

논문 2014-51-9-9

메모리 크기에 효율적인 적분영상 하드웨어 설계 연구

(A Study of Integral Image Hardware Design for Memory Size Efficiency)

이 수 현*, 정 용 진**

(Su-hyun Lee[Ⓒ] and Yong-jin Jeong)

요 약

적분영상은 입력영상의 픽셀 값을 기준좌표부터 순차적으로 누적하여 만든 영상으로, Haar-like features와 같은 네모난 박스 모양의 필터 연산을 효율적으로 처리하기 위하여 사용된다. 그러나 적분영상은 입력영상보다 3배 이상 많은 메모리를 소모하기 때문에, 메모리 자원이 제한적인 하드웨어 설계 환경에서는 사용이 어렵다. 본 논문에서는 효율적인 메모리 사용을 위한 적분영상 하드웨어 설계 방법을 제안한다. 해당 방법은 적분영상 이외에 세로적분영상과 가로적분영상을 생성하고, 입력영상을 재사용 하는 방법을 사용한다. 그리고 박스 필터의 크기에 따라 modulo 연산을 적용하여 적분영상의 데이터 크기를 줄이는 방법을 함께 적용하였다. 적분 영상 데이터를 읽기 위해 나누어진 영상 데이터를 다시 덧셈해야하는 연산 오버헤드가 발생하지만, 4개의 데이터를 단순히 더하는 연산이므로 병렬처리가 가능한 하드웨어 환경에서는 큰 영향을 미치지 않는다. Xilinx사의 Virtex5-LX330T를 대상으로 실험한 결과 640×480 크기의 8bit gray-scale 입력영상에서 최대 32×32 크기의 필터사용을 기준으로 50%의 적분영상 메모리를 감소시킬 수 있다.

Abstract

The integral image is the sum of input image pixel values. It is mainly used to speed up processing of a box filter operation, such as Haar-like features. However, large memory for integral image data can be an obstacle on an embedded hardware environment with limited memory resources. Therefore, an efficient method to store the integral image is necessary. In this paper, we propose a memory size reduction hardware design for integral image. The hardware design is used two methods. It is the new integral image memory and modulo calculation for reducing integral image data. The new integral image memory has additional calculation overhead, but it is not obstacle in hardware environment that parallel processing is possible. In the Xilinx Virtex5-LX330T targeted experimental result, integral image memory can be reduced by 50% on a 640×480 8-bit gray-scale input image.

Keywords : Integral Image, Image Processing, Image Recognition, Embedded Hardware.

* 학생회원, ** 정회원, 광운대학교 전자통신공학과

(Department of Electronic Communication Engineering , Kwangwoon University)

Ⓒ Corresponding Author(E-mail:metal477@kw.ac.kr)

※ This research was supported by the Industrial Core Technology Development Program (10049192, Development of a smart automotive ADAS SW-Soc for a self-driving car) funded by the Ministry of Trade, Industry & Energy and the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the Human Resource Development Project for SoC support program (NIPA-2014-H0601-14-1001) supervised by the NIPA(National IT Industry Promotion Agency)

접수일자: 2014년4월22일, 수정일자: 2014년8월06일, 수정완료: 2014년9월02일

I. 서론

얼굴 검출 및 인식, 영상 특징점 추출 및 매칭 등 영상 데이터를 이용한 인식기술이 다양한 방향으로 연구되고 있다. 이러한 영상 인식 기술은 PC기반에서 점점 임베디드 개발 환경으로 이식되어 다른 기술들과 융합을 하고자 하는 노력을 볼 수 있다. 영상처리 및 인식 기술은 대용량의 데이터를 반복적으로 연산하는 특성상 임베디드 소프트웨어 환경에서 실시간 처리에 어려움이 있어 이를 해결하기 위하여 하드웨어 설계를 이용한다. 임베디드 하드웨어 환경은 메모리 자원의 사용에 있어 제한적인 측면이 있어 영상처리를 위한 하드웨어 설계시에 효율적인 메모리의 사용이 요구된다. 본 논문은 영상인식 기술에서 자주 사용되는 적분영상(Integral Image)을 임베디드 하드웨어 환경에서 쉽게 구현하고자 적분영상에 사용되는 메모리를 효율적으로 설계하는 방법에 대하여 제안한다.

적분영상의 아이디어는 “Summed-area tables for texture mapping”^[1]에서 처음 제안 되었다. 적분영상은 입력영상의 픽셀 값을 누적한 영상으로 Haar feature와 같은 사각형의 필터 연산을 빠르게 하기위해 사용된다. 영상인식 분야의 대표적인 알고리즘인 SURF(Speeded Up Robust Features)^[2] 알고리즘^[2]과 Haar-like feature를 사용한 얼굴검출^[3] 등에서 적분영상은 사용된다.

적분영상은 사각영역 내의 픽셀 합을 빠르게 계산할 수 있지만 입력영상의 몇 배에 달하는 메모리를 필요로 하는 단점이 있다. 특히 하드웨어 자원의 제약이 많은 임베디드 환경에서 대용량 메모리를 요구하는 적분영상의 특징은 큰 단점으로 작용하게 된다. [4]와 같은 하드웨어 설계에서도 대용량의 적분영상 메모리 때문에 큰 사이즈의 입력영상을 사용하지 못한 문제가 있다. 본 논문에서는 이러한 단점을 보완하기 위하여, 적분영상 처리에 사용되는 메모리의 사용량을 줄이기 위한 설계 방법을 사용한다. 적분영상의 저장에 필요한 데이터를 줄이기 위한 방법으로 [5]와 [6]의 방법을 함께 사용하였으며, 그 결과를 비교 분석하였다.

II. 본론

1. 적분영상

적분영상은 그림 1과 같이 영상 내에서 기준 좌표

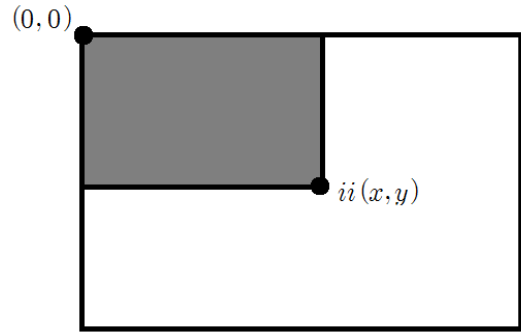


그림 1. 적분 영상
Fig. 1. Integral Image.

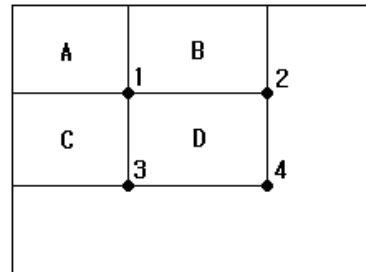


그림 2. 적분영상을 이용한 연산의 예
Fig. 2. Example of Integral Image usage.

(0, 0)와 지정 좌표(x,y)를 두 꼭짓점으로 하는 사각형 영역 내에 존재하는 모든 픽셀의 합으로 변환한 것이다. 적분영상은 식 (1)과 같이 표현되며, 식(1)에서 $i(x,y)$ 는 좌표 (x,y)상의 입력영상 픽셀 값이며, $ii(x,y)$ 는 좌표 (x,y)상의 적분영상 픽셀 값이다.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

적분영상을 이용하면 그림 2와 같이 특정 영역의 픽셀의 합을 식 (2)와 같이 간단히 연산 할 수 있다. 식(2)에서 *Space D*는 그림 2의 D공간의 모든 픽셀 값의 합이다.

$$Space D(Sum of Pixel) = ii(4) + ii(1) - ii(2) - ii(3) \quad (2)$$

2. 적분영상 메모리의 크기

적분영상 데이터의 크기는 입력영상의 모든 값이 최대치를 가지는 최악의 상황(worst case)을 고려하여야 한다. 입력영상의 가로 크기를 W , 세로 크기를 H , 픽셀 값의 word length를 L_i 라고 할 때, 적분영상의 최대 값 ii_{max} 은 수식 (3)과 같이 표현된다.

$$ii_{max} = (2^{L_i} - 1)WH \quad (3)$$

적분영상 픽셀값의 word length L_{ii} 는 수식 (4)를 만족해야 한다.

$$(2^{L_{ii}} - 1) \geq (2^{L_i} - 1)WH \quad (4)$$

적분영상을 저장하기 위한 메모리 크기는 수식 (5)와 같다.

$$integral\ image\ memory\ size = WHL_{ii}bits \quad (5)$$

640×480의 8bit 입력영상의 경우, 적분영상은 27bit의 word length를 가지고 640×480×27bit(8,294,400bits)의 메모리 용량을 필요로 한다.

3. 적분영상 메모리 감소 방법

가. 새로운 적분영상 저장 방법

해당 방법은 [5]에서 제안된 것으로 적분영상을 좌표 별로 4분할하여 저장한다. 적분영상의 수식은 식 (6)과 같이 변환이 가능하다.

$$\begin{aligned} ii(x, y) &= \sum_{x' \leq x, y' \leq y} i(x', y') \\ &= \sum_{x' \leq x-1, y' \leq y} i(x', y') + \sum_{y' \leq y} i(x, y') \\ &= \sum_{x' \leq x, y' \leq y-1} i(x', y') + \sum_{x' \leq x} i(x', y) \\ &= \sum_{x' \leq x-1, y' \leq y-1} i(x', y') + \sum_{y' \leq y-1} i(x, y') \\ &\quad + \sum_{x' \leq x-1} i(x', y) + i(x, y) \end{aligned} \quad (6)$$

세로적분영상(column integral image)과 가로적분영상(row integral image)의 수식 (7)과 (8)에 표현하였다.

$$ii_{column}(x, y) = \sum_{y' \leq y} i(x, y') \quad (7)$$

$$ii_{row}(x, y) = \sum_{x' \leq x} i(x', y) \quad (8)$$

수식 (7)과 (8)을 이용하여 수식 (6)을 수식 (9)와 같이 변환 할 수 있다.

$$\begin{aligned} ii(x, y) &= \sum_{x' \leq x, y' \leq y} i(x', y') \\ &= ii(x-1, y) + ii_{column}(x, y) \\ &= ii(x, y-1) + ii_{row}(x, y) \\ &= ii(x-1, y-1) + ii_{column}(x, y-1) \\ &\quad + ii_{row}(x-1, y) + i(x, y) \end{aligned} \quad (9)$$

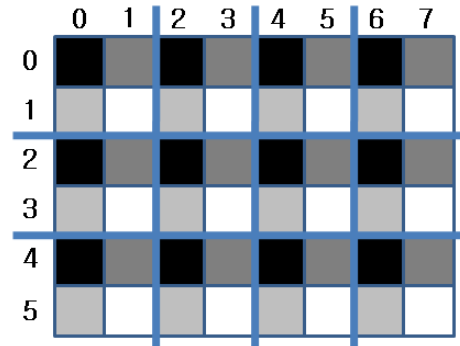
수식 (9)를 이용하면 적분영상을 4등분하여 적분영상, 세로적분영상, 가로적분영상, 입력영상으로 나누어 저장할 수 있다. 그림 3에 분할된 적분영상의 재구성 방법을 표현하였다.

해당 방법을 이용할 경우 영상크기 1/4의 입력영상 데이터를 재사용 하므로, 영상크기 1/4의 적분영상, 세로적분영상, 가로적분영상의 저장 공간만 필요하게 된다. 그리고 세로적분영상과 가로적분영상의 word length 와 L_{cii} 는 L_{rii} 수식 (10)과 (11)을 만족 하여야 한다.

$$(2^{L_{cii}} - 1) \geq (2^{L_i} - 1)H \quad (10)$$

$$(2^{L_{rii}} - 1) \geq (2^{L_i} - 1)W \quad (11)$$

따라서 새로 구성된 적분영상의 메모리 크기는 수식



$$\begin{aligned} ii(0,0) &= \text{Integral Image} \\ ii(0,1) &= \text{Integral Image} + \text{Column Integral Image} \\ ii(1,0) &= \text{Integral Image} + \text{Row Integral Image} \\ ii(1,1) &= \text{Integral Image} + \text{Column Integral Image} + \text{Row Integral Image} + \text{Original Image} \end{aligned}$$

그림 3. 적분영상의 재구성 방법
Fig. 3. Reconstruction of the integral image.

(12)와 같이 표현할 수 있다.

$$\begin{aligned}
 & \text{new integral image memory size} \\
 &= \frac{1}{4} WHL_{ii} + \frac{1}{4} WHL_{cii} + \frac{1}{4} WHL_{rii} \\
 &= \frac{1}{4} WH(L_{ii} + L_{cii} + L_{rii}) \text{ bits} \tag{12}
 \end{aligned}$$

640×480 크기의 8bit 입력영상에서 새로운 방법으로 적분영상을 저장하기 위해 320×240×27bit (2,073,600 bits)의 적분영상 메모리, 320×240×18bit (1,382,400 bits)의 가로적분영상 메모리, 320×240×17bit(1,305,600 bits)의 세로적분영상 메모리를 필요로 하므로, 전체 5,061,600bits의 메모리를 사용한다.

나. Modulo 연산을 이용한 방법

[6]에서는 적분영상이 박스필터의 연산에 사용되는 점을 이용하여, 박스필터 내부 픽셀 합의 최대치를 기준으로 Modulo 연산을 통하여 적분영상의 데이터에서 overflow되는 상위 비트를 줄이는 방법을 제안하였다.

박스 필터 연산에 있어 해당영역에 존재하는 픽셀의 정보는 하위 비트에 존재하며 식(2)와 같이 연산될 때 상위 비트는 overflow되어 소거된다. 따라서 박스필터의 최대크기를 연산하는데 있어 필요 없는 상위 비트는 제거 할 수 있다. 박스필터의 최대 가로 크기를 w_{max} , 최대 세로 크기를 h_{max} , 입력영상의 픽셀데이터의 word length를 L_i 라고 할 경우 박스필터 연산에 필요한 적분영상의 저장에 필요한 word length L_{ii} 는 식 (13)을 만족해야 한다.

$$(2^{L_{ii}} - 1) \geq (2^{L_i} - 1)w_{max}h_{max} \tag{13}$$

예를 들어 그림 4. (a)와 같은 4bit의 8×4영상 데이터를 이용하여 4×2 필터를 연산한다고 가정하였을 때, 일반적으로 그림 4. (b)와 같이 적분 영상을 저장할 경우 수식(2)를 이용하여 적분영상의 저장에 필요한 word length L_{ii} 는 9bit가 된다. 그리고 overflow되는 데이터를 제거하여 그림 4. (c)와 같이 저장할 경우 수식 (13)을 이용하여 L_{ii} 는 bit가 된다. 7bit가 된다.

그림 4에서 진하게 표시된 박스 영역내의 픽셀 합 S를 연산하였을 때, 각각의 연산은 수식 (14)와 같이 이루어지며, 값이 모두 같음을 확인할 수 있다.

5	10	11	2	1	9	4	8
6	6	12	3	0	4	11	1
7	7	13	5	8	3	12	2
8	9	6	7	14	13	15	3

(a) 입력영상
(a) Input image

5	15	26	28	29	38	42	50
11	27	50	55	56	69	84	93
18	41	77	87	96	112	139	150
26	58	100	117	140	169	211	225

(b) 적분영상
(b) Integral image

5	15	26	28	29	38	42	50
11	27	50	55	56	69	84	93
18	41	77	87	96	112	11	22
26	58	100	117	12	41	83	97

(c) modulo 연산이 적용된 적분영상
(c) Integral image with modulo calculation

그림 4. modulo 연산이 적용된 적분영상의 예제
Fig. 4. Example of integral image with modulo calculation.

$$\begin{aligned}
 (a) \quad & S = 5 + 8 + 3 + 12 + 7 \\
 & \quad \quad \quad + 14 + 13 + 15 = 77 \\
 (b) \quad & S = 50 + 211 - 84 - 100 = 77 \tag{14} \\
 (c) \quad & S = (50 + 83 - 84 - 100)(\text{mod}128) = 77
 \end{aligned}$$

해당 방법은 단순한 modulo 연산을 이용하여 필터 연산에 불필요한 상위 비트를 제거하는 방법으로 앞에서 설명한 새로운 구조의 적분영상과도 함께 적용이 가능하다.

4. 하드웨어 구조

수식(1)을 이용하여 일반적인 적분영상 하드웨어를 설계하면 그림 5와 같이 입력영상 메모리와 적분영상 메모리를 이용하는 단순한 구조를 가진다.

[5]에서 제안한 방법을 이용하여 하드웨어를 설계 하면 그림 6과 같이 적분영상 데이터 저장을 위하여 3개

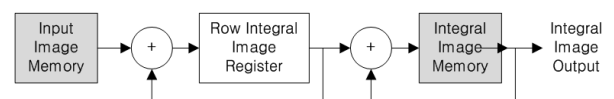


그림 5. 적분영상 하드웨어 구조
Fig. 5. Integral image hardware architecture.

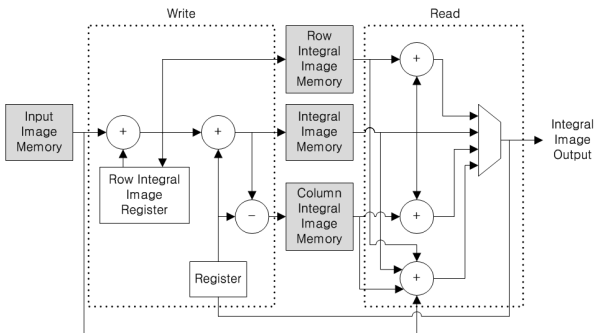


그림 6. 새로운 적분영상 하드웨어 구조
Fig. 6. New Integral image hardware architecture.

의 메모리를 나누어 사용하고, 입력영상을 적분영상 데이터 복원에 재사용하는 구조를 가진다.

그림 6의 구조는 그림 5보다 복잡한 구조를 가진다. 하지만 병렬연산과 파이프라인 설계를 이용하여 data throughput을 1pixel/clock으로 동일하게 처리할 수 있으므로 추가적인 연산시간에 대한 부담이 없다.

[6]의 방법은 2^n 으로 modulo 연산을 하므로 저장되는 적분영상의 데이터의 상위 비트를 제거하는 것으로 하드웨어에 적용이 가능하다. 따라서 추가적인 하드웨어 구조의 변경이 필요하지 않다. [5]와 [6]의 방법을 함께 사용할 시에도 마찬가지로 세로적분영상과 가로적분영상의 word length가 L_{ii} 보다 큰 경우 L_{ii} 의 크기에 맞추어 상위 비트를 제거하여 저장하면 된다. 즉, 그림 6의 하드웨어 구조에서 메모리의 word length 만 줄여주면 [5]와 [6]의 함께 적용이 가능하다.

III. 실험 및 분석

1. 메모리 감소율

[5]의 구조 적용시 메모리 감소율을 계산하기 위하여 식 (5)와 식 (12)를 사용하여, 식 (15)에 나타내었다.

$$\begin{aligned}
 & \text{reduced memory size} \\
 &= \text{integral image memory size} \\
 &\quad - \text{new integral image memory size} \\
 &= WHL_{ii} - \frac{1}{4} WH(L_{ii} + L_{cii} + L_{rii}) \quad (13) \\
 &= \frac{1}{4} WH(3L_{ii} - L_{cii} - L_{rii})
 \end{aligned}$$

그리고 메모리 감소율은 수식 (16)에 나타내었다

memory reduction rate

$$\begin{aligned}
 &= \frac{\frac{1}{4} WH(3L_{ii} - L_{cii} - L_{rii})}{WHL_{ii}} \times 100 \quad (14) \\
 &= 75 - 25 \frac{L_{cii} + L_{rii}}{L_{ii}} \%
 \end{aligned}$$

식(14)에 의해 메모리의 감소율은 가로적분영상과 세로적분영상의 word length에 영향을 받음을 알 수 있다. 가로적분영상과 세로적분영상의 word length는 각각 영상의 가로, 세로의 크기에 영향을 받게 된다. 즉 감소되는 메모리의 크기는 영상의 가로, 세로의 크기에 따라 메모리의 감소율은 달라질 수 있다. 메모리의 감소율을 주로 사용되는 표준영상의 크기에 적용하여 표 1에 나타내었다. 비교 결과 영상의 가로 세로 크기 및 비율에 따라 약간씩 차이가 있지만 기본적으로 40% 이상의 감소율을 보임을 알 수 있다.

[6]의 방법은 박스 필터의 최대 크기에 따라 감소되는 메모리의 양이 달라진다. 표 2에 다양한 영상의 크기와 박스 필터 크기에 대하여 비교하였다. 비교 결과 필터 크기가 클수록 감소율은 낮아지며, 영상의 크기가 클수록 감소율이 높아짐을 알 수 있다.

앞서 설명한 바와 같이 [5]와 [6]의 방법은 함께 적용이 가능하다. 두 가지 방법을 모두 사용한 경우에 대한 메모리 감소율을 표 3에 나타내었다.

[5]의 메모리 구조만 적용하는 것 보다 [6]의 방법을 함께 적용한 경우 평균적으로 5% 정도의 추가적인 감

표 1. 새로운 적분영상 메모리의 감소율
Table 1. Reduction rate of new integral image memory.

영상크기 (8bit word length)	영상비율 (가로:세로)	적분영상을 저장하기 위한 메모리 용량 (단위 : bytes)		감소율 (%)
		일반적인 방법	[5]	
320×240	4:3	240,000	139,200	42.0
640×480	4:3	1,036,800	595,800	42.6
800×600	4:3	1,620,000	945,000	41.7
1024×768	4:3	2,752,512	1,572,864	42.9
1600×1200	4:3	7,440,000	4,140,000	44.4
1280×1024	5:4	4,751,360	2,703,360	43.1
1280×720	16:9	3,255,600	1,872,000	42.0
2048×1080	16:9	8,294,400	4,700,160	43.3

표 2. Modulo 연산을 사용한 적분영상의 메모리 감소율

Table 2. Reduction rate of integral image memory using modulo calculation.

영상크기 (8bit word length)	박스 필터 최대 크기	적분영상을 저장하기 위한 메모리 용량 (단위 : bytes)		감소율 (%)
		일반적인 방법	[6]	
320×240	32×32	240,000	172,800	28.0
	64×64		192,000	20.0
	128×128		211,200	12.0
640×480	32×32	1,036,800	691,200	33.3
	64×64		768,000	25.9
	128×128		844,800	18.5
2048×1080	32×32	8,294,400	4,976,640	39.8
	64×64		5,529,600	33.1
	128×128		6,082,560	26.4

표 3. Modulo 연산을 적용한 새로운 적분영상 메모리의 감소율

Table 3. Reduction rate of new integral image memory using modulo calculation.

영상크기 (8bit word length)	박스 필터 최대 크기	적분영상을 저장하기 위한 메모리 용량 (단위 : bytes)		감소율 (%)
		일반적인 방법	[5]+[6]	
320×240	32×32	240,000	122,400	49.0
	64×64		127,200	47.0
	128×128		132,000	45.0
640×480	32×32	1,036,800	508,800	50.9
	64×64		528,000	49.0
	128×128		547,200	47.2
2048×1080	32×32	8,294,400	3,732,480	55.0
	64×64		4,008,960	51.7
	128×128		4,147,200	50.0

표 4. FPGA 환경에서 적분영상 메모리 사용량 비교

Table 4. Integral image memory usage comparison in FPGA environment.

영상크기 (8bit word length)	박스 필터 최대 크기	적분영상을 저장하기 위한 메모리 용량 (단위 : Kbits)				감소율 (%)		
		일반적인 구조	[5]	[6]	[5]+[6]	[5]	[6]	[5]+[6]
320×240	32×32	1,962	1,206	1,350	1,026	38.5	31.19	47.71
	64×64			1,531	1,080		21.96	44.95
	128×128			1,692	1,116		13.76	43.12
640×480	32×32	8,100	4,734	5,400	4,050	41.5	33.3	50.0
	64×64			6,084	4,230		24.9	47.8
	128×128			6,750	4,392		16.7	45.8

소율을 보인다. 그리고 2048×1080 크기의 영상에서 32×32 크기의 필터를 사용하는 경우 11.7%의 높은 추가 감소율을 보인다. 해당 경우는 modulo 연산에 의해 줄어든 word length L_{ii} 가 18bit로, 가로와 세로적분영상에 필요한 word length인 L_{cii} , L_{rii} 19bit보다 작기 때문에 L_{cii} , L_{rii} 를 L_{ii} 와 동일한 크기인 18bit로 사용가능하기 때문에 다른 조건보다 감소율이 높아진 것이다.

2. FPGA 합성

실제 하드웨어 설계에서 사용되는 메모리의 크기는 1bit 단위의 word length 적용이 어렵다. 앞서 분석한 이론적인 수치와 하드웨어 설계에 적용된 실제 감소율을 비교하기 위하여 FPGA를 타겟으로 해당 설계를 합성하여 그 결과를 비교하였다. 합성툴은 ISE 12.4를 이용하였으며, II. 4장에서 설명한 구조를 사용하였다. 실험은 Xilinx사의 Virtex5-LX330T를 타겟으로 이루어졌으며, 다양한 입력영상과 박스필터의 크기에 따라 합성을 하였다. 합성 결과에 따른 메모리의 사용량을 표 4에 정리 하였다. 적분영상 메모리의 크기 비교를 위하여 하드웨어 구조에서 입력영상 메모리는 제외하고 합성하였다.

실험 결과를 보면 전체적으로 메모리의 사용량은 이론적인 값거나 비슷하게 사용되었다. 메모리 감소율 또한 이론적 수치와 가깝지만 다소 평균적으로 2% 정도 낮게 나왔다. Virtex5 FPGA의 내부 블록 메모리는 18Kbits단위로 구성되어 있으며, 16k×1, 8k×2, 4k×4, 2k×9, 1k×18, 512×36, 256×72와 같은 사이즈로 생성되어 사용된다^[7-8]. 해당 bit width와 depth가 정확하게 맞아 떨어지지 않으면 보다 많은 용량을 소모하게 되는

것이다. 따라서 이론적인 용량 보다 다소 높은 메모리를 사용하게 되며, 감소율 또한 다소 낮게 나오는 것이다. 하지만, 결과적으로 [5]와 [6]의 방법을 이용한 하드웨어 설계는 이론적인 수치와 비슷한 40% 이상의 효율적인 메모리 감소율을 보였다.

IV. 결 론

본 논문에서는 효율적인 메모리 사용을 위한 적분영상 하드웨어 설계 방법을 제안하였다. 적분영상을 저장하기 위한 메모리의 용량을 줄이기 위해 적분영상을 분할하여 저장하는 새로운 방법^[5]과 박스 필터 연산에서 필요 없는 상위 비트를 제거하여 modulo 연산을 하는 방법^[6]을 함께 사용 하였다. FPGA 환경에서 실험한 결과 이론적인 수치와 차이는 있지만 일반적인 설계보다 효율적인 메모리의 사용량을 보였다.

적분영상은 영상처리 및 인식 분야에서 연산의 효율성을 위하여 주로 사용되며, 영상처리 및 인식 알고리즘들은 각종 융합기술과 함께 임베디드 시스템에서 활발하게 연구가 진행되고 있다. 하드웨어 자원이 제한적인 임베디드 환경에서 대용량 데이터를 저장하는 적분영상의 특징은 단점으로 작용한다. 본 논문에서 제안한 설계를 이용할 경우 FPGA 환경에서 40% 이상의 메모리 감소율을 보였다. 해당 연구 결과로 인하여 적분영상을 이용하는 영상처리 및 인식 알고리즘을 임베디드 하드웨어 환경에서 구현하는데 큰 도움을 줄 것으로 예상된다.

REFERENCES

[1] Crow, F., "Summed-area tables for texture mapping," in SIGGRAPH'84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, New York, NY, USA, 1984, pp. 207-212, ACM.

[2] Bay, H., Tuytelaars, T. and Gool, Luc V., "Speeded Up RobustFeatures (SURF)," in Computer Vision and Image Understanding, Vol. 110, No. 3, pp. 346-359, June 2008.

[3] Viola, P. and Jones, M. "Rapid Object Detection using a BoostedCascade of Simple Features," in Proceedings of CVPR 2001, pp. 511-518.

[4] Su-hyun Lee, Yong-jin Jeong, "A design and

implementation of Face Detection hardware", Journal of the Institute of Electronics Engineers of Korea Vol. 51, NO. 9, pp.43-54, April 2007.

[5] Su-hyun Lee, Yong-jin Jeong., "A New Integral Image Structure for Memory Size Reduction," IEICE Transactions on Information and Systems, Volume E97-D No.4, pp. 998-1000, 2014.

[6] Belt, H.J.W., "Word length reduction for the integral image," ICIP 2008, 15th IEEE International Conference on Image Processing, San Diego, USA, pp. 805 -808, 2008.

[7] "Virtex-5 Family Overview", DS100(v5.0), ©XILINX, February 6, 2009.

[8] "LogiCORE IP Block Memory Generator v4.3", DS512, ©XILINX, September 21, 2010.

저 자 소 개



이 수 현 (학생회원)
2005년 광운대학교 전자공학부
학사 졸업.
2007년 광운대학교 전자통신
공학과 석사 졸업.
2007년 3월~현재 광운대학교
전자통신공학과
박사과정.

<주관심분야 : SoC 설계, 영상처리 및 인식, 임베디드 시스템 설계>



정 용 진 (정회원)
1983년 서울대학교 제어계측
공학과 학사 졸업.
1983년 3월~1989년 8월
한국전자통신연구원.
1995년 미국 UMASS 전자전산
공학과 박사 졸업.

1995년 4월~1999년 2월 삼성전자 반도체 수석 연구원.

1999년 3월~현재 광운대학교 전자통신공학과 교수.

<주관심분야 : 무선통신, 정보보호, SoC 설계, 영상처리 및 인식, 임베디드 시스템>