

# 중심점 기준의 융선추적 검증을 이용한 지문 인증 시스템

조영우\*, 조병호\*\*, 유기영\*

\*경북대학교 컴퓨터공학과 정보보호 연구실

\*\*디도정보기술

hiro@infosec.knu.ac.kr, bhcho@dittotec.com, yook@knu.ac.kr

## FingerPrint Verification System Using Core Centered Ridge Frequency Validation

Youngwoo Jo\*, Byungho Jo\*\*, and Kee-Young Yoo\*

\*Dept of Computer Engineering, Kyungpook National University

\*\*Ditto Information Technology Inc

### 요 약

Window Frame 방향성 추출법을 보완한 Extended Window Frame을 제시하고, 높은 보완성을 필요로 하는 곳에 사용할 수 있도록 하기 위해서 지문의 중심점으로부터 매칭 된 각각의 특징점까지 이르는 선상에 있는 융선을 추적하는 방법을 이용한 검증 방법을 제안한다. 제안된 검증 방법의 성능을 비교하기 위해서 방향성을 추출하는 방법, 특징점을 추출하는 방법을 기존의 여러 가지 방법들로 구현하고 그 결과를 비교, 분석한다.

### 1. 서 론

생체인식 시스템은 다른 사용자 인증 시스템에 비해서 여러 가지 장점을 가지고 있기 때문에 요즘 사용자 인증의 수단으로 많이 사용되고 있는 추세이다. 그 중에는 지문인식, 음성인식, 얼굴인식, 홍채인식 등이 연구되어 오고 있으며, 그 중에서도 지문인식이 실제 시스템으로 많이 사용되고 있다. 한국의 경찰청이나 미국의 FBI에서는 지문을 이용한 수사를 하고 있으며, 매우 큰 규모의 지문 데이터베이스를 갖추고 있다. 그리고 최근에는 개인이나 사무실 등에도 지문인증 자물쇠 등을 많이 사용하는 추세이다.

이런 실제 보완 시스템 상에서는 신원확인(1:n)과 인증(1:1)의 응용이 있다. 신원확인(1:n)은 지문을 이용한 범인검색과 같이 많은 매칭을 해야 하는 작업이고, 인증은 적법한 사용자가 맞는가를 검사하는 작업이다. 신원확인은 따라서 성능을 중요시하는 응용부분이며, 사용자 인증은 보완성을 중요시하는 응용부분이다. 본 논문에서는 사용자 인증 응용에서 사용될 수 있는 검증법과 방향성 추출법을 제안하였고, 이로써 보완성을 높일 수 있음을 보일 것이다.

### 2. 기존의 지문인식 방법

지문은 융선과 골로 표시되는 이미지로 볼 수 있다. 융선과 골은 사람마다 다른 위치와 방향의 특징점을 갖는다. 따라서 이 특징점을 이용해서 지문을 구분할 수 있다. 특징점의 종류는 끝점(Ending point), 분기점(bifurcation), 중심점(core), 삼각주(delta) 등으로 여러 가지가 있다. 이 가운데 지문인식에 있어서는 위에 나열한 네 가지가 주로 사용되

고, 특히 중심점과 삼각주는 지문을 분류하는데 주로 사용된다.

지문인식을 위해 처리하는 과정은 일반적으로 다음과 같다.

1. 입력이미지는 각 픽셀이 0-255의 명암으로 표시되는 흑백 영상이미지이다.
2. 흑백영상에서 이미지 보정작업을 수행한다.
3. 이미지를 적정크기의 블록으로 나누고, 각 블록의 방향성을 계산한다. 방향성은 360도의 방향을 8개의 대표 방향으로 양자화 시킨 값들 중에 하나이다.
4. 명암으로 표시되는 흑백 이미지를 각 픽셀을 0-1의 흑 또는 백으로만 표시하는 2진이미지로 변환한다.
5. 융선의 흐름을 1픽셀로 나타내도록 수정한다.
6. 5번까지의 결과 이미지로부터 특징점을 추출한다.
7. 얻어진 특징점들을 가지고 매칭 대상 이미지로부터 추출된 특징점들과 매칭을 수행한다.
8. 매칭 결과를 매칭점수(matching-score)로 변환시킨다. 그리고 미리 정해진 기준값(threshold)과 비교해서 한 쌍의 지문이 일치하는 지를 결정한다.

위의 과정들을 처리하는데 있어서 매칭의 성능을 높이는데 관련이 큰 부분은 특징점들을 추출하는 부분과 추출된 특징점들을 매칭하는 부분이다. 제안된 시스템에서는 특징점 추출에 필요한 블록의 방향성 추출 방법들을 제안하였고, 특징점을 추출하는 기존의 여러 방법들과 함께 구현하였다. 그리고 매칭 정확도를 높이기 위해서 매칭한 후에 실행하는 융선추

적 검증 방법을 제안하였다.

### 2.1 기존의 특징점 추출법

a. 기본 방법 : 용선 1픽셀 굵기 이미지에서 작업한다. 용선인 모든 픽셀들에 대해서 각각 그 점을 중심으로 하는 3x3 크기의 마스크를 적용시킨다. 그리고 주변의 8개의 픽셀값의 변화를 계산함으로써 그 결과가 2 이면 끝점, 6 이면 분기점으로 결정하는 방법이다.

b. 용선 추적(Ridge Following[4]) : 위의 방법은 특징점을 추출하기 전에 이미지에 많은 처리를 해야만 한다. 그러나 이 방법에서는 흑백영상 이미지에 곧바로 작업하므로써 효율을 높인다. 각각의 용선을 따라 가면서 검은색이 이어 지다가 끊어지는 곳이나 나오면 끝점으로 여기고, 용선이 갈라지는 곳이 나오면 분기점으로 여기는 방법이다.

c. 골 추적(Valley Following[5,6]) : 용선을 이용하는 대신에 골을 사용하는 방법이며, 진행 방법은 용선 추적과 흡사하다. 이미지 상에서 골에 대한 왜곡이 용선 보다 상대적으로 덜한 것이 보통이다. 따라서 골을 추적함으로써 정확도와 속도 측면에서 향상이 된 방법이다.

### 2.2 기존의 방향성 추출법

지문이미지를 적정 크기의 블록들로 나누고, 그 각각의 블록마다 그 안에 있는 용선들의 방향이 어디를 향하는가를 찾고, 그것을 그 블록의 방향성으로 정한다. (orientation). 그리고 그 블록이 지문 이미지를 벗어난 부분인가를 결정하는 것이다 (segmentation). 이러한 방향성 추출 방법들로는 Sobel operator를 사용하는 방법과 Window Frame을 사용하는 방법이 있다. Sobel operator를 사용하는 방법은 이 식을 사용함으로써 각 방향에 해당하는 픽셀들의 변화량이 측정하고, 그 변화량을 이용해서 방향성을 결정하는 방법이다. 그리고 Window Frame 방법은 5x5크기의 블록을 잡고 그 안에서 8가지의 각 방향에 해당하는 픽셀 중에 양 끝에 잇는 픽셀간의 값의 차이를 이용해서 방향을 결정하는 방법이다.

### 3. 제안한 시스템

제안한 지문인증 시스템은 등록단계와 인증단계의 2가지 과정을 거친다. 그림 1과 같이 등록단계에서는 오프라인 상태에서 사용자의 지문을 입력받아서 특징점을 추출한 후 저장매체에 저장을 하는 작업이다. 그리고 인증단계에서는 온라인 상태에서 사용자의 입력을 등록된 기존의 지문과 비교해서 일치하는 지를 결정하는 과정이다. 자세한 과정은 그림1과 같다.

지문 분류는 매칭 후에 사용될 검증 방법을 사용할 수 있게 하기 위해서 지문을 분류한다. 중심점이 있는 지문인지를 먼저 판별한 후에 중심점이 있는 지문에 대해서는 제시한 매칭 검증을 적용시킬 수 있다. 매칭은 elastic 매칭을 사용하였다.

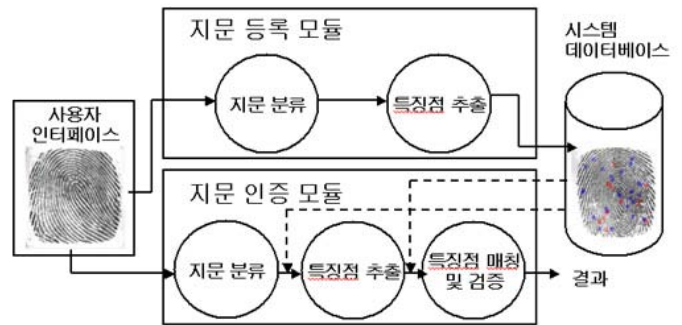


그림 1. 제안한 시스템.

### 3.1 제안한 Extended Window Frame

Window Frame을 확장한 방법이다. 방향을 계산하는 픽셀을 선택할 때, 양 끝점들만을 사용하지 않고, 그 사이에 있는 점들도 포함시켜 계산하는 방법이다.

$a_6$	$a_5$	$a_4$	$a_3$	$a_2$
$a_7$	$b_3$	$b_2$	$b_1$	$a_1$
$a'_0$	$b'_0$	$(i, j)$	$b_0$	$a_0$
$a'_1$	$b'_1$	$b'_2$	$b'_3$	$a'_7$
$a'_2$	$a'_3$	$a'_4$	$a'_5$	$a'_6$

$$S_k = \sum_{i=0}^{BS-1} \sum_{j=0}^{BS-1} \{ |a_k - b_k| + |b_k - (i, j)| + |(i, j) - b'_k| + |b'_k - a'_k| \}$$

단,  $k=0, 2, 4, 6$ , 이고 블록 크기(BS) = 8이다.

$$S_k = \sum_{i=0}^{BS-1} \sum_{j=0}^{BS-1} \{ |a_k - c_k| + |c_k - (i, j)| + |(i, j) - c'_k| + |c'_k - a'_k| \}$$

$$c_k = (b_{k/2} + b_{k/2+1})/2, \quad c'_k = (b'_{k/2} + b'_{k/2+1})/2$$

단,  $k=1, 3, 5, 7$ .

그림 2. Extended Window Frame Orientation.

그림 2의 식과 같이 각각의 방향에 있는 5개 픽셀 값들 간에 4개의 차이를 계산해서 그 중심이 되는 픽셀에서의 8가지의 각 방향마다 그 값을 계산해서 기록해 둔다. 만약 방향이 1, 3, 5, 7인 경우에는 그 방향에 해당하는 중간 픽셀을 정할 수가 없으므로, 올바른 방향 주변에 있는 2개의 픽셀의 평균으로 그 방향에 있는 중간 픽셀의 값으로 정한다. 구하려고 하는 블록의 모든 픽셀에 대해서 아래의 계산을 하

면, 그 블록 전체 픽셀들이 갖는 8가지 방향 가운데 가장 많이 나온 방향과 가장 적게 나온 방향을 결정할 수 있다. 이 두 가지 방향값을 가지고 그 블록의 방향을 결정한다.

### 3.2 제안한 매칭 검증

매칭 검증의 기본 생각은 지문의 중심점을 찾아서 그 위치로부터 매칭된 각각의 특징점을 잇는 직선을 따라 가면서 용선을 몇 개나 지나는 지를 체크하고, 매칭된 특징점 쌍들 간에 그 차이를 비교해서 잘못 매칭된 특징점의 쌍을 매칭 결과에서 제거하는 것이다.

#### 3.2.1 중심점 찾기

다음의 3가지 과정을 거친다.

a. 지문의 방향성은 360도 방향을 8가지로 나눈 것 중의 하나로 결정된다. 그림 3은 방향성이 X축에 평행인 방향을 표시하고 있으며 그림과 같이 방향성의 특성상 방향성이 한군데로 몰리는 지점이 있는데, 그 주위에 지문의 중심점이 있다. 이렇게 해서 중심점이 있는 예상 구역을 찾는다.



그림 3. 방향성을 이용한 중심점 예상구역 찾기.

b. 중심점 예상 구역에서 PoinCare Index[1]를 사용해서 중심점을 찾는다.

c. 찾은 중심점에서 대상 지문의 중심점의 이진 이미지 영역을 이용해서 Moving Window[2] 알고리즘을 사용한다. 픽셀 단위의 매칭을 수행한 후에 그 결과로 일치하는 영역을 찾는다. 일치하는 각각의 영역의 중심좌표를 2개의 지문에서 서로 일치하는 중심점으로 정한다. 그림 4와 같다.

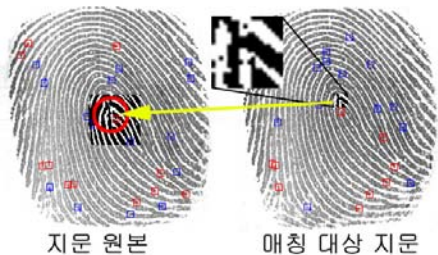


그림 4. Moving Window를 사용한 일치하는 중심점 찾기.

#### 3.2.2 중심점으로부터 각각의 특징점까지의 용선추적

찾은 중심점으로부터 매칭된 특징점까지 선을 긋는다. 선을 따라서 용선을 거칠 때 마다 계수해서 몇 개의 용선을 거쳤는가를 계산한다. 하지만 일반적으로 이미지는 왜곡되어 있는 상태이기 때문에 선 상에 있는 용선의 위치가 왜곡되어 있을 수 있다. 이것을 보정하기 위해서 그림 5와 같이 특징점의 쌍을 잇는 선에 1픽셀 거리차이를 갖고, 평행인 4개의 선을 추가해서 계수를 한다. 그리고 Gaussian 테이블의 값을 이용한 { 1, 2, 5, 2, 1 } 의 가중치를 두어서, 용선을 계수하는 데 있어서 가운데 있는 선일수록 계수하는 데 많은 반영을 하고, 가운데서 멀리 떨어져 있을 수록 적게 반영한다.

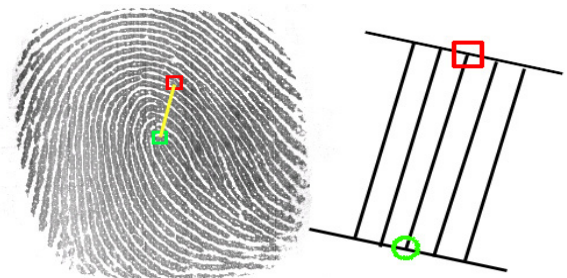


그림 5. 5개의 평행한 선을 따라 용선 추적.

#### 3.2.3 검증 단계

모든 매칭된 쌍에 대해서 용선을 추적한 결과에 따라서 기준값(threshold)을 벗어나는 매칭된 쌍은 제외시킴으로써 검증을 한다.

### 4. 실험결과 및 분석

실험대상 데이터베이스는 표2 와 같이 Veridicom FPS 110 반도체 센서 지문 765개와 Secugen HFDU01 광학센서 지문 1000개를 사용하였다. 실험은 방향성 추출 방법에 따라 나누어서 실험해 보았다.

표 2. 실험 데이터베이스.

타입	반도체 센서(DB1)	광학 센서(DB2)
개수	765	1,000
이미지크기	300 x 300	292 x 248

아래의 결과는 방향성 추출 방법에 따라서 매칭 후에 검증을 한 것과 하지 않은 것으로 나누어서 실험한 결과이다. 그리고 10%와 15%의 에러율을 가질 때로 나누어서 결과를 표시하였다.

표 3. 방향성 추출법과 검증 유무에 따른 타인 수락률.

방향성 추출법	10%		15%	
	No	Yes	No	Yes
Sobel	0.1210	0.0743	0.00636	0.00465
Window Frame	0.2451	0.1592	0.02149	0.01331
Extended Window Frame	<b>0.1023</b>	<b>0.5890</b>	<b>0.00501</b>	<b>0.00450</b>

표 4. 방향성 추출법과 검증 유무에 따른 본인 거부률.

방향성 추출법	10%		15%	
	No	Yes	No	Yes
Sobel	10.0204	10.0195	17.0940	17.0911
Window Frame	10.4115	10.3998	16.6529	16.6458
Extended Window Frame	<b>8.3126</b>	<b>8.2912</b>	<b>14.9959</b>	<b>14.9017</b>

위의 결과와 같이 어떤 방향성 추출법을 사용했을 때라도, 본인 거부률은 큰 변화가 없지만 타인 수락률이 많이 감소했음을 알 수가 있다. 따라서 같은 보완 수준에서 더 낮은 본인 실패 확률을 제공할 수 있으며, 따라서 더 높은 보완 수준이 요구되는 분야에서 사용할 수 있다. 그리고 특히 Extended Window Frame 방법으로 방향성을 추출하였을 때 가장 좋은 결과를 보이고 있다.

표 5. 특징점 추출법과 검증 유무에 따른 본인 거부률.

특징점 추출법	DB	Security Level (FAR)			
		0.001%		0.0001%	
		No	Yes	No	Yes
기본 방법	DB 1	<b>19.80</b>	<b>15.60</b>	<b>26.06</b>	<b>21.75</b>
	DB 2	<b>17.88</b>	<b>14.23</b>	<b>24.35</b>	<b>20.87</b>
융선 추적	DB 1	29.72	25.23	32.94	27.41
	DB 2	28.22	24.14	31.29	27.12
골 추적	DB 1	27.14	21.68	29.35	24.53
	DB 2	25.65	20.52	28.27	22.09

표 6의 결과는 특징점 추출 방법에 따라서 위와 같은 실험을 한 결과이다. 방향성 추출 방법으로는 Extended Window Frame을 사용한 결과가 가장 좋았기 때문에 위의 방법을 사용한 결과만 표시하였다. 보완 수준을 결정하는 타인 수락률이 0.001%, 0.0001%일 때, 각각의 실험에서의 본인 거부률을 표시하고 있다. 역시 마찬가지로 검증을 했을 때 같은 보완 수준에서 타인 수락률이 많이 떨어진 것을 확인할 수 있고, 기본 방법을 사용했을 때 가장 좋은 결과를 보이고 있다. 그리고 융선 추적에 비해서 골 추적이 조금 더 좋은 성능을 보이고 있다. 융선 추적이나 골 추적보다 기본 방법의 성능이 더 좋지만, 속도적인 측면에서는 융선 추적이나 골 추적이 더 빠르기 때문에

사용되는 분야에 따라서 다르게 쓰일 수 있을 것이다. 그리고 DB 1 보다 DB 2 에서 더 좋은 결과가 나온 것은 광학 센서가 반도체 센서 보다 더 좋은 화질을 갖기 때문이다.

## 5. 결론

새로운 검증 방법과 개선된 방향성 추출 방법을 제안하였고, 그 결과는 앞에서와 같이 보완수준을 많이 높여 주었다. 높은 보완 수준을 필요로 하는 분야에서 사용될 수 있을 것이고, 속도를 개선시킨다면 사용자 인증 이외의 신원확인 분야에서도 사용될 수 있을 것이다. 속도 개선이 향후 연구 과제 이다.

## 참고 문헌

- [1] M. Kawagoe and A. Tojo, "Fingerprint Pattern Recognition," *Pattern Recognition*, Vol. 17, No. 3, pp. 295-303, 198
- [2] H. Yahagi, S. Igaki, and F. Yamagishi, "MovingWindow Algorithm for Fast Fingerprint Verification," *Proc. 1990 Southeastcom*, vol. 1, pp. 343-348, 1990.
- [3] Dario Maio, and Davide Maltoni, "Direct Gray-Scale Minutiae Detection In Fingerprints" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 1
- [4] I. G. Bae, B. H. Cho, "On-line Fingerprint Verification System using Direct Minutiae Extraction," *Proceedings of CAINE*, pp. 120-123, Hawaii, November, 2000.
- [5] I. G. Bae, B. H. Cho, "An Automatic fingerprint Verification System using Direct Minutiae Extraction," *Proceedings of KISS*, pp. 553-555, Spring, 2000.