

최소화 알고리즘 기반의 지문 인식 시스템의 설계 및 구현

이호현^o, 조범준
조선대학교 컴퓨터공학과

Design and Implementation of Minutiae Extract Minimize Algorithm System

Ho-Hyun Lee, Beom-Joon Cho
Dept. of Computer Engineering, Chosun Univ.

요 약

본 논문에서는 기존 지문 인식 시스템에서 사용하는 특이점 추출 알고리즘과 매칭 알고리즘을 개선한 알고리즘들을 설계하고 구현함으로서 새로운 지문 인식 시스템 개발에 목표를 두었다. 현재, 이진화된 지문 이미지를 표현하는 자료구조인 Union and Division을 고안하였고, Union and Division에 기반한 특이점 추출 알고리즘을 설계 및 구현하였다. 또한 특이점 추출에서의 과다한 추출 단계로 인하여 발생되는 문제점들을 줄이기 위하여 기존의 6 단계인 특이점 추출 과정을 개선하여 3단계로 줄이면서도 정확성을 높이는 특이점 추출 알고리즘을 적용 하였다.

1. 서론

사람에게는 다른 사람들과 구별되는 특징들이 여러 가지가 있다. 얼굴모양과 얼굴에 있는 눈이나 코, 입 등의 위치와 눈에서의 융채, 그리고 손가락의 지문 등으로 사람과 사람을 구별할 수 있다. 그 중에서도 특히 손가락의 지문은 개개인이 다른 모양을 가지고 있어 확실한 구별이 되기 때문에 옛날부터 지문을 이용하여 신원을 확인하는 시스템이 운영된 것으로 알려지고 있다[1][2].

F.Galton (1843~1930), H.Faulds (1843~1930), H.Wilder(1864~1928)은 지문을 이용한 신원확인 시스템에 과학적으로 접근하였고, H.Poll (1877~1939),

Galton은 지문의 유일성과 시간이 흘러도 모양이 변하지 않다는 점을 지적하였다[3].

최근에는 지문을 이용하는 분야에서 경찰은 모든 범죄자나 일반 사람들에게서 지문을 획득하여 데이터베이스화하고 있는데, 이 과정에서 손가락에 잉크를 묻혀 종이에 찍는 방법으로 데이터베이스화하고 있다. 그런데 이 방법으로써는 많은 잡음을 생성시켜 지문 인식 시 많은 어려움을 줄 수가 있다. 예를 들자면 잉크의 양이 많거나 적을 수도 있고 종이 표면이 고르지 못해서 이미지 상에 잡음이 생긴다. 그리고 지문을 획득할 때의 강약이라든가 손가락의 위치 등에서도 많은 잡음이 생겨 인식 시 오류를 범할 수도 있다. 지문은 융선(Ridge)과 골(Valley)로 이

루어져 있다고 할 수 있다. 지문인식은 보통 지문의 형태를 구분 짓는 융선의 형태에 의해 정해지는데 그 형태에 따라서 코어(중심점), 레타 같은 특이점을 추출하게 된다. 기존의 지문 인식 시스템은 특이점을 추출하기 위해 전처리 단계, 방향추출, 이진화, 세선화, 특이점 추출, 오류 특이점 제거 단계를 거친다. 이런 기존의 시스템들은 실시간 처리를 목적으로 하는 지문 인식 시스템에서 인식하는 속도를 저하 할 뿐 아니라 메모리의 요구량(위치와 방향, 특이점 종류 정보 저장)이 많고 원 영상에 몇 가지 단계를 거치면서 이미지의 불필요한 변형이 생길 수도 있다.

본 논문에서는 특이점 추출 단계에서의 과다한 추출 단계로 인하여 발생되는 문제점들을 줄이기 위하여 기존의 일곱 단계인 특이점 추출 과정을 개선하여 3단계로 줄이면서도 정확성을 높이는 특이점 추출 알고리즘을 설계/구현하고자 한다.

2. 관련연구

- 최소화 알고리즘

- ① 전처리를 한다.
- ② 입력 지문의 중심 블록에 대하여 X-축과 Y-축의 Gray scale 값의 분포를 구한다.(중심 블록이 잡음이 가장 작음) 각 화소(i, j)에 대하여 융선과 꿀을 포함하는 $n \times n$ 화소크기의 영역을 설치하고 식(1)과 같이 Gray scale 값의 분포를 찾는다.

$$K(i, j) = \text{Min} \sum [C(i_k, j_k) - C(i, j)] \quad \text{식(1)}$$

- ③ 현재 Pixel에 대하여 X-축 값과 Y-축 값, X-축과 Y-축에서의 이전 융선과의 거리를 고려하여 현재 Pixel이 융선에 해당하는 가를 판정한다. (기존의 방법 중 융선의 거리를 이용한 방법은 존재하지만 단계 ③과 같은 요소들을 고려하여 한 Pixel의 융선 여부를 바로 판정하는 방법은 없음.)

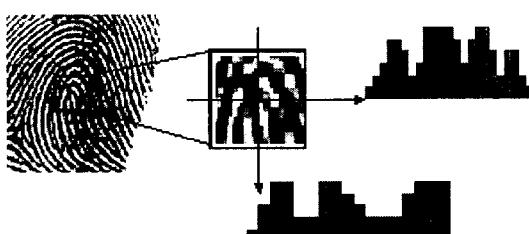


그림.4 단계 ③에 대한 다이어그램

- ④ 특이점을 추출하기 위하여 현재 Pixel이 특이점이기 위한 조건을 만족하는지의 여부를 판단한다. 예를 들어 현재의 Pixel이 끝점이기 위한 조건은 주변 융선들(과)의 거리 등 간격에 가까워야 한다.



그림.5 끝점이기 위한 조건

- ⑤ 기존 방법들을 이용하여 오류 특이점을 제거한다.
 - ⑥ 단계 ②~단계 ④ 과정을 전체 입력 지문으로 확장하여 처리한다.
- 제안한 방법에 의해 기존의 6 단계 특이점 추출 과정을 3 단계로 줄일 수 있다.

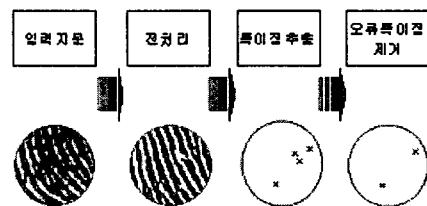


그림.6 제안 시스템에서의 특이점 추출 단계

3. 개발 지문 인식 시스템

3.1 기존 지문 인식 시스템

현재까지 지문 이미지를 표현하는 자료구조는 제안된 바 없다. 본 연구에서는 이진화된 지문 이미지를 표현하는 Union and Division이라 불리는 자료구조를 도출하였다. Union and Division을 이용하여 지문 인식 시스템에서 특이점을 추출하기 위해 소요되는 시간은 Union and Division을 사용하지 않는 지문 인식 시스템에서 특이점을 추출하기 위하여 소요되는 시간의 1/3 밖에 되지 않는다. 소요되는 시간 측정은 기존 특이점 추출 방식에 대하여 측정하였으며, 평균 시간을 얻기 위하여 5명이 각각 500번의 지문을 입력하였다.

Union and Division을 이용하는 경우, 기존 특이점 추출 과정을 (1) 이진화 단계 (2) 잡음제거 단계 (3) 특이점 추출 단계로 축약할 수 있다. Union and Division을 이용하는 특이점 추출의 잡음제거 단계는

기준 오류 특이점 제거 단계를 포함하는 것으로 간주할 수 있으며, 잡음제거 단계에서 평활화가 이루어 진다. 그리고, 평활화가 이루어진 이후에는 세션화 단계를 필요로 하지 않게 된다.

기준 매칭 방법에서 가장 큰 문제점은 무엇을 기준점으로 설정할 것인가 하는 점이다. 본 연구에서는 지문 이미지의 테두리에 있는 융선에서 시작하여 그 융선과 연결된 모든 융선을 융선 줄기로 정의하고, 동일한 융선 줄기 상에 있는 두 특이점을 잇는 선분을 기준점으로 사용한다. 이와 같이, 동일한 융선 줄기 상에 있는 두 특이점을 잇는 선분을 기준점으로 사용하는 경우, 사용자가 지문을 어떤 방향에서 입력하는지에 상관없이 매칭을 할 수 있고, SDB에 있는 모든 지문은 둘 이상의 특이점을 갖는 하나 이상의 융선을 포함하고 있었다.

개발한 Union and Division을 이용한 특이점 추출 알고리즘과 동일 융선 줄기 상에 있는 두 특이점을 잇는 선분을 기준점으로 사용하는 지문 인식 시스템과 기존 지문 인식 시스템과의 비교를 위하여 기존 지문 인식 시스템을 구현하였다.

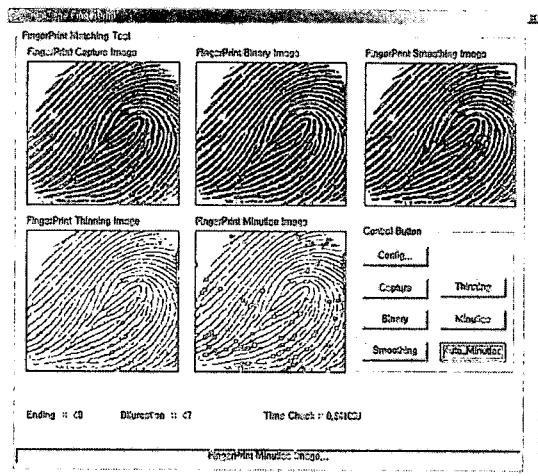


그림.7 기존 알고리즘을 이용하는 지문 인식 시스템

3.2 Union and Division과 특이점 추출

Union and Division의 개요는 다음과 같다.

- ① 이진화된 지문 이미지를 $B[R][W]$ 라 하자.
- ② $B[R][W]$ 에 있는 모든 값은 0 또는 1이다.
- ③ $B[R][W]$ 의 한 열에 있는 비트 스트링에 대하여

그 비트 스트링은 0의 런(들)과 1의 런(들)으로 구성되어 있다. 예를 들어, $B[R][W]$ 에 있는 한 열에 있는 비트 스트링이 000111000011111이라 하면 000111000011111은 0의 런 000과 1의 런 111, 0의 런 0000, 1의 런 1111로 구성되어 있다.

④ $B[R][W]$ 의 열에 있는 모든 런은 선분으로 간주될 수 있다. 예를 들어, $B[R][W]$ 의 i 열에 있는 j 번째 런이 12 행부터 18 행 사이에 걸쳐져 있다고 할 때, 그 런은 X-축 값이 i이고 Y-축의 작은 값이 12, Y-축의 큰 값이 18인 선분으로 간주될 수 있다.

⑤ Union and Division은 2 차원 배열 $UD[MAX][W]$ 이다. 여기에서, MAX는 열에 있는 런의 수들 중에서 최대 런의 수이다.

⑥ $UD[MAX][W]$ 의 i 행 j 열 원소는 다음과 같은 정보를 저장하고 있다.

$B[R][W]$ 의 i 열에 있는 j 번째 런이 Y_{min} 행에서 시작하여 Y_{max} 행에 걸쳐 있다고 할 때, Y_{min} 값과 Y_{max} 값이 저장되어 있다. $B[R][W]$ 의 i 열에 있는 j 번째 런이 0의 런인가 또는 1의 런인가를 나타내는 정보가 저장되어 있다. $B[R][W]$ 의 i 열에 있는 j 번째 런을 나타내는 선분과 (i+1)열에 있는 j' 번째 런을 나타내는 선분을 Y-축으로 사영하였을 때, 두 선분이 겹친다고 하자. 그러면, $B[j][i]$ 에는 j'가 저장되어 있고, $B[j'][i]$ 에는 j가 저장되어 있다. 어떤 글이나 융선을 나타내는 선분의 근원지 위치 정보가 저장되어 있다. 여기에서 근원지란 융선 또는 글이 시작되는 곳을 말한다. 이 정보의 저장 여부는 선택 사항이다.

<그림 8>은 Union and Division의 원소에 저장되는 정보를 그림으로 표현한 것이다.

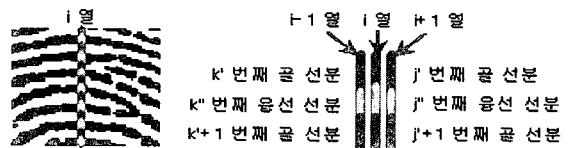


그림 8 Union and Division에 저장되는 정보

Union and Division은 다음과 같은 특징을 갖는다.

① 이진화된 지문 이미지와 $UD[MAX][W]$ 는 1:1 대응한다.

② $UD[MAX][W]$ 는 지문 이미지를 나타내는 그래프이지만, 첨자만 사용하여 연산이 가능하므로 구현이 쉽고 고속이다.

Union and Division의 추상 데이터 타입의 가독성을 위하여 자세한 언급은 하지 않는다. <표 1>는 추상 데이터 타입에 있는 함수가 특이점 추출 알고리즘과의 관계를 나타내고 있다.

함수	역할
Union	잡음제거
Division	잡음제거
Extension	잡음제거 및 평활화
Reduction	잡음제거 및 평활화
Tracing	특이점 추출

표 1. 데이터 타입 함수와 특이점 추출 알고리즘과의 관계

① Union 함수의 경우, 잡음으로 간주되는 것으로 인하여 하나의 선분이 세 개의 선분으로 나누어졌다고 판단되는 경우, 잡음을 제거한다. 이때, 잡음을 제거하면 세 개의 선분이 하나의 선분으로 된다.

② Division 함수의 경우, 잡음으로 간주되는 것으로 인하여 세 개의 선분이 하나의 선분으로 되었다고 판단되면, 잡음을 제거한다. 이때, 잡음을 제거하면 하나의 선분이 세 개의 선분으로 나누어진다.

③ Extension 함수의 경우, 잡음으로 인하여 선분의 길이가 짧아진 경우에 선분의 길이를 길게 함으로서 잡음제거를 하도록 하고, 서로 다른 선분의 길이 또는 넓이를 동일하게 함으로서 평활화를 한다.

④ Reduction 함수의 경우, 잡음으로 인하여 선분의 길이가 길어진 경우에 선분의 길이를 짧게 함으로서 잡음제거를 하도록 하고, 서로 다른 선분의 길이 또는 넓이를 동일하게 함으로서 평활화를 한다.

⑤ j 열 k 번째 런을 나타내는 선분과 $B[R][W]$ 의 $(j+1)$ 열에 있는 두 선분을 Y -축으로 사영하였을 때, 세 개의 선분이 겹친다고 하자. 그러면, $B[k][j]$ 는 분기점의 위치가 된다. j 열 k 번째 런을 나타내는 선분과 $B[R][W]$ 의 $(j+1)$ 열에 있는 선분을 Y -축으로 사영하였을 때, 어떤 선분도 겹치지 않는다고 하자. 그러면, $B[k][j]$ 는 끝점의 위치가 된다.

<그림 9>는 선분 분할에 의해 잡음이 제거되는 경우를 나타내고 있다. <그림 9>의 좌측 그림을 참고하여, $(i-\Omega-1)$ 열과 $(i'+1)$ 열의 $(j'-1)$ 번째 선분과 j' 번째 선분은 모두 융선을 나타내는 선분인데, $(i-\Omega)$ 열과 i' 번째 열에서 융선에 해당하는 부분이 잡음으로 인하여 골로 변하였다. <그림 9>의 우측 그림은 Division 함수를 이용하여 잡음이 제거된 후의 결과를 나타내고 있다.

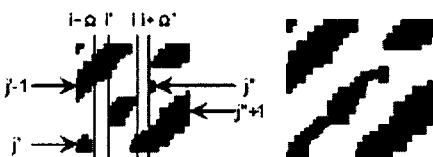


그림 9 잡음제거

잡음 제거를 위하여 선분을 추적 하는게 필요하다. 이웃해 있는 선분을 추적하는 과정에서 두 선분의

간격이 변하는 경향에 따라 잡음 여부를 결정하였다. 예를 들어, 두 선분을 추적 하는 과정에서 한 선분이 갑자기 끊어져 버렸는데, 끊어지기 이전에 두 선분의 간격에 큰 변동이 없다면 잡음으로 간주되었다.

<그림 10>은 Union and Division 과 Union and Division 추상 테이터 타입에서 정의된 함수들을 이용하여 특이점을 추출하는 과정을 나타내고 있다.

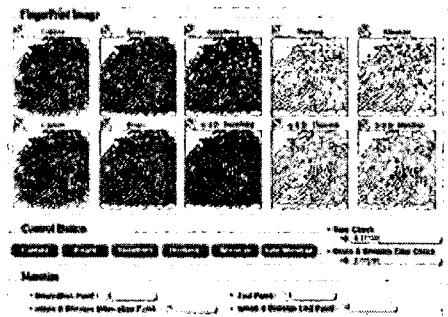


그림 10 Union and Division 을 이용한 특이점 추출

Union and Division 을 사용하는 경우 다음과 같은 장점이 있었다.

① 런을 선분으로 간주하여 처리하므로 0 또는 1 각각을 처리하는 기존 특이점 추출 알고리즘보다 고속이었다. 실제로 모의실험 결과, Union and Division 을 사용하여 특이점을 추출하는데 소요되는 시간은 Union and Division 을 사용하지 않는 경우의 특이점을 찾는데 소요되는 시간의 1/3 이었다.

② 기존 전처리=>이진화=>평활화=>세션화=>특이점 추출=>오류 특이점 제거 단계로 구성된 특이점 추출 과정을 전처리=>이진화=>잡음제거=>특이점 추출 단계로 줄였다. 잡음 제거 단계는 오류 특이점 제거 단계를 필요 없게 하였고, 잡음제거에서 선분 확장과 축소는 평활화를 필요 없게 하였으며, 이진화된 지문 이미지의 각 런을 선분으로 간주하여 처리하므로 세션화 과정을 수행하지 않고도 특이점을 추출할 수 있었다.

③ 기존의 고정된 방법에서 탈피하여 잡음제거라는 단계를 포함하는 특이점 추출 알고리즘을 설계함으로서 특이점을 보다 정확하게 추출할 수 있었다.

4. 구현 및 고찰

본 논문에서는 제시한 알고리즘의 성능을 평가하기 위해 NIST 의 SDB 에 있는 27,000 여종의 지문을 사용하였다. 지문 인식 시스템은 펜티엄

PC 에 Windows 2000 운영체제를 사용하고, 알고리즘의 구현은 C++ 언어를 사용하였다.

판정	데이터 개수	백분율(%)
인식	26,190	97
오인식	810	3

표. 1 제안 시스템의 인식률

위의 표.1에서 보는 바와 같이 제안된 시스템에서의 인식률은 97%로 만족할 만한 성능을 보였고, 개개의 지문 인식 속도에서도 1 인당 평균 0.5 초의 빠른 속도를 보였다.

이 실험에서 나타난 오인식 판정은 영상을 확인한 바 대부분이 입력 영상이 풍개짐으로 응선의 흐름이 뚜렷이 나타나지 않은 상태이거나, 영상에 회전 왜곡이 발생한 경우임을 알 수 있었다. 지문영상 획득장비의 개선이 이루어진다면 양질의 입력 영상을 제공 받아 더욱 향상된 인식률과 시스템의 안정성을 기대할 수 있을 것으로 생각된다.

5. 결론

본 논문에서는 기존 지문 인식 시스템에서 사용하는 특이점 추출 알고리즘과 매칭 알고리즘을 개선한 알고리즘들을 설계하고 구현함으로서 새로운 지문 인식 시스템 개발에 목표를 두었다. 현재, 이진화된 지문 이미지를 표현하는 자료구조인 Union and Division을 고안하였고, Union and Division에 기반한 특이점 추출 알고리즘을 설계 및 구현하였다. Union and Division의 경우, 지문 이미지를 표현하는 자료구조가 없었던 것을 개선한 것으로 평가할 수 있다.

특이점 추출 알고리즘은 잡음제거 단계를 수행하도록 함으로서 특이점 추출을 효율적이고 정확하게 할 수 있도록 하였다. 특히, Union and Division을 사용하는 경우 기존 특이점 추출 단계를 획기적으로 줄이는 효과가 있었다. 매칭 알고리즘의 경우, 기준에 기준점을 설정하기가 어렵거나 트리나 그래프를 구성하는 것과는 달리 동일 응선줄기를 정의하고, 동일 응선줄기 상에 있는 두 특이점을 잇는 선분을 기준점으로 선택함으로서 효율적이고 정확하게 매칭하도록 하였다. 앞으로 이러한 회전왜곡과 데이터량 축소, 인식률의 향상이라는 문제점을 보완할 수 있는 알고리즘에 대한 연구를 진행할 것이다..

[참고문헌]

- [1]B. Moayer, K. S. Fu, "A syntactic approach to fingerprint pattern recognition", Pattern Recognition 7, 1-23, 1975.
- [2]D. K. Isenor, S. G. Zaky, Fingerprint identification using graph matching, pattern Recognition 19, 113-122, 1986.
- [3]F. Galton, Finger Prints, MacMillan, London, 1892.
- [4]Q. Xiao, H. Rafat, A combined statistical and structural approach for fingerprint image postprocessing proceedings of the IEEE International Conference on systems, Man and Cybernetics Conference, pp. 331-335, 1990.
- [5]The Science of Fingerprints: Classification and Uses United States Department of justice, Federal Bureau of Investigation, Washington, rev. 12-84, 1988.
- [6]W. C. Lin, R. C. Dubes, A review of ridge counting in dermatoglyphics, Pattern Recognition 16, 1-8, 1983.
- [7]안도성, “실시간 지문영상 자동인식을 위한 알고리즘 개발”, 인하대학교 석사학위논문, 1994.
- [8]안도성외, “블럭 FFT를 이용한 실시간 지문 인식 알고리즘”, 전자공학회 논문지, 제32권 16호, pp910-912, 1995.
- [9]문명권외, “구조적 방법에 근거한 지문 영상의 효율적인 분류”, 정보과학회 논문지, 제24권 2호, pp170-171, 1997.
- [10]제종태, “Ridge-line을 이용한 계층적 지문인식”, 서강대학교 석사학위논문, 1990.
- [11]L. O'Gorman and J. V. Nickerson, "An approach to fingerprint filter design," Pattern Recognition 22, pp. 28-38, 1989.
- [12]Zs. M. Kovacs-Vajna, R. Rovatti, and M. Frazzoni, "Fingerprint ridge distance computation methodologies," Pattern Recognition 33, pp. 69-80, 2000.