다중 접근 메모리 시스템을 이용한 고속 지문인식 특징추출 시스템

박종선[†], 김재희^{**}, 고경식^{***}, 박종원^{****}

요 약

최근 보안 시스템 중에서 지문인식을 이용한 보안 시스템은 다른 보안 시스템에 비해 유일성과 편의성 등의 장점을 가짐으로써 많은 사람들이 관심을 갖는 분야이다. 지문인식 시스템에 있어서 가장 중요한 사항은 실제 지문과 영상을 통해 얻어진 지문간의 정합에서의 정확성과 지문 인식을 위해 사용하는 영상처리 알고리즘들을 신속하게 처리하는 데 있다. 기존의 지문인식 시스템은 특징 추출을 위해 사용하는 알고리즘들의 처리 시간을 줄이기 위해 전체 처리과정 중 일부 과정들을 생략함으로써 처리시간을 단축한다. 하지만 이 방식은 처리시간을 단축시킬 수 있는 반면 특징 추출에 대한 정확도가 떨어진다. 따라서 본 논문에서는 특징 추출에 대한 정확도를 높이기 위해 전체 처리 과정을 사용하면서 동시에 처리시간도 단축시킬 수 있는 다중 접근 메모리 시스템을 이용한 지문인식 특징 추출 알고리즘을 구현하였고, 구현된 시스템을 사용하였을 때 다중접근 메모리 시스템과 시리얼 프로세서의 결과에 대해 correlation을 이용한 검증을 통해 제안된 방법의 신뢰도를 확인하였으며, 시리얼 프로세서에 비해 MAMS-PP64를 이용한 방법은 수행시간에서 약 1.56배 향상되었음을 확인하였다.

Feature Extraction System for High-Speed Fingerprint Recognition using the Multi-Access Memory System

Jong Seon Park[†], Jea Hee Kim^{**}, Kyung-Sik Ko^{***}, Jong Won Park^{****}

ABSTRACT

Among the recent security systems, security system with fingerprint recognition gets many people's interests through the strengths such as exclusiveness, convenience, etc, in comparison with other security systems. The most important matters for fingerprint recognition system are reliability of matching between the fingerprint in database and user's fingerprint and rapid process of image processing algorithms used for fingerprint recognition. The existing fingerprint recognition system reduces the processing time by removing some processes in the feature extraction algorithms but has weakness of a reliability. This paper realizes the fingerprint recognition algorithm using MAMS(Multi-Access Memory System) for both the rapid processing time and the reliability in feature extraction and matching accuracy. Reliability of this process is verified by the correlation between serial processor's results and MAMS-PP64's results. The performance of the method using MAMS-PP64 is 1.56 times faster than compared serial processor.

Key words: Parallel Processing(병렬처리), Multi-Access Memory System(다중접근 메모리 시스템), Image Processing(영상처리), SIMD Computer(SIMD 컴퓨터)

※ 교신저자(Corresponding Author): 박종원, 주소: 대전 유성구 공동 220번지 충남대학교 공대3호관 408호(305-764), 전화: 042) 821-6863, FAX: 042) 825-7792, E-mail: jwpark@cnu.ac.kr

접수일: 2013년 5월 8일, 수정일: 2013년 7월 15일

완료일: 2013년 7월 18일

[†] 정회원, 충남대학교 정보통신공학과

(E-mail: cyburbia@daum.net)

^{**} 정회원, 충남대학교 정보통신공학과

(E-mail: jehe@crow.cnu.ac.kr)

^{***} 정회원, 충남대학교 정보통신공학과

(E-mail: haidy1223@daum.net)

^{****} 정회원, 충남대학교 정보통신공학과

※ 이 연구는 충남대학교 학술연구비에 의해 지원되었음

1. 서 론

최근 정보통신 및 컴퓨터 기술의 급격한 발전에 따른 온라인 금융거래에 대한 보안 및 시설물 출입 및 관리 분야의 보안에 대한 관심이 증가하고 있다. 현재 개인 신원 확인을 위한 일반적인 수단인 열쇠, ID Card, 비밀번호 등의 보안장비는 복제나 도용 및 분실의 위험이 있다. 하지만 생체 인식을 이용한 방법은 별도로 기억하거나 보관할 필요가 없으며 도용이 힘들다는 점에서 강력한 보안 및 인증 분야의 해결책으로 제시되고 있다. 생체인식 시스템에서 주로 이용하는 것으로 지문, 손 모양, 얼굴, 눈, 정맥, 음성 등이 있는데, 그 중에서 지문은 개개인마다 모양이 다르고 일반적인 보안 방식과 달리 휴대의 필요성이 없기 때문에 유일성, 보편성, 편의성 등 여러 장점을 갖는 개인인증 방법으로 이용되고 있다[1].

지문인식 시스템의 알고리즘은 크게 지문에 대한 특징 추출 알고리즘과 지문 정합 알고리즘 두 부분으로 구분한다. 특징추출 알고리즘은 지문정합 단계에서 사용할 특징 데이터 파일을 생성하는 것이며 세부적으로는 전처리, 특징 추출, 후처리의 과정을 거친다[2].

지문인식 특징 추출 알고리즘과 관련된 기존 연구에서 지문의 특징 추출에 대한 검출 속도를 향상시키기 위한 방법들 중 하나는 전처리 과정이 없이 가보 필터(Gabor filter)[3]만을 사용하여 코어를 찾아내고 검출된 코어를 이용하여 정합하는 방식이 있으며, 다른 하나는 전처리 과정 중에서 이진화(binanzation)와 세선화(thinning) 과정 없이 단지 회색 음영 영상만을 이용하여 특징을 추출하는 방식이 있다[4,5]. 하지만 두 방법 모두 전체 알고리즘에서 일부 처리과정을 생략함으로 인해서 지문의 정합에 있어서 특징으로 사용되는 코어 검색의 정확도가 떨어지는 단점이 있다. 최근 연구된 그리드 기반의 매칭 알고리즘은 앞의 두 방법에 비해 특징 추출에 대한 정확도는 높아졌지만 처리시간이 길다는 단점이 발생하였다[6]. 이러한 단점들을 해결하기 위해 본 논문에서는 전처리, 특징추출, 후처리 과정들을 모두 수행하면서 빠른 처리가 가능하게 하기위해 이전 연구에 의해 개발된 다중 접근 메모리 시스템인 MAMS-PP64를 사용하였다. MAMS-PP64는 64개의 프로세서를 포함하며 각 프로세서에서 메모리에 동시에 다중 접근이

가능하도록 설계된 시스템이다[7].

지문인식 특징추출 알고리즘은 “지문인식 기술 동향[2]”에서 정의한 방식에 따라 전처리, 특징 추출, 후처리 3단계로 구분하여 알고리즘을 완성하였다. 다음으로 완성된 지문인식 특징 추출 알고리즘이 MAMS-PP64에서 동작하도록 하기 위해 병렬처리가 가능한 알고리즘으로 변환하였다[8]. 본 논문에서는 시리얼 프로세서를 통해 처리한 지문의 특징 추출에 대한 결과와 MAMS-PP64를 통해 처리한 데이터의 결과에 대해 correlation을 이용한 검증을 통해 제안된 방법의 신뢰도를 확인하였고, 처리속도 또한 1.56배가 향상되었음을 보인다.

2. 시리얼 프로세서를 이용한 특징 추출 알고리즘

본 논문에서는 MAMS-PP64가 병렬 처리 방식을 가지므로 이와 상대적인 개념인 일반적인 범용 프로세서에 대해 시리얼 프로세서로 명명하는 것으로 한다. 시리얼 프로세서에서 지문인식 시스템은 사용자의 지문을 읽고 이를 데이터베이스에 저장되어 있는 데이터와 비교하여 개인을 인증하는 방식이다. 그림 1은 시리얼 프로세서를 이용한 지문인식 시스템의 흐름도이다. 지문인식 시스템의 특징 추출 알고리즘은 전처리, 특징 추출, 후처리 단계로 구분하며. 전처리 단계에서는 배경과 지문을 분리하기 위해 분할(Segmentation) 후 이진화(Binarization)를 수행하

그림 1. 지문인식 시스템 흐름도

며, 그리고 지문의 융선(Ridge)과 골(Valley)의 두께를 이용하여 방향 계산(Directional Calculation)을 수행하며, 가보 필터(Gabor filter)를 이용하여 노이즈가 제거된 융선을 추출한다. 다음으로 특징 추출 단계에서는 융선은 검은색, 골은 흰색으로 처리하는 세선화(Thinning)를 수행한다. 마지막으로 후처리 단계에서는 지문에 대한 특징을 추출하기 위해 Minutiae Extraction을 수행한다[2].

2.1 시리얼 프로세서의 전처리 단계

2.1.1 분할 (Segmentation)

분할(Segmentation)은 영상에서 물체와 배경을 분리하는 과정으로써 그림 2의 (a)에서 (d)의 순으로 처리하여 분할에 대한 결과 영상을 획득하였다. 그림 2는 분할을 위해서 처리하는 일반적인 과정으로써 그림 2에서 (a)는 스캐너로부터 획득한 컬러 영상, (b)는 (a)를 회색음영으로 변환한 영상, (c)는 (b)에서 임계값(threshold)을 통해 이진화한 영상, 그리고 (d)는 (c)영상에 침식(erosion) 연산을 수행한 결과이다.

2.1.2 이진화 (Binarization)

이진화(Binarization)는 회색음영 영상에서 이진화 영상을 얻기 위한 과정으로써 그림 3의 (a)에서 (f)의 순으로 처리하였다. 그림 3에서 (a)는 회색음영 영상, (b)는 (a)를 반전시킨 영상, (c)는 (b)에 스무딩 필터(Smoothing filter)를 적용한 영상, (d)는 (c)와 (a)를 ADD연산한 영상, (e)는 (d)에서 임계값을 통

그림 2. 시리얼 프로세서의 세그멘테이션 처리 결과

그림 3. 시리얼 프로세서의 이진화 처리 결과

해 이진화한 결과이다. 그리고 (f)는 (e)와 그림 2의 (d)를 AND연산을 통해 얻은 결과 영상이다.

2.1.3 방향 계산(Direction Calculation)

융선과 골 두께의 비율은 블록 단위로 계산[9]하며 각 블록의 융선 방향에 수직인 방향에 놓여있는 픽셀의 명암값을 이용하여 계산할 수 있다. 각 블록에서 구해진 융선 두께와 골 두께를 이용하여 각 블록의 융선과 골 두께의 비율을 구한 후, 전체 영상에서 이 값의 평균을 얻는다.

방향성 변화도는 블록 방향성을 이용하여 방향성의 변화를 측정한 값이다. 방향성 정보는 다음과 같이 계산한다[10].

- ㉠ 지문영상을 $W \times W$ 크기의 블록으로 나눈다.
- ㉡ Sobel 연산을 이용하여 각 픽셀의 gradient $\partial_x(i, j)$ 와 $\partial_y(i, j)$ 를 계산한다.
- ㉢ 각 블록의 방향성을 식 (1), (2), (3)을 이용하여 계산한다[11].

$$V_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u, v)\partial_y(u, v) \quad (1)$$

$$V_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u, v)\partial_y^2(u, v)) \quad (2)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(i, j)}{V_x(i, j)} \right) \quad (3)$$

여기서 식 (1)과 (2)를 이용하여 (i, j) 번째 픽셀을 중심으로 하는 블록의 수평방향 변화량 $V_x(i, j)$ 와 수직방향 변화량 $V_y(i, j)$ 를 구한 후 식 (3)을 이용하여 (i, j) 번째 픽셀을 중심으로 하는 블록의 방향성과 수직인 값 $\theta(i, j)$ 를 구한다. 본 논문에서는 블록 크기를 모두 $w=16$ 으로 하였으며, 이렇게 추출된 특징 값들은 실험에 사용하기 위해 특징 별로 0에서 1사이의 값으로 정규화 하였다.

2.1.4 가보 필터 (Gabor filter)

가보 필터는 주파수와 방향성을 가지며 앞서 구현한 방향성 정보를 이용하여 가보 필터 계수를 구하여 filtering을 수행한다. 그 결과가 양수이면 골 (Valley), 음수이면 융선(Ridge)으로 결정한다.

융선과 골을 보호하면서 노이즈를 제거하는 융선 추출방법으로 단순히 영상 밝기값을 임계값으로 이진화하는 방법에 비해 그 결과가 우수하다. 또한 방향성과 주파수 대역에 맞는 부분은 증폭되고 그렇지 않은 부분은 감쇄된다[11,12]. 그 결과 그림 4와 같이 시리얼 프로세서 결과를 볼 수 있다.

그림 4. 가보 필터 결과 영상 (필터적용 전 : 좌, 필터적용 후: 우)

2.2 특징 추출 단계

2.2.1 세선화 (Thinning)

세선화는 T.Y. Zang과 C.Y.Suen에 의해 제안된 병렬 세선화 알고리즘(ZS)을 2개의 서브 반복 단계로 구성하였다. 여기서 $N(P1)$ 은 그림 5에서와 같이 $P1$ 화소의 이웃하는 8방향의 화소들 중 1(영역)의 값을 갖는 화소의 개수이고, $S(P1)$ 은 8방향 화소들의 set $P2 \sim P9$ 에서의 0-1로의 천이 개수이다[13].

1 단계: 다음의 식의 조건이 만족되는 화소를 제거한다.

- 조건 : 1) $2 \leq N(P1) \leq 6$
- 2) $S(P1) = 1$
- 3) $P2 * P4 * P6 = 0$
- 4) $P4 * P6 * P8 = 0$

2 단계: 1단계 조건 중 3), 4)만 다음과 같이 바뀐다.

- 조건 : 3) $P2 * P4 * P8 = 0$
- 4) $P2 * P6 * P8 = 0$

이와 같은 알고리즘을 구현한 결과 그림 6과 같은 시리얼 프로세서 결과를 볼 수 있다.

P9	P2	P3
P8	P1	P4
P7	P6	P5

그림 5. ZS 알고리즘에서 3×3마스크

그림 6. 시리얼 프로세서의 thinning 결과

2.3 후처리 단계

2.3.1 Minutiae Extraction

Minutiae Extraction은 영상에서 명암값의 평균 및 분산, 융선 방향성의 연속성 정도, 골 두께 비율, 지문의 선명도를 나타내주는 블록 방향성 차이를 추출한다. 영상의 평균과 분산은 명암값을 이용하여 아래와 같이 계산한다. 평균값(Mean)은 영상 전체의 명암에 대한 평균을 나타내며, 분산은 영상 전체에서 각 픽셀 간의 균일성 정도를 나타낸다[14].

$$Mean = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I(i, j) \quad (4)$$

$$Variance = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i, j) - Mean)^2 \quad (5)$$

식 (4)와 (5)에서 입력영상은 $N \times M$ 크기이며 $I(i, j)$ 는 영상의 (i, j) 번째 픽셀의 명암값을 의미한다. Minutiae Extraction을 수행한 영상은 그림 7과 같다.

그림 7. Minutiae Extraction의 처리 결과

3. FPGA를 이용한 MAMS-PP64

일반적으로 디지털 영상은 픽셀 $I(i, j)$ 들의 $M \times N$ 행렬로 나타낸다. MAMS-PP64에서 임의의 정수 p, q 를 영상 크기에 대한 설계 상수로 하면, $p \times q$ 개의 영상의 점을 블록, 행, 열의 형태로 동시에 접근할 수 있는 다중접근기억장치로 구성할 수 있다. 이 때 주소 생성회로는 메모리 모듈을 배치하는 모듈 할당 함수와 배치된 모듈 내의 주소를 결정하는 주소 할당 함수를 이용한다. 병렬처리시스템의 병렬처리는 PE(Process Elements)와 파이프라인 SIMD(Pipeline Single Instruction Multiple Data) 구조를 갖는 MAMS로 구성되어 있다[7,15]. 본 논문에서 지문인식 시스템의 특징 추출 알고리즘을 위하여 MAMS-PP64를 사용하였으며, 그 구조는 명령어 관리를 하는 CU(Control Unit), 64개의 PE, 그리고 67개의 MM(Memory Module)으로 이루어져 있다. 본 논문에서 지문인식 시스템의 특징 추출 알고리즘에 이용하는 MAMS-PP64의 구성도는 그림 8과 같고, 이를 16개의 PE로 구성된 MAMS-PP16 보드에서 그림 9와 같이 64개의 PE로 확장한 MAMS-PP64 보드로 제작하였다[16].

그림 8에서 MAMS-PP64는 호스트 컴퓨터에서 Instruction을 받아 64개의 Processing Elements (PE)가 동시에 병렬 처리 하는 SIMD(Single Instruction Multiple Data) 구조로써, MAMS-PP64를 이용한 특징 추출 알고리즘을 구현하기 위해서 시리얼 프로세서의 알고리즘을 병렬 처리가 가능한 구조로 알고리즘을 재구성하고, 8개의 Sub-array Type [7]을 통해 마스크 처리를 효과적으로 처리 할 수 있

그림 8. MAMS-PP64 구성도

그림 9. MAMS-PP64 보드

도록 지문인식 특징추출 알고리즘을 명령어 코드 (Instruction Code)로 변환하여 처리한다.

3.1 MAMS 병렬처리를 이용한 마스크 처리

MAMS의 병렬 처리 및 마스크 처리 방법은 Sub-array Type, 하나의 PE가 처리할 마스크의 크기 그리고 PE가 처리해야하는 연산으로 구성할 수 있다. 만약 그림 10에서의 (a)와 같은 영상에서 MAMS-PP4를 이용하여 3×3 마스크 크기로 침식(Erosion) 연산에 대한 처리를 구현한다면, 8개의 Sub-array Type중에 SEB(South East Block)을 사용하고, 식 (6)과 같이 PE가 실행할 수 있는 병렬처리 알고리즘을 구현할 수 있다.

$$M(i, j) = \sum_{x=i-1}^{x=i+1} \sum_{y=j-1}^{y=j+1} I(x, y) = 9 ? 1 : 0 \quad \begin{cases} SEB \\ 3 \times 3 \end{cases} \quad (6)$$

그림 10과 같이 지금 현재 처리할 위치가 (1, 3)라고 가정하면 Sub-array Type의 SEB이므로 $PE_0 = (1, 3)$, $PE_1 = (2, 3)$, $PE_2 = (1, 4)$, $PE_3 = (2, 4)$ 로 설정되고 그림 10의 (b)와 같이 설정된다. 그림 10의 (c)는 PE_0 , (d)는 PE_1 , (e)는 PE_2 , (f)는 PE_3 가 처리할 마스크의 위치이며, 9 clock에서 모든 PE들이 연산을 종료하게 되고, 그 결과로 (g)와 같은 결과 영상을 보여준다.

4. MAMS-PP64를 이용한 특징 추출 알고리즘

4.1 전처리 단계

4.1.1 분할 (Segmentation)

MAMS-PP64에서의 Segmentation은 시리얼 프

그림 10. MAMS-PP4 병렬처리를 이용한 마스크 처리 과정

로세서의 처리과정과 순서가 동일하고, 3.1과 같이 병렬처리 및 마스크 처리가 가능한 알고리즘으로 수정하였고, 알고리즘의 단계는 다음과 같다.

1 단계 : 입력된 지문 Color영상을 식 (7)을 이용하여 Gray영상 $G(i, j)$ 를 계산한다.

$$G(i, j) = (R(i, j) + G(i, j) + B(i, j)) / 3 \quad \left\{ \begin{matrix} ROW \\ 1 \times 1 \end{matrix} \right. \quad (7)$$

2 단계 : 1단계에서 계산된 $G(i, j)$ 를 식 (8)을 이용하여 이진화영상 $b0(i, j)$ 를 계산한다.

$$b0(i, j) = G(i, j) \geq Threshold ? 1 : 0 \quad \left\{ \begin{matrix} ROW \\ 1 \times 1 \end{matrix} \right. \quad (8)$$

3 단계 : 2단계에서 계산된 $b0(i, j)$ 를 식 (9)를 이용하여 Segmentation영상 $S(i, j)$ 를 계산한다.

$$S(i, j) = \sum_{x=i-1}^{x=i+1} \sum_{y=j-1}^{y=j+1} b0(x, y) = 9 ? 1 : 0 \quad \left\{ \begin{matrix} ROW \\ 3 \times 3 \end{matrix} \right. \quad (9)$$

위와 같은 단계를 실행한 결과 그림 11은 MAMS-PP64의 분할(Segmentation) 결과이며, 시리얼 프로

그림 11. MAMS-PP64를 이용한 세그멘테이션 처리 결과

세서의 분할 결과와 같음을 확인하였다.

4.1.2 이진화 (Binarization)

MAMS-PP64에서의 이진화는 시리얼 프로세서의 처리과정과 순서가 동일하고, 3.1과 같이 병렬처리 및 마스크 처리가 가능한 알고리즘으로 수정하였고, 알고리즘의 단계는 다음과 같다.

1 단계 : Segmentation에서 계산된 $G(i, j)$ 영상을 식 (10)을 이용하여 반전영상 $Re(i, j)$ 를 계산한다.

$$Re(i, j) = 255 - G(i, j) \quad \left\{ \begin{matrix} ROW \\ 1 \times 1 \end{matrix} \right. \quad (10)$$

2 단계 : 1단계에서 계산된 $Re(i, j)$ 를 식 (11)을 이용하여 스무딩 필터(Smoothing filter) 처리된 영상 $Sf(i, j)$ 를 계산한다.

$$Sf(i, j) = \sum_{x=i-1}^{x=i+1} \sum_{y=j-1}^{y=j+1} (Re(x, y) \times \frac{1}{9}) \quad \left\{ \begin{matrix} SEB \\ 3 \times 3 \end{matrix} \right. \quad (11)$$

3 단계 : 2단계에서 계산된 $Sf(i, j)$ 와 $G(i, j)$ 를 식 (12)를 이용하여 ADD영상 $Ad(i, j)$ 를 계산한다.

$$Ad(i, j) = Sf(x, y) + G(x, y) \quad \left\{ \begin{matrix} ROW \\ 1 \times 1 \end{matrix} \right. \quad (12)$$

4 단계 : 3단계에서 계산된 $Ad(i, j)$ 를 식 (13)을 이용하여 이진화 영상 $b1(i, j)$ 를 계산한다.

그림 12. MAMS-PP64의 이진화 처리 결과

$$b1(i, j) = Ad(i, j) \geq Threshold ? 1 : 0 \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (13)$$

5 단계 : 4단계에서 계산된 $b1(i, j)$ 과 분할 단계에서 계산된 $S(i, j)$ 를 식 (14)를 이용하여 이진화 영상 $B(i, j)$ 를 계산한다.

$$B(i, j) = b1(x, y) \& S(x, y) \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (14)$$

위와 같은 단계를 실행한 결과 그림 12은 MAMS-PP64의 이진화 결과이며, 시리얼 프로세서의 결과와 같음을 확인하였다.

4.1.3 방향 계산(Direction Calculation)

MAMS-PP64에서 방향 계산은 시리얼 프로세서의 처리 과정에서 각 블록의 윤선 방향의 수직인 방향에 놓여있는 명암값을 이용한 블록 단위로 계산하는 방식에서 병렬성과 마스크 처리를 위하여 마스크를 사용하는 방식으로 알고리즘을 변경하여 사용하

그림 13. 방향검출을 위한 8방향 템플릿

였고, 방향을 계산하는 방식은 그림 14와 같은 방향 템플릿을 마스크로 이용하여 지문 윤선을 처리하고 그 결과 8방향에 대한 방향성 정보를 가지는 0~1의 정규화 영상을 추출한다. 이와 같이 MAMS-PP64에서 방향 계산 처리를 위한 알고리즘은 식 15~22와 같이 계산을 이용하여 D0~D7의 8방향 영상을 구할 수 있다. 그리고 수식에서의 m 은 그림 13에서의 각 방향성들의 템플릿 마스크를 뜻한다.

$$D0(i, j) = \sum_{x=i-2}^{i+2} \sum_{y=j-1}^{j+1} m \times B(x, y) = 9 ? 1 : 0 \begin{cases} x \neq i-1, x \neq i+1 : m = 0 \\ other : m = 1 \\ \{ ROW \\ 5 \times 3 \end{cases} \quad (15)$$

$$D1(i, j) = \sum_{x=i-2}^{i+2} \sum_{y=j-2}^{j+2} m \times B(x, y) = 9 ? 1 : 0 \begin{cases} x=i-2, j-2 \leq y \leq j : m = 1 \\ x=i, j-1 \leq y \leq j+1 : m = 1 \\ x=i+2, j \leq y \leq j+2 : m = 1 \\ other : m = 0 \\ \{ ROW \\ 5 \times 6 \end{cases} \quad (16)$$

$$D2(i, j) = \sum_{x=i-3}^{i+3} \sum_{y=j-2}^{j+2} m \times B(x, y) = 9 ? 1 : 0 \begin{cases} y=j-2, i-3 \leq x \leq i-1 : m = 1 \\ y=j, i-1 \leq x \leq i+1 : m = 1 \\ y=j+2, i+1 \leq x \leq i+3 : m = 1 \\ other : m = 0 \\ \{ ROW \\ 7 \times 5 \end{cases} \quad (17)$$

$$D3(i, j) = \sum_{x=i-2}^{i+2} \sum_{y=j-2}^{j+2} m \times B(x, y) = 9 ? 1 : 0 \begin{cases} y=j-2, i-2 \leq x \leq i : m = 1 \\ y=j, i-1 \leq x \leq i+1 : m = 1 \\ y=j+2, i \leq x \leq i+2 : m = 1 \\ other : m = 0 \\ \{ ROW \\ 5 \times 5 \end{cases} \quad (18)$$

$$D4(i, j) = \sum_{x=i-1}^{i+1} \sum_{y=j-2}^{j+2} m \times B(x, y) = 9 ? 1 : 0$$

$$\begin{cases} y=j-2 : m=1 \\ y=j : m=1 \\ y=j+2 : m=1 \\ \text{other} : m=0 \\ \{ROW \\ 3 \times 5 \end{cases} \quad (19)$$

$$D5(i, j) = \sum_{x=i-1}^{i+1} \sum_{y=j-2}^{j+2} m \times B(x, y) = 9 ? 1 : 0$$

$$\begin{cases} x=i-1, j+1 \leq y \leq j+3 : m=1 \\ x=i, j-1 \leq y \leq j+1 : m=1 \\ x=i+1, j-3 \leq y \leq j-1 : m=1 \\ \text{other} : m=0 \\ \{ROW \\ 3 \times 7 \end{cases} \quad (20)$$

$$D6(i, j) = \sum_{x=i-3}^{i+3} \sum_{y=j-2}^{j+2} m \times B(x, y) = 9 ? 1 : 0$$

$$\begin{cases} y=j-2, i+1 \leq x \leq i+3 : m=1 \\ y=j, i-1 \leq x \leq i+1 : m=1 \\ y=j+2, i-3 \leq x \leq i-1 : m=1 \\ \text{other} : m=0 \\ \{ROW \\ 5 \times 5 \end{cases} \quad (21)$$

$$D7(i, j) = \sum_{x=i-3}^{i+3} \sum_{y=j-1}^{j+1} m \times B(x, y) = 9 ? 1 : 0$$

$$\begin{cases} y=j-1, i+1 \leq x \leq i+3 : m=1 \\ y=j, i-1 \leq x \leq i+1 : m=1 \\ y=j+1, i-3 \leq x \leq i-1 : m=1 \\ \text{other} : m=0 \\ \{ROW \\ 7 \times 3 \end{cases} \quad (22)$$

4.1.4 가보 필터 (Gabor filter)

MAMS-PP64에서 가보 필터는 앞서 구현한 8개의 방향성 정보를 이용하여 4개의 방향성에 대하여 식 (23)~(26)을 이용하여 계산하고, 계산된 방향성에 따라 그림 14에서 4 방향 7×7 가보 필터 마스크를 중 하나를 선택하여 식 (27)과 같이 컨볼루션(Convolution) 연산을 처리하여 가보필터 영상을 구한다. 알고리즘의 처리 단계는 다음과 같다.

1단계 : 이진화 영상 $B(i, j)$ 와 방향 계산단계에서 계산된 $D0(i, j) \sim D7(i, j)$ 을 이용하여 4개의 방향성을 가지는 $D[4](i, j)$ 를 구한다. $D[0]$ 에는 수평 방향성, $D[1]$ 에는 수직 방향성, $D[2]$ 에는 대각 방향성, $D[3]$ 에는 역대각 방향성을 가지는 배열을 구성한다.

$$D[0] = (B(i, j) > 0 \text{ and } D0(i, j) = 1) ? 1 : 0 \quad \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (23)$$

$$D[1] = (B(i, j) > 0 \text{ and } D4(i, j) = 1) ? 1 : 0 \quad \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (24)$$

-11	-82	-80	-90	-80	-82	-11
-2	-7	-22	-22	-22	-7	-2
22	82	82	82	82	82	22
88	100	100	100	100	100	88
22	82	82	82	82	82	22
-2	-7	-22	-22	-22	-7	-2
-11	-82	-80	-90	-80	-82	-11

(a) 가로방향 마스크

-11	-2	22	88	22	-2	-11
-82	-7	82	100	82	-7	-82
-80	-22	82	100	82	-22	-80
-80	-22	82	100	82	-22	-80
-80	-22	82	100	82	-22	-80
-82	-7	82	100	82	-7	-82
-11	-2	22	88	22	-2	-11

(b) 세로방향 마스크

-90	-80	-82	-11	-2	22	88
-80	-22	-22	-7	82	100	22
-82	-22	82	82	100	82	-2
-11	-7	82	100	82	-7	-11
-2	82	100	82	82	-22	-82
22	100	82	-7	-22	-22	-80
88	22	-2	-11	-82	-80	-90

(c) 대각선방향 마스크

88	22	-2	-11	-82	-80	-90
22	100	82	-7	-22	-22	-80
-2	82	100	82	82	-22	-82
-11	-7	82	100	82	-7	-11
-82	-22	82	82	100	82	-2
-80	-22	-22	-7	82	100	22
-80	-80	-82	-11	-2	22	88

(d) 역대각선방향 마스크

그림 14. 4방향 7×7 가보필터 마스크

$$D[2] = (B(i, j) > 0 \text{ and } (D1(i, j) = 1 \text{ or } D2(i, j) = 1) ? 1 : 0 \quad \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (25)$$

$$D[3] = (B(i, j) > 0 \text{ and } (D5(i, j) = 1 \text{ or } D6(i, j) = 1 \text{ or } D7(i, j) = 1) ? 1 : 0 \quad \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (26)$$

2단계 : 1단계에서 구성된 $D[4](i, j)$ 배열을 이용하여 식 (27)과 같이 컨볼루션(Convolution) 연산 처리하여 그림 15와 같은 가보필터 결과 $Gf(i, j)$ 를 볼 수 있다.

$$Gf(i, j) = \sum_{v=0}^4 (D[v] = 1) ? \left(\sum_{x=i-3}^{x=i+3} \sum_{y=j-3}^{y=j+3} (Mv(x, y) \times B(x, y)) : 0 \quad \begin{cases} ROW \\ 7 \times 7 \end{cases} \quad (27)$$

그림 15. MAMS-PP64의 가보필터 결과

4.2 특징 추출 단계

4.2.1 세선화 (Thinning)

MAMS-PP64에서의 세선화(Thinning)는 시리얼 프로세서의 처리과정과 순서가 동일하고, 3.1과 같이

병렬처리 및 마스크 처리가 가능한 알고리즘으로 구현하였고, 알고리즘의 단계는 다음과 같다.

1단계: 가보 필터 단계에서 계산된 $Gf(i, j)$ 를 이용하여 (x, y) 위치의 주변에 계수를 카운터하기 위해 식 (28)과 같이 계산하여 $c(x, y)$ 를 구한다.

$$c(x, y) = \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} Gf(i, j) = 1? 1 : 0 \quad \begin{cases} ROW \\ 1 \times 1 \end{cases} \quad (28)$$

2단계: 1단계에서 계산된 $c(x, y)$ 를 이용하여 식 (29)를 계산하여 $T1(x, y)$ 를 구한다.

$$T1(x, y) = 2 < c(x, y) < 8? \\ ((Gf[i][j+1]*Gf[i-1][j]*Gf[i][j-1]) \equiv 0 \text{ and } \begin{cases} ROW \\ 1 \times 1 \end{cases} \\ Gf[i-1][j]*Gf[i+1][j]*Gf[i][j-1] \equiv 0)? \quad 0: 1: 1 \quad (29)$$

3단계: 1단계에서 계산된 $c(x, y)$ 를 이용하여 식 (30)을 계산하여 $T2(x, y)$ 를 구한다.

$$T2(x, y) = 2 < c(x, y) < 8? \\ ((Gf[i-1][j] \equiv 0 * Gf[i][j+1]*Gf[i+1][j]) \equiv 0 \text{ and } \begin{cases} ROW \\ 1 \times 1 \end{cases} \\ Gf[i][j+1]*Gf[i+1][j]*Gf[i][j-1] \equiv 0)? \quad 0: 1: 1 \quad (30)$$

이와 같은 알고리즘을 구현한 결과 그림 16과 같은 MAMS-PP64 결과를 볼 수 있고, 시리얼 프로세서의 결과와 같음을 확인하였다.

그림 16. MAMS-PP64의 thinning 결과

4.3 후처리

4.3.1 Minutiae Extraction

MAMS-PP64에서는 시리얼 Process에서 사용하였던 수식에 그림 17과 같은 3×3 mask를 이용하여 Minutiae Detection을 구현하고, 식 (31)과 같이 알고

$$F(i, j) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad I(i, j) = \begin{cases} 1 \text{ at Ridge Ending} \\ 2 \text{ at Ridge Bifurcation} \end{cases}$$

그림 17. 특징 추출을 위한 3×3 마스크

그림 18. Minutiae Extraction의 처리 결과

리즘을 구성하였다. 그 결과 그림 18과 같은 MAMS-PP64 결과를 볼 수 있고, 시리얼 프로세서의 결과와 같음을 확인 하였다.

$$Me(x, y) = \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} Gf(i, j) \quad (31) \\ \begin{cases} \text{if}(Me(x, y) \equiv 3) \\ \quad Me(x, y) = 1 & //bifurcation \\ \text{elseif}(Me(x, y) \equiv 1) \\ \quad Me(x, y) = 1 & //end \\ \text{else} \\ \quad Me(x, y) = 0 \\ \quad SEB \\ \quad 3 \times 3 \end{cases}$$

5. 실험 및 성능 평가

지문에 대한 특징 추출 결과를 영상을 통해 확인하기 위하여 특징 추출 알고리즘에 대하여 시리얼 프로세서와 MAMS-PP64에서 동작하는 프로그램을 그림 19과 같이 구성하였다.

상단은 시리얼 프로세서에서 지문의 특징을 추출하는 기능을 수행하며, 하단은 MAMS-PP64에서 지문의 특징을 추출하는 기능을 수행한다. 지문인식 특징 추출 알고리즘을 MAMS-PP64를 이용하여 처리한 결과의 신뢰성을 확인하기 위해서 MAMS-PP64에서 생성된 Minutiae Extraction 영상과 시리얼 프로세서에서 생성된 Minutiae Extraction 영상을 Image Correlation 방법을 사용하여 두 영상간의 유사도를 확인하였다. 이 프로그램을 이용하여 지문에 대한 특징 추출 영상을 획득하여 MAMS-64와 시리얼 프로세서를 통해 처리한 영상을 Correlation을 수행한 결과, 표 1과 같이 시리얼 프로세서를 통해 처리한 영상과 MAMS-PP64를 통해 처리한 영상 간에 99.99%의 유사도를 가짐을 확인하였다. 이것은 MAMS-PP64를 통해 지문에 대한 특징을 추출한 결과가 범용 프로세서인 시리얼 프로세서를 통해 처리

그림 19. 지문인식 프로그램

표 1. Minutiae Extraction 영상 유사도 결과

NO.	Original image 256 X 256	Input Image 128 X 128	Processor	Result image 128 X 128	Correlation (%)
1			Serial processor		99.999987%
			MAMS-PP64		
2			Serial processor		99.999972%
			MAMS-PP64		

한 결과와 일치함을 증명하는 것이다.

그리고 MAMS-PP64(100Mhz)와 시리얼 프로세서(intel @ Core2 Quad 2.66Ghz)의 처리속도를 확인하기 위하여 전처리, 특징 추출, 후처리의 각 단계에

서의 속도를 측정하였다. 표 9에서 Binary Image의 수행 시간은 전처리 단계의 영상처리 알고리즘 수행 결과 시간을 의미하며, Thinning Image의 수행 시간은 특징 추출 단계의 영상처리 알고리즘 수행 결과

표 2. MAMS-PP64와 시리얼 프로세서 실행 결과 비교

1/1000초(millisecond)

처리 횟수	MAMS-PP64				Serial Processor				Ratio
	Binary Image (Segmentation Directional)	Thinning Image	Minutiae Extraction Image	Total	Binary Image (Segmentation Directional)	Thinning Image	Minutiae Extraction Image	Total	
1	1712	227	104	2043	2732	372	120	3224	1.57
2	1712	227	104	2043	2720	373	108	3201	1.56
3	1712	227	104	2043	2724	357	115	3196	1.56
4	1712	227	104	2043	2714	353	119	3186	1.55
5	1712	227	104	2043	2717	353	104	3174	1.55
6	1712	227	104	2043	2724	356	109	3189	1.56
7	1712	227	104	2043	2717	366	118	3201	1.56
8	1712	227	104	2043	2718	356	133	3207	1.56
9	1712	227	104	2043	2769	363	123	3255	1.59
10	1712	227	104	2043	2729	366	118	3213	1.57
Avg	1712	227	104	2043	2726.4	361.5	116.7	3204.6	1.56

시간을 의미한다. 그리고 Minutiae extraction Image의 수행 시간은 후처리 단계의 영상처리 알고리즘 수행 결과를 의미한다.

표 2는 Binary image, Thinning image, Minutiae extraction image를 생성하는 알고리즘에 대하여 10회 수행한 결과로써 시리얼 프로세서의 결과와 MAMS-PP64의 결과를 확인할 때 시리얼 프로세서는 Binary image는 2714~2769ms, Thinning image는 353~373ms, 그리고 Minutiae extraction image는 104~133ms로써 처리 시간이 일정하지 않음을 확인할 수 있고, MAMS-PP64의 결과를 확인하여 보면 Binary image는 1712ms, Thinning image는 227ms, 그리고 Minutiae extraction image는 104ms로써 처리 시간이 항상 일정함을 확인할 수 있다. 또한 전체 알고리즘에 대한 평균적인 처리시간을 비교해 볼 때 MAMS-PP64가 시리얼 프로세서보다 1.56배만큼 처리 속도가 빠르다는 사실을 확인하였다. 만약 현재 FPGA로 구성되어 있는 MAMS-PP64는 실행 클럭(clock)이 100Mhz이지만 ASIC으로 구성하여 시리얼 프로세서와 동일한 클럭(clock)인 2.66Ghz로 제작된다면 시리얼 프로세서보다 40.56배만큼 더 빠르게 처리할 수 있을 것으로 예상된다.

6. 결 론

생체인식 분야중 하나인 지문인식 시스템의 특징 추출 알고리즘에 대한 처리 속도를 향상시키기 위해 많은 연구가 이루어지고 있지만 기존의 방법들은 속도를 향상시키기 위해 특징 추출 알고리즘에서 단순히 전체 알고리즘 중에서 일부 단계를 생략함으로써 처리 속도를 향상시키는 방법들을 사용하였다. 하지만 이러한 방식은 속도는 향상되었지만 정합에 대한 정확성을 보장할 수 없었다.

본 논문에서는 지문의 특징 추출 알고리즘 중 모든 과정을 포함하여 정합에 대한 정확성을 보장하면서 동시에 처리 속도 또한 단축시키기 위한 방법으로써 병렬 처리 방식의 다중 접근 메모리 시스템인 MAMS-PP64를 이용하여 특징 추출 알고리즘을 처리함으로써 지문의 특징 추출에 대한 정확성과 처리 속도의 단축을 가능하게 하였다.

참 고 문 헌

- [1] N.K. Ratha, K. Karu, S. Chen, and A.K. Jain, "A Real-time Matching System for Large Fingerprint Database," *IEEE Transaction on*

- Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 799-813, 1996.
- [2] 반성범, 문지현, 정용화, 김학말, “지문 인식 기술 동향,” 전자통신동향분석, 제16권, 제5호, pp. 46-54, 2001.
- [3] C.J. Lee and S.D. Wang, “A Gabor Filter-Based Approach to Fingerprint Recognition,” *IEEE Workshop on Signal Processing Systems*, pp. 371-378, 1999.
- [4] D. Maio and D. Maltoni, “Direct Gray-Scale Minutiae Detection In Fingerprints,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 1, pp. 27-40, 1997.
- [5] X. Jiang, W.Y. Yau, and W. Ser, “Minutiae Extraction by Adaptive Tracing the Gray Level Ridge of the Fingerprint Image,” *Proc. International Conference on Image Processing*, Vol. 2, pp. 852-856, 1999.
- [6] 양승룡, 김성민, 조성배, “고성능 지문인식을 위한 그리드 기반 매칭 알고리즘,” 한국정보과학회, 학술발표논문집 Vol. 31, No. 1B, pp. 727-729, 2004.
- [7] J.W. Park, “Multiaccess Memory System for Attached SIMD Computer,” *IEEE Transaction on Computers*, Vol. 53, No. 3, pp. 439-452, 2004.
- [8] 이유진, 김재희, 박종원, “16개의 처리기를 가진 다중접근기억장치를 위한 영상처리 알고리즘의 구현에 대한 성능평가,” 대한전자공학회논문지, 제49권, 제3호, pp. 8-14, 2012.
- [9] E. Lim, X. Jiang, and W. Yau, “Fingerprint Quality and Validity Analysis,” *IEEE International Conference on Image Processing*, Vol. 1, pp. 22-25, 2002.
- [10] K. Karu and A. Jain, “Fingerprint Classification,” *Pattern Recognition*, Vol. 29, No. 3, pp. 389-404, 1996.
- [11] A. Rao, *A Taxonomy for Texture Description and Identification*, Springer-Verlag, New York, 1990.
- [12] A.K. Jain and N.K. Ratha, “Object Detection using Gabor Filters,” *Pattern Recognition*, Vol. 30, No. 2, pp. 295-309, 1997.
- [13] 이현우, 김남명, 소병호, 이지혜, 문송철, “스마트폰 카메라 기반의 템플릿 매칭을 활용한 문자 인식,” 한국정보기술응용학회학술대회, pp. 69-72, 2011.
- [14] 윤은경, 대규모 지문인식시스템을 위한 지문영상개선 및 분류, 연세대학교 석사학위 논문, 2004.
- [15] J.W. Park, “Efficient Image Analysis Memory System,” *Journal of The Korea Information Science Society*, Vol. 16, No. 6, pp. 559-566, 1989.
- [16] 임재호, 박성미, 박종원, “영상처리용 16개의 처리기를 위한 다중접근기억장치 및 병렬처리기의 칩 설계,” 멀티미디어학회논문지, 제14권, 제11호, 1401-1408, 2011.

박 종 선

2004년 우송대학교 컴퓨터과학과
학사졸업
2013년 충남대학교 정보통신공학
과 석사졸업
관심분야: 영상처리, 병렬처리

김 재 희

1998년 영산대학교 컴퓨터공학과
학사졸업
2006년 울산대학교 교육대학원
전자계산학과 석사졸업
2010년 충남대학교 정보통신공학
과 박사과정
관심분야: 영상처리, 병렬처리

고 경 식

1986년 동국대학교 전자계산학과
학사졸업
2006년 충남대학교 산업대학원 전
자정보통신공학과 석사졸
업
2006년 충남대학교 정보통신공학
과 박사과정

관심분야: 영상처리, 병렬처리

박 종 원

1979년 충남대학교 전자공학과
학사졸업
1981년 한국과학기술원 전산학과
석사졸업
1991년 한국과학기술원 전산학과
박사졸업

1983년~1993년 충남대학교 전산학과 부교수
1994년~현재 충남대학교 정보통신공학과 교수
관심분야: 영상처리, 병렬처리, 의학영상처리