

논문 2009-46SD-3-11

# 임베디드 환경에서 SIFT 알고리즘의 실시간 처리를 위한 특징점 검출기의 하드웨어 구현

( A Hardware Design of Feature Detector for Realtime Processing of  
SIFT(Scale Invariant Feature Transform) Algorithm in Embedded  
Systems )

박 찬 일\*, 이 수 현\*, 정 용 진\*\*

( Chan-Il Park, Su-Hyun Lee, and Yong-Jin Jeong )

## 요 약

SIFT(Scale Invariant Feature Transform) 알고리즘은 영상 데이터로부터 객체의 꼭지점이나 모서리와 같이 색상 성분의 차가 심한 영역에서 특징점을 찾아 벡터성분을 추출하는 알고리즘으로, 현재 얼굴인식, 3차원 객체 인식, 파노라마, 3차원 영상 복원 작업의 핵심 알고리즘으로 연구 되고 있다. 본 논문에서는 SIFT 알고리즘을 임베디드 환경에서 실시간으로 처리하기 위해 가장 연산량이 많은 특징점 위치 결정 단계를 Verilog HDL 언어를 이용하여 FPGA로 구현하고 그 성능을 분석한다. 하드웨어는 100MHz 클럭에서 1,280×960영상기준 25ms, 640×480영상기준 5ms의 빠른 연산속도를 보인다. 그리고 Xilinx Virtex4 XC4VLS60 FPGA를 타겟으로 Synplify Pro 8.1i합성툴을 이용하여 합성시 약 45,792LUT(85%)의 결과를 나타낸다.

## Abstract

SIFT is an algorithm to extract vectors at pixels around keypoints, in which the pixel colors are very different from neighbors, such as vertices and edges of an object. The SIFT algorithm is being actively researched for various image processing applications including 3D image reconstructions and intelligent vision system for robots. In this paper, we implement a hardware to sift feature detection algorithm for real time processing in embedded systems. We estimate that the hardware implementation give a performance 25ms of 1,280×960 image and 5ms of 640×480 image at 100MHz. And the implemented hardware consumes 45,792 LUTs(85%) with Synplify 8.1i synthesis tool.

**Keywords :** SIFT, Keypoint, Feature Detect, Object Recognition

## I. 서 론

특징점은 객체 인식이나 파노라마와 같이 입력된 영상과 레퍼런스 영상끼리 매칭하기 위한 공통된 중요한 위치 또는 그 주변의 정보를 말한다. 이러한 특징점을

검출하기 위한 방법이 컴퓨터 비전 분야에서 여러 가지 방법<sup>[3-4]</sup>으로 활발히 연구중에 있다. SIFT 알고리즘은 그 중 객체의 모서리나 꼭지점과 같은 부분에서 생성되는 특징점에 대한 벡터성분을 추출하는 알고리즘으로 패턴인식 방법과 달리 비교 영상의 크기 변화 또는 회전으로 인한 변형에 대해 뛰어난 매칭성능을 갖는 알고리즘이다. 하지만, 대용량의 영상데이터를 반복적으로 연산하는 과정이 많아 실시간으로 처리하기가 어렵다. 따라서 SIFT 알고리즘을 객체인식 또는 추적에 활용하기 위해서는 하드웨어로의 구현이 필수적이다.

SIFT 알고리즘을 하드웨어로 구현한 예로는 [3]과

\* 학생회원, \*\* 정회원, 광운대학교 전자통신공학과  
(Department of Electronics and Communication  
Engineering, Kwangwoon University)

※ 본 논문은 IDEC/IT-SOC 사업단의 톨 지원과 서울시  
나노 혁신 클러스터(나노 IP-SOC)사업 및 광운대학교  
2008 교내학술 연구 지원으로 이루어졌습니다.

접수일자: 2008년12월9일, 수정완료일: 2009년3월3일

[4]가 있으며, 이들 하드웨어는 SIFT와 유사한 알고리즘을 이용하여 특징점 검출까지 하드웨어로 구현하였으며, 수행속도는 각각 512×512크기의 영상기준 46fps와 640×480크기의 영상기준 30fps를 보인다. 하지만 두 가지 모두 특징점의 위치를 결정하는 단계까지를 하드웨어로 구현한 것으로 SIFT 전체 실시간 처리를 위해서는 보다 빠른 속도로 설계 되어야 한다.

본 논문은 SIFT 알고리즘 전체의 실시간 처리를 위해 특징점 검출 기능까지 640×480영상 기준 5ms의 성능을 가질 수 있도록 하드웨어로 구현한다.

내용의 구성은 II장에서 SIFT 특징점 위치결정 알고리즘에 대하여 설명하고, III장에서는 특징점 검출기의 하드웨어 구조에 대해 설명하며 IV장에서 성능 분석을, 마지막으로 V장에서 결론을 맺는다.

## II. SIFT 특징점 검출 알고리즘 설명

본 논문에서는 알고리즘의 변환과 고정 소수점 모델링을 위한 성능분석 및 비트크기 결정을 위해 UCLA Vision Lab 에서 공개한 알고리즘<sup>[3]</sup>을 참고하였다. 그림 1은 SIFT 알고리즘의 흐름도를 나타내며, 각 단계별 동작은 다음과 같다.

- (1) 영상 확대/축소 단계(Resize Image)  
: 입력 영상(640×480)에 대해서 2배,1배, 1/2배, 1/4배 영상을 만들고 각 영상에 대해 (2)~(7)단계를 반복한다. 그리고 생성된 각 영상의 최대값과 최소값을 추출한다.
- (2) 히스토그램 평활화 단계 (Histogram Equalization)  
: 해당 영상 확대/축소 단계에서 추출된 최대값과 최소값을 이용하여 히스토그램 평활화 작업을 수행한다.

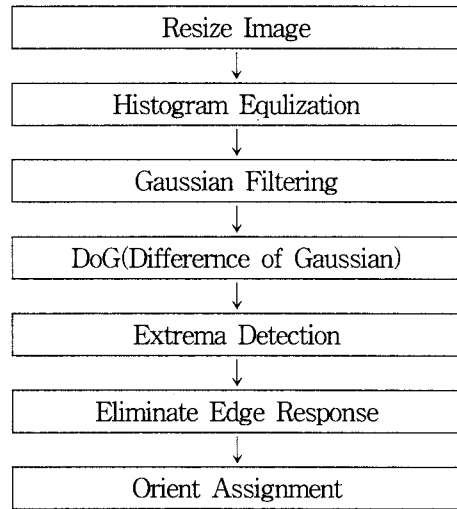


그림 1. SIFT 특징점 검출 알고리즘의 흐름도  
Fig. 1. SIFT Feature Detection Algorithm Flow.

- (3) 가우시안 필터 영상 생성 단계 (Gaussian Filtering)  
: 히스토그램 평활화 영상을 이용하여 총 10단계의 가우시안 영상을 생성한다.
- (4) DoG 영상 생성단계 (Difference of Gaussian)  
: 두 개의 연속된 가우시안 필터영상의 차를 구한다.
- (5) 극값 추출 단계(Extrema Detection)  
: 3개의 연속된 DoG 영상에서 3×3×3마스크를 이용하여 최대 또는 최소가 되는 극값을 가지는 지점을 추출한다.
- (6) 엣지 성분 제거 단계 (Eliminate Edge Response)  
: 특징점 후보좌표의 해당 가우시안 영상에서 주변값을 이용하여 에지성분인지 아닌지 검사한다.
- (7) 방향 설정 단계 (Orient Assignment)  
: 에지성분이 제거된 가우시안 영상에서 특징점의 좌표 주변값을 이용하여 방향성분을 추출한다.

표 1. SIFT 특징점 검출 알고리즘의 평균 연산시간  
Table 1. Operation Time of SIFT Feature Detection Algorithm.

구동 환경 및 모델 단계	펜티엄 4 2.13GHz		ARM9 400MHz
	부동 소수점 모델	고정 소수점 모델	고정 소수점 모델
영상입력 (with resize)	3 ms	2 ms	942 ms
히스토그램 평활화	60 ms	48 ms	2,053 ms
가우시안 필터링	2,724 ms	1,203 ms	103,390 ms
DoG	1,599 ms	219 ms	5,262 ms
특징점 위치결정	75 ms	219 ms	18,732 ms
방향설정	245 ms	406 ms	24,601 ms
전체 구동시간	4,709 ms	2,097 ms	152,941 ms

앞에서 설명한 SIFT 알고리즘을 임의로 정한 100장의 레퍼런스 영상(640×480크기)을 이용하여 PC환경(펜티엄4 E4400 2.13GHz)에서 구동한 결과 표 1과 같은 각 단계별 평균 수행시간이 나타나며, 부동 소수점 모델의 전체 구동시간은 약 4.7초, 고정 소수점 모델은 약 2초로 측정되었다. 그리고 임베디드 환경(ARM9 400MHz)에서의 고정 소수점 모델의 수행시간은 약 153초로 펜티엄 PC에서 구동시켰을 때보다 약 76배의 연산시간이 더 소요되었다. 따라서 임베디드 환경에서 실시간 구동을 위해서는 하드웨어 구현이 필수적이다.

본 논문에서는 방향설정 단계까지 고정 소수점으로 변환된 모델<sup>[5]</sup>을 하드웨어로 구현하고 그 성능을 분석한다. 표 1은 본 논문에서 다루고 있는 SIFT 특징점 검출 알고리즘의 평균 연산시간을 나타낸다.

### III. 하드웨어 설계

하드웨어 모듈은 그림 1에서 설명한 알고리즘 흐름의 구분과 동일하게 Resize, Histogram Equalization, Gaussian Filtering, DoG, Extrema Detection, Eliminate Edge Response, Orient Assignment의 총 7개의 모듈을 중심으로 이루어진다. 그리고 각 모듈의 SDRAM의 접근을 스케줄링 하기 위한 2개의 SDRAM 스케줄러와 SDRAM 제어를 위한 SDRAM 컨트롤러가 구성된다. 그 중 Resize Image 모듈은 외부로부터 전송받은 영상 데이터를 이용하여 리사이즈 영상(1,280×960, 640×480, 320×240, 160×120)을 생성하는 동시에 프레임 버퍼(SDRAM)에 Resize SDRAM Scheduler를 이용하여 저장하며, Feature Detection 모듈에서는 Feature Detection SDRAM Scheduler를 이용하여 프레임 버퍼에서 영상 데이터를 읽는 동시에 히스토그램 평활화 모듈을 거쳐 생성된 히스토그램 평활화 영상 데이터를 이용하여 가우시안 필터영상을 만들고, DoG 모듈을 통해 DoG 영상 데이터를 생성한다. 그리고 DoG 영상을 3줄씩 파이프라인 구조로 저장을 하면서 3×3영역에서의 극값을 찾고, 에지제거 모듈을 거쳐 최종 결정된 특징점 좌표와 그에 해당하는 가우시안 영상의 주변 픽셀을 참조하여 방향정보를 최종 결과 버퍼 메모리에 저장하는 방식으로 구성된다.

#### 1. Resize Image

Resize Image 단계<sup>[6]</sup>에서는 외부로부터 전송되는 8비

트 영상 데이터를 이용하여 총 4개 크기의 리사이즈 영상데이터(1,280×960 : 9bit, 640×480 : 8bit, 320×240 : 10bit, 160×120 : 12bit)를 생성하는 동시에, FPGA에 연결된 SDRAM에 리사이즈 영상 데이터를 입력한다. 이때, SDRAM에 전송하기 위한 Resize SDRAM Scheduler는 32bit의 Burst8모드로 동작하며, 영상 확대 모듈과 영상 축소 모듈로 구성된다.

#### 가. 영상확대

영상 확대 알고리즘 종류에는 여러 가지 방법이 있지만, 본 논문에서는 크기 변화에 대해서 영상의 훼손이 가장 낮은 양선형 보간법<sup>[7]</sup>을 이용하였다. 640×480크기의 입력 영상을 1,280×960크기의 2배 영상으로 확대하기 위해서 그림 1과 같이 주변픽셀을 이용하여 새로 생성된 픽셀을 구현하였으며, 3×3 영역에서 새로 생성된 5개의 픽셀은 8비트 입력 CLA(Carry Look-ahead Adder) 5개를 이용하여 최적화 시켰다. 그림 1은 영상 확대 알고리즘의 하드웨어 연산 구조를 나타낸다.

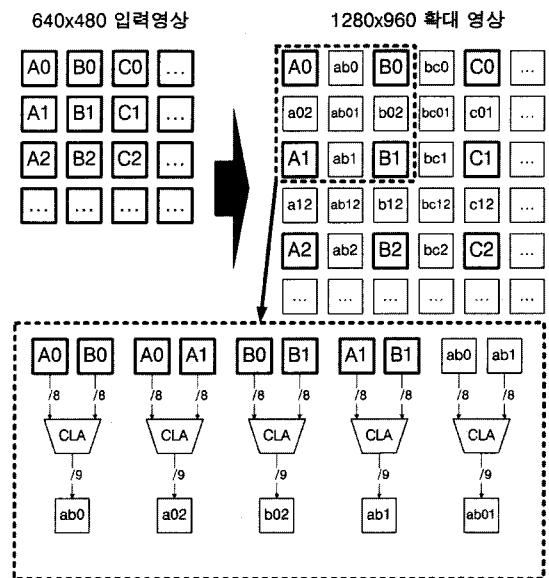


그림 2. 영상 확대 알고리즘의 하드웨어 연산 구조<sup>[6]</sup>  
Fig. 2. Hardware Architecture of Image Extension.

#### 나. 영상축소

영상 축소 알고리즘은 주변픽셀의 평균값을 이용하여 영상의 크기를 변환하는 평균 보간법<sup>[7]</sup>을 이용하여 하드웨어로 구현한다.

1/2영상을 생성하기 위해서 4개의 픽셀을 더하고 1/4 영상을 생성하기 위해 1/2영상에서 다시 4개의 픽셀을

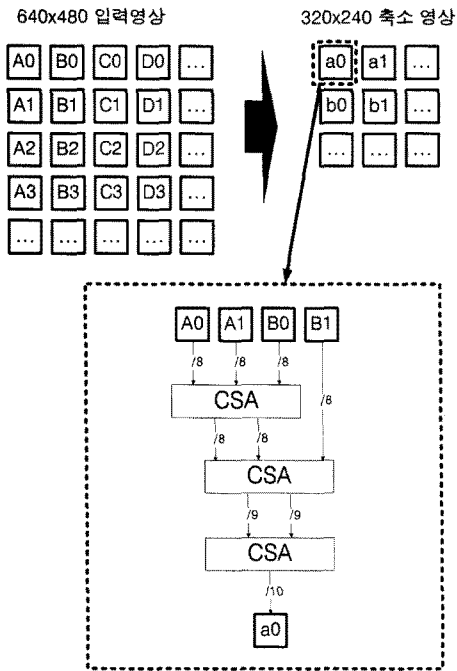


그림 3. 영상 축소 알고리즘의 하드웨어 연산 구조<sup>[6]</sup>  
 Fig. 3. Hardware Architecture of Image Reduction.

표 2. 버퍼 메모리 구성도  
 Table 2. Buffer Memory Table.

설명	비트수	길이	개수
2배 영상 버퍼 메모리	9bit	1,280	8 EA
1배 영상 버퍼 메모리	16bit	320	4 EA
1/2배 영상 버퍼 메모리	10bit	320	2 EA
1/4배 영상 버퍼 메모리	12bit	160	1 EA

더하였다. 4개의 픽셀값을 빠르게 더하기 위해 CSA (Carry Save Adder)를 이용한 구조로 그림 3과 같이 설계하였다. 그림 3은 1/2영상을 생성하기위한 경우의 영상 축소 알고리즘의 하드웨어 연산구조를 나타낸다.

Resize Image를 처리하는데 필요한 내부 메모리는 총 13.8kbit 로, Octave0(2배 영상) 메모리 9bit 1,280×8EA, Octave1(1배 영상) 메모리 16bit 320×4EA, Octave1(1/2배 영상) 메모리 10bit 320×2EA, Octave3(1/4배 영상) 메모리 12bit 160×1EA 로 구성된다.

Octave1 원본 영상 메모리가 4EA(4Line만큼을 임시로 저장)인 이유는 1/4영상을 만들기 위함이며, 8비트 데이터이기 때문에 16비트 저장 공간에 2개의 픽셀씩 입력한다. 그리고 총 15개의 메모리 셀은 효율적인 메모리 관리를 위해 병렬로 연결하여 개별적인 접근이 가능하다. 표 2는 버퍼 메모리 구성도를 나타낸다.

## 2. Histogram Equalization

히스토그램 평활화는 입력 영상이 너무 밝거나 어두운 영상을 보정하기 위한 과정이며, 이를 수행하여 픽셀차가 크지 않은 밝기가 비슷한 부분에서 생기는 세밀한 부분을 표현할 수 있다. 히스토그램 평활화 과정을 수행하기 위해서 아래의 네 단계를 거친다. 아래는 히스토그램 평활화 영상을 생성하기 위한 순서를 설명한다.

- (1) 입력된 영상의 최대 픽셀값과 최소 픽셀값을 구한다.
- (2) 전체 픽셀의 각 픽셀값을 최소 픽셀값으로 뺀다.
- (3) 픽셀 레벨을 정규화 하기 위하여 각 픽셀을 최대 픽셀값으로 나눈다.
- (4) 각 픽셀을 0부터 1사이의 값으로 만들어 최종 픽셀값을 결정한다.

위에서 설명한 내용을 바탕으로 하드웨어를 구성할 때에 필요한 최대값과 최소값은 Resize 모듈에서 SDRAM으로 저장할 때 레지스터에 저장해놓은 최대값(Max)과 최소값(Min)을 이용하여 구현한다. 히스토그램 평활화 모듈은 SDRAM에서 Resize Image를 읽어들이는 동시에 Histogram Equalization Buffer Memory에 저장하게 되는데 이때 Sequential Divider의 연산 클럭수를 포함하여 22클럭 파이프라인 구조로 설계하여 640×480영상 기준 약 총 10,560 클럭으로 100MHz 클럭에서 약 100.5us밖에 소요되지 않는다.

히스토그램 평활화 모듈의 메모리는 히스토그램 평활화 영상을 버퍼에 저장하기위한 메모리(1EA) + 가우시안 필터 마스크의 크기(17EA)의 총 18EA가 필요하며 영상의 최대 가로크기인 1,280의 Width와 16비트의 버스크기를 가지므로 총 368.64kb(1,280pixel×16bit×18EA)가 필요하다. 그리고 State에 따라서 각각의 메모리가 디코딩되어 1개의 메모리가 입력 모드일 때 나머지

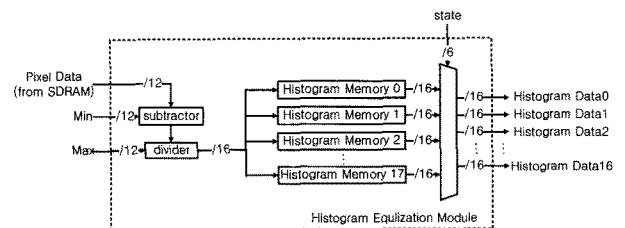


그림 4. 히스토그램 평활화 모듈의 하드웨어 구조도  
 Fig. 4. Hardware Architecture of Histogram Equalization Module.

지 17개의 메모리는 출력 모드로 설정되어, 가우시안 필터링과 히스토그램 평활화 영상의 입력이 동시에 이루어질 수 있다. 그림 4는 히스토그램 평활화 모듈의 하드웨어 구조도이다.

### 3. Gaussian Filtering

가우시안 필터 모듈은 히스토그램 평활화 모듈에서 출력되는 영상데이터를 미리 저장된 가우시안 필터 마스크데이터를 이용하여 필터연산을 수행하는 모듈이다. 필터 마스크의 크기는 17×17이며 세로 픽셀 연산 후 가로 픽셀 연산과정을 거쳐서 한 픽셀 씩 출력하는 방식으로 설계한다. 아래의 그림 5는 가우시안 필터 모듈의 하드웨어 구조를 나타낸다.

빠른 처리를 위하여 가우시안 필터 모듈 4개를 병렬로 구성하며, 각각의 가우시안 필터 모듈은 미리 저장

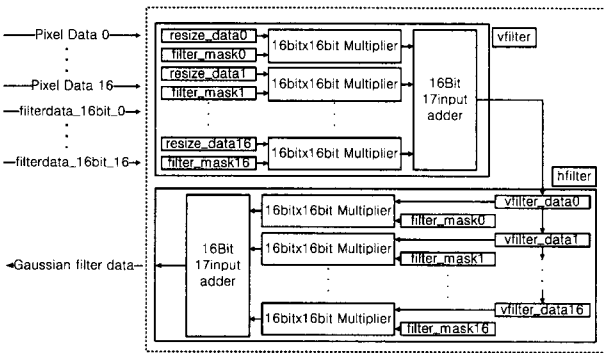


그림 5. 가우시안 필터모듈의 하드웨어 구조도  
Fig. 5. Hardware Architecture of a Gaussian Filter Module.

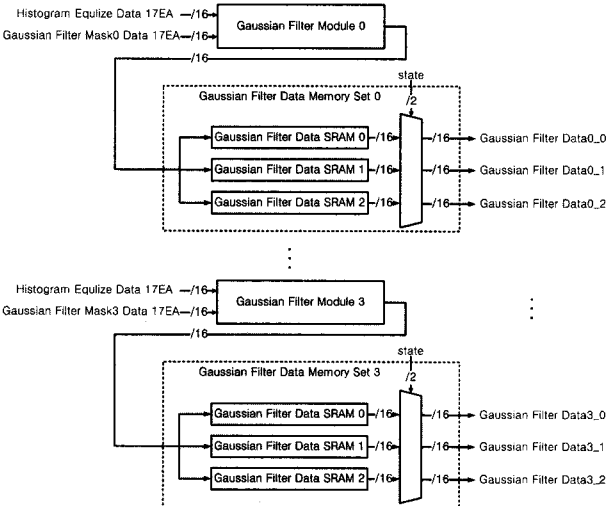


그림 6. 가우시안 필터 메모리의 하드웨어 구조도  
Fig. 6. Hardware Architecture of Gaussian Filter Memory.

된 서로 다른 필터 데이터를 입력받을 수 있도록 필터 메모리를 나누어 구성한다. 그림 6은 가우시안 필터 메모리의 하드웨어 구조도를 보인다.

### 4. DoG (Difference of Gaussian)

DoG영상은 두 개의 가우시안 영상의 차 영상을 말한다. 본 논문에서는 16bit의 Gaussian Filter 영상의 차를 구하면 부호비트를 포함한 17bit의 DoG 영상이 생성되는 고정 소수점 모델을 바탕으로 하드웨어를 구성한다. 그리고 극값을 검출하기 위해 필요한 3×3×3영역에 해당하는 DoG 픽셀영역을 생성하기위해 DoG 모듈을 3개로 병렬로 연결 후, 각각의 DoG 모듈에 두 개의 파이프라인용 레지스터를 추가하여 가로방향의 연속된 3픽셀을 구현한다. 아래의 그림 7은 3개의 DoG 모듈을 병렬로 구성한 하드웨어 구조도이다.

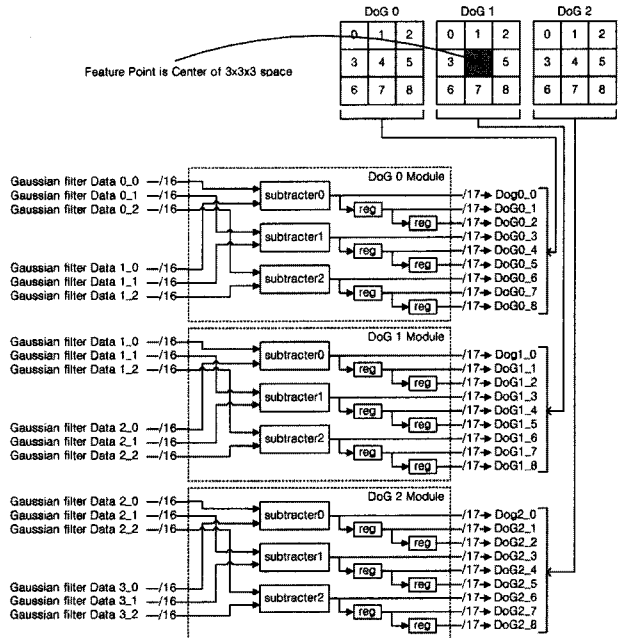


그림 7. DoG 모듈의 하드웨어 구조도  
Fig. 7. Hardware Architecture of 3 DoG Modules.

### 5. Extrema Detection

Extrema Detection 모듈에서는 DoG 영상으로부터 입력된 총 27픽셀 중에서 가운데 값이 나머지 26픽셀 값보다 가장 크거나 작고 Threshold에 만족하는 값인 특징점의 후보를 추출 하는 모듈이다. 모듈의 구성은 DoG1 영상의 4번째 픽셀 데이터가 나머지 26픽셀과 비교하여 극값일 경우 Local Maxima or Local Minima Detector 모듈에서 High 신호를 출력하며, DoG1 영상

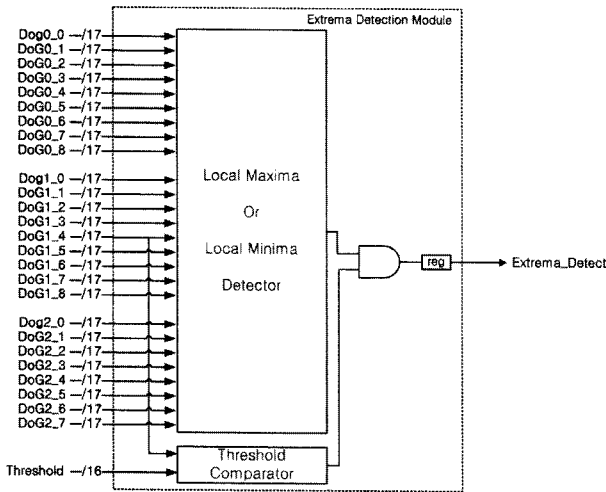


그림 8. Extrema Detection Module의 하드웨어 구조도  
 Fig. 8. Hardware Architecture of Extrema Detection Module.

의 4번째 픽셀 데이터가 Threshold에 만족할 경우 Threshold Comparator 모듈에서 High신호를 출력한다. 그리고 두 개의 결과 신호를 AND 게이트를 통과 시켜서 최종 레지스터에 저장한다. 본 논문에서는 16bit 비교기 54개를 병렬로 구성하여 한 클럭만에 연산 가능하도록 하였다. 그림 8은 Extrema Detection 모듈의 하드웨어 구조도이다.

6. Eliminate Edge Response

특징점의 위치가 결정되기 위해서는 특징점이 위치한 좌표에 해당하는 위치가 객체의 모서리나 평면부면이면 특징점이 될 수 없게 해야 한다. 그 이유는 매칭단계에서 모서리나 평면부면에 있는 데이터는 거의 비소

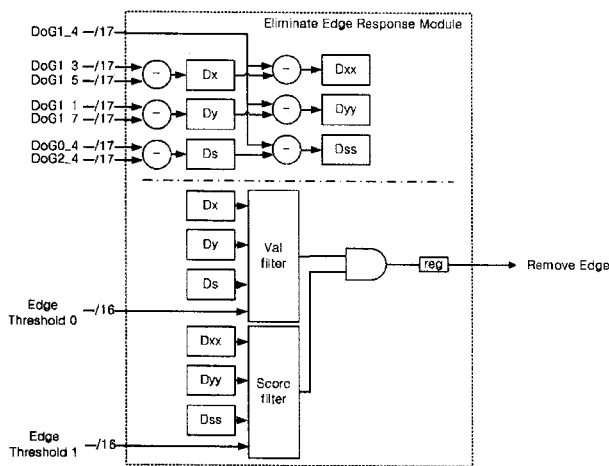


그림 9. 에지제거 모듈의 하드웨어 구조도  
 Fig. 9. Hardware Architecture of Eliminate Edge Response Module.

한 벡터성분을 지니고 있으므로 매칭성능을 현저히 저하시키는 원인이 되기 때문이다. 따라서 에지성분 제거 모듈은 [5]에서 나타낸 Score Filter 모듈과 Val Filter 모듈을 이용하여 하드웨어를 구성한다. 그림 9는 에지 제거 모듈의 하드웨어 구조도를 나타낸다.

7. Orient Assignment

방향설정 모듈은 해당 가우시안 영상에서 특징점 위치의 경사도를 8방향으로 표현한 데이터를 출력하는 모듈이다. 수식 (1)은 특징점에 대한 방향설정 과정을 나타내었는데, 삼각함수와 나눗셈 연산 때문에 하드웨어로 구현하기가 힘들다.

$$D_x = \frac{GaussianImage(s, x + 1, y)}{2} - \frac{GaussianImage(s, x - 1, y)}{2}$$

$$D_y = \frac{GaussianImage(s, x, y + 1)}{2} - \frac{GaussianImage(s, x, y - 1)}{2}$$

$$THETA = fmod(atan2(D_y, D_x) + 2\pi, 2\pi)$$

$$bin = (int)(\frac{36 \times THETA}{2\pi}) \tag{1}$$

따라서 본 논문에서는 모듈의 입력부분인 Dx, Dy와 출력의 bin의 상관관계를 이용하여 최종 방향설정 데이터를 추출한다. [5]에서 나타내듯이 Dx, Dy의 변수길이는 8비트로 고정되며, 이에 따라 최종 Dx/Dy 연산식이 bin으로 출력되기 위한 조건으로 이루어진다는 사실을 알아냈다. 방향설정 모듈은 크게 Dx/Dy의 나눗셈 모듈(Sequential Divider)과 그 결과를 이용하여 데이터의 구간이 어느 영역에 속하는지 구분하는 비교기 모듈 (Region Comparator)로 구성된다. 아래 그림 10은 방향 설정 모듈의 하드웨어 구조도를 나타낸다.

앞에서 설명한 각 모듈을 통합하여 검증하기 위해 그림 11과 같이 구성을 하였다. 우선 검증을 위한 ARM9 CPU의 External Memory Interface에 연결하기 위해 Bridge 모듈을 구성하였으며, ARM으로부터 입력된 명

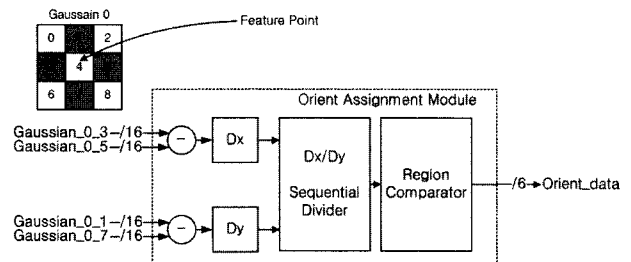


그림 10. Orient Assignment 모듈의 하드웨어 구조도  
 Fig. 10. Hardware Architecture of Orient Assignment Module.

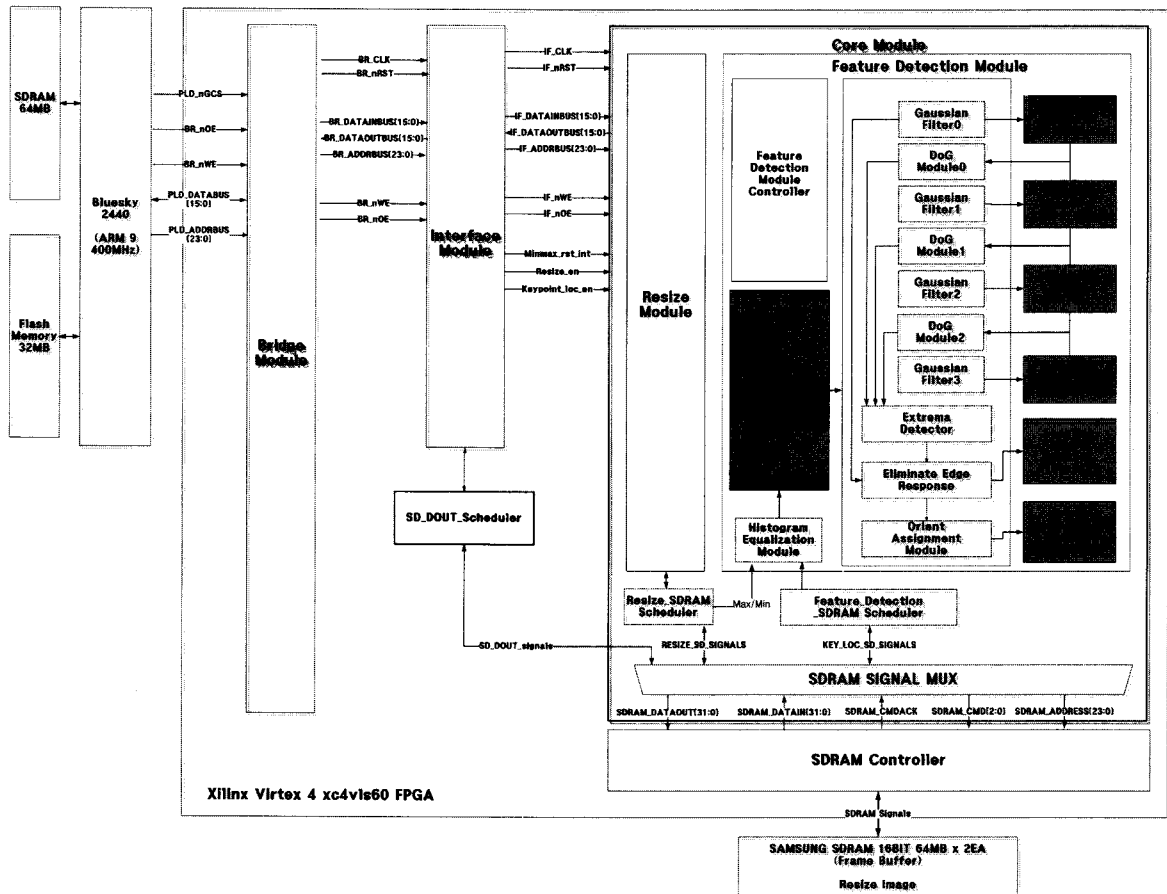


그림 11. SIFT 특징점 검출기의 하드웨어 구조도  
Fig. 11. Hardware Architecture of SIFT Feature Detector.

령 및 Data를 입력받고, 칩의 상태를 출력하기 위한 Interface 모듈을 설계하였다. 그리고 Core 모듈은 영상 데이터를 입력받는 동시에 Resize영상을 SDRAM으로 전송하는 Resize 모듈과 SDRAM에서 Resize 영상 데이터를 읽어 들여서 히스토그램 평활화, 가우시안 필터링, DoG의 연산과정을 거쳐 특징점의 위치를 결정하여 내부 SRAM에 결과 데이터를 저장하는 Feature Detection 모듈로 구성한다. 이때 가우시안 필터 영상을 3줄씩 출력하기 위해 각 가우시안 필터모듈에 1,280×3 EA의 SRAM을 추가하였다. 그리고 SD\_DOUT\_Scheduler 모듈을 구성하여 ARM 소프트웨어에서 SDRAM의 영상데이터를 읽고 디버깅을 할 수 있도록 설계하였다. 그림 11은 SIFT 특징점 검출기의 하드웨어 구조를 보인다.

IV. 성능 분석

하드웨어 설계에 대한 검증은 SAMSUNG S3C2440 ARM9 Processor 를 탑재한 EmLinux사의 Bluesky

2440보드와 Xilinx사의 Virtex4 XC4VLS60 FPGA 이용 하였다. Bluesky2440 Board는 128MB의 SDRAM과 64MB의 Flash Memory를 내장하고 있으며, 10/100 Ethernet Port를 탑재하여 Network File System 환경에서 검증이 가능하다. 그리고 XC4VLS60 FPGA는 총 59,904LUT와 최대 2,880Kbit의 고정된 메모리를 사용이 가능하며, 16bit의 데이터 버스를 갖는 SAMSUNG SDRAM 64MB×2EA가 연결되어 있어서 32bit로 접근이 가능하다. 아래의 그림 12는 검증환경을 나타낸다.

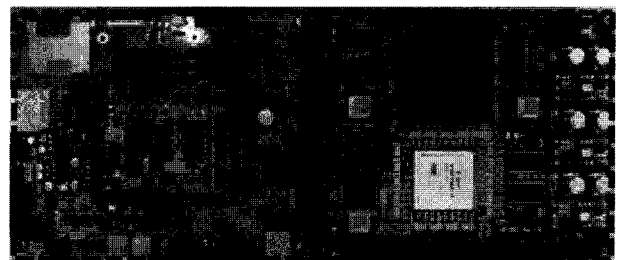


그림 12. 검증환경  
Fig. 12. Verification Environment.

Verilog HDL 언어로 작성한 RTL 코드는 FPGA를 타겟으로 Synplify 8.1i를 이용하여 합성한 결과 445,792/59,904LUT를 사용하였으며, 가우시안 Filter Module이 26,175LUT로 가장 많은 자원을 사용했다. 그리고 최대 동작 속도는 103.5MHz로, 실제 에뮬레이션 상에서는 100MHz 클럭을 적용한 결과 문제없이 동작함을 확인했다.

본 논문에서 구현한 하드웨어와 비슷한 사례로 [3]과 [4]가 있으며, 표 4는 [3]과 [4]를 본 논문에서 설계한 하드웨어와의 비교를 나타낸다. [3]에서는 내부 메모리만을 이용하여 데이터를 저장하고, 7×7크기의 가우시안 필터 마스크 적용하여 특징점 위치결정 단계까지를 FPGA로 구현 하였을 때, 512×512 크기의 영상을 약 46fps의 속도로 구동이 가능하다. 그리고 이와 비슷한 사례로, [4]에서는 입력된 영상에 대해 5×5 크기의 Laplacian 필터링 된 영상에서 Harris Corner Detector를 적용하여, 코너성분을 검출하는 알고리즘을 고정소수점 모델로 변환 후 FPGA로 구현하였다. 해당 논문에서는 ALTERA사의 Stratix S80 FPGA를 4개로 연결한 구조를 가지며 총 70,159,179 Logic Element의 Resource가 사용되었으며 2개의 1GB DDR SDRAM bank를 프레임 버퍼로 이용하였고, 12MHz부터 60MHz까지의 개별 Clock을 적용 하였을 때 640×480영상 기준 30fps의 연산속도를 보인다.

하지만, 위에서 설명한 내용들은 모두 Feature

표 3. 고정 소수점 모델과 하드웨어의 연산속도 비교 표

Table 3. Comparison of Fixed Point Model and Hardware.

동작 환경 모듈명	고정 소수점 모델 (ARM9 400MHz)	하드웨어 (100MHz)
영상 입력 모듈 (with resize)	942 ms	220 ms
히스토그램 평활화 모듈	2,053 ms	212.1 ms
가우시안 필터 모듈	103,390 ms	
DoG 모듈	5,262 ms	
특징점 위치결정 모듈	18,732 ms	
방향 설정 모듈	24,601 ms	
전체 구동 시간	152,941 ms	432.1 ms

Detection 기능까지를 하드웨어로 구현할 때 약 30fps 이상의 속도를 가지므로, 표 1과 같이 연산량의 약 50%에 해당하는 서술자 생성부분까지 하드웨어로 구현할 경우 SIFT 알고리즘 전체를 실시간으로 동작시킬 수 없다. 하지만 본 논문에서는 640×480영상을 기준으로 약 5ms의 연산시간을 나타내므로 이러한 단점을 극복할 수 있다.

Verilog HDL언어로 구현된 하드웨어 모델을 FPGA의 100MHz 클럭에서 구동 하였을 때 SDRAM에 저장된 리사이즈 영상데이터를 FPGA로 전송 후 연산과정을 거쳐 결과 데이터를 내부 SRAM에 저장할 때까지의 시간이 특징점의 개수와 상관없이 1,280×960영상기준 약 24ms, 640×480영상기준 약 5ms, 320×240영상기준

표 4. 성능 비교

Table 4. Performance Comparison.

내용	[3]	[4]	본 논문
영상 크기	512×512	640×480	1,280×960, 640×480, 320×240, 160×120
출력데이터	x, y	x, y	x, y, orient data
마스크 크기	7×7	11×11	17×17
디바이스	Xilinx Spartan 3	Altera Stratix s80	Xilinx Virtex4 xc4vls60
클럭	정보 없음	18MHz~60MHz (모듈별 다중 클럭 입력)	100MHz (단일 클럭 입력)
연산시간	512×512 : 30 fps	640×480 : 46 fps	1,280×960 : 40 fps 640×480 : 200 fps 320×240 : 1000 fps 160×120 : 1000 fps 이상
자원	DSP : 5 EA LUT : 224 LUTs	Multiscale Harris : 78,728 LUT (99.6%) Iteration Block : 110,227 LUT (140%)	DSP : 64 EA Total LUT : 45,792 (85%)
외부메모리	사용안함	1GB DDR SDRAM × 2 EA	64MB SDRAM × 2 EA
내부메모리	16 block rams	71.5 Kbits	284.8 Kbits



약 1ms, 160×120영상기준 약 0.3ms의 시간이 걸리므로 소프트웨어와 동일하게 반복수행 할 경우 총 212.1ms ((24ms+5ms+1ms+0.3ms)×7번 반복수행)의 연산시간이 소요된다. 따라서 소수점 모델을 ARM9 프로세서 환경에서 구동 시킬 때에 비하여 표 3과 같이 약 355배 빠른 연산속도를 갖는다. 이러한 연산속도는 표 4에서 나타내듯이 기존논문에서 제시하는 하드웨어모델에 비하여 빠른 속도로 동작이 가능하며, 서술자 생성부분과 특징점 끼리의 매칭 단계를 하드웨어로 구현할 경우 SIFT 전체 연산의 실시간 처리가 가능할 것으로 예상된다. 아래의 표 3은 ARM9 환경에서의 고정 소수점 모델과 7회 반복수행한 하드웨어 연산속도의 비교표를 나타내며, 표 4는 다른 논문과의 성능비교표를 나타낸다.

## V. 결 론

SIFT알고리즘을 임베디드 환경에서 구동시키기 위해서는 속도의 개선이 필요하다. 본 논문에서는 SIFT 알고리즘의 임베디드 환경에서의 실시간 구동을 위해서 SIFT 특징점 검출기를 설계하였다.

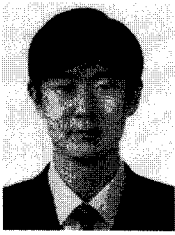
본 논문에서는 연산 수행 시간을 빠르게 처리하기 위하여 리사이즈 데이터를 SDRAM에서 FPGA로 전송 후 히스토그램 영상을 버퍼에 저장하는 동시에 기존에 저장된 히스토그램 영상을 생성하며, 가우시안 필터 모듈, DoG 모듈, 극값추출 모듈, 에지성분 제거모듈, 방향 설정 모듈, 그리고 결과데이터를 저장하는 SRAM을 파이프라인 구조로 설계하였다. 1,280×960영상일 경우 40fps, 640×480영상일 경우 200fps, 320×240영상일 경우, 1000fps, 160×120영상일 경우 1000fps이상의 빠른 연산결과를 보인다. 그리고 본 논문과 유사한 알고리즘을 하드웨어로 구현한 논문<sup>[3-4]</sup>과 비교한 결과, 보다 빠른 속도로 동작함을 확인했다.

완성된 하드웨어는 FPGA에물레이션 결과 고정 소수점 모델을 ARM9 환경에서 구동한 결과와 비교하여 약 2분 30초 이상 빨라지는 효과를 보았으며, Xilinx Virtex4 XC4VLS60 FPGA를 타겟으로 합성한 결과 약 85%의 면적을 차지하였다. 본 논문에서 제시한 하드웨어 구조를 이용하여 SIFT전체 알고리즘을 하드웨어로 구현할 경우 임베디드 환경에서 실시간 객체 인식 및 추적시스템 개발이 가능할 것으로 예상된다.

## 참 고 문 헌

- [1] D. G. Lowe. "Object recognition from local scaleinvariant features." In Proceedings of the Seventh International Conference on Computer Vision (ICCV'99), pages 1150-1157, Kerkyra, Greece, September 1999.
- [2] D. G. Lowe, "Distinctive image features from scale invariant keypoints", International Journal of Computer Vision, vol. 60, no. 2, 2004.
- [3] Paolo Giacon, Saul Saggin, Giovanni Tommasi, Matteo Bausti, "Impelmenting DSP Algorithms Using Spartan-3 FPGA", p16~p18, DSP Magazine, October, 2005.
- [4] Cristina Cabani, W. James MacLean, " A Proposed Pipelined-Architecture for FPGA-Based Affine-Invariant Feature Detectors." accepted for presentation at The Second IEEE Workshop on Embedded Computer Vision Systems, New York, June 2006.
- [5] 박찬일, 이수현, 정용진, "SoC 하드웨어 설계를 위한 SIFT 특징점 위치 결정 알고리즘의 고정 소수점 모델링 및 성능분석", 전자공학회 논문지 제 45 권 SD편 제 6호, 2008.
- [6] 박찬일, 이수현, 정용진, "SIFT 알고리즘의 하드웨어 구현을 위한 리사이즈 모듈 설계", 2008 SoC Conference,대한전자 공학회 SoC 설계 연구회, 2008.
- [7] 정성태, Visual C++을 이용한 실용 영상처리, 생능출판사, 2007.

저 자 소 개



박 찬 일(학생회원)  
 2004년 중부대학교 정보통신  
 공학과 학사 졸업.  
 (컴퓨터공학과 복수전공)  
 2009년 전자통신공학과 석사  
 졸업.  
 2008년 12월~현재 (주)썬이미지  
 MTS

<주관심분야 : 영상처리 및 인식, 임베디드 시스  
 템, 통신, 신호처리>



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

<주관심분야 : SoC 설계, 영상처리 및 인식, 임베  
 디드 시스템 설계>



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