. K. Ratha, S. Chen, and Anil K. Jain, "Adaptive Flow Orientation-based Feature Extraction in Fingerprint Images", Pattern

Recognition, vol.28, no.11, pp.1657-1672, 1995.

[11] 김두현, 박래홍, 컨백스구조를 이용한 지문의 특징점 추출", 전자공학회 논문지, 제40권, SP편, 제6호, pp.427-435, 2003.

[12] S. D. Kim, J. H. Lee, and J. K. Kim, "A new chain coding algorithm for binary images using run-length codes", Computer Vision, Graphics, and Image Processing 41, pp.114-128, 1988.

[13] D. W. Jung and R. H. Park, Robust Fingerprint Identification based on Hybrid Pattern Recognition Methods", in Hybrid Methods in Pattern Recognition, H. bunke and A. Kandel, Eds., World Scientific Pub. Co., pp.275-300, 2002.

[14] D.Maltoni, D.Maio, A.K.Jain, and S.Prabhakar, "Handbook of Fingerprint Recognition", Springer, 2003.

저자소개

이정환(Jeong-Hwan Lee)



1984년 경북대학교 전자공학과 학사
 1986년 한국과학기술원 전기 및 전자공학과 석사
 1993년 한국과학기술원 전기 및 전자공학과 박사

2001~2002 미국 조지워싱턴대학교 객원연구원
 1990년~현재 안동대학교 전자정보산업학부 교수
 ※ 관심분야 : 영상처리, 멀티미디어, 워터마킹 등

박세현(Se-Hyun Park)



1980년 경북대학교 전자공학과 학사
 1982년 경북대학교 전자공학과 석사
 1985년 아주대학교 전자공학과 박사

1997년 국민포장 수여
 현재 안동대학교 전자정보산업학부 교수
 1997년~1999년 국립안동대학교 공과대학 학장
 1999년~2000년 미시간주립대학 전기컴퓨터공학과 겸임교수
 ※ 관심분야 : FPGA설계, 컴퓨터구조, 영상처리하드웨어 등