

논문 2008-45CI-1-1

얼굴 특징 검출 알고리즘의 하드웨어 설계

(Hardware Implementation of Facial Feature Detection Algorithm)

김 정 호*, 정 용 진**

(Jung-ho Kim, and Yong-jin Jeong)

요 약

본 논문에서는 기존에 얼굴 검출에 사용된 ICT(Improved Census Transform) 변환을 이용하여 눈, 코, 입 등의 얼굴 특징을 검출하는 하드웨어를 설계하였다. 파이프라인 구조를 이용하여 동작 속도를 높였고, ICT 변환, 메모리 공유, 동작 과정의 세분화를 통하여 메모리 사용량을 줄였다. 본 논문에서 사용한 알고리즘을 얼굴 검출 및 인식 분야에서 테스트용으로 주로 쓰이는 BioID 데이터베이스(database)를 이용하여 테스트한 결과 100%의 검출률을 보였고, 설계한 하드웨어의 결과도 이와 동일하였다. 또한, Synopsys사의 Design Compiler와 동부아남사의 0.18 μm library를 이용하여 합성한 결과 총 376,821 μm^2 의 결과를 얻었고, 78MHz의 동작 클럭 하에서 17.1msec의 검출 속도를 보였다. 본 논문은 소프트웨어 형태의 알고리즘을 임베디드 하드웨어로 구현함으로써 실시간 처리의 가능성을 보였고, 저가격, 높은 이식성에 대한 가능성을 제시하였다.

Abstract

In this paper, we designed a facial feature (eyes, a mouth, and a nose) detection hardware based on the ICT transform which was developed for face detection earlier. Our design used a pipeline architecture for high throughput and it also tried to reduce memory size and memory access rate. The algorithm and its hardware implementation were tested on the BioID database, which is a worldwide face detection test bed, and its facial feature detection rate was 100% both in software and hardware, assuming the face boundary was correctly detected. After synthesizing the hardware on Dongbu 0.18 μm CMOS library, its die size was 376,821 μm^2 with the maximum operating clock 78MHz.

Keywords : Face detection, Facial feature detection, Embedded system

I. 서 론

지문 인식, 홍채 인식, 정맥 인식과 같은 생체 인식 기술은 오래전부터 다양한 연구가 진행되어 왔다. 이러한 생체 인식 기술 중에서도 얼굴 및 얼굴 특징 검출 기술은 다른 생체 정보에 비해 사용자에게 대해 강제성이 적으며, 비 접촉식으로 거부감이 덜하기 때문에 각광을 받고 있는 연구 분야이다. 또한, 얼굴 특징 검출 기술은 검출된 얼굴 영역을 검증하고, 얼굴 인식을 위한 정보

를 검출하는 등 여러 응용분야에 적용할 수 있는 중요한 기술이다.

각 개인을 구별시켜주는 얼굴의 특징으로는 눈, 눈썹, 코, 입, 귀, 얼굴의 윤곽선 등을 들 수 있다. 이러한 얼굴 특징 정보는 얼굴 검증^[4], 얼굴 인식, 아바타(avatar) 제어 시스템^[5], 감정 인식^[6] 등 다양한 용도로 응용이 가능하다. 본 논문에서는 여러 가지 얼굴 특징 중 눈, 코, 입을 검출하고자 하는 얼굴의 특징으로 정의하였고, 검출 결과는 해당 특징이 존재하는 영역으로 하였다.

이러한 얼굴 특징을 검출하는 방법은 본 논문에서 이용한 방법인 CT(Census Transform) 변환^[1]을 응용한 방법, 여러 개의 특징점을 훈련된 매쉬(mesh)를 이용하여 실시간으로 검출하는 방법^[7], PCA(Principal Components Analysis)를 기반으로 컬러 정보를 이용하여 검출하는 방법^[8] 등 여러 가지 방향으로 연구되어지

* 학생회원, ** 정회원, 광운대학교 전자통신공학과
(Dept. of Electronics and Communications Engineering, Kwangwoon University)

※ 본 연구는 IDEC/IT-SOC 톨 지원과 정보통신부 서울시 혁신 클러스터 (나노 IP-SOC) 사업 및 2007 정보통신부 선도기반기술개발 사업의 지원으로 이루어졌습니다.

접수일자: 2007년2월22일, 수정완료일: 2008년1월11일

고 있다.

얼굴 특징 검출은 사전에 검출된 얼굴 영역에서 눈, 코, 입 등의 사람의 얼굴이 갖고 있는 특징을 검출하는 것이므로 선행 과정으로 얼굴 영역 검출이 필요하다. 이 과정은 하드웨어 설계의 효율성을 고려하여 얼굴 검출에서 사용한 방법과 유사한 방법으로 검출 필터를 달리하여 수행하였다^[3]. 본 논문에서는 얼굴 검출에 사용된 ICT(Improved Census Transform) 변환을 얼굴 특징 검출에 적용시켜 효율적인 하드웨어를 설계하였고, 우수한 성능 결과를 얻었다.

본 논문에서 이용한 얼굴 특징 검출 알고리즘의 특징은 다음과 같다. 검출 속도의 향상을 위하여 검출할 얼굴의 특징에 적합한 영역을 정의하였고, 검출하고자 하는 특징에 맞는 검출 필터를 적용하여 효율적이고 신뢰도 높은 검출 결과를 얻었다. 또한, 메모리를 최대한 공유하도록 설계하고, 검출 단계를 세분화하여 메모리 사용량을 줄였고, 파이프라인 방식을 도입하여 검출 속도를 향상시켰다.

II. 배경 지식 및 얼굴 특징 검출 알고리즘

1. ICT 변환

얼굴 및 얼굴 특징 검출에 있어 가장 큰 문제는 조명 성분 및 포즈이다. MCT변환은^[2] 이러한 조명성분의 변화에 강인하고, CT변환^[1]이 갖고 있는 단점을 보완한 알고리즘이다. 하지만 이는 변환 후 9-비트의 비트스트림(bitstream) 결과가 나오므로, 하드웨어 구현 시 메모리의 낭비가 심하여 적합하지 않은 방법이다. 따라서 이를 개선하여 하드웨어 구현에 좀 더 적합한 형태로 만든 것이 ICT 변환이다^[3].

3×3 픽셀 크기 영역의 중심 픽셀 x 에서의 ICT 변환을 정의하면 다음 식과 같다.

$$T(x) = \bigotimes_{y \in N''} \xi(I(x), I(y)) \quad (1)$$

$$\left\{ \begin{array}{l} N'' : x \text{를 제외한 } x \text{와 맞닿아 있는} \\ \quad 3 \times 3 \text{ 픽셀 크기 영역 내의 이웃 픽셀} \\ \otimes : \text{연결 (concatenate) 연산자} \\ \xi(I(x), I(y)) : \\ \quad = 0 \text{ if } I(x) \leq (I(y) - \text{threshold}) \\ \quad = 1 \text{ if } (I(y) - \text{thre.}) < I(x) < (I(y) + \text{thre.}) \\ \quad = 2 \text{ if } I(x) \geq (I(y) + \text{threshold}) \end{array} \right.$$

ICT 변환을 하게 되면, 3×3 픽셀 크기 영역내의 중

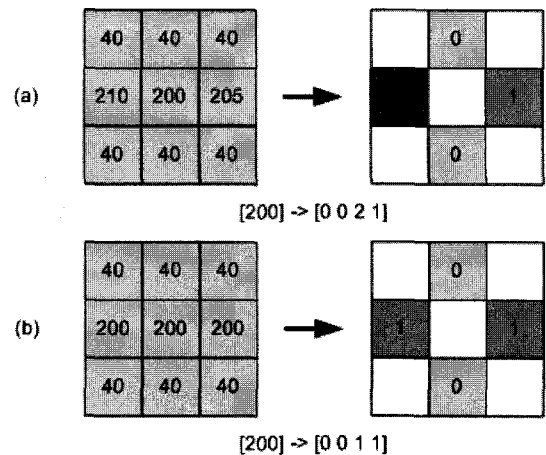


그림 1. ICT 변환의 예

Fig. 1. Example of ICT transform.

심 픽셀 값이 변환된 값으로 대체된다. 변환 결과 네 개의 점이 0, 1, 2의 값을 가지므로, 총 34=81개의 특징이 나온다. 이는 7-비트의 비트스트림으로 나타낼 수 있으므로 효율적인 메모리 사용이 가능하다. 실험을 통하여 threshold의 값은 8이 가장 적합하다고 판단되었고, 이를 이용한 ICT 변환의 예를 그림 1에 나타내었다.

2. 표본 생성 및 필터 학습

본 논문에서 이용한 얼굴 특징 검출 알고리즘은 필터를 이용한 방법이다. 따라서 이 검출 필터를 학습을 통하여 생성하여야 한다. 검출하고자 하는 얼굴의 특징을 왼쪽 눈, 오른쪽 눈, 코, 입으로 정의하였고, 이 중 눈의 경우는 왼쪽 눈과 오른쪽 눈의 형태가 다르므로 정확성을 높이기 위하여 각각 다른 필터를 사용하였다.

얼굴 특징들을 검출하기 위한 필터의 학습 방법은 다음과 같다. 임의의 얼굴 영상들에서 학습을 시키고자 하는 크기에 맞춰 얼굴 영역을 잘라내고, 검출하고자 하는 각각의 특징이 있는 영역들을 추출해낸다. 다음으로, 임의의 얼굴이 아닌 영상들에서도 학습을 시키기 위한 각각의 필터 크기에 맞춰 잘못된 영상을 추출해낸다. 추출된 영상들을 ICT 변환을 통하여 변환하고, 이를 통

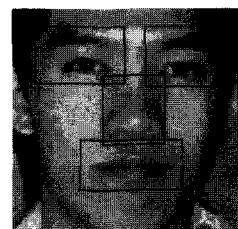


그림 2. 필터 학습용 얼굴 영역

Fig. 2. Facial area for filter training.

하여 나온 분류자(classifier)들을 부스팅(boosting) 알고리즘을 이용하여 얼굴 특징 검출용 필터들을 학습시킨다^[2]. 각 얼굴 특징 검출용 필터의 학습을 위한 영역을 그림 2에 나타내었다.

3. 필터 테이블 정보

학습된 필터 테이블의 구성은 다음과 같다. 검출 속도의 향상을 위하여 각 필터별로 검출 단계를 세분화하였고, 각 단계는 얼굴 특징 검출 윈도우 크기 내의 검사를 위한 픽셀의 X-좌표와 Y-좌표를 포함한다. 또한, 각 좌표에 해당하는 픽셀마다 81개의 분류자를 가지고 있다. 표 1에 얼굴 특징 검출 필터 테이블의 구성을 나타내었다.

표 1. 얼굴 특징 검출 필터 테이블의 구성
Table 1. Organization of facial feature detection filter table.

1단계	X ₁ 좌표 Y ₁ 좌표	00000000	분류자 값	81개
		00000001	분류자 값	
		...		
	10100000	분류자 값	81개	
	00000000	분류자 값		
	00000001	분류자 값		
X ₂ 좌표 Y ₂ 좌표	
	10100000	분류자 값		
	...			
2단계	X ₁ 좌표 Y ₁ 좌표	00000000	분류자 값	81개
		00000001	분류자 값	
		...		
	10100000	분류자 값	...	
	...			
	...			
...	

4. 얼굴 특징 검출 알고리즘

본 논문에서 이용한 얼굴 특징 검출 알고리즘은 검출된 얼굴 영역을 입력받아 눈, 코, 입 영역을 검출하는 알고리즘으로, 사전에 검출된 얼굴 영역에서 각 얼굴 특징을 나타내는 영역을 정의한다. 다음으로 검출하고자 하는 각 영역에 필터를 적용시켜^[2] 눈, 코, 입 후보 영역들을 검출한다. 검출된 각 후보 영역들을 각각 그룹화를 통하여 여러 그룹으로 분류하고, 각 그룹들 간의 신뢰도를 비교하여 가장 높은 신뢰도를 갖는 후보를 대표 특징으로 한다.

모든 얼굴 특징이 검출되지 않는다면, 잘못된 얼굴영

```

// Facial feature detection algorithm
// Input : facial area
// Output : facial feature area

facial_feature_area_calculate (most confidential face,
                              facial feature area);

// 1 = left eye, 2 = right eye, 3 = nose, 4 = mouth
for (i = from 1 to 4){
    for (ratio = begin; ratio >= end; ratio =
ratio*proportion){
        resize (input image, ratio, facial feature area,
                resized image);
        ICT (resized image, transformed image);
        find_candidate (transformed image, ith filter data,
                        facial feature area candidates);
    }
    grouping (facial feature area candidates,
              grouped candidates);
}

for (i = 0; i < (number of grouped left eye *
                number of grouped right eye *
                number of grouped mouth); i++){
    facial_feature_combinator (grouped candidates,
                              facial feature pair);
    affine_transform (input image, facial feature pair,
                     corrected image);
    ICT (corrected image, transformed image);
    find_candidate (transformed image, verification filter
                    data,
                    confidence);
}

verification (confidence, result);
    
```

그림 3. 얼굴 특징 검출 알고리즘의 의사코드

Fig. 3. Pseudo code of facial feature detection algorithm.

역이 입력된 것 혹은 얼굴 특징이 불완전하게 나타나 검출되지 않은 것으로 판단을 하고 얼굴 특징 검출을 실패로 끝낸다. 얼굴 특징이 모두 검출된 경우, 검출된 특징들을 조합하여 얼굴 특징 후보 쌍을 만들어 낸다. 각 얼굴 특징 후보 쌍들을 이용하여 검증에 필요한 형태로 어파인 변환을 하고, 검증 필터를 적용하여 올바르게 검출된 특징들인지 검증한다. 검증이 성공한 경우 각 얼굴 특징이 존재하는 영역을 결과로 내보내고, 검증이 실패한 경우 얼굴 특징들의 검출이 잘못 이루어

진 것으로 판단하여 얼굴 특징 검출을 실패한 것으로 결과를 내보낸다. 전체 알고리즘의 의사 코드를 그림 3에 나타내었다.

III. 하드웨어 설계

1. 전체 하드웨어의 구성

설계된 하드웨어의 구조를 그림 4에 나타내었다. 입력 영상은 외부의 DRAM에 저장을 하였고, 자주 쓰이는 필터 정보는 내부 메모리인 SRAM에 저장하였으며, 나머지 필터 정보는 DRAM에 저장하였다. 또한 각 모듈에 필요한 레지스터(register)들을 정의하여 전체 메모리 맵을 설정하였다. 설정된 메모리 맵을 그림 5에 나타내었다.

전체 하드웨어 시스템과 내부의 각 모듈별 입·출력 포트를 설정하였다. 각 세부 모듈은 8 가지로 나누었고, SRAM과 레지스터를 이용하여 데이터를 공유하며, 알고리즘의 흐름에 따라 순차적으로 동작을 하도록 설계하였다. 디버깅을 위하여 모듈별로 동작을 시킬 수 있도록 설계하였고, 임의로 레지스터 및 SRAM 세팅을 외부에서 할 수 있도록 설계하였다.

영상처리를 위한 임베디드 하드웨어 설계 시에는 빠른 수행속도와 적은 메모리 사용에 초점을 맞추어 설계하여야 한다. 먼저 빠른 수행 속도를 위하여 덧셈기의 경우 CSA(carry save adder)로 피연산자를 빠르게 줄

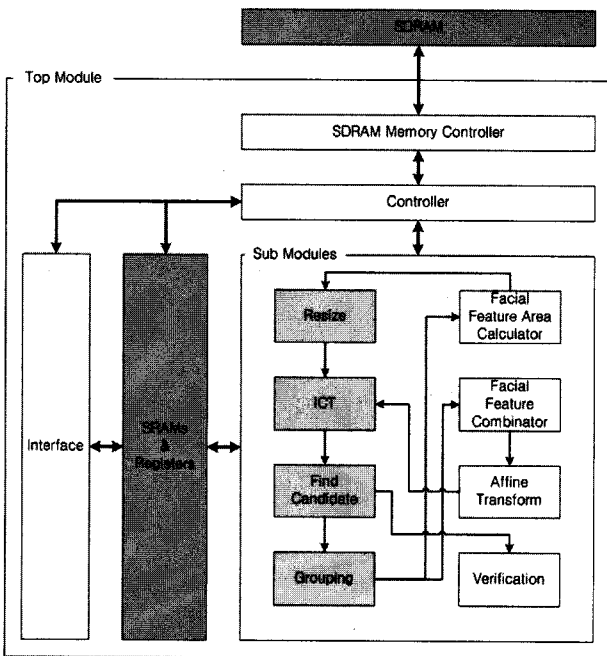


그림 4. 하드웨어 구조
Fig. 4. Hardware architecture.

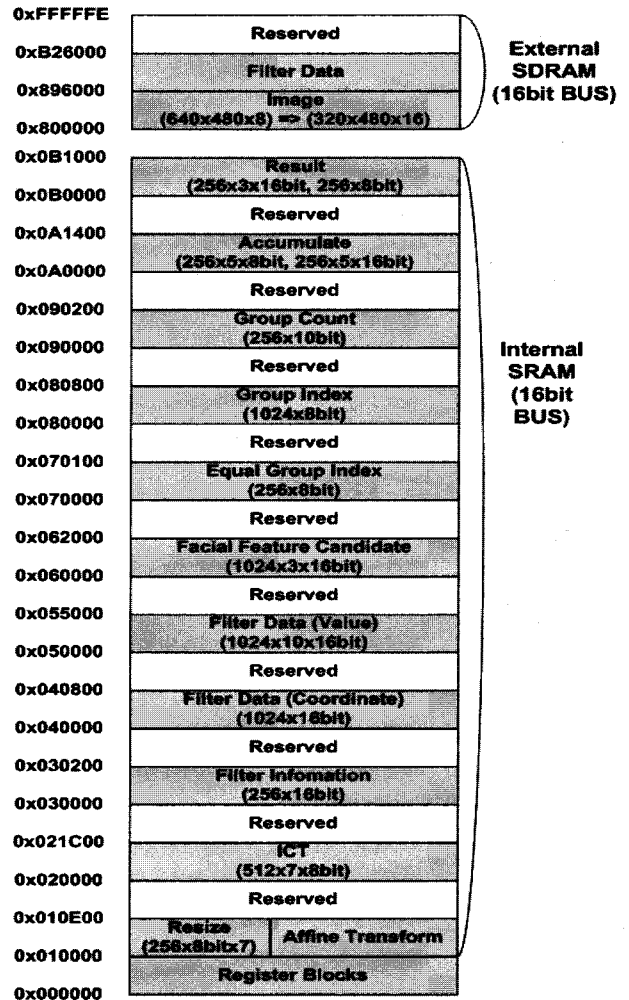


그림 5. 메모리 맵
Fig. 5. Memory map.

여주고 최종단에서 CLA(carry lookahead adder)를 이용하여 더하는 방법으로 속도를 높였다. 또한, 각 덧셈기 및 곱셈기를 높은 클럭 속도로 동작시키기 위하여 내부에 레지스터를 삽입하였고, 효율적인 클럭 사용으로 동작 속도를 높일 수 있는 방법인 파이프라인 구현을 위하여 레지스터를 이용하여 데이터패스를 나누어주었다. 다음으로 적은 메모리 사용을 위하여 모듈간의 데이터를 최대한 공유하도록 설계하였다. 또한 한 영상 전체에 대하여 연산을 하지 않고 영상의 일부분씩을 가져와 연산을 하는 방법을 통하여 보다 적은 메모리를 사용하였다. 다음 절부터 각 세부 모듈에 대한 설명을 나타내었다.

2. 얼굴 특징 영역 계산 모듈

얼굴 특징 영역 계산(facial feature area calculator) 모듈은 기 검출된 얼굴 영역에서 얼굴 특징들을 검출하기 위한 영역을 설정하는 모듈이다. 검출된 얼굴 영역

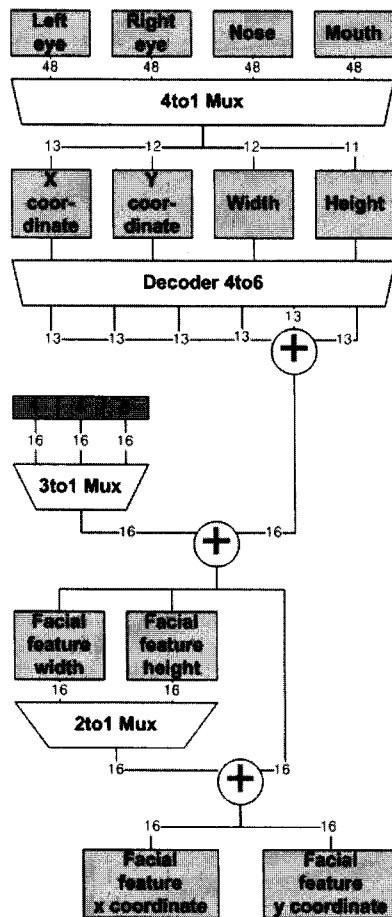


그림 6. 얼굴 특징 영역 계산 모듈의 데이터패스
Fig. 6. Datapath of facial feature area calculator module.

에서 왼쪽 눈, 오른쪽 눈, 코, 입이 위치하는 영역을 계산해내고, 이로 인하여 얼굴 특징 검출에 소요되는 계산량을 줄일 수 있다.

전체 영상에서 검출된 얼굴의 중심 좌표, 가로 길이와 세로 길이를 입력받아 각각의 얼굴 특징이 존재하는 영역의 위치에 대한 중심 좌표, 가로 길이와 세로 길이를 출력으로 한다. DRAM에 있는 입력 영상을 공통적으로 사용하기 위하여 입·출력 좌표의 기준은 전체 영상을 대상으로 한다.

각각의 얼굴 특징 영역들을 계산하는 연산은 같은 수식을 사용하고, 계수만 다르므로 이 4 가지 수치들만 멀티플렉서(multiplexer)를 이용하여 선택하고, 나머지 계산 부분은 공유를 하도록 설계하였다. 이를 통하여 설계된 하드웨어의 면적을 줄일 수 있다. 설계된 하드웨어의 데이터패스를 그림 6에 나타내었다.

3. 영상 축소 모듈

영상 축소(resize) 모듈은 주어진 영상에 존재하는 다양한 크기의 얼굴 특징을 검출해내기 위하여 전체 영상

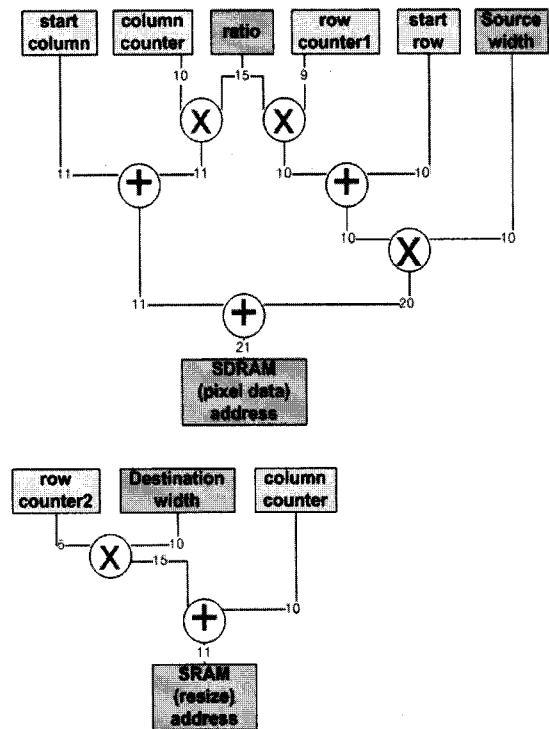


그림 7. 영상 축소 모듈의 데이터패스
Fig. 7. Datapath of resize module.

의 크기를 줄여나가는 기법인 multi-resolution 기법을 구현한 모듈이다. 얼굴 영역을 입력으로 받고, 축소된 영상을 결과로 한다. 결과는 임시로 사용되므로 SRAM에 저장을 하도록 설계하였다.

입력된 영상의 좌표를 기준으로 축소될 영역의 좌표를 계산한 후, 해당 부분의 픽셀 값을 읽어 SRAM에 저장하는 방법을 이용하였다. 이를 통하여 메모리 접근 횟수를 줄일 수 있다. 또한, 다음으로 수행되어지는 모듈인 ICT 모듈이 3×3 픽셀 크기의 영역을 기준으로 수행되어지므로 보다 적은 메모리를 사용하기 위하여 영상을 3줄씩 축소해나간다. 이를 통하여 메모리 사용량을 줄일 수 있었다. 설계된 하드웨어의 데이터패스를 그림 7에 나타내었다.

4. ICT 변환 모듈

ICT 변환(ICT) 모듈은 축소된 영상을 ICT 변환하는 모듈이다. SRAM에서 축소된 영상을 입력으로 읽어와 ICT 변환을 한 후, 변환된 영상을 다시 SRAM으로 써넣는 동작을 한다.

3×3 픽셀 크기 영역을 입력으로 받아 ICT 변환을 하여 변환된 비트스트림을 결과로 내보낸다. 3×3 픽셀 크기 영역 내에서 십자 형태의 5픽셀을 이용하며 변환 결과는 7-비트의 비트스트림으로 나온다. 먼저, 가로 방

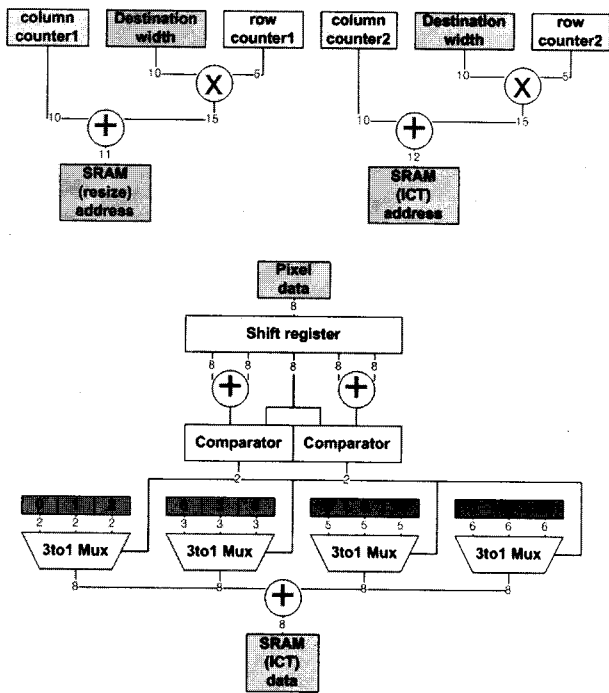


그림 8. ICT 변환 모듈의 데이터패스
Fig. 8. Datapath of ICT transform module.

향으로 한 픽셀 크기만큼씩 슬라이딩을 하면서 변환을 해 나가므로, 하나의 픽셀을 재사용 할 수 있게 된다. 다음으로 세로방향으로 한 픽셀을 슬라이딩 하는데 이 때는 가로 방향의 최종적으로 변환한 곳에서 아래로 슬라이딩을 함으로 하나의 픽셀을 재사용 하게 된다. 따라서 슬라이딩의 형태는 지그재그 방식이 된다. 이와 같이 재사용을 할 수 있는 픽셀을 최대한 활용하는 방식으로 메모리 접근 횟수를 줄일 수 있다. 설계된 하드웨어의 데이터패스를 그림 8에 나타내었다.

5. 후보 영역 검출 모듈

후보 영역 검출(find candidate) 모듈은 변환된 영상에 각각의 검출 필터를 적용하여 후보 SRAM에서 변환된 영상을 읽어오고, SRAM과 DRAM에서 필터 정보를 읽어온다. 이 데이터를 이용하여 검출을 하고, 검출 결과를 SRAM에 써 넣는다^[3].

각 검출 필터의 크기에 맞게 resize모듈과 ICT 변환 모듈을 반복 수행한다. 우선적으로 SRAM에 저장된 필터 정보를 이용하여 얼굴이 아닌 영역을 빠르게 걸러내는 방법을 사용하여 수행 시간을 단축시킬 수 있다. 설계된 하드웨어의 데이터패스를 그림 9에 나타내었다.

설계된 하드웨어의 구조를 보면 나눗셈 연산을 하는 부분이 존재한다. 이것은 피제수(dividend)를 한 비트씩

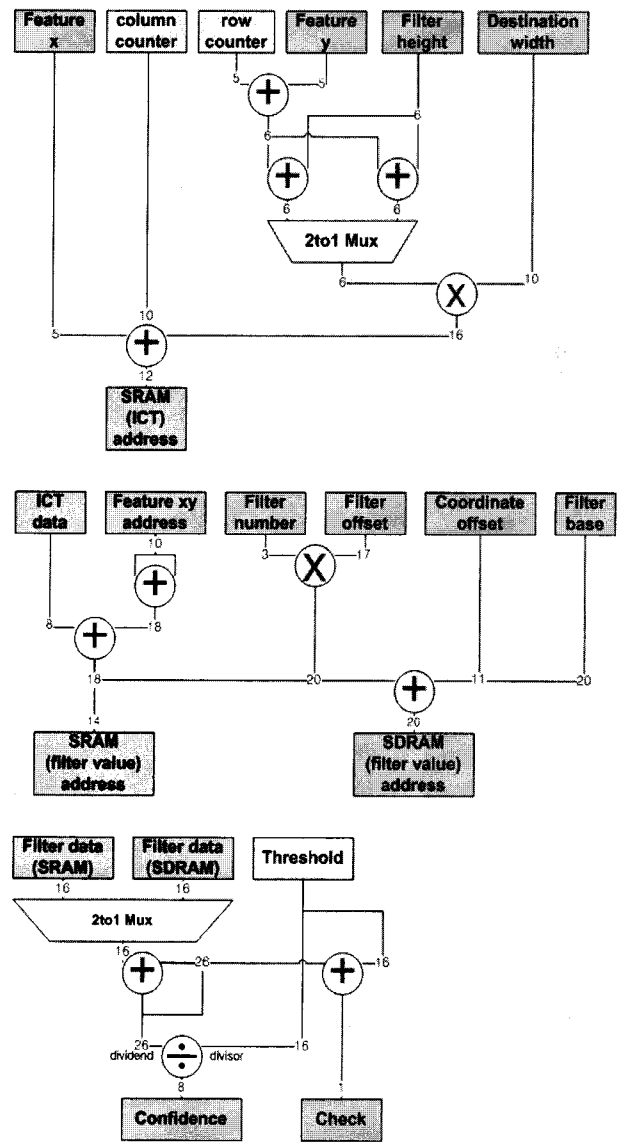


그림 9. 후보 영역 검출 모듈의 데이터패스
Fig. 9. Datapath of find candidate module.

이동(shift) 하면서 제수(divisor)를 빼주어, 매 클럭마다 한 비트 씩의 몫(quotient)이 나오도록 설계하였다.

6. 그룹화 모듈

그룹화(grouping) 모듈은 검출된 후보 영역들을 몇 개의 그룹으로 나눠주는 모듈이다. 영상에 존재하는 다양한 크기들의 후보를 multi-resolution기법을 사용하여 검출을 하게 되므로 동일한 후보가 여러 번 검출될 수 있다. 따라서 이 후보들을 그룹화를 통하여 동일한 후보로 묶어주는 동작을 한다.

SRAM에서 후보 영역에 대한 데이터들을 읽어와 그룹화 동작을 수행한 후, 그룹 결과를 다시 SRAM에 써 넣는다. 그룹 결과는 그룹에 소속된 영역들의 누적 신

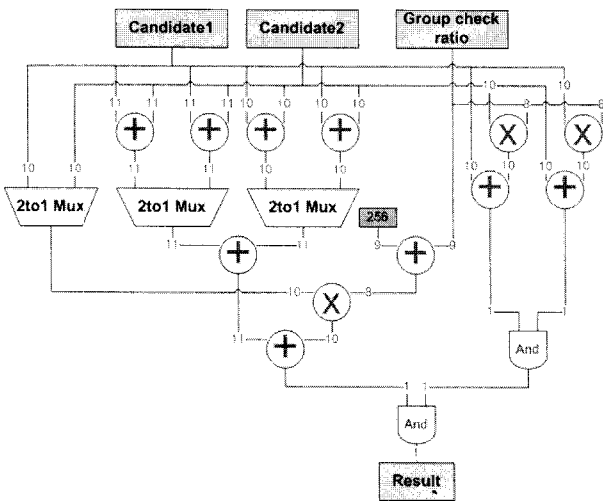


그림 10. 그룹화 모듈의 데이터패스
Fig. 10. Datapath of grouping module.

회도를 통한 가중 평균을 이용하여 계산하였다. 설계된 하드웨어의 데이터패스를 그림 10에 나타내었다.

7. 얼굴 특징 조합 모듈

얼굴 특징 조합(facial feature combinator) 모듈은 검출된 얼굴 특징들을 다시 한 번 검증하기 위하여 조합을 하는 모듈이다. 검출된 왼쪽 눈, 오른쪽 눈, 입에 대한 정보를 SRAM으로부터 읽어 들여 조합을 만들어 낸

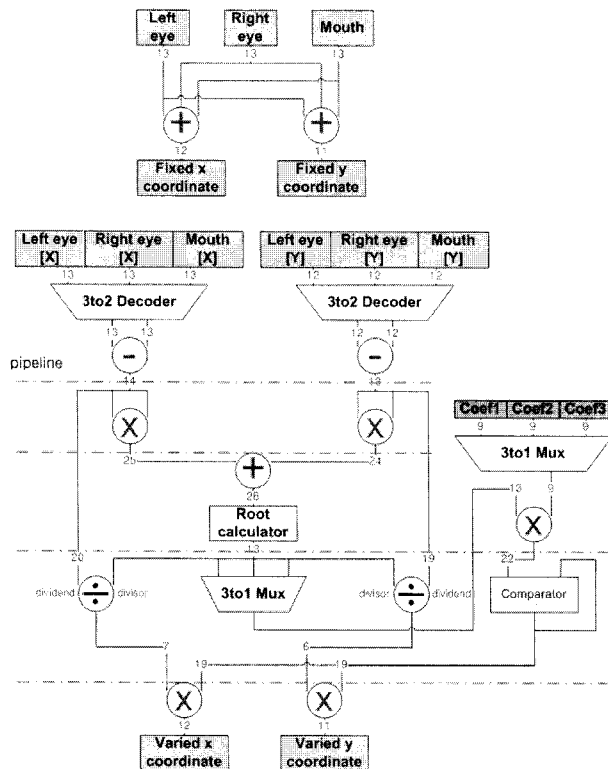


그림 11. 얼굴 특징 조합 모듈의 데이터패스
Fig. 11. Datapath of facial feature combinator module.

다. 또한, 다음 과정인 어파인 변환을 위한 입력 데이터를 파이프라인 형식으로 계산한다. 설계된 하드웨어의 데이터패스를 그림 11에 나타내었고, 파이프라인 단계를 점선으로 구별했다.

8. 어파인 변환 모듈

어파인 변환(affine transform) 모듈은 조합된 얼굴 특징을 이용하여 검증에 필요한 형태로 변환을 하는 모듈이다. 계산된 좌표를 이용하여 SDRAM에서 픽셀 값을 읽어온 후, SRAM에 써 넣는다. SDRAM은 SRAM과 달리 ACK 신호를 이용하여 컨트롤 신호를 제어한다. 이러한 SDRAM과 SRAM의 동작 방식의 차이를 이용하여 SDRAM에서 데이터를 읽는 동안 다음 SRAM의 주소를 미리 계산하는 방식으로 파이프라인 구조를 이용하였다. 이를 통하여 파이프라인 방식을 도입하기 전보다 메모리를 읽고 쓰는 동작 클럭 수를 약 반으로 줄일 수 있다. 설계된 하드웨어의 데이터패스를 그림 12에 나타내었고, 각 파이프라인 단계를 점선으로 구분해 놓았다.

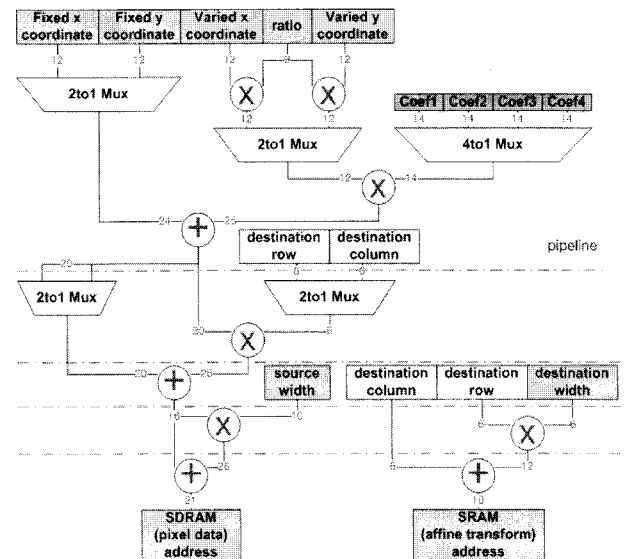


그림 12. 어파인 변환 모듈의 데이터패스
Fig. 12. Datapath of affine transform module.

9. 검증 모듈

검증(verification) 모듈은 검출된 얼굴 영역과 각각의 얼굴 특징 영역들이 올바르게 검출되었는지 여부를 검증하는 모듈이다. 이는 검증용 필터를 사용하여 신뢰도를 계산해내고, 이를 임계치와 비교하는 방법으로 수행한다. 설계된 하드웨어의 데이터패스를 그림 13에 나타내었다.

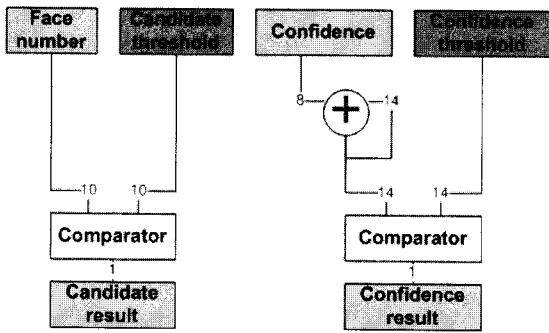


그림 13. 검증 모듈의 데이터패스
Fig. 13. Datapath of verification module.

IV. 결과 및 성능 분석

본 논문에서 설계한 하드웨어를 얼굴 검출 및 인식 분야에서 테스트용으로 이용되는 BioID 데이터베이스^[9]를 이용하여 테스트하였다. 테스트는 검출된 얼굴 영역을 입력으로 하여, 얼굴 특징 검출과정을 수행한 후 모든 얼굴 특징 영역들이 찾아졌는지의 여부를 이용하여 판단하였다. BioID 영상에서 임의로 얼굴 영역을 추출한 후, ICT 변환을 이용하여 얼굴 특징들을 검출 한 결과 소프트웨어와 하드웨어 모두 100%의 검출 결과를 얻었다.

본 논문에서 사용한 검출 필터의 학습에는 정상적인 얼굴 특징을 이용하였다. 따라서 눈이 감겨있거나 색깔이 들어있는 안경을 쓴 경우, 마스크를 쓴 경우, 입을 크게 벌린 경우 등 얼굴 특징에 왜곡이 있는 영상을 입력으로 하였을 경우에는 검출이 불가능하였다.

검출 속도는 소프트웨어의 경우 펜티엄 2GHz의 프로세서로 BioID 데이터베이스를 이용하여 테스트한 결과 약 74.9 msec/frame의 성능을 보였고, ARM9 400MHz의 프로세서로 동일 환경에서 테스트한 결과 약 718.89 msec/frame의 성능을 보였다. 또, 하드웨어의 경우 FPGA를 이용하여 25MHz의 동작 클럭으로 동일

표 2. 성능 결과 비교

Table 2. Comparison of performance result.

프로세서	테스트 내용			
	펜티엄4	ARM9	FPGA	
클럭 속도	2GHz	400MHz	78MHz	25MHz
테스트 데이터베이스	BioID			
검출 속도 (msec/frame)	약 74.9	약 718.89	약 17.1	약 49.9

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops:	5,119	53,248	9%
Number of 4 input LUTs:	15,819	53,248	29%
Logic Distribution:			
Number of occupied Slices:	9,627	26,624	36%
Number of Slices containing only related logic:	9,627	9,627	100%
Number of Slices containing unrelated logic:	0	9,627	0%
Total Number 4 input LUTs:	15,911	53,248	29%
Number used as logic:	15,819		
Number used as a route-thru:	92		
Number of bonded IOBs:	87	640	13%
Number of BUFG/BUFGCTRLs:	1	32	3%
Number used as BUFGs:	1		
Number used as BUFGCTRLs:	0		
Number of FIFO16/RAMB16s:	47	160	29%
Number used as FIFO16s:	0		
Number used as RAMB16s:	47		
Number of DSP48s:	32	64	50%

그림 14. FPGA 자원 사용률

Fig. 14. FPGA resource utilization.

환경에서 테스트한 결과 약 49.9 msec/frame의 성능을 보였고, 78MHz의 동작 클럭으로 테스트한 결과 약 17.1 msec/frame의 성능을 보였고, 이를 표 2에 나타내었다.

임베디드 환경에서의 실시간 처리를 고려하여 ARM9에서 소프트웨어로 실행한 결과와 비교 시 약 14~42배의 수행 성능을 보이며, 이는 임베디드 환경에서 실시간 처리가 가능함을 나타낸다.

설계한 하드웨어 모듈을 FPGA에서 검증하기 위하여 Xilinx사의 ISE를 이용하여 합성 및 다운로드를 하였고, 검증에서 사용된 자원의 사용률을 그림 14에 나타내었다. 검증 플랫폼은 Xilinx사의 Virtex-4 vlx60-1148 FPGA^[10]와 ARM9이 내장된 Bluesky S3C2440A^[11] 프로세서를 이용하여 구성하였다. 검증 결과 소프트웨어와 하드웨어의 결과가 동일하게 나타났으며, 이를 실제 영상에 적용 시킨 결과 만족스러운 결과를 얻을 수 있

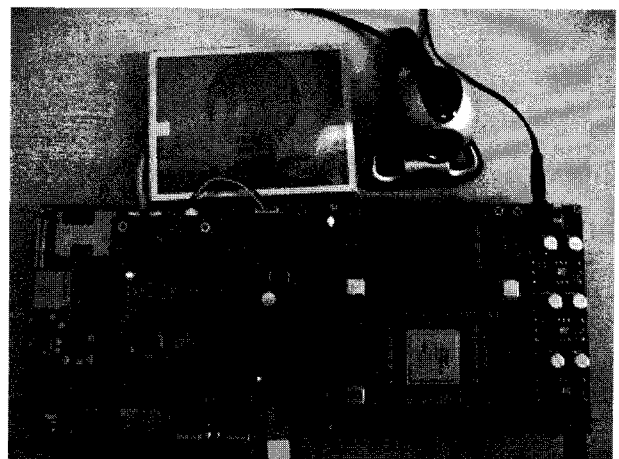


그림 15. 하드웨어 검증 플랫폼

Fig. 15. Hardware verification platform.

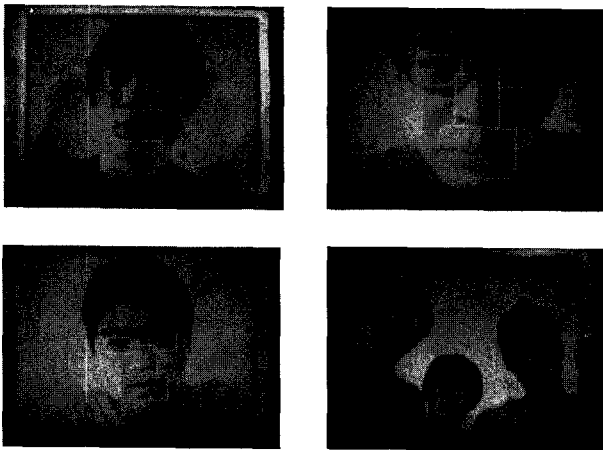


그림 16. 얼굴 특징 검출 결과 데모 영상

Fig. 16. Demonstration image of facial feature detection result.

었다. 검증 플랫폼 및 데모 영상을 그림 15와 16에 나타내었다. 또한, Synopsys사의 Design Compiler와 동부아남사의 0.18 μm 라이브러리를 사용하여 합성한 결과 조합 영역(combination area) 71,596 μm^2 , 비조합 영역(noncombination area) 247,628 μm^2 , 넷 상호연결 영역(net interconnect area) 57,597 μm^2 로 총 376,821 μm^2 가 나왔다.

현재 얼굴 특징 검출과 관련한 하드웨어의 설계는 비교 가능한 연구 논문이 없으므로 본 논문의 가치가 있다고 할 수 있다. 또한, 본 논문에서 사용한 알고리즘은 동일한 과정이 반복적으로 일어나므로 하드웨어 구현에 적합하고, 알고리즘의 핵심이자 메모리를 가장 많이 사용하는 부분인 영상의 변환 결과가 7-비트의 비트스트림 데이터로 이루어져 있어 효율적인 메모리 사용이 가능하다. 따라서 추후, 얼굴 인식 등의 다른 알고리즘과 같이 사용하여도 충분한 성능을 보일 것으로 예상된다.

VI. 결 론

본 논문에서는 얼굴 검출을 위해 개발된 ICT 변환 기반 알고리즘의 핵심 아이디어를 눈, 코, 입 등의 얼굴 특징을 검출하는데 적용하여 매우 우수한 결과를 얻을 수 있었다. 또한 이는 얼굴 특징 검출의 하드웨어 설계에 있어 얼굴 검출에 이용된 일부 모듈이 공유 가능한 형태로 효율적이고 메모리를 절약할 수 있다는 매우 커다란 이점이 있다.

멀티미디어 관련 하드웨어 설계에 요구되는 가장 중요한 요소는 수행 속도와 메모리 사용량이다. 따라서 이를 위하여 파이프라인 구조를 최대한 이용하여 설계

하였고, 모듈간의 데이터를 최대한 공유하도록 설계하였다. 또한 메모리 사용량을 줄이기 위하여 전체 영상을 대상으로 연산하지 않고, 영상을 여러 부분으로 나누어 수행되도록 설계하였다.

본 연구를 통해 제안된 얼굴 특징 검출 하드웨어는 얼굴 인식 시스템, 얼굴 추적 시스템 등 다양한 분야에 접목시켜 이용이 가능하다. 이와 관련하여 다른 얼굴 특징의 검출 혹은 얼굴 특징을 이용한 응용 시스템에 관한 연구로의 확대가 요구된다.

참 고 문 헌

- [1] Ramin Zabih and John Woodfill, "A non-parametric approach to visual correspondence," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996.
- [2] Bernhard Fröba, Andreas Ernst, "Face detection with the modified census transform," in Proc. of IEEE Conf. on Automatic Face and Gesture Recognition, pp. 91-96, May 2004.
- [3] 김운구, 정용진, "임베디드 시스템 적용을 위한 얼굴검출 하드웨어 설계," 대한전자공학회 논문지 제44권 SD편 제9호, Sep. 2007.
- [4] Haiyuan Wu, Qian Chen, "Facial feature extraction and face verification," in Proc. of IEEE Conf. on Pattern recognition, pp.484-488, 1996.
- [5] 양애경, 최형일, "얼굴 특징 추출에 의한 아바타 제어 시스템," 정보과학회논문지(B), 제25권, 제9호, 1998.
- [6] De Silva, L. C, "Real-time facial feature extraction and emotion recognition," in Proc. of IEEE Conf. on Multimedia, pp. 1310-1314, 2003.
- [7] Zhiwei Zhu, Qiang Ji, "Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time," in Proc. of IEEE Conf. on Pattern Recognition, pp. 1092-1095, 2006.
- [8] Hasan Demirel, "Color-space analysis for facial feature detection," IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications, 16-19, June 2002.
- [9] <http://www.facedetection.com/facedetection/datasets.htm>, BioID Face Detection Database.
- [10] http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex4/index.htm, Virtex-4 FPGA.
- [11] "Bluesky2410/2440 Reference Manual v1.0," eLOGICs, 2004.

저 자 소 개



김 정 호(학생회원)
 2006년 광운대학교 전자공학부
 학사 졸업.
 2006년 3월~현재 광운대학교
 전자통신공학과 석사과정
 <주관심분야 : SoC 설계, 얼굴 검
 출 및 인식, 임베디드 시스템 설
 계>



정 용 진(정회원)
 1983년 서울대학교 제어계측
 공학과 학사 졸업.
 1983년 3월~1989년 8월 한국전자
 통신연구원.
 1995년 미국 UMASS 전자전산
 공학과 박사 졸업.
 1995년 4월~1999년 2월 삼성전자 반도체 수석
 연구원.
 1999년 3월 광운대학교 전자통신공학과 부교수
 <주관심분야 : 무선통신, 정보보호, SoC 설계,
 영상처리 및 인식, 임베디드 시스템>