얼굴인식을 위한 실시간 하드웨어 설계

서기범* · 차선태**

A Realtime Hardware Design for Face Detection

Ki-bum Suh* · Sun-tae Cha**

이 논문은 2012년도 시스템반도체 설계인력양성 칩설계 공동연구 사업 연구비를 지원받았음

요 약

본 논문에서는 Adaboost알고리즘을 이용한 얼굴인식 하드웨어 시스템의 구조를 제안하였다. 제안된 하드에어 구조는 초당 30프레임을 가지며 실시간 처리가 가능하다. 또한 Adaboost알고리즘을 이용하여 얼굴 특징 데이터를 학습하였고, 영상 크기 축소부와 적분 영상 추출부 그리고 얼굴 비교부, 메모리 인터페이스부, 데이터 그룹화, 검출결과 표시부 등으로 구성되었다. 제안된 하드웨어 구조는 사이클당 1포인트를 계산 할 수 있는 구조로 속도의 향상을 가져오며 full HD(1920×1080)의 경우에는 총 사이클 수 2,316,087×30=69,482,610로 약 70MHz의 속도를 가진다. 제안된 하드웨어 구조는 Verilog HDL로 디자인되었고, Mentor Graphics Modelsim을 이용하여 검증하였으며, 합성은 FPGA Xilinx Virtex5 XC5VLX330을 이용하여 칩의 대략 35%인 74,757 Slice LUT와 45MHz의 주파수에서 동작한다.

ABSTRACT

This paper propose the hardware architecture of face detection hardware system using the AdaBoost algorithm. The proposed structure of face detection hardware system is possible to work in 30frame per second and in real time. And the AdaBoost algorithm is adopted to learn and generate the characteristics of the face data by Matlab, and finally detected the face using this data. This paper describes the face detection hardware structure composed of image scaler, integral image extraction, face comparing, memory interface, data grouper and detected result display. The proposed circuit is so designed to process one point in one cycle that the prosed design can process full HD(1920x1080) image at 70MHz, which is approximate 2316087 x 30 cycle. Furthermore, This paper use the reducing the word length by Overflow to reduce memory size. and the proposed structure for face detection has been designed using Verilog HDL and modified in Mentor Graphics Modelsim. The proposed structure has been work on 45MHz operating frequency and use 74,757 LUT in FPGA Xilinx Virtex-5 XC5LX330.

키워드

Adaboost 알고리듬, 실시간, 얼굴인식, 하드웨어설계, FPGA

Key word

Adaboost Algorithm, Real-Time, Face Detection, Hardware Design, FPGA

* 종신회원 : 우송대학교(kbsuh@wsu.ac.kr)
** 준회원 : 우송대학교

접수일자 : 2012. 12. 24
심사완료일자 : 2013. 01. 14

Open Access http://dx.doi.org/10.6109/jkiice.2013.17.2.397

Ⅰ. 서 론

최근 얼굴, 지문 및 홍채 등의 인체를 이용하여 개인의 인증, 정보 보호 및 신분 확인을 수행 할 수 있는 생체 인식 기술들에 대한 연구가 활발히 진행되었다. 특히 얼굴인식의 경우 얼굴의 변화에 따라 인식률이 낮다는 단점에도 얼굴인식에 있어 사용자에게 특별한 행위를 요구할 필요가 없으며, 비접촉 식으로 인식을 수행할 수 있어 편리하다는 장점을 가지므로 공공기관 및 기업 등 높은 보안을 요구하는 곳에서 사용되고 있다[1]. 본 논문에서는 Viola와 Jones가 Adaboost알고리즘을 이용하여 얼굴인식을 하였고, 얼굴 검출에 사용된 특징인 하알 유사특징과 속도향상을 위해 고안된 적분영상과 약한 분류기에서 강한 분류기로 부스팅하는 방법 그리고 단계 검사 방법을 사용하였다. 먼저 하알 유사 특징은 Viola와 Jones가 얼굴 검출에 적용한 것으로, 단순 합 이미지를 이용하여 특징 값을 표현하는 것이다[2]. 이는 위치, 모양 크기에 따라서 수많은 형태를 나타낼 수 있으므로 생성된 특징 값은 얼굴의 특징을 잘 나타낸다는 장점을 가지고 있다. 이런 특징은 계산 방식이 단순 합으로 이루어져 있기 때문에 연산 속도가 빠르므로, 동영상에서와 같은 실시간 처리가 가능하다. 본 논문의 구성에는 2장에서는 Adaboost학습 과정을 통한 특징벡터 추출에 대하여 설명하고, 3장에서는 Adaboost알고리즘을 구현하기 위한 전체적인 하드웨어의 세부적인 구조와 특징 및 메모리 현황에 대해 설명하고, 4장에서는 시뮬레이션 결과와 기능 검증 결과를 설명한다. 그리고 5장에서는 본 논문의 결론에 대해 설명한다.

Ⅱ. 본 론

본 논문에서는 Viola와 Jones의 Adaboost알고리즘을 이용하여 영상에서 얼굴 영역을 검출하여 비교하는 하알 유사 특징을 사용하였다. 그림 1은 Viola와 Jones의 Adaboost알고리즘 학습을 통해 얻은 특성 벡터와 본 논문에서 사용한 특성벡터를 나타낸다. 본 논문에서 사용한 특성벡터의 개수는 총 57,500개이고 그중 200개의 특성벡터만을 얼굴인식을 위해 사용하였다.

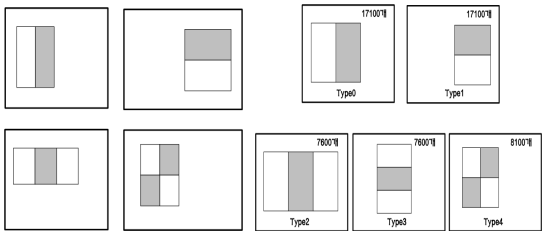

그림 1. 특성벡터의 집합
Fig. 1 The set of feature vector

Ⅲ. 제안된 얼굴인식 하드웨어 전체 모듈

3.1. 얼굴인식 하드웨어 전체 모듈

이 카메라로부터 입력되는 영상(480×272)의 YCbCr 신호에 대하여 VIM 모듈에서 흑백 신호인 Y신호만을 검출하여 얼굴인식을 위한 모듈에 데이터를 입력하기 위해 첫 번째 메모리(org_sram)에 저장한다. 저장된 데이터는 영상 축소부와 적분영상 추출부 과정을 거치고 얼굴영역 비교부에서 얼굴영역 위치 값을 가진 후 마지막으로 그룹화부 과정 후에 데이터를 Vertex5 LX330보드의 화면에 출력하는 과정을 그림 2에서 보여준다.

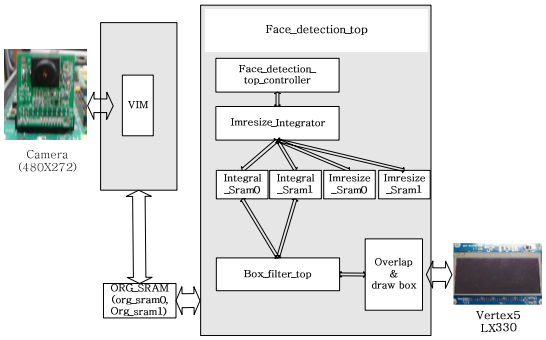

그림 2. 얼굴인식 하드웨어 전체 블록도
Fig. 2 The overall diagram of face detection hardware

또한 본 논문에서는 기본 영상의 크기를 480×272로 정하였고 영상 축소를 이전 영상×3/4라고 하였기 때문

에 얼굴영역을 검출 하는데 총 9번을 동작하게 되는데, 전반적인 데이터 흐름은 첫 번째 시작 신호에 영상 축소 과정과 적분영상 추출 과정을 동시에 실시하고, 두 번째 시작 신호에 얼굴영역 비교부와 영상축소, 적분영상 추출 과정을 반복 수행하게 된다. 본 논문은 파이프 라인 구조를 형성하였기 때문에 데이터 처리과정에서 시간적 낭비를 최소화 하여 효율적인 코딩이 되었다. 그림 3에서는 얼굴인식을 위한 데이터의 흐름도를 나타낸다.

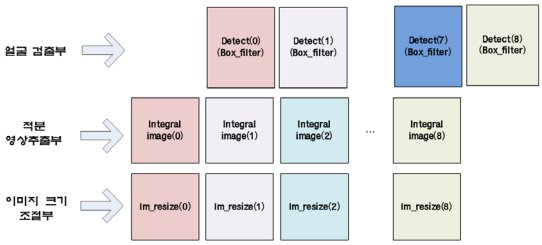

그림 3. 얼굴 인식 파이프라인 흐름도
Fig. 3 Pipeline flow of face detection

3.2. 적분영상 추출(Integral image)

적분영상 처리에서 평균 필터를 사용하는 경우에 각각의 픽셀에서 일정 크기의 윈도우를 설정하고, 윈도우 내의 픽셀들의 평균값을 취하여 새로운 픽셀 값을 설정하게 된다. 이처럼 많은 영상 처리의 필터들이 윈도우 연산을 하고, 윈도우가 커지면 시간 복잡도도 그에 따라 커지게 되는데 이러한 시간 복잡성을 줄이는 방법 중의 하나가 적분영상 처리 방식이다. 적분영상 처리 방식을 이용하면 임의의 크기의 사각형에 대한 픽셀 값의 합을 크기에 관계없이 일정 시간 내에 계산할 수 있게 한다. 또한 영상에서 특정한 객체를 검출하기 위한 과정은 반복적인 하알 유사 특징 값 계산을 요구하기 때문에 원영상의 데이터 값을 적분영상으로 변환한 후에 이용하면 그 특징 값을 빠르게 얻을 수 있다는 장점이 있다.

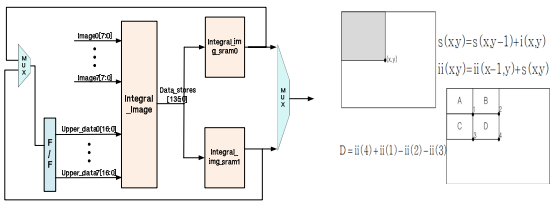

그림 4. 적분 영상 연산과 데이터 흐름도
Fig. 4 Integral image calculator and data flow

그림 4에서는 본 논문에서 설계한 적분영상 처리의 데이터 흐름과 영역 D의 픽셀 값을 구하기 위한 연산에 대해 나타낸다.

3.3. 영상 축소(Image resize)

얼굴인식 과정에 있어서 보다 정확한 얼굴인식을 위하여 본 논문에서는 영상 축소 모듈을 설계하였다. 본 논문에서 적용한 영상 축소 과정은 그림 4와 같이 입력된 데이터에 대해 4픽셀씩 나누어 첫 번째 픽셀 값에 3/4 두 번째 픽셀 값에 1/4를 곱한후 얻어진 결과 값을 다시 더한다. 다음으로는 두 번째 픽셀과 세 번째 픽셀에 대해 각각 1/2를 곱하고 더한다. 마지막으로 세 번째 픽셀 값에 1/4, 네 번째 픽셀 값에 3/4를 곱 한 후 얻은 결과 값을 더하며 영상의 픽셀 값 마다 축소 과정을 실시하여 기본 4×4의 이미지를 3×3으로 축소하였다.

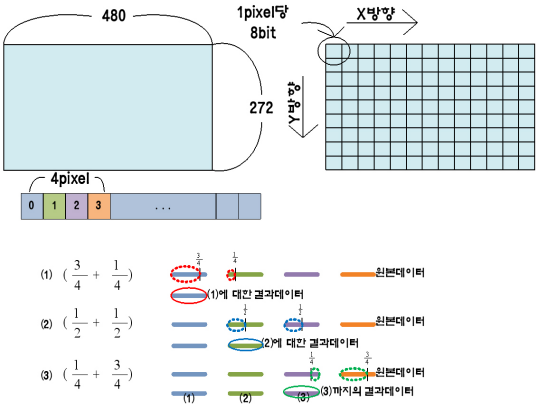

그림 5. 영상 축소 연산 과정
Fig. 5 Image resizing operation

그림5와 같이 영상 축소 과정 후 원 영상 크기(480×272)의 데이터는 다음 영상 크기인 360×204의 값을 가진다. 이러한 과정을 통해 처음 원 영상 크기의 경우에는 정확한 다음 영상 사이즈를 가질 수 있지만, 두 번째 영상 크기인 360×204의 경우에는 여분의 값이 존재 하게 된다. 본 논문에서는 4픽셀 단위로 계산하기 때문에 하드웨어 시스템 에서 연산이 가능하도록 나머지 값은 영향을 미치지 않으므로, 사용하지 않고 영상의 크기를 표 1의 실제 입력 축소 크기 부분과 같이 정수 값으로만 표현하였다.

표 1. 영상 축소 계산과 결과
Table. 1 Image reduction size and results

계산 축소 크기		실제 입력 축소 크기		읽어 들이는 횟수		저장 x 넓이	저장 y 넓이
화면 size_x	화면 size_y	읽어들일 크기		X방향	y 방향		
		x	y	Floor (x/8)	Floor (y/4)		
480	272						
360	204	480	272	60	68	360	204
270	153	360	204	45	51	264	152
202.5	114.75	264	152	33	38	192	112
151.875	86.0625	192	112	24	28	144	84
113.90625	64.54688	144	84	18	21	104	60
85.429688	48.41016	104	60	13	15	72	44
64.072266	36.30762	72	44	9	11	48	32
48.054199	27.23071	48	32	6	8	32	24
36.040649	20.42303	32	24	4	6	24	18
27.030487	15.31728	24	18	3	4	16	12

본 논문에서 제안한 영상 축소 과정을 통한 실제 하드웨어 출력 화면은 그림 6과 같이 나타낸다.

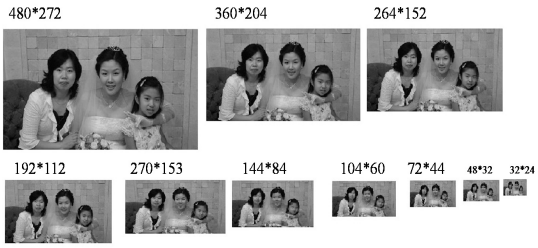

그림 6. 영상 축소 결과
Fig. 6 Image resizing results

3.4. 윈도우 기본 크기 저장

적분영상 처리 과정을 거친 데이터 값이 메모리에 저장된 값을 이용하여 본 논문에서 사용되는 기본 윈도우 크기 20×20만큼씩 추출해 내어 저장 후 얼굴영역 검출 모듈로 보내지는 역할을 수행한다. 이 모듈은 상위 모듈인 얼굴영역 검출 제어기에서 입력되는 1비트 시작 신호 발생시 20×20만큼씩 추출하는 동작을 그림 7과 같이 수행하게 된다. 수행과정은 시작 신호 발생 시, register19번에서 0번까지, 역으로 저장하게 되며, 각 register에 데이터가 전부 채워지고 나면, 얼굴영역 검출 모듈로 이동하여, 얼굴 인식 과정을 수행 하게 된다.

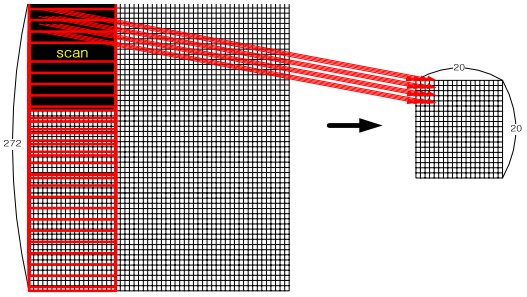

그림 7. 윈도우 사이즈 20x20의 이미지 추출
Fig. 7 Image extraction for window size 20x20

표 2는 480×272크기 영상에 대한 소요 시간을 계산한 결과를 보여준다. 실시간 처리 방식을 기준으로, 초당 30 프레임을 처리해야 하므로, 총 사이클 수는 132,606×30=3,978,180으로, 약 4MHz에서 동작 하게 된다.

표 2. 480×272 영상에 대한 사이클 계산
Table. 2 Cycle calculation for 480x272 image

단계	넓이	높이	x조사 점	y조사 점	첫 행처리 시 필요 사이클(A)	행변환 사이클 수(B)	총 사이클 수
계산방법			(넓이-20)/2	(높이-20)/2	20 × x조사 점	2 ×남은 점	A+B
1	480	272	230	126	4370	57500	61870
2	360	204	170	92	3230	30940	34170
3	264	152	122	66	2318	15860	18178
4	192	112	86	46	1634	7740	9374
5	144	84	62	32	1178	3844	5022
6	104	60	42	20	798	1596	2394
7	72	44	26	12	494	572	1066
8	48	32	14	6	266	140	406
9	32	24	6	2	114	12	126
사이클의 합/frame					14402	118204	132606

표 3. 1920×1080 영상에 대한 사이클 계산
Table. 3 Cycle calculation for 1920x1080 image

단계	넓이	높이	x조사 점	y조사 점	첫 행처리 시 필요 사이클(A)	행변환 사이클 수(B)	총 사이클 수
계산방법			(넓이-20)/2	(높이-20)/2	20 × x조사 점	2 × 남은 점	A+B
1	1920	1080	950	530	18050	1005100	1023150
2	1440	810	710	395	13490	559480	572970
3	1080	607	530	294	10070	310580	320650
4	810	455	395	218	7505	171430	178935
5	607	341	294	161	5586	94080	99666
6	455	255	218	118	4142	51012	55154
7	341	191	161	86	3059	27370	30429
8	255	143	118	62	2242	14396	16638
9	191	107	86	44	1634	7396	9030
10	143	80	62	30	1178	3596	4774
11	107	60	44	20	836	1672	2508
12	80	45	30	13	570	720	1290
13	60	33	20	7	380	240	620
14	45	24	13	2	247	26	273
사이클의 합/frame					68383	2247098	2316087

표 3에서는 표 2와 같은 방법으로 full HD에 대한 사이클 수는 2,316,087×30=69,482,610으로 대략 70MHz에서 동작하며 실시간 처리가 가능함을 나타낸다.

3.5. 얼굴 영역 검출

Adaboost 알고리즘을 이용한 얼굴 검출을 얼굴과 비얼굴 영상으로 구성된 학습용 영상 집합에서, 추출한 특징 집합을 이용하여 학습 된다. 학습 과정은 반복적인 계산으로 학습이 이루어지는데, 특징 선택과 관측 값에 해당하는 에러 율 산출 그리고, 가중치 수정의 단계로 이루어져 있다. 반복마다 분류 능력이 강해지는데, 이는 단계를 거치면서 약한 분류기가 결합해 강한 분류기로 만들어지기 때문이다. 잘못 분류된 학습 영상은 가중치를 증가 시키고, 바르게 분류된 학습 영상은 가중치를 감소시키는 과정을 반복하면서, 최소의 에러 율을 내는 하알-유사 특징만을 선택한다.

$$h_{st}(x)=\begin{Bmatrix}1 & \alpha_1 h_1(x)+\ldots+\alpha_n h_n(x)\geq\frac{1}{2}(\alpha_1+\ldots+\alpha_n)\\ 0 & otherwise\end{Bmatrix}$$

$$\alpha_t=\log(\frac{1}{\beta_t}),\ \beta_t=\frac{\epsilon_t}{1-\epsilon_t} \tag{1.1}$$

식 1.1은 강한 분류기로, n개의 특징을 선형적으로 조합하여 분류함을 의미한다. 단계적으로 반복하면서 분류기의 가중치를 수정하는데, 각 단계마다 특징 집합 중에서 얼굴을 검출하는데 결정적인 역할을 하는, 특징만을 남기고 나머지는 제거 하는 방식으로 학습이 진행된다.

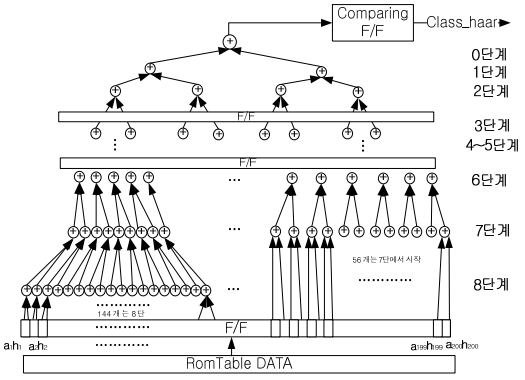

그림 8. 사이클을 줄이기 위한 데이터 처리 방법
Fig. 8 Data processing for reducing the cycle

그림 8은 얼굴 영역 검출 모듈에서 사용하는 사이클 수가 많아짐에 따라, 사이클의 수를 줄이기 위해 개발한 알고리즘이며, 이를 통해 얼굴 영역 검출 모듈 에서 사용되는 사이클 수를 줄 일수 있었고, 표 2와 같은 성능을 확인 하였다.

3.6. 데이터 길이 감소 방법

본 논문에서는 실제 하드웨어 동작 보드에 데이터를 입력하고, 테스트 하는 과정에서 메모리 과부하가 발생하였다. 메모리 과부하 발생 원인은 얼굴 영역 추출(Box filter)를 제공하는 많은 시스템들은 많은 크기의 메모리를 소비하기 때문이다. 해결방법은 모듈로 연산을 통한 계산방법을 사용하면 된다.[3]

$$\begin{aligned}S^{(x_1,y_1)}_{(x_0,y_0)}&=\sum_{x=x_0}^{x_1}\sum_{y=y_0}^{y_1} i[x,y]\\&=ii[x_0-1,y_0-1]+ii[x_1,y_1]\\&\quad -ii[x_0-1,y_1]-ii[x_1,y_0-1]\end{aligned} \tag{1.2}$$

식 1.2, 1.3, 1.4는 적분 영상 처리에서 필요로 하는 데이터의 길이(word length)값을 구하기 위한 조건식을 나타낸다.

$$(2^{L_{ii}}-1)\geq(2^{L_i}-1)w_{\max}h_{\max} \tag{1.3}$$

$$L_{ii}=[Log_2((2^{L_i}-1)w_{\max}h_{\max}+1] \tag{1.4}$$

위의 등식을 이용하여 예제를 통해 살펴보도록 할 것이다. 임의의 4비트 입력 영상을 그림 9와 같이 표현하였다.

적분영상 처리과정은 픽셀에 존재 하는 모든 값을 더해야 하므로 그림 9에서 첫 번째를 모두 합하면 5비트가 필요하고, 4비트 연산 처리 방식이므로 등식 1.3에 대입하게 되면 $L_{ii}=9$만큼 필요하게 되고 $L_i=4$일 때 등식 1.4에 대입을 해보면 L_{ii}는 7비트를 넘어 설수 없게 된다. 그림 9의 두 번째에서는 9비트에 대한 값을 표현하였고, 세 번째에서는 7비트에 대한 값을 표현하였다.

i : ($L_i = 4$)

12	10	13	12	5	11	3	10
10	12	3	4	3	10	5	8
5	1	1	9	5	12	12	14
8	2	5	14	15	7	10	12

ii_9 : ($L_{ii} = 9$)

12	22	35	47	52	63	66	76
22	44	60	76	84	105	113	131
27	50	67	92	105	138	158	190
35	60	82	121	149	189	219	263

ii_7 : ($L_{ii} = 7$)

12	22	35	47	52	63	66	76
22	44	60	76	84	105	113	3
27	50	67	92	105	10	30	62
35	60	82	121	21	61	91	7

그림 9. 7bit(128)으로 Modulo연산
Fig. 9 Module operation by 128(7bit)

그림 9의 데이터 값을 등식 1.2에 9비트의 임의의 값과 7비트의 임의의 값을 대입하면 1.5, 1.6과 같은 결과 값을 얻을 수 있다.

$$S_{(x_0,y_0)}^{(x_1,y_1)} = \sum_{x=x_0}^{x_1}\sum_{y=y_0}^{y_1} i[x,y] = \sum_{x=4}^{7}\sum_{y=2}^{3} i[x,y]$$

$$S = ii_9[3,1] + ii_9[7,3] - ii_9[7,1] - ii_9[3,3] \quad (1.5)$$

$$= 76 + 263 - 131 - 121 = 87$$

$$S = ii_7[3,1] + ii_7[7,3] - ii_7[7,1] - ii_7[3,3]$$

$$= 76(\mathrm{mod}128) + 7(\mathrm{mod}128) - 3(\mathrm{mod}128) - 121(\mathrm{mod}128)$$

$$= 87(\mathrm{mod}128) \quad (1.6)$$

식 1.6은 9비트에서의 데이터 값을 연산한 결과 이며, 등식 1.7은 7비트에서의 Modulo연산에 의한 결과를 나타내며 비록 등식 1.7의 $ii[7,3]$과 $ii[7,1]$에서 과부하 현상이 발생하였지만, 같은 결과 값을 가지고 있다. 이는 Modulo연산을 사용하게 되면 데이터 값의 길이를 줄일 수 있다는 결론을 활용하였다.

3.7. 겹침 현상 제거(Overlap)

겹침 현상 제거(Overlap)모듈은 같은 포지션 내부에 둘이상의 사각형이 위치할 때 하나의 사각형으로 처리함으로써, 간단한 표현을 할 수 있도록 설계 되었으며 그림 10과 같이 수행된다. 얼굴인식 제어기 모듈에서 출력되는 x위치, y위치 값을 가지고 내부 사각형의 위치를 찾게 된다. 그림 10에서 x위치와 y위치인 (1),(2),(3)과 같은 사각형 상자가 포함 되는지를 알기 위해 얼굴 인식 제어기에서 출력되는 x위치와 y의 좌표를 20개까지 저장하고, 얼굴 인식 제어기 모듈이 동작이 끝이라는 신호와 함께 내부 사각형을 찾는 구조로 설계되었다.

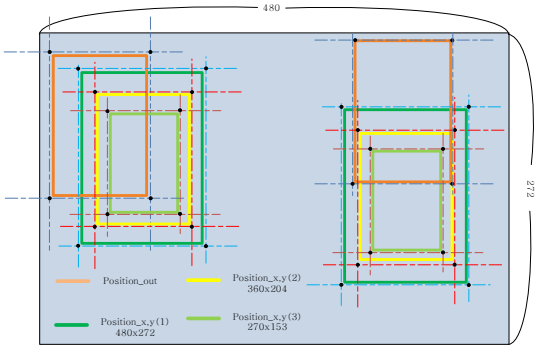

그림 10. 겹침 현상 제거
Fig. 10 Removing the overlap

겹침 현상 제거 모듈을 적용하여 얻어진 결과 값은 목표 보드의 화면에 출력하게 되며, 그림 11은 실제 하드웨어 출력 화면을 보여준다.

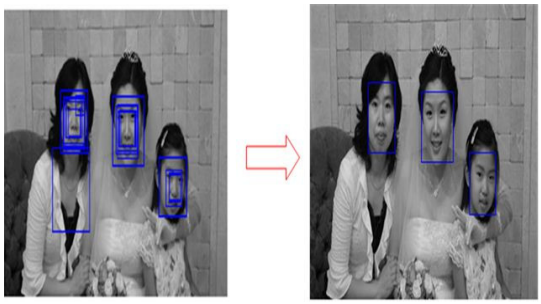

그림 11. 겹침 현상 제거 적용 사례
Fig. 11 Example of overlap removing

Ⅳ. 실험결과 및 분석

논문은 Modelsim을 이용하여 그림 12의 검증 순를 바탕으로 각 모듈별 출력 데이터 값인 적분 영상부 결과 값과 영상 축소부 결과 값 그리고 마지막 얼굴 영역 검출부 결과 값을 비교함으로써 신뢰성 있는 하드웨어임을 증명하였다.

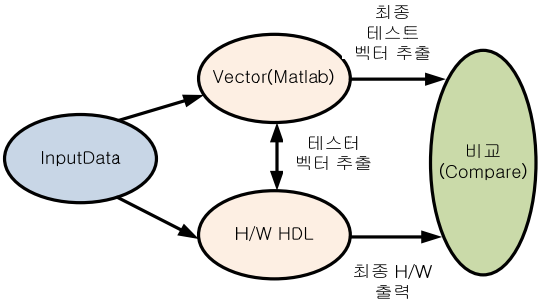

그림 12. 검증 순서도
Fig. 12 Verification Flow

영상 축소부의 수직 방향과 수평 방향의 결과 값 비교를 그림 13에서 나타낸다.

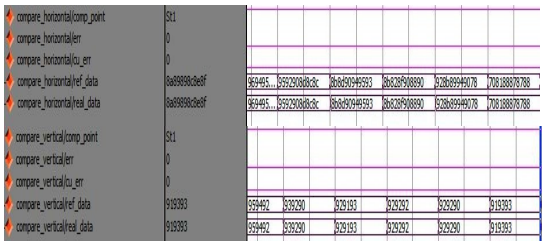

그림 13. 수직/수평 필터 방향 결과 비교
Fig. 13 Result comparison for verifical and horizontal filtering

적분 영상부의 결과 값 비교는 그림 14에서 나타낸다.

그림 14. 적분 영상부 결과 비교
Fig. 14 Result caparison for integral image

마지막으로 본 논문의 최종 출력 값이며, 얼굴 영역부의 출력 값 비교를 그림 15에서 나타낸다.

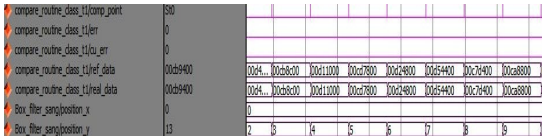

그림 15. 최종 출력 결과 비교
Fig. 15 Result comparison for final results

표 4. 검증결과 비교 분석
Table. 4 The comparison with others

	Logic elements	QVGA	VGA	1920x1080
한동일[4]	80,796	30 fps	NA	
Cho[5]	62,890	27 fps	7 fps	
Our System	74,757		112fps (480×272)	16 fps

표 4에서는 다른 시스템과 본 논문을 비교 분석한 결과를 나타내며, 본 논문은 480×272(VGA) 영상과 클럭 주파수 24MHz를 기준으로 사용하였다.

Ⅴ. 결 론

본 논문에서는 Viola와 Jones의 Adaboost알고리즘을 이용하여 영상에서 사람의 얼굴을 검출 하여 비교하는 DVR 시스템용 얼굴인식 하드웨어 시스템을 설계하였다. 얼굴 인식을 위하여 본 논문에서는 크게 적분 영상, 영상 축소, 얼굴영역 검출로 나누어 각 모듈의 기능에 설명하였고, Matlab을 이용하여 참조 벡터를 생성하여 설계된 하드웨어를 Modelsim을 이용하여 검증하였고 합성은 Xilix사의 Xilinx ISE Design Suit 12.4을 사용하여 FPGA검증과 Synopsys사의 Design Compiler를 통해 0.18um공정에서 Vertex5 LX330의 35%인 Slice LUT 74,757를 사용하였다.[6] 동작 주파수 45MHz의 속도로 동작이 가능하도록 설계되어 VGA급 영상에 대해 초당 112프레임 처리가 가능하며, full HD영상에 대해 초당 16프레임을 처리 가능하다.

감사의 글

본 연구는 2012년도 시스템반도체 인력양상 칩 설계 공동연구사업 지원에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

참고문헌

[1] 한학용, 패턴인식 개론, 한빛 미디어, 2009.

[2] P. Viola and M. Jones, "Robust Real-time Object Detection", International J. Computer Vision, pp. 137-154, 2004.

[3] H. J. W. Belt, "Storage Size Reduction for the Integral Image", Koninikljke Philips Electronics N.V, pp.7-9, Dec, 2007.

[4] 한동일, 조현종, 최종호, 조재일, "고성능 실시간 얼굴 검출 엔진의 설계 및 구현", 대한 전자 공학회 논문지, Vol. 제 47권SP 제 호, pp.37-39, Feb. 2010.

[5] J. Cho, S. Mirzaei, J. Oberg, and R. Kastner, "FPGA-Based Face Detection System Using Haar Classifiers", Dept, of Computer Sci. Eng. Univ. of CA, 2009.

[6] Xilinx Inc, "Virtex-5 Data sheets: Virtex-5 Family Overview", Sep.2008. DOI=http://www.xilinx.com

저자소개

서기범(Ki-bum Suh)

한양대학교 전자공학과 공학사
한양대학교 전자공학과 석사.
한양대학교 전자공학과 박사
한국전자통신연구원 선임연구원
우송대학교 철도전기정보통신학부 교수.
※관심분야 : 컴퓨터 비전, 영상신호통신, 집적회로 설계, 패턴 인식, 반도체

차선태(Sun-Tae Cha)

우송대학교 전자정보통신공학과 학사 졸업.
우송대학교 철도 전자정보통신 공학과 석사 과정.
※관심분야 : 컴퓨터 비전, 영상신호통신, 집적회로 설계, 패턴 인식, 반도체