

인근 특징 정보를 이용한 임베디드용 지문인식 알고리즘

정회원 박태근*, 정선경**

Embedded Fingerprint Verification Algorithm Using Various Local Information

Tea geun Park*, Sun kyung Jung** *Regular Members*

요약

본 논문에서는 지문의 영상 품질, 특징점의 구조, 융선의 주파수와 방향성 등의 정보들을 이용하여 추출된 특징점들을 기반으로 임베디드 시스템에 적합한 지문인식 알고리즘을 제안하였다. 추출된 특징점들은 전처리 및 후처리 과정을 통해 참 특징점만을 선택함으로써 정합과정의 신뢰도를 높였다. 정합과정에서는 지문 획득시 천이, 회전, 늘림 등으로 인한 오차를 고려함으로써 신뢰도를 개선하였으며, 인근 특징점만의 부분 정합을 통하여 전체정합을 위한 참조쌍을 택함으로써 수행시간을 단축하였다. 제안된 지문인식 알고리즘은 Arm920T 프로세서 환경하에서 구현되어 검증되었으며 실시간처리를 위한 다양한 방법이 적용되었다. 지문 등록부터 인식까지 처리시간은 0.541초로 실시간 임베디드 응용분야에 적용 가능함을 보여주었으며 이때 FRR(본인 거부율)과 FAR(타인 승인율)은 각각 0.079과 0.00005로 높은 신뢰도를 갖는다.

Key Words : biometrics, fingerprint verification, embedded systems

ABSTRACT

In this paper, we propose a fingerprint verification algorithm for the embedded system based on the minutia extracted using the image quality, the minutia structure, and the frequency and the orientation of ridges. After the pre- and the post-processing, the true minutia are selected, thus it shows high reliability in the fingerprint verification. In matching process, we consider the errors caused by shift, rotation, and pressure when acquiring the fingerprint image and reduce the matching time by applying a local matching instead of a full matching to select the reference pair. The proposed algorithm has been designed and verified in Arm920T environment and various techniques for the realtime process have been applied. Time taken from the fingerprint registration through out the matching is 0.541 second that is relevant for the realtime applications. The FRR (False Reject Rate) and FAR (False Accept Rate) show 0.079 and 0.00005 respectively.

I. 서론

정보사회로 발전할수록 교류되는 정보의 양은 급격히 많아지고 신원확인의 정확성과 신원정보 보안

이 더욱 중요해지고 있다. 이를 위하여 얼굴인식, 홍채인식, 지문인식, 정맥인식 및 음성인식 등의 생체인식 기술과 그의 응용분야가 더욱 확대되고 있다. 그 중에서도 지문은 모든 사람이 교유의 문양을

* 가톨릭대학교 정보통신전자공학부 (parktg@catholic.ac.kr), ** 넥서스칩스 (powerjisk@nexuschips.com)

논문번호 : KICS2004-10-250, 접수일자 : 2004년 10월 27일

※본 연구는 2004년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

가지고 평생토록 변하지 않는 것으로 알려지고 있으며, 지문인식을 위한 알고리즘이 상대적으로 간단하고 정확하며 저가의 시스템으로 구현 가능하다는 장점으로 많이 연구되고 있다^{[1][6]}. 초기에 지문은 범죄수사에 이용되었으나 최근에는 출입관리, 근태, 금융, 신분확인 등으로 다양하게 그의 응용분야를 넓히고 있다. 그러나 여러가지 응용분야에 적용하려면 지문 인식 과정에서 영상처리(전처리와 후처리) 및 특징점 추출, 정합 등의 효과적인 알고리즘이 필수적이다. 이에 대한 실시간 처리와 신뢰도 향상 등 성능 개선 및 실시간 임베디드 시스템에 대한 연구가 진행되고 있다^[9].

다른 사람과 구별되어지는 대표적인 정보는 지문의 끝점과 분기점의 위치, 방향, 종류를 특징량으로 하는 특징점 기반 방법^{[1][6]}과 필터링 결과 패턴으로 정합하는 필터 기반의 방법^{[7][8]} 등이 있다. 많이 사용되는 특징점 기반 방법에는 특징량을 검출하는 과정으로 회색조 영상에서 바로 용선을 따라 끝점과 분기점을 검출하는 방법^[1]과 전처리 후 얻은 세션화 영상에서 용선의 교차점 수로 찾아내는 방법이 있다. 전처리 없이 바로 용선 추적으로 특징점을 찾는 방법^[2]은 전처리를 하지 않는다는 장점을 가지고 있지만, 채취영상의 품질 저하에 따른 용선의 끊어짐과 배경과 지문의 경계에서 끊어짐으로 용선 추적이 어렵고 추적과정에서 많은 샘플링이 요구된다. 많이 사용되는 세션화 영상 기반의 방법은 획득된 영상의 품질뿐만 아니라 전처리 과정의 정확한 용선 추출이 중요하다.

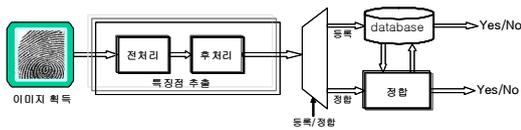


그림 1. 지문인식시스템
Fig 1. Fingerprint verification system

지문인식 시스템은 그 사용이 편리하기 때문에 가장 보편적으로 응용되고 있다. 다대일 시스템은 방대한 양의 기존에 구축된 데이터베이스에서 현재 입력된 지문과 가장 유사한 지문을 골라내는 조회 시스템이며, 일대일 시스템은 기존에 구축된 지문 데이터베이스에서 인증 받을 사람이 자신의 ID 등을 알려주고 본인의 지문인지를 판단해주는 인증 시스템이라 할 수 있다. 그림 1의 지문인식 시스템은 지문영상으로부터 추출된 특징점을 저장하는 등록단계와 등록된 지문과 입력지문의 동일 여부를

검사하는 인증단계로 구성된다. 특징점 기반 방법에서 지문의 끝점과 분기점을 추출하기 위해서는 회색조지문 영상을 이진화하고, 용선 두께가 한 픽셀을 갖도록 세션화하여 개인 지문고유의 정보인 특징점(끝점과 분기점)을 추출한다. 이렇게 추출된 정보는 등록되거나 이미 등록된 특징점과 비교하여 동일여부를 판별하는 인증에 사용된다.

본 논문에서는 인근 특징 정보를 이용한 임베디드용 지문인식 알고리즘을 제안한다. 획득된 지문 영상으로부터 용선 주파수와 방향 정보를 추출하여 영상을 이진화한 후 지문의 특징점들을 추출한다. 이 후, 획득된 지문 영상의 품질 정보를 체계적이고 효율적으로 이용하여, 세 단계에 걸쳐 효과적으로 참특징점과 의사특징점을 구별하여 참특징점을 의사특지점으로 오인하는 오추출률을 최소화하면서 정추출률을 높이는 효율적인 후처리과정을 거친다. 지문 정합 시에는 모든 가능한 쌍을 참조 쌍으로 하여 전체 정합을 시도하는 것은 실시간 시스템에는 적합하지 않으므로 부분정합을 통해 후보 참조쌍을 구하여 전체 정합을 수행하였다. 248×292 영상크기를 갖는 다양한 품질의 지문 영상에 대해, 알고리즘 적용에 따른 오추출률과 정추출률의 변화와 타인을 승인하는 FAR과 본인을 거부하는 FRR의 변화를 통해 후처리 알고리즘의 필요성과 성능을 검증하였을 때, FRR(본인 거부율)과 FAR(타인 승인율)은 각각 0.079과 0.00005로 신뢰성을 갖는다. 또한 제안된 알고리즘을 Arm920T에 포팅하였을 때 지문 정보 추출 및 정합까지 걸린 시간은 0.541초로 실시간 임베디드 응용분야에 적용 가능함을 보여주었다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 지문인식 알고리즘 중에서 획득된 지문 영상에 대한 전처리 및 후처리 과정을 살펴보고, 3장에서는 추출된 특징점을 바탕으로 지문인식 알고리즘을 설명한다. 4장에서는 임베디드 지문인식 시스템 및 데이터 베이스 구조를 설명하고 5장에서는 실험을 통해 알고리즘의 성능을 분석하고, 6장은 본 논문의 결론을 서술한다.

II. 지문영상에 대한 전처리/후처리 알고리즘

2.1 전처리 알고리즘

각 픽셀의 영상 밝기 정보를 기준 평균과 분산값으로 용선과 골의 구조는 바뀌지 않고, 용선과 골을 따라 변화의 폭을 줄이면서 골과 용선의 대조를 분명히 하는 정규화 과정을 수행한다. 정규화된 영상

으로 일정크기를 갖는 블록으로 나누어 용선방향과 용선주파수를 계산한다. 블록의 용선 방향은 계산 복잡도가 낮은 Sobel 연산자를 이용하였고, 그렇게 구한 용선의 방향은 잡음에 의한 오차를 줄이기 위해 이웃블럭 방향정보를 이용하여 필터연산을 추가 수행한다. 블록의 용선주파수 계산 단계는 각 블록을 중심으로 일정크기의 윈도우에 대해 그 블록의 용선방향으로 윈도우를 회전, 용선의 방향으로 투영하여 일차원 파형으로부터 봉우리를 찾고 봉우리간의 평균거리를 계산하여 그 역수를 그 블록 주파수로 한다. 이것 또한 블록간 주파수 보정단계를 거친다.

이진화 단계는 앞서 구한 블록 방향정보와 주파수정보를 이용하여 가보필터 계수를 결정하여 각 픽셀에 대해서 필터링한 후, 그 결과의 부호에 따라 0과 1을 결정하는 단계이다. 이 가보필터는 주파수와 방향 선택적 필터로, 그림 2에서와 같이 용선과 골을 보호하면서 잡음을 제거하는 효과를 갖는다. 영상 밝기값을 임계치로 이진화하는 방법에 비해 그 결과가 우수하다. 이진화 영상의 용선을 한 픽셀 두께의 용선으로 만드는 세션화 단계를 거쳐, 그 영상으로부터 교차점의 수를 세서 끝점과 분기점을 추출한다.



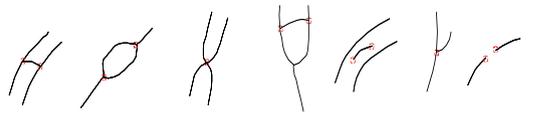
(a) 원 영상 (a) Original image (b) 이진화 영상 (b) Binarized Image
 그림 2. 가보필터를 이용한 이진화 결과 영상
 Fig 2. Binarized result image using Gabor Filter

2.2 후처리 알고리즘

후처리 과정은 전처리 단계에서 추출된 특징점 중에서 지문 영상의 왜곡, 전처리 과정에서의 정보 손실 등 다양한 원인으로 생성된 의사특징점을 제거하는 단계이다. 이런 의사특징점은 지문인식 시스템의 성능을 저하시키고, 또한 특징점의 수 증가로 매칭 시간과 데이터베이스 공간을 증가시킨다. 참특징점의 비율을 최대한으로 향상시키기 위해 특징점 각각에 대해 특징점의 구조적 특징, 지문의 고유한 특성, 그리고 획득된 지문 영상의 품질 정보 등을 이용하여 세단계를 거쳐 의사특징점을 제거한다.

2.2.1 이웃하는 특징점들간의 관계에 의한 의사 특징점 제거

잡음이 섞인 영상에 대한 세션화 결과영상에서 추출한 후보특징점 중에는 그림 3에서 정의한 패턴을 갖는 의사특징점들이 많이 포함되어있다. 후보특징점들 간의 거리와 방향정보, 그리고 용선 간의 연결 정보로부터 정의된 패턴에 해당하는 특징점을 의사특징점으로 분류한다.

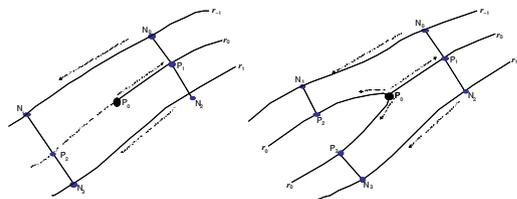


(a)다리, (b)원형구조, (c)십자가구조, (d)삼각형구조, (e)단선, (f)잔가지, (g)절선
 (a)Bridge, (b)Hole, (c)Cross, (d)Triangle, (e)Broken ridge, (f)Short branch, (g)Broken ridge
 그림 3. 의사특징점 패턴 정의

Fig 3. Pattern definitions for the false minutiae

2.2.2 특징점을 이루는 용선과 이웃 용선을 이용한 참특징점 추출

특징점은 세션화 영상에서 분기점과 끝점을 포함하는 용선과 그 이웃 용선이 그림 4에서와 같이 이루어져 있다. 이 구조를 갖는 특징점을 참특징점으로 분류한다. 이 때, 원 영상에서 용선의 구성으로 판단하는 것 보다 세션화 영상을 이용하는 것이 참특징점을 의사특징점으로 오인하는 일을 줄일 수 있다.



(a) 끝점 (a) Endpoint (b) 분기점 (b) Bifurcation

그림 4. 끝점과 분기점의 용선
 Fig 4. Endpoint and bifurcation ridges

2.2.3 특징점 주위의 영상 품질과 지문특성에 따른 의사 특징점 제거

앞 두 단계에서 분류되지 않은 후보특징점을 대상으로 주위의 영상 품질과 지문특성 정보를 이용하여 참과 의사특징점을 가려낸다. 배경과 저 품질의 지문 영역에 대한 평균과 분산은 골과 용선이 뚜렷하게 분포하는 양질의 지문 영상과 상이하게 다르다.

$$SCORE = \alpha O_{score} + \beta I_{score} + \gamma F_{score}$$

III. 지문 정합 알고리즘

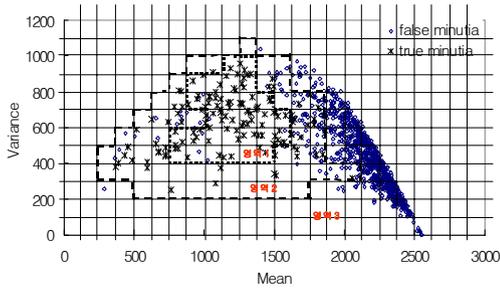


그림 5. 참특징점과 의사특징점 주위 영역의 평균과 분산 분포
 Fig 5. Mean and standard deviation distribution of the true and the false minutiae



(a) 원영상 (b) 가보 필터링 (c) 세션화 및 특징점 추출
 (a) Original image (b) Gabor filtering (c) Thinned image and extracted minutiae
 그림 6. 영상 센서를 통해 얻은 영상에 대한 처리 결과
 Fig 6. Process result obtained by image sensor

그림 5는 품질이 다양한 지문영상 140개에서 첫 번째, 두 번째에서 결정되지 않는 후보 특징점 1000 개에 대해 의사특징점과 참특징점 주위의 회색 영상의 평균과 분산을 구하여 얻은 그래프이다. 이 데이터를 근거로 세 영역을 구분하였고, 특징점을 중심으로 일정영역에 대하여 I_{score} 점수를 부여한다. 두번째, 특징점을 중심으로 블록단위 주파수를 구하는 방법과 동일하게 $w \times h$ 윈도우에 대한 일차원 파형을 구한다. F_{score} 는 특징점이 속해있는 블록의 주파수 $BF(i,j)$ 와 봉오리 간 평균거리의 역수 $F(i,j)$ 와의 오차에 정도에 따라 할당된다. 세번째, 용선 추출을 위해 미리 구한 블록의 대표 방향 중 그 특징점이 속해있는 방향 $BO(i,j)$ 와 특징점의 방향 $\phi_{i,j}$ 간의 차이로 $Od(i,j)$ 에 따라 O_{score} 는 할당된다. 총 점수는 평균과 분산, 주파수, 그리고 방향에 대한 점수에 가중치를 달리하여 더한 점수이다. 그 특징점의 SCORE가 이상일 때 참특징점으로 분류된다. 이와 같은 후처리 과정을 거쳐 그림 6과 같은 특징점을 추출한다.

3.1 지문 등록

지문등록 단계는 영상획득, 전처리, 그리고 후처리를 통해 얻은 최종 특징점에 대한 특징점 위치(x, y), 특징점의 방향(θ), 특징점의 종류(k), 일정 반경안의 이웃 특징점 정보 등을 정합에 사용하기 위해 메모리 공간에 저장한다.

3.2 지문 정합

마지막으로 입력된 지문과 등록된 지문이 동일한지를 인증하는 정합 과정을 수행한다. 정합의 어려움은 지문 영상이 일정크기의 센서 창을 통해 획득되어서 특징점에 대한 기준 좌표가 없다는 것이다. 즉, 동일 지문이라도 영상 획득될 때 마다 천이, 회전, 늘림 등의 원인으로 특징점이 동일한 위치와 각도를 갖지 못한다. 따라서 정합을 할 때 대상지문과 입력지문의 기준이 되는 특징점 쌍을 결정해야 한다. 결정된 참조 쌍을 중심으로 모든 특징점을 극좌표 변환하여 제 정렬한다. 그 후에 두 영상의 특징점들 간의 거리, 각도, 특징점의 방향, 그리고 특징점의 종류를 비교하여 동일지문 여부를 결정한다.

두 지문의 특징점이 가질 수 있는 모든 가능한 쌍을 참조 쌍으로 하여 전체 정합을 시도하는 것은 실시간 시스템에는 적합하지 않다. 따라서 두 지문 영상에 대해서 참조 쌍을 정확하고 빨리 찾는 것이 중요하다. Jain^[3]과 Luo^[4]의 정렬을 기반으로 하는 정합 알고리즘에서는 특징점을 따라 이루어진 일정 길이의 용선의 곡률 차를 정량화하여 참조 쌍을 구하였다^[1]. 그러나 지문을 중심으로부터 같은 거리에 있는 용선의 곡률의 차이가 작아서 참조쌍을 찾는 데 오류를 가질 수 있다. 본 논문에서는 부분정합을 통해 후보 참조쌍을 구하여 전체 정합을 수행하였다.

3.2.1 이웃 특징점과 부분 정합

후처리를 통해 얻은 각 특징점에 대해서 일정 반경 안의 포함된 이웃 특징점들 구하고, 특징점 $M_i = (x_i, y_i, \theta_i, t_i)$ 을 중심으로 이웃 특징점 $M_{ik} = (x_{ik}, y_{ik}, \theta_{ik}, t_{ik})$ 을 극좌표로 변환하여 거리 r_{ik} 와 각도 ϕ_{ik} 를 구한다. 그리고 중심 특징점의 방향에 대한 이웃 특징점 방향과의 상대적인 각도 차 θ'_{ik} 을 계산한다.

$$r_{ik} = \sqrt{(x_{ik} - x_i)^2 + (y_{ik} - y_i)^2},$$

$$\phi_{ik} = \tan^{-1} \left(\frac{y_{ik} - y_i}{x_{ik} - x_i} \right), \quad \theta'_{ik} = \theta_i - \theta_{ik}$$

이때 반지름 R 이 너무 크면 이웃 특징점 수가 증가하여 저장 공간 증가 및 부분정합 시간이 증가하게 되고, 너무 작으면 정확한 참조 쌍을 찾는 데 어려움을 갖는다. 따라서 본 논문에서는 실험을 통해 R 을 20으로 고정하였다.

효율적으로 이웃 특징점을 구하기 위해 블록 단위로 블록에 속한 특징점의 수와 특징점의 색인 정보를 갖는 $Bx(u,v)$ 를 생성하였다. 블록단위로 현재 블록과 그 이웃 블록의 특징점 개수 합이 a 이상인 블록에 대해서 그림 7과 같이 이웃 블록에 포함된 특징점의 거리를 계산하여 반경 R 에 속한 특징점을 극좌표로 변환하고, 이웃 특징점 정보를 ϕ_{ik} 가 증가하는 순으로 정렬된다. 등록지문에 대해서는 이웃 특징점 정보는 등록 시 계산되어 저장된다.

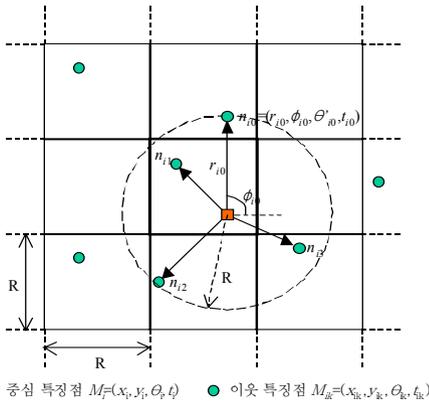


그림 7. 반경 R 안에 속한 이웃 특징점
Fig 7. Neighbor minutiae into a radius R

이렇게 저장된 이웃특징점 정보와 입력 지문영상의 이웃특징점 정보를 이용하여 부분 정합을 수행한다. 모든 특징점이 이웃특징점을 갖지 않으므로 부분 정합 계산 횟수가 감소하고, 등록정보의 경우 극좌표로 변환되어 있으므로 변환 시간이 따로 요구되지 않으며, 또한 전체 정합에 비해 매우 적은 계산 시간을 요한다. 모든 쌍에 대해서 부분정합을 한 후에 점수가 상위에 속하는 쌍을 참조쌍으로 정의하고 전체 정합을 시도한다. 부분정합에서 이웃특징점에 대한 정합 계산은 전체 정합방법과 동일하다.

3.2.2 전체 정합

등록된 지문에 대한 그림 8에서와 같이 특징점 정보를 Q , 입력된 지문의 특징점 정보를 Q' 이라고 정의한다. Q , Q' 의 각 특징점 D_i 는 벡터정보 $(x_b, y_b, \theta_b, t_i)^T$ 로 구성되어있다.

$$Q = \{D_1, D_2, \dots, D_M\}, \quad Q' = \{D'_1, D'_2, \dots, D'_N\}$$

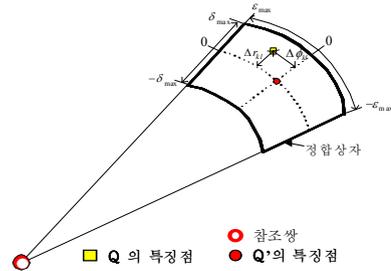


그림 8. Q 와 Q' 의 특징점 정합
Fig 8. Minutia matching of Q and Q'

전체 정합단계에서는 부분정합 결과 점수가 상위를 차지한 특징점 쌍들을 중심으로 전체 정합을 수행한다. 먼저 등록지문에 대한 특징점 D_i 와 입력 지문의 특징점 D'_j 를 중심으로 나머지 특징점을 극좌표로 변환한다. $\Delta\theta_{ij}$ 은 $\theta_i - \theta'_j$ 으로 상대적인 입력지문의 회전각을 의미한다.

$$\begin{pmatrix} r_{ik} \\ \phi_{ik} \\ \theta_{ik} \end{pmatrix} = \begin{pmatrix} \sqrt{(x_{ik} - x_i)^2 + (y_{ik} - y_i)^2} \\ \tan^{-1}\left(\frac{y_{ik} - y_i}{x_{ik} - x_i}\right) \\ \theta_{ik} - \theta_i \end{pmatrix},$$

$$\begin{pmatrix} r'_{jl} \\ \phi'_{jl} \\ \theta'_{jl} \end{pmatrix} = \begin{pmatrix} \sqrt{(x'_{jl} - x'_j)^2 + (y'_{jl} - y'_j)^2} \\ \tan^{-1}\left(\frac{y'_{jl} - y'_j}{x'_{jl} - x'_j}\right) + \Delta\theta_{ij} \\ (\theta'_{jl} - \theta_j) + \Delta\theta_{ij} \end{pmatrix}$$

반지름 각도가 증가하는 순서로 정렬된 극좌표계를 갖는 Q_i 과 Q'_j 를 갖게 된다.

$$Q_i = \{P_1, P_2, \dots, P_{M-1}\}, \quad Q'_j = \{P'_1, P'_2, \dots, P'_{N-1}\}$$

두 특징점 P_{ik} , P'_{jl} 에 대해서 그림 8에서 정의된 정합상자 안에서의 반지름의 오차 Δr_{kl} , 각도의 오차 $\Delta\phi_{kl}$, 그리고 특징점 방향의 오차 $\Delta\theta_{kl}$ 를 구한다. 그리고 P'_{jl} 이 정합상자 밖에 존재한다면 $-\infty$ 을 부여한다.

$$\Delta r_{kl} = \begin{cases} |r_{ik} - r'_{jl}|, & \text{for } 0 \leq |r_{ik} - r'_{jl}| < \delta_{\max} \\ -\infty, & \text{otherwise} \end{cases}$$

$$\Delta\phi_{kl} = \begin{cases} |\phi_{ik} - \phi'_{jl}|, & \text{for } 0 \leq |\phi_{ik} - \phi'_{jl}| < \epsilon_{\max} \\ -\infty, & \text{otherwise} \end{cases}$$

$$\Delta\theta_{kl} = \begin{cases} |\theta_{ik} - \theta'_{jl}|, & \text{for } 0 \leq |\theta_{ik} - \theta'_{jl}| < \omega_{\max} \\ -\infty, & \text{otherwise} \end{cases}$$

두 특징점 P_{ik}, P_{jl} 의 정합 점수는 $S_{kl} = b \cdot (S_{kl}^r + S_{kl}^\phi + S_{kl}^\theta)$ 으로 b 는 특징점 종류가 같지 않을 경우 0.5, 같을 경우 1.0의 정합점수 가중치이다. 전처리 단계에서 발생한 정보 손실에 따른 특징점 종류 오류를 보상해 주기 위해서 가중치 b 를 두었다. S_{kl}^r, S_{kl}^ϕ , 그리고 S_{kl}^θ 의 정합 점수는 오차 값에 비례하다. S_{min}^r, S_{max}^r 은 Δr_{kl} 에 대한 최소 점수와 최대 점수를 의미한다. 본 논문에서는, $S_{min}^r, S_{min}^\phi, S_{min}^\theta$ 는 각각 25, 25, 20으로 적용하였고, $S_{max}^r, S_{max}^\phi, S_{max}^\theta$ 은 각각 35, 35, 30으로 결정하였다.

$$S_{kl}^r = S_{min}^r + (S_{max}^r - S_{min}^r) \cdot \frac{\delta_{max} - \Delta r_{kl}}{\delta_{max}}$$

각 특징점에 대한 정합점수의 합 TS_{ij} 를 두 지문의 특징점 수 중 큰 수로 나누어 정합 점수 MS_{ij} 를 구한다. 부분정합에서 결정된 참조쌍을 기준으로 하여 구한 정합점수 중에서 본인 여부를 결정하는 임계 점수보다 크거나 같으면 동일 지문이라고 결정한다.

$$MS_{ij} = \frac{TS_{ij}}{MAX(M, N)} \times 100$$

IV. 임베디드 지문인식 시스템

휴대 전화기나 PDA 이용자가 증가함에 따라 온라인 금융 거래 및 무선 인터넷을 통한 거래가 빈번해지고 이에 대한 보안의 중요성이 강조되고 있다. 따라서 휴대하기 용이하고 기억하지 않아서 편리하며 위·변조가 불가능한 생체인식 시스템이 요구된다. 지문인식 시스템을 비롯한 생체 인식시스템은 높은 메모리 용량과 이미지 필터링, 그리고 신호처리를 위한 강력한 처리능력이 요구되지만 일반 개인 컴퓨터와 달리 소형의 휴대 기기는 처리성능 및 메모리 등의 하드웨어 리소스가 충분하지 못하다. 그러므로 임베디드 환경에 적합하도록 최적화하여야 하는데 이 방안으로 먼저 부동소수점 연산부분을 고정소수점 연산으로 변환한다. 부동소수점 연산기가 구현되어 있지 않은 경우 부동소수점 에뮬레이터 연산기를 이용하는 경우 많은 수행시간을 요구하게 된다. 두 번째, 전처리와 후처리 그리고 정합까지 삼각함수 $\sin()$ 과 $\cosin()$ 을 많이 사용한다. 이 삼각함수를 라디안 방식이 아닌 degree 방식의 LUT (Look Up Table)로 대체하고 $atan()$ 함수는 비선형

적인 점을 고려한 이진 트리 검색 방법을 사용하였다.

$$h(x, y : \phi, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\phi^2}{\delta_x^2} + \frac{y_\phi^2}{\delta_y^2}\right]\right\} \cos(2\pi f X_\phi) \quad (1)$$

$$x_\phi = x \cos \phi + y \sin \phi$$

$$y_\phi = -x \sin \phi + y \cos \phi$$

세 번째, 필터계수를 효율적으로 계산하도록 최적화한다. 이진화 영상을 추출하기 위해 수행하는 가보 필터링에서 각 픽셀에 적용되는 가보필터계수는 앞서 구한 방향과 주파수에 의해 결정되어지며 매 픽셀마다 구해야한다. 가우시안 식(1)에서 지수함수 $\exp()$ 값을 고정시키고 블록의 응선 주파수로부터 블록의 필터계수를 계산하고 응선의 방향으로 영상을 회전하여 필터링을 수행한다. 영상에 대한 모든 픽셀에 대해서 가보필터링을 하는 것이 아니라 샘플링하여 처리하고 나온 결과에 대해서 사이 값 보정을 함으로써 수행시간은 1/4로 줄어든 반면에 영상의 질에는 큰 영향이 없었다. 그러나 중간 결과를 저장하기 위해 $4 \times w \times h$ 바이트 크기의 메모리공간이 필요하다. 메모리의 제약이 있을 때 가보필터계수는 블록 단위이므로 저장 공간을 줄이기 위해 연산시간이 다소 증가하지만 블록단위로 연산을 수행할 수 있다.

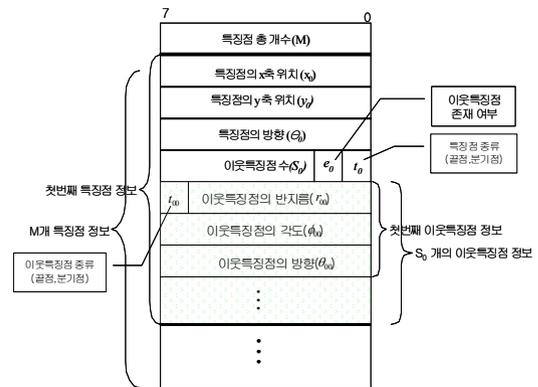


그림 9. 등록된 지문에 대한 데이터베이스 구조
Fig 9. Database Structure about registered fingerprint

넷째, 지문에 대한 특징량을 효율적인 저장하는 등록방법이다. 임베디드 환경에서는 메모리의 사용이 제한되므로 효율적으로 저장하여야 한다. 지문이 M개의 특징점을 갖는다면 그림 9에서와 같이 저장된다. 예를 들어 지문의 총 특징점이 25 개이고 이

웃특징점을 갖는 특징점 수가 9개, 이웃특징점 수가 35개라면 $94+353+1=142$ 바이트 만큼의 저장공간이 필요하다.

V. 실험 및 결과 분석

본 논문에서 제안하는 지문인식 알고리즘의 성능을 분석하기위해서 30명으로부터 시간차를 주면서 5개씩의 지문을 획득하여 150개의 지문영상을 얻어 실험하였다. 248,292 크기를 갖는 실험 영상은 좋은 품질의 영상이 30%, 중간이45%, 나쁜 영상이 25%로 구성되어 있다. 실험에 사용된 지문은 니트젠(㈜)에서 제작한 해상도 450dpi의 광원센서로부터 얻었으며 지문 인식 알고리즘은 ARM 프로세서의 소프트웨어 에뮬레이터인 ARMulator에서 알고리즘을 검증하였다. 현재 블록과 그 이웃 블록의 특징점 개수 합 a는 15, 그리고 참조쌍을 구하기 위한 임계값은 6으로 고정하였다. 본 논문에서 제안한 지문인식 알고리즘을 Arm920T(클럭 주파수: 70MHz) 환경에서 구현하였을 때 알고리즘의 모듈별 수행시간을 표 1에 나타내었다. 가보필터를 이용한 이진화 과정과 세션화 과정의 수행시간이 상대적으로 많은 비중을 차지하였으며 전체적으로 0.541초로 실시간 임베디드 응용분야에 적용 가능성을 보여주었다.

표 1. 지문인식 처리 시간(Arm920T, 70MHz)
Table 1. Fingerprint verification processing time(Arm920T, 70MHz)

단 계	수행시간(초)
정 규 화	0.131
블 력 방 향	0.024
블 록 주 파 수	0.044
이 진 화	0.178
세 셴 화	0.137
특 징 점 추 출	0.008
의 사 특 징 점 제 거	0.009
정 합	0.010
계	0.541

그림 10은 제안된 임베디드용 지문인식 알고리즘을 검증하기 위하여 제작된 보드를 보여주며 그림 11은 이때 이용된 Arm920T에 대한 메모리 맵의 구조를 나타낸다. 그림 12는 본인 대 본인, 본인 대 타인의 정합 점수 분포 그래프로 30명으로부터 각각 5개씩 획득한 지문영상 150개에 대해 총 22,350 (150,149) 쌍을 매칭하여 얻는 그래프이다. 매칭 점

수를 25점을 기준으로 하였을 때 FRR(본인 거부율)과 FAR(타인 승인율)은 각각 0.079과 0.00005로 신뢰성을 갖는다.

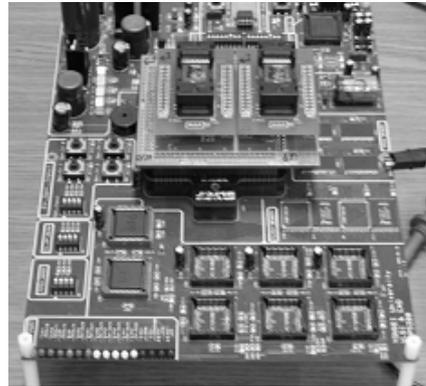


그림 10. 테스트 보드 구조
Fig 10. Test Board Structure

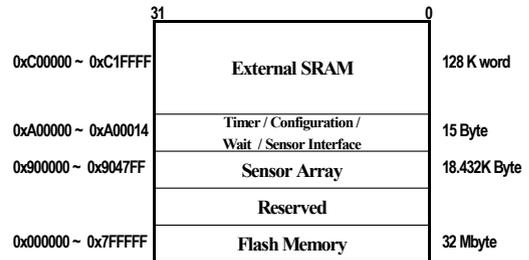


그림 11. 메모리 맵
Fig 11. Memory Map

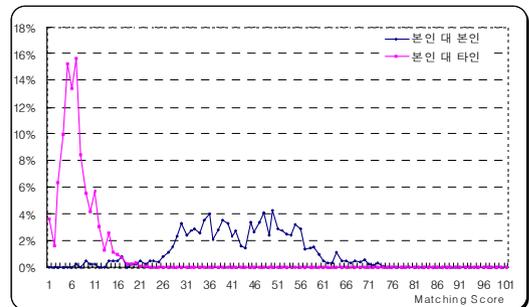


그림 12. 제안한 알고리즘 적용에 따른 정합율
Fig 12. Matching rate with the proposed algorithm

VI. 결론

본 논문에서는 실시간 응용분야에 적합한 지문인식 알고리즘을 제안하였다. 제안된 알고리즘은 전처리 및 후처리 과정을 거쳐 의미있는 참특징점을 추출한 후 부분 정합을 이용하여 효과적으로 지문을

인식한다. 전처리 및 후처리 과정에서는 획득된 영상 및 옵션의 주파수와 방향 등의 다양한 정보를 이용하여 참 특징점을 선택하였으며 정합 과정에서는 영상 획득 시에 있을 수 있는 오차를 고려하여 신뢰도를 높였다. 제안된 알고리즘은 Arm920T 환경에서 검증되었으며 이때 실시간 처리를 위한 다양한 방법이 적용되었다. 추출된 지문 정보를 위한 효과적인 데이터 구조와 실시간 시스템을 위한 메모리 맵이 제안되었다. 제안된 알고리즘의 수행시간은 영상획득 후 전처리과정부터 정합까지 0.541초로 실시간 임베디드 응용분야에 적용 가능성을 보여주었으며 매칭 점수 25점을 기준으로 하였을 때 FRR(본인 거부율)과 FAR(타인 승인율)은 각각 0.079과 0.00005로 신뢰성을 갖는다.

감사의 글

저자들은 본 연구를 위하여 설계 소프트웨어를 제공하여 준 IDEC(IC Design Education Center)에 감사드립니다.

참 고 문 헌

[1] A. Jain and L. Hong, "An identity-authentication system using fingerprints," *Proceedings of the IEEE*, vol.85, no.9, pp.1365-1388, 1997

[2] D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.19, no.1, pp.27-40, 1997

[3] A. Jain, L. Hong, and R. Bolle, "On-Line Fingerprint Verification," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp.302-313, 1997.

[4] X. Luo, J. Tian, and Y. Wu, "A Minutiae Matching Algorithm in Fingerprint Verification," *15th International Conference on IEEE Pattern Recognition*, vol. 4, pp.833-836, 2000

[5] D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp.27-40, 1997

[6] 김현, 김학일, "RSTI 불변 지문 인식 알고리즘," *전자공학회 논문지*, 35권, S편, 6호, pp. 838-850, 1998

[7] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Trans. Image Processing*, vol.9, no.5, pp.846-859, 2000

[8] C. Lee and S. Wang, "Fingerprint feature extraction using gabor filters," *Electronics Letters*, vol.35, no.4, pp.288-290, 1999

[9] N. Ratha, K. Karu, S. Chen, and A. 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

[10] M. Alam and M. Akhteruzzaman, "Real time fingerprint identification," *Proc. of the IEEE NAECON 2000*, pp.434-440, 2000

박 태 근(Tea geun Park)

정회원



1985년 연세대학교 전자공학 학사
 1988년 Syracuse Univ. Computer 공학석사
 1993년 Syracuse Univ. Computer 공학박사
 1991년~1993년 Coherent Research Inc. VLSI 설계 엔지니어
 1994년~1998년 현대전자 System IC 연구소 책임연구원
 1998년~현재 가톨릭대학교 정보통신 전자공학부 부교수
 <관심분야> VLSI 설계, CAD, 병렬처리 등

정 선 경(Sun kyung Jung)

정회원



2001년 가톨릭대학교 정보통신 공학 학사
 2003년 가톨릭대학교 컴퓨터공학과 석사
 2003년~현재 넥서스칩스 연구원
 <관심분야> VLSI 설계, 패턴인식, 디지털 신호처리 등