

퍼지 로직을 이용한 지문의 지역적 특성을 효율적으로 반영하는 지문 특징점 추출

Detecting fingerprint features with immediate adaptation to local fingerprint quality using fuzzy logic

이기영 · 김세훈 · 정상갑 · 이광형 · 원광연
한국과학기술원 전자 전산학과

Ki-young Lee, Se-hoon Kim, Sang-gab Jeong, Kwang H. Lee
and Kwangyun Wohn
Department of Computer Science, KAIST

E-mail : kylee@if.kaist.ac.kr, payton@vr.kaist.ac.kr
sgjeong@if.kaist.ac.kr, and {khlee, wohn}@cs.kaist.ac.kr

요약문

본 논문은 지문 이미지에 존재하는 애매함을 퍼지 로직을 이용한 표현으로 기존의 융선 추적법의 단점을 보완한다. 지문의 근방의 질을 퍼지 집합의 상대 크기와 근방 명암의 분산을 이용하여 판단한 후 근방의 지문의 질이 좋고 나쁨에 즉각적으로 다른 방법을 사용하여 지문의 융선을 추적하는 새로운 융선 추적법을 제안 설계한다.

KeyWords : Fingerprint, Fuzzy logic, Cardinality, Edge following, Adaptation

1. 서론

현대에는 컴퓨터 및 통신망의 급속한 보급과 전자 상거래의 확산에 따라 보안의 중요성이 크게 부각되고 있다. 기존의 개인 식별 번호나 암호 등은 누출, 분실, 망각, 도용 등의 위험이 존재하므로 이러한 위험으로부터 해방된 다른 방법의 보안 대책이 요구되고 있다. 이러한 보완책들 중 지문인식이 최근에 각광받고 있다[10].

자동화된 지문 인식 시스템은 지문에 존재하는 여러 특징점들의 위치 및 방향 성분들을 추출하여 이러한 특징점들의 정합 정도로써 개인을 식별이나 검증을 하게 된다[9]. 지문의 특징점으로는 단점, 분기점, 삼각주, 교차점, 짧은 융선, 고립점 등이 있으나 자동화 시스템에서는 주로 단점이나 분기점을 사용한다. 그 이유는 다른 특징점들이 이것들로 표현 가능하기 때문이다[2]. 따라서 이러한 지문 인식 시스템에서 관건은 단점 및 분기점 등의 특징점들의 방향성 및 위치 정보를 빠르고 정확하게 추출할 수 있는 방법론의 개발에 달려 있다[9].

특징점 추출을 바탕으로 지문 정합을 시도하는 시스템은 사용 목적이나 의도에 따라 고전적 방법, 융선 추적 방법, 퍼지 로직을 사용한 방법 등이 있다[1][3][4][5][6][7]. 고전적인 방법은 지문 이미지로부터 전처리, 방향성 추출, 이진화, 세선화, 후처리를 거쳐 특징점을 추출한 후 이러한 특징점으로 지문 정합을 한다[8]. 그러나 이 방법은 과도한 시간적 소요와 각 단계마다 정보의 손실이 존재할 수 있는 단점이 있다[1][3].

이에 대한 보안책으로 융선 추적 방법이 고안되었다. 이 방법은 융선 방향을 추적하면서 바로 특징점을 추출하여 위에서 언급된 문제점을 해결하고자 하였다[1].

그리고 지문의 이미지에 존재하는 복잡성과 애매함을 처리하기 위해서 퍼지 로직을 사용한 방법이 제안되었다[3].

본 논문에서는 기존의 융선 추적 방법의 단점을 보완하기 위해 퍼지 로직을 이용하여 지문 근방의 질을 즉각적으로 반영하여 지문 이미지로부터 바로 특징점을 추출하는 새로운 융선 추적법을 제안한다. 즉, 지문의 융선을 추적하면서 현재 위치된 융선의 질을 퍼지 로직을 사용한 로컬 윈도우로 판단하여 지문의 이미지로부터 바로 특징점을 추출한

감사의 글 : 본 연구는 AITrc 프로젝트에 의해 지원받았습니다.

다.

본 논문의 구성은 2절에서 기존의 방법들에 대해서 알아보고 3절에서는 제안하는 시스템에 대해서 설명한다. 그리고 4절에는 실험의 결과를 보이고, 마지막으로 5절에서는 본 논문의 결론을 맺는다.

2. 기존 관련 연구

자동화된 지문 인식 시스템은 특징점을 추출하는 방법에 따라 고전적인 방법, 음선을 추적하는 방법, 퍼지 로직을 이용한 방법 등으로 나눌 수 있다.

가장 일반적으로 사용되는 고전적인 방법은 입력된 지문 이미지로부터 이미지의 질에 따라 필터링 등 필요에 따라 전처리를 수행하고[5][7], 그 결과로 방향성 추출[6][8] 및 각 점의 값을 0 또는 1로 바꾸는 이진화 수행하고[5], 그 이진화된 이미지로 지문의 음선의 두께를 1로 바꾸는 세선화를 거친다[8]. 이런 세선화된 결과에서 특징점을 추출하고[6] 휴리스틱 방법을 적용해서 의사 특징점을 배제한[11] 후 추출된 특징점을 근거해서 지문 정합을 수행한다[12].

그러나 이 방법은 필터링 및 세선화 등에서 과도한 시간이 소요되며 이진화를 비롯해 각 단계를 거치면서 많은 정보의 손실을 야기하며, 지문의 질이 좋지 않을 경우 잘못된 결과를 도출할 수 있는 문제점이 있다[1].

그에 대한 보완책으로 음선 방향을 따라 가면서 바로 특징점을 추출하는 음선 추적 방법이 제안되었다. 이 방법은 방문하지 않은 음선을 찾은 후 그 음선 방향으로 몇 개의 화소의 산술평균과 음선의 수직 방향으로 고정된 가중치를 부여해서 음선 가운데 화소를 찾는다. 그리고 이 화소를 기준으로 고정된 길이 만큼 떨어진 다음 점을 찾는 과정을 종료 조건에 부합 할 때까지 반복한다. 이러한 방식으로 지문의 이미지로부터 바로 특징점을 추출하는 방법을 사용한다[1].

이 방법은 지문의 이미지로부터 이진화 및 세선화를 생략시켜 시간적인 향상을 이끌어 내었고 지문의 이미지로부터 바로 특징점을 추출함으로써 여러 단계를 거치면서 발생할 수 있는 정보의 유실을 막아 정확한 특징점을 찾을 수 있는 장점이 있다.

그러나 이 방법은 음선 가운데 화소를 찾는 과정에서 지문 음선의 방향성 계산에서 발생하는 에러와 음선 방향의 수직 방향으로 가중치를 적용 시 공간적인 에러로 인하여 음선의 가운데 화소를 찾는데 오류가 있을 수 있고 지문에 존재하는 구조적인 특징의 고려가 미비하다.

다른 한편으로는 퍼지 로직을 이용한 접근 방법이 있는데 이 방법은 지문에 존재하는 밝음(음선)과 어두움(골)을 퍼지 언어항을 사용해서 모델링한다. 그리고 지문의 이미지에 사각형이나 육각형 또는 팔각형의 고정된 윈도우 마스크를 씌워서 윈

도우 마스크의 각 부영역별 밝고 어두움을 이용해서 퍼지 규칙을 적용시켜 지문의 이미지로부터 바로 특징점을 찾는 방법이 제안되어있다[3].

이 방법도 지문의 이미지로부터 바로 특징점을 추출함으로써 여러 단계를 거칠 때 발생하는 정보의 유실을 제거하였다. 또한 지문의 이미지 자체에 존재하는 애매함을 수학적으로 잘 고려해 줄 수 있는 장점이 있다. 그러나 고정된 윈도우 마스크 사용으로 인한 음선과 음선 사이의 겹침 현상을 고려해 주기가 어렵고 퍼지 추론시 어느 정도 시간적인 부담이 증가 될 수 있다.

3. 제안된 시스템

본 논문은 지문 음선에 윈도우 마스크를 씌워서 그 부분을 퍼지 집합으로 표현하는 방법을 정의하고, 이 표현된 집합의 상대 크기로 음선의 질을 판단하여 음선을 추적하는 새로운 음선 추적법을 제안한다.

3.1 절에서는 기본 아이디어를 설명하고 3.2절에서는 시스템의 전체 구조를 그리고 3.3절에서는 세부 알고리즘을 설명한다.

3.1 음선 상태와 퍼지집합의 상대 크기

[그림 1]과 같은 확대된 지문의 음선이 존재한다고 하자. 이때 지문 이미지 전체적인 정보나 지역적인 정보를 이용해서 멤버쉽 함수 $f(x)$ 를 도출하면 각 화소마다 음선에 속할 정도를 구할 수 있다.

그리고 [그림 2]에서 보는바와 같이 지문 음선 위에 윈도우를 씌운 후 각 화소마다 위에서 사용한 방법을 적용시키면 위치정보와 음선에 속할 정도를 쌍으로 하는 원소들의 퍼지 집합으로 표현된다.

$$A = \sum_{i=1}^{M^2} \mu_A(x_i)/x_i, \quad (식1)$$

x_i : 명암, $\mu_A(x_i)$: 음선에 속할 정도

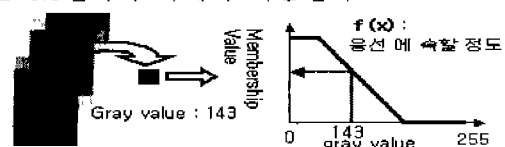
완성된 퍼지 집합으로 α -cut한 후 집합(A_α)의 상대 크기($Card_{(i,j)} : \|A_\alpha\|$)를 구할 수 있다.

$$A_\alpha = \{x_i \in X \mid \mu_A(x_i) \geq \alpha\}, \quad X = [0, 255] \quad (식2)$$

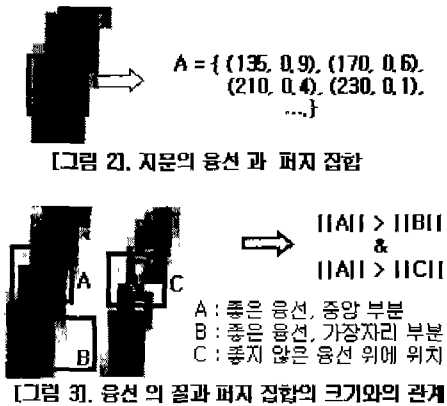
$$\|A_\alpha\| = \frac{|A_\alpha|}{|A_0|}, \quad |A_\alpha|: A_\alpha \text{의 크기}, \alpha \in [0, 1] \quad (식3)$$

이 때 좋은 음선의 중간 부분에 윈도우 마스크가 놓여 있는 경우는 그렇지 않은 경우의 상대 크기보다 크게 나타난다[그림 3].

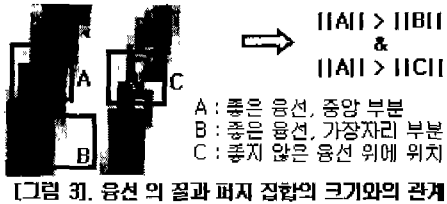
이 때 음선에 속할 정도를 결정하는 멤버쉽 함수와 상대 크기에 영향을 미치는 윈도우 마스크의 크기에 대한 결정이 중요한 요소로 고려된다. 이 문제는 3.3절에서 자세히 기술한다.



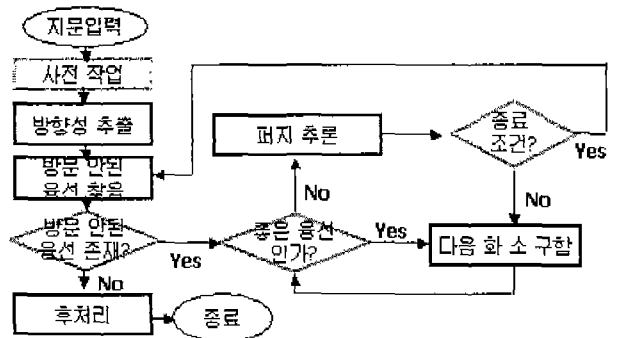
[그림 1]. 지문의 음선과 멤버쉽 함수 관계



[그림 2]. 지문의 용선과 퍼지 집합



[그림 3]. 용선의 결과 퍼지 집합의 크기와의 관계



[그림 4]. 제안된 방법의 시스템 구조

3.2 제안된 시스템의 전체 구조

제안된 시스템의 전체적인 구조는 [그림 4]와 같다.

지문 이미지를 입력받은 후 다음 단계들에서 사용될 근방의 명암의 평균 및 분산 등 몇 가지의 사전 작업을 수행하고 각 부영역별 대표 방향성을 추출한다. 이 작업이 끝난 후 지문 이미지를 스캔해서 방문하지 않은 새로운 용선을 찾는다. 그 후 용선의 질 판단과 용선 대표점 추출을 위 3.1절에서 정의한 방법에 따라 퍼지 집합으로 표시한 후 이 퍼지 집합의 $Card_{(i,j)}$ 와 $Vari_{(i,j)}$ 를 이용하여 결정한다. 이 때 좋은 용선으로 판명 날 경우 가변 길이 만큼 떨어진 다음 화소를 구하고 위의 작업을 반복한다. 만약 용선의 질이 좋지 않거나 특징점이 될 수 있을 시 퍼지추론을 사용하여 좀 더 정확한 판단으로 종료 조건에 속하는지 결정하고 종료 조건에 속할 시 다시 새로운 방문 안한 용선을 찾고 위의 작업을 반복한다.

3.3 각 모듈별 알고리즘 설계

3.3.1 사전 작업

여기서는 다음 단계들에서 사용되어지는 각 화소를 중심으로 하는 정사각형의 윈도우 마스크를 씌워서 그 윈도우 마스크의 명암값의 평균($LM_{(i,j)}$), 이 평균과 각 화소의 명암값과의 차이($Differ_{(i,j)}$), 그리고 $Differ_{(i,j)}$ 를 바탕으로 하나의 화소당 평균 명암값의 분산($Vari_{(i,j)}$)을 아래의 식을 이용해서 구한다.

$$LM_{(i,j)} = \sum_{(m,n) \in W} gray_{(m,n)} / N^2, \quad (식4)$$

$(m,n) \in LocalWindow, N^2$: 윈도우안의 화소수

$$Differ_{(i,j)} = LM_{(i,j)} - gray_{(i,j)}, \quad (식5)$$

$$Vari_{(i,j)} = \sum_{(m,n) \in W} |Differ_{(m,n)}| / N^2, \quad (식6)$$

그리고 Max-Vari, Min-Vari, TM을 다음의 식으로 구한다.

$$Max - Vari = \max_{(i,j) \in I} \{ Vari_{(i,j)}, I, Image \} \quad (식7)$$

$$Min - Vari = \min_{(i,j) \in I} \{ Vari_{(i,j)}, I, Image \} \quad (식8)$$

$$TM = \sum_{(m,n) \in I} gray_{(m,n)} / N^2, \quad (식9)$$

$(m,n) \in I, N^2$: 이미지 안의 전체 픽셀 수

이 때 윈도우를 슬라이딩시키면서 새롭게 첨가되는 열과 제거되는 열만 계산함으로써 계산의 양을 줄일 수 있다. 이 때 윈도우 마스크의 크기(N)를 용선의 굵기(R)와 골의 굵기(V)의 합, 즉 용선의 주기(T)로 잡았을 때 가장 좋은 결과를 도출한다.

그 이유는 밝고 어두움은 주위의 화소와의 상대적인 값의 차이로서만 의미가 있고 윈도우 마스크의 크기가 작으면 작을수록 주위의 상황을 잘 판단할 수 있는 좋은 기준을 제시해 준다. 또한 최소한 용선의 굵기와 골의 굵기를 합한 것보다는 커야만 의미가 있기 때문이다.

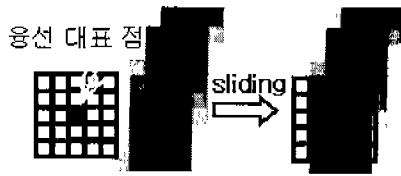
$$N = R + V = T, \quad (식10)$$

3.3.2. 부영역별 방향성 추출

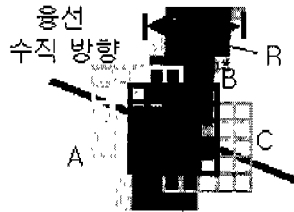
지문에는 일정한 방향성이 있고 또한 특정 부영역별의 방향성은 모두 같다. 이러한 일정한 방향성을 추출하는 기존 방법들 중 시간대 성능비가 우수한 소벨 방정식을 사용한다. 그리고 각 부영역별 방향성은 모두 같다는 가정 하에서 부영역에 속하는 모든 화소들에 대한 소벨 방정식으로 방향성을 구한 다음 가장 많은 빈도의 방향으로 그 부영역의 대표 방향으로 선정한다. 그리고 좀더 정확한 방향성을 계산하기 위해서 특정 부영역의 방향성을 구할 때 좀더 넓은 범위의 화소를 계산한다.

3.3.3. 방문하지 않은 용선 찾기

지문 이미지를 스캔하면서 방문하지 않은 새로운 용선을 찾는다. 이 때 적용하는 방법은 아래의 [그림 5]과 같이 윈도우를 마스크를 슬라이딩시키면서 윈도우 안의 각 화소에 대해 위 3.1절에서 정의한 방법을 사용하여 용선 대표점을 찾는다. 이 때 용선 대표점이란 용선을 용선 수직 방향으로 절단했을 시 용선의 가장 가운데 점을 말한다. 그리고 이



[그림 5]. 방문하지 않은 용선 찾기



[그림 6]. 용선 대표 점 찾기

것은 고전적인 접근 방법에서 용선이 세선화 되었을 때의 픽셀을 의미한다. 그리고 만약 더 이상 방문하지 않은 용선이 없을 경우 후처리를 한다.

3.3.3. 용선의 질 판단

용선의 질 판단에 앞서 먼저 윈도우 마스크를 용선의 가장 가운데 위치하도록 만든다. 이것은 [그림 6]에서 보는바와 같이 윈도우 마스크를 용선의 수직 방향으로 용선의 굵기(R)의 반의 크기로 좌우로 이동시키면서 가장 큰 퍼지 집합의 상대 크기를 갖는 곳을 찾는다.

그 후 현재 위치의 용선의 질 판단은 위 3.1에서 정의한 방법을 사용한다. 이 때 Vari_(i,j)값 및 Card_(i,j)값이 어느 수준 이상이면 현재의 위치는 좋은 용선의 가운데 위치한 결과가 된다.

그러나 용선의 질이 좋지 않을 경우 좀더 많은 요소를 고려한 퍼지 추론을 하게 된다.

이 때 중요한 이슈는 윈도우 마스크의 크기(W)와 용선에 속하는 정도를 나타내는 멤버쉽 함수(f(x))를 결정하는 방법이다. 이 때 W는 용선 대표 점을 찾기 위해선 홀수이어야 하고 가장 합당한 크기는 용선 굵기(R)와 같아야한다. 지문의 이미지는 일반적으로 비슷한 굵기의 용선으로 구성되어 있으므로 본 연구에서는 5x5 크기의 윈도우를 사용했고 또한 윈도우 마스크를 지문의 수직 방향으로 움직일 때 집합의 상대 크기의 변화량의 차이에 따라 윈도우의 크기를 조정(3x3, 7x7)할 수 있도록 했다.

f(x)를 결정하는 문제에서 사용한 방법은 [그림 7]과 같이 미리 구한 LM_(i,j)와 Vari_(i,j)를 이용하여 결정한다. 즉 지문의 이미지가 밝고 어두운 정도는 주위의 이웃의 명암과의 상대적인 차이에만 관여하고[13], Vari_(i,j)는 각 화소 당 평균적 명암 차이를 나타내므로 이 값을 기준으로 하여 근방의 멤버쉽 함수를 구한다.

$$W = R, \quad (\text{식 11})$$

$$f(x) = \begin{cases} 1 & , x < LM_{(i,j)} - Vari_{(i,j)} \\ 0 & , x > LM_{(i,j)} + Vari_{(i,j)} \\ \frac{(LM_{(i,j)} + Vari_{(i,j)} - x)}{2Vari_{(i,j)}} & , \text{otherwise} \end{cases}$$

(식 12)

3.3.4. 다음 화소 구하기

기존의 연구에서는 용선을 따라 가면서 다음 화소를 구할 때 고정된 길이 만큼 떨어진 점을 구했다[1]. 그러나 지문의 용선은 일반적으로 규칙성을 가지고 일정한 방향으로 뻗어 있으므로 이러한 구조적인 특성을 고려해서 가변 길이를 적용시킬 수 있다. 이러한 가변 길이는 시간과 성능 면에서 좀더 뛰어난 결과를 이끌어 낼 수 있다. 이 때 사용하는 구조적 특성은 용선의 각도의 변화량을 사용한다. 즉 아래의 [그림 8]과 같이 "거리2"가 "거리1"보다 클 수 있다. 이 때 다음 화소는 용선 내부에 존재해야 된다.

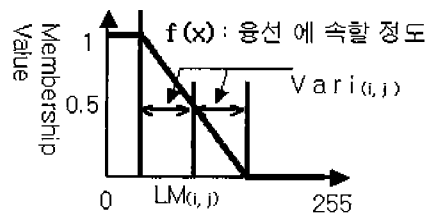
3.3.5. 퍼지 추론 및 종료 조건

용선의 질이 좋지 않을 때에는 좀더 많은 정보로 퍼지 추론을 하게 되는데 여기에 사용되는 정보로는 LM_(i,j), Vari_(i,j), TM 그리고 Card_(i,j)이다. 그리고 아래와 같은 퍼지 규칙을 사용하여 종료 조건에 속하는지 알아 본다.

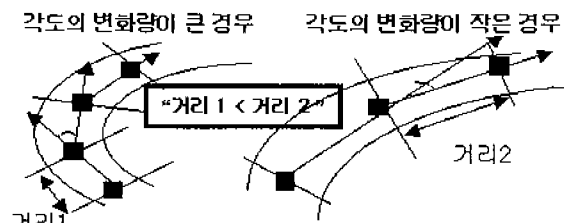
종료 조건으로는 1)지문의 이미지를 벗어난 경우, 2)백그라운드 도착한 경우 3) 끊어진 경우 4) 지나간 용선과 만났을 경우이다.

그러면 다음과 같은 퍼지 규칙을 만들 수 있다.

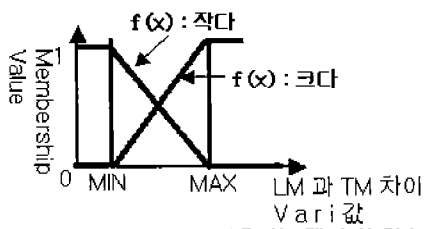
- A) If Vari_(i,j)가 충분히 작고, LM_(i,j)가 TM 보다 충분히 크면 then 백그라운드.
- B) If Vari_(i,j)가 충분히 작고, LM_(i,j)가 TM 보다 충분히 작으면 then 잉크가 많이 묻은 곳.
- C) If Vari_(i,j)가 충분히 크고, Card_(i,j)가 충분히 작으면 then 끊어진 점.



[그림 7]. 용선에 속할 멤버쉽 함수



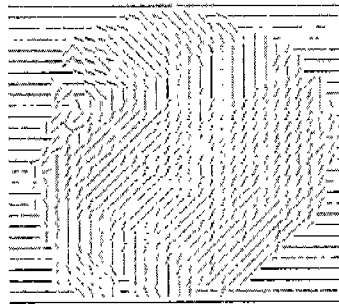
[그림 8]. 각도 변화량에 따른 가변 길이 사용



[그림 9]. 퍼지 추론에 사용되는 멤버쉽 함수



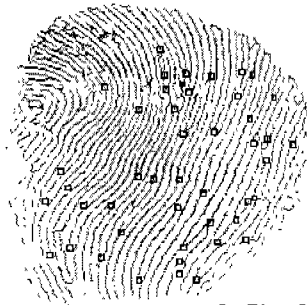
[그림 10].



[그림 11].



[그림 12].



[그림 13].

[그림 10 - 13]. 실험 결과

이 때 A)와 C)의 경우는 종료될 하게 되고 B)의 경우에는 그 지점을 의사 특징점의 가능성과 함께 마크를 한 후 계속 다음 화소를 구하게 된다.

여기에 사용되는 '작다'와 '크다'의 퍼지 언어항에 대한 멤버쉽 함수는 사전 작업에서 $Vari(i, j)$, $LM(i, j)$ 의 최대 최소를 이용해서 [그림 9]과 같이 멤버쉽 함수를 미리 정의한 것을 사용한다.

3.3.6. 후처리

앞 단계에서 추출된 특징점에는 잡영이나 지문 자체의 상처 그리고 잉크의 양에 따라 의사 특징점이 발생 할 수 있다. 이러한 의사 특징점을 아래의 휴리스틱 방법으로 제거 한다.

- 1) 일정 거리 안에 있는 같은 방향의 단점 제거
- 2) 일정 방향 안에 있는 같은 방향의 분기점 제거
- 3) 분기에서 융선이 일정 거리 이하일 경우
- 4) 두 평행한 융선사이에 존재하는 특징점 제거

4. 실험 및 결과

시스템은 자바를 사용하여 구현하였고 실험 환경은 펜-3 850Mhz의 사양에 Window 2000 Pro.의

O.S.에서 실험을 하였다.

실험 데이터는 10명(남자 7명, 여자 3명)의 사람의 검지를 한 사람 당 엄지를 5번씩 찍어 총 50개의 지문 데이터를 사용했다.

실험 결과는 [그림10-13]과 같다. [그림10]에서 보는 바와 같이 지문 이미지의 질이 좋지는 않다. 또한 지문의 상단 부분에는 잉크가 적게 묻어서 잘 나오지 않은 부분이 존재한다. 이러한 지문의 이미지를 사용했을 때 각각의 결과는 그림에 나와 있다. [그림 11]는 지문의 이미지로부터 지문의 방향성을 추출한 것이고 [그림12]는 실제로 윈도우 마스크가 지문의 융선을 따라간 흔적을 나타낸 것이다.

그리고 [그림 13]는 모든 단계를 거친 후 특징점을 찾은 결과를 나타낸 것이다.

그리고 [표 1]은 50개의 지문에 대한 각 이미지의 특징점 마다 평균 찾지 못한 특징점의 개수와 찾은 의사 특징점의 수를 나타낸 것이다.

	Miss된 특징점 수	의사 특징점 수
남자	0.02	0.03
여자	0.025	0.02

[표 1] - 실험 결과

5. 결론 및 향후 과제

본 논문에서는 퍼지 로직을 이용해서 융선의 질을 퍼지 집합의 상대 크기로 나타내어 융선의 질에 따라 각각 다른 방법을 적용해서 융선을 따라가는 새로운 융선 추적 방법을 제안하고 구현한 후 실험을 하였다.

현재는 많은 실험을 하지 않은 상태인데 앞으로 좀더 실험을 하고 기존의 다른 방법들과 비교 실험을 통해 결과를 분석하고 좀더 지문이 질이 떨어질 경우에도 특징점을 잘 찾을 수 있도록 퍼지 추론 및 후처리를 강화하여 지문 정합에 사용해야 할 것이다.

참 고 문 헌

- [1] Dario Maio, Davide Maltoni, "Direct Gray-Scale Minutiae Detection In Fingerprints," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 1, pp. 27-40, 1997.
- [2] Federal Bureau of Investigation, "The Science of Fingerprints : Classification and Uses," Washington DC, 1984.
- [3] V. K. Sagar, "Fuzzy Control for Fingerprint Feature Selection," ACCV'95 Conference, vol. 3, pp. 767-771, Dec 5-8, 1995.
- [4] M.R. Verma, A.K. Majumdar and B. Chatterjee, "Edge Detection in Fingerprints," Pattern Recognition,

vol. 20, no. 5, pp. 513-523, 1987.

[5] Louis Coetzee and Elizabeth C. Botha, "Fingerprint Recognition In Low Quality Images," Pattern Recognition, vol. 26, no. 10, pp. 1441-1460, 1993.

[6] Nalini K. Ratha, Shaoyung Chen and Anil K. Jain, "Adaptive Flow Orientation-Based Feature Extraction In Fingerprint Images," Pattern Recognition, vol. 28, no. 11, pp. 1657-1672, 1995.

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

[8] 김현, "RSTI-Invariant Feature Volume Extraction and Recognition of Fingerprints and Their Applications," Inha University, M.S. Thesis, 1998.

[9] Nalini K. Ratha, Shaoyun Chen and Anil K. Jain, "Adaptive Flow Orientation-Based Feature Extraction In Fingerprint Images," Pattern Recognition, vol. 28, no. 11, pp. 1657-1672, 1995.

[10] E. Newham, "The biometric Report," New York, NY : SJB services, 1995.

[11] Qinghan Xiao and Hazem Raafat, "Fingerprint Image Postprocessing : A Combined Statistical and Structural Approach," Pattern Recognition, vol.24, no. 10, pp. 985-992, 1991.

[12] Nalini K. Ratha, Shaoyun Chen and Anil K. Jain, "A Real-Time Matching System for Large Fingerprint Databases," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.18, no.8, pp. 799-813, 1996.

[13] Maria Petrou and Panagiota Bosdogianni, "Image Processing : The Fundamentals," John Willey & Sons Ltd, 2000.