

논문 2008-45SD-6-8

SoC 하드웨어 설계를 위한 SIFT 특징점 위치 결정 알고리즘의 고정 소수점 모델링 및 성능 분석

(Fixed-Point Modeling and Performance Analysis of a SIFT Keypoints
Localization Algorithm for SoC Hardware Design)

박 찬 일*, 이 수 현*, 정 용 진**

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

요 약

본 논문에서는 SIFT(Scale Invariant Feature Transform) 알고리즘을 임베디드 환경에서 실시간으로 처리하기 위해 가장 연산량이 많은 특징점 위치 결정 단계를 고정 소수점 모델로 설계 및 분석하고 그에 근거한 하드웨어 구조를 제안한다. SIFT 알고리즘은 객체의 꼭지점이나 모서리와 같이 색상 성분의 차가 심한 구역에서 얻어진 특징점 주위 픽셀의 벡터성분을 추출하는 알고리즘으로, 현재 얼굴인식, 3차원 객체 인식, 파노라마, 3차원 영상 복원 작업의 핵심 알고리즘으로 연구 되고 있다. 본 알고리즘에 대한 최적의 하드웨어 구현을 위해 특징점 위치(Keypoint Localization)와 방향(Orient Assignment)에 대한 정확도, 오차율을 사용하여 고정 소수점 모델에서 각 중요 변수들의 비트 크기를 결정 한다. 얻어진 고정 소수점 모델은 원래의 부동 소수점 모델과 비교했을 때 정확도 93.57%, 오차율 2.72%의 결과를 보이며, 고정 소수점 모델은 부동 소수점 모델과 비교하여 제거된 특징점의 대부분이 두 영상에서 추출된 특징점끼리의 매칭과정에서 불필요한 객체의 모서리 영역에 몰려있음을 확인 했다. 고정 소수점 모델링 결과 ARM 400MHz 환경에서 약 3시간, Pentium Core2Duo 2.13GHz 환경에서 약 15초의 연산시간을 갖는 부동 소수점 모델이 동일한 환경에서 약 1시간과 10초의 연산시간을 가지며, 최적화된 고정 소수점 모델을 하드웨어로 구현 시 10~15 frame/sec의 성능을 보일 것으로 예상된다.

Abstract

SIFT(Scale Invariant Feature Transform) 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 3-D image constructions, and its most computation-intensive stage is a keypoint localization. In this paper, we develop a fixed-point model of the keypoint localization and propose its efficient hardware architecture for embedded applications. The bit-length of key variables are determined based on two performance measures: localization accuracy and error rate. Comparing with the original algorithm (implemented in Matlab), the accuracy and error rate of the proposed fixed point model are 93.57% and 2.72% respectively. In addition, we found that most of missing keypoints appeared at the edges of an object which are not very important in the case of keypoints matching. We estimate that the hardware implementation will give processing speed of 10 ~ 15 frame/sec, while its fixed point implementation on Pentium Core2Duo (2.13 GHz) and ARM9 (400 MHz) takes 10 seconds and one hour each to process a frame.

Keywords : SIFT, Keypoint, Object recognition, Fixed point modeling

I. 서 론

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

※ 본 논문은 IDEC의 톨 지원과 IT-SoC 사업단 설계
실습 프로젝트 및 서울시 혁신 클러스터 (나노
IP-SoC)사업의 지원으로 이루어졌습니다.

접수일자: 2007년1월23일, 수정완료일: 2008년5월26일

객체 인식은 생체 인식과 더불어 미래 생활에 편리함을 더해줄 수 있는 인식 기술 중 하나이다. 이러한 객체 인식을 위해 사용 되는 알고리즘은 특징점 벡터 추출 방법^[1], 패턴인식 방법^[11], 공간적 컬러 모델을 이용하는

방법^[12], 동영상에서 차분 분할을 이용한 움직이는 객체 검출 방법^[13] 등으로 나누어진다. 그 중 SIFT (Scale Invariant Feature Transform) 알고리즘^[1]은 입력된 영상에서 객체의 모서리 부분에서 생성되는 특징점의 벡터성분을 추출하는 알고리즘이며, 비교 영상의 크기 변화 또는 회전으로 인한 변형에 대해 매칭단계에서 독특하고, 강인한 특징을 갖고 있다.

SIFT 알고리즘은 객체 인식^[1]뿐만 아니라 3D 영상 복원^[5], 모션 트래킹^[14] 등 다양한 분야에서 활발한 연구가 이루어지고 있다. 하지만 반복 연산 과정이 많아 일반 PC환경에서 구동 시킬 경우 실시간 처리가 힘들다. 따라서 실시간 객체 인식을 위해서는 하드웨어로의 구현이 필수적이다.

이러한 SIFT 알고리즘은 성능, 실행속도, 활용분야등 다각도에서 많은 연구가 이루어지고 있으며, 최근 구현된 사례는 다음과 같다. 먼저 [2]에서는 레퍼런스 영상에서 얻어진 SIFT 특징점과 현재 입력영상에서 얻어진 SIFT 특징점끼리의 매칭과정을 통해 3개 이상의 매칭되는 특징점이 있을 경우 Affine Transform을 이용하여 객체의 윤곽을 표시하였고, [9]에서는 II장의 연속된 사진으로부터 SIFT 특징점을 추출한 후, 1개의 파노라마 영상으로 병합했다.

[5]는 스테레오 카메라를 통해서 입력된 두 개의 영상에서 공통된 특징점을 찾아서 데이터베이스에 저장하고, 다음 프레임에서 얻어진 공통된 특징점끼리 연결하여 3차원 지도를 구현했다. 이러한 작업을 수행하기 위해 해당 환경에서 SIFT 특징점을 추출하기 위해 부동 소수점 모델로 구현 후, Simulink라는 툴을 이용하여 Xilinx사의 VIRTEX 2 FPGA에서만 동작이 가능한 고정 소수점 모델로 변환했다. 하지만 해당 FPGA에 적합한 형태로 변환되었기 때문에 SoC 구현에는 메모리의 크기나, 동작 소모 전류 등의 많은 제약이 있다.

그래서 본 논문에서는 SIFT 알고리즘 중 가장 연산량을 많이 차지하는 특징점 위치 결정 단계를 SoC에 직접 이식이 가능한 형태의 하드웨어 설계를 위해 해당 과정 까지를 고정 소수점 모델로 구현하고 성능을 분석한다. 그리고 부동 소수점 모델의 연산구조를 하드웨어에 적합한 구조로 변환한다.

내용의 구성은 II장에서 SIFT 특징점 위치결정 알고리즘에 대하여 설명하고, III장에서는 SIFT 특징점 위치결정 알고리즘의 부동 소수점 모델과 고정 소수점 모델의 유사성을 판단하기 위해 특징점 위치와 방향성분의 정확도를 이용하여 성능을 분석하고 고정 소수점 모

델을 구현 하였으며, IV장에서는 SIFT 특징점 위치결정 알고리즘의 하드웨어 구현을 위한 구조를 제안한다. 마지막으로 V장에서 결론을 맺는다.

II. SIFT 특징점 위치결정 알고리즘

본 논문에서 사용된 SIFT 특징점 위치 결정 알고리즘은 UCLA Vision Lab 에서 공개한 알고리즘^[3]을 참고했다. 아래에 해당 알고리즘의 실행 순서를 설명하였으며, 그림 1에 흐름도를 나타내었다.

- (1) 영상 확대/축소 단계(Resize Image)
- (2) 히스토그램 평활화 단계 (Histogram Equalization)
- (3) 가우시안 필터 영상 생성 단계 (Gaussian Filtering)
- (4) DoG 영상 생성단계 (DoG Processing)

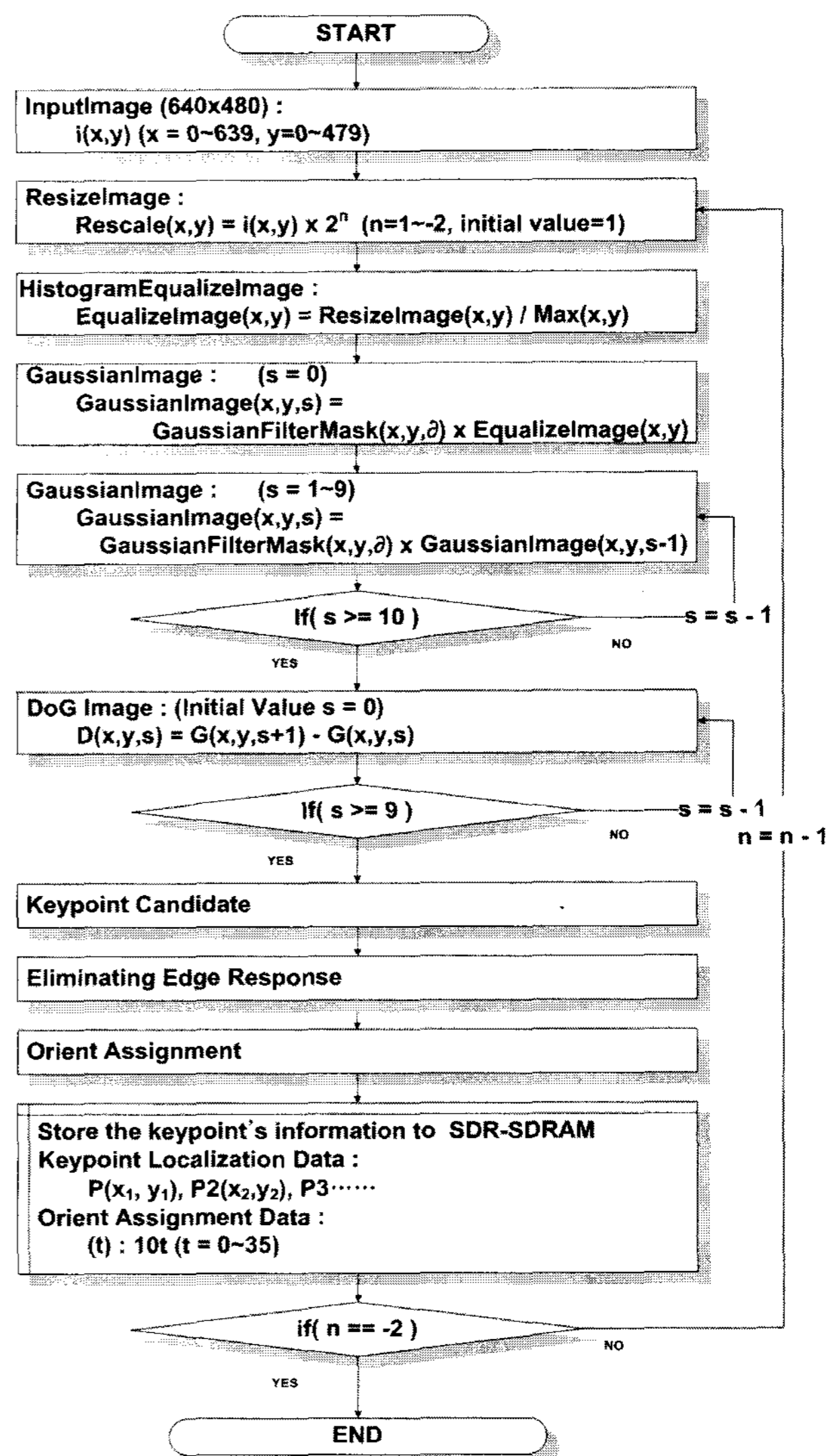


그림 1. SIFT 특징점 위치 결정 알고리즘의 흐름도
Fig. 1. SIFT keypoints localization algorithm flow.

- (5) 특징점 후보 선정 단계 (Keypoints Candidate)
- (6) 엣지 성분 제거 단계 (Eliminate Edge Response)
- (7) 방향 설정 단계 (Orient Assignment)

1. 영상 확대/축소

본 논문에서 사용된 SIFT 특징점 위치 결정 알고리즘은 입력된 영상을 4단계의 배율로 영상의 크기를 변환하여, 서로 다른 크기를 갖는 영상에서 특징점을 추출하는 과정을 갖게 된다. 이러한 입력된 영상에서 크기를 4단계로 확대/축소하는 작업은 객체 인식에 있어서 멀리 있는 객체가 작아 보이거나 커 보일 때 인식성

표 1. 양선형 보간법 알고리즘
Table 1. Bilinear interpolation algorithm.

```

for(y = 0; y < 960; y++)
  for(x = 0; x < 1280; x++)
  {
    src_x = x / (float)src_x;
    src_y = y / (float)src_y;
    alpha = src_x - x / src_x;
    beta = src_y - y / src_y;
    Ax = x / 2;    Ay = y / 2;
    Bx = Ax + 1;  By = Ay;
    Cx = Ax;      Cy = Ay + 1;
    Dx = Ax + 1  Dy = Ay + 1;
    E=(int) (InputImage[Ay*640+Ax]*(1-alpha)+
             InputImage[By*640+Bx]*alpha );
    F=(int) (InputImage[Cy*640+Cx]*(1-alpha) +
             InputImage[Dy*640 +Dx]*alpha );
    ResizeImage[y][x]=(float)(E*(1-beta)+F*beta);
  }
  
```

표 2. 영상축소 알고리즘
Table 2. Algorithm of image reduction.

```

ResizeImage = 0;
k = reduction parameter;

for(y=0; y<Height; y=y-k)
  for(x=0; x<Width; x=x-k)
  {
    for(ky=y; ky<k; ky++)
      for(kx=x; kx<k; kx++)
        ResizeImage(k,x,y)=ResizeImage(k,x,y) +
                             InputImage(k,kx,ky);
    ResizeImage(k,x,y)=ResizeImage(k,x,y) / 2k;
  }
  
```

능을 높이는 효과를 얻을 수 있다.

입력된 영상을 2배 크기로 확대하기 위해서 양선형 보간법을 이용하는데, 이는 영상을 확대할 때 생기는 영상손실을 최대한 줄여준다. 표 1은 해당 알고리즘에 이용된 양선형 보간법 알고리즘^[10]을 나타낸다. 그리고 영상 축소는 근접 픽셀의 평균값을 이용하여 입력된 영상(640x480, 정수부 8비트)의 1/2영상(320x240, 정수부 8비트, 소수부 1비트), 1/4영상 (160x120, 정수부 8비트, 소수부 2비트)을 생성한다. 표 2는 영상 축소 알고리즘을 설명한다.

2. 히스토그램 평활화

히스토그램 평활화는 입력 영상에 대해서 확대/축소된 영상이 너무 밝거나 어두운 영상을 보정하기 위한 과정이며, 이를 수행하여 영상의 보다 세밀한 부분을 표현할 수 있다.

입력된 640x480픽셀 크기의 8비트 흑백 영상의 밝기를 보정하기 위해 영상의 각 픽셀 값의 특성을 누적 분포하고 이를 정규화 하여, 정규화 된 값을 참조하여 픽셀값을 보정한다.

히스토그램 평활화 과정을 수행하기 위해서 다음의 다섯 단계를 거친다.

- (1) 입력된 영상의 확대/축소과정을 거친 영상을 한 픽셀씩 불러들여 전체픽셀의 최소값을 구한다.
- (2) 전체 픽셀의 각 픽셀값을 최소값으로 뺀다.
- (3) 전체영상을 다시 한 픽셀씩 불러들여 최대값을 구한다.
- (4) 픽셀 레벨을 정규화 하기 위하여 각 픽셀을 최대 픽셀값으로 나눈다.
- (5) 각 픽셀을 0부터 1사이의 값으로 만들어 최종 픽셀값(EqualizeImage)을 결정한다.

3. 가우시안 필터 영상

SIFT 알고리즘은 특징점의 위치를 결정하기 위해 DoG 영상을 이용하는데, DoG 영상을 만들기 위한 가우시안 필터 영상을 생성하는 단계는 특징점의 위치를 결정하는 과정에서 가장 중요한 부분이다.

본 논문에서 다루고 있는 SIFT 특징점 위치 결정 알고리즘에서는 10단계의 가우시안 필터 영상을 만드는데, 가우시안 필터 영상에 적용되는, 가우시안 필터 마스크는 입력된 영상의 특성에 따라서 달라질 수 있는 부분이기도 하다. 현재 최적화 되었다고 판단된 가우시안 필터 영상은 수식 (1)과 같이 σ 값에 의해 생성

된 10개의 서로 다른 필터 마스크를 사용한다.

$$w = \text{ceil}(4 \times \text{sigma})$$

$$\text{GaussianMaskWidth} = (\text{int})(2w + 1)$$

$$\text{Mask}(x) = \exp\left(-0.5 \times \frac{(x-w)^2}{\text{sigma}^2}\right)$$

$$\text{acc} = \sum_{x=0}^{\text{length}} \text{Mask}(x)$$

$$\text{GaussianFilterMask}(x) = \frac{\text{Mask}(x)}{\text{acc}} \quad (1)$$

표 3은 본 SIFT 특징점 위치 결정 알고리즘에 적용된 10개의 가우시안 필터 영상을 생성하기 위해 각 단계별로 σ 값을 생성하고 적용하는 방법을 나타낸다.

표 3. 10개의 가우시안 필터 영상을 생성하기 위한 σ 값 생성 과정과 필터 적용 알고리즘

Table 3. Calculate σ and application filter mask for making the ten slice of gaussian filter image.

$\sigma_0 = 1.6 \times 2^{\frac{1}{3}}$ $\sigma_n = 0.5$ $k = 2^{\frac{1}{7}}$ $\sigma = \sqrt{(\sigma_0 \times k^{-1})^2 - \sigma_n^2}$ $\text{GaussianImage}[0] = \text{GaussianFilterFunc}(\sigma, \text{EqualizedImage})$ $\text{for}(s=1; s<10; s++)$ $\{$ $\sigma = k^{(s-1)} \times \sigma_0 \times \sqrt{1 - \frac{1}{k^2}}$ $\text{GaussianImage}[s] = \text{GaussianFilterFunc}(\sigma, \text{GaussianImage}[s-1])$ $\}$
--

4. DoG 영상

DoG(Difference of Gaussian) 영상은 두 개의 가우시안 영상의 차 영상을 말한다. 0부터 1사이의 픽셀값으로 이루어진 영상끼리의 차를 구하면 -2부터 2사이의 픽셀값으로 이루어진 DoG 영상이 생성된다.

본 SIFT 특징점 위치 결정 알고리즘은 10개의 가우시안 영상을 통하여 9개의 DoG 영상을 생성한다. 수식 (2)는 두 개의 가우시안 영상에 의해 DoG 영상이 생성되는 방법을 설명한다.

s = Scale Space Element

$$\text{DoG}(s, x, y) = \text{GaussianImage}(k\sigma, s, x, y) - \text{GaussianImage}(\sigma, s-1, x, y) \quad (2)$$

5. 특징점 후보

현재 해당픽셀의 기준으로 3개의 DoG 영상의 3x3x3 영역에서 극값이 *Threshold*에 만족하는 값이 특징점의 후보가 된다. 수식 (3)은 특징점 후보를 구하는 방법을 보여준다.

$$\text{if}(\left((i(\text{DoG}(s, x, y))) == \max(\text{DoG}(s+l, x+m, y+n)) \right. \\ \left. \text{or } \min(\text{DoG}(s+l, x+m, y+n)) \right) \\ \&\& (i(\text{DoG}(s, x, y)) \leq \text{Threshold}) \\ \{ \\ \text{KeypointCandidate}(s, x, y) = i(\text{DoG}(s, x, y)) \\ \} \\ \left(-1 \leq l, m, n \leq 1 \quad l, m, n \text{은 정수} \right) \\ \left(\text{Threshold} = 0.004 \right) \quad (3)$$

6. 엣지 성분(Edge response) 제거

입력된 영상에서 객체의 모서리에 해당되는 좌표에 자리 잡은 특징점끼리의 매칭단계에서 불필요한 특징점 후보를 제거하는 단계로, Hessian Edge Detector^[2]에 s(scale space)성분을 추가 하는 변형된 방법을 이용하여, 해당 특징점 후보가 엣지 성분인지 아닌지를 판단하여 엣지 성분을 제거하게 된다. 표 4는 특징점 후보에 대한 엣지 성분을 제거하는 과정을 나타낸다.

7. 방향 설정

객체인식 단계에서 영상회전에 강인한 특성을 발휘하기 위해 특징점이 갖고 있어야할 정보 중에 하나인 방향 정보는 해당 가우시안 영상에 위치한 특징점 주위 픽셀의 경사도를 말한다. 경사도는 36개의 방향으로 표현하며, 정해진 방향은 [1]에서 설명하고 있는 서술자를 표현하기 위한 기준으로 쓰인다. 수식 (4)는 본 논문에서 다루고 있는 SIFT 특징점 위치 결정 알고리즘의 특징점에 대한 방향성분(bin)을 구하기 위한 공식을 나타내었다.

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

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

$$\text{THETA} = \text{fmod}(\text{atan2}(\text{Dy}, \text{Dx}) + 2\pi, 2\pi)$$

$$\text{bin} = (\text{int})\left(\frac{36 \times \text{THETA}}{2\pi}\right) \quad (4)$$

표 4. 특징점 후보에 대한 엣지 성분 제거 과정
Table 4. Eliminating edge response of keypoint candidates.

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

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

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

$$D_{xx} = \frac{DoG(s, x, y+1) + DoG(s, x, y-1) - 2(DoG(s, x, y))}{2}$$

$$D_{yy} = \frac{DoG(s, x+1, y) + DoG(s, x-1, y) - 2(DoG(s, x, y))}{2}$$

$$D_{ss} = \frac{DoG(s+1, x, y) + DoG(s-1, x, y) - 2(DoG(s, x, y))}{2}$$

$$D_{xy} = \frac{DoG(s, x+1, y+1) + DoG(s, x-1, y-1) - DoG(s, x+1, y-1) - DoG(s, x-1, y+1)}{4}$$

$$D_{xs} = \frac{DoG(s+1, x, y+1) + DoG(s-1, x, y-1) - DoG(s+1, x, y-1) - DoG(s-1, x, y+1)}{4}$$

$$D_{ys} = \frac{DoG(s+1, x+1, y) + DoG(s-1, x-1, y) - DoG(s+1, x-1, y) - DoG(s-1, x+1, y)}{4}$$

$$dx = ((-D_x > 0.6) ? 1 : 0) + ((-D_x < -0.6) ? -1 : 0)$$

$$dy = ((-D_y > 0.6) ? 1 : 0) + ((-D_y < -0.6) ? -1 : 0)$$

$$val = DoG(s, x, y) - \frac{D_x^2 + D_y^2 + D_s^2}{2}$$

$$score = \frac{(D_{xx} + D_{yy})^2}{D_{xx} \times D_{yy} - D_{xy} \times D_{xy}}$$

```

if(dx==0 && dy==0)
    if(fabs(Val) > 0.005)
        if(score < 12.1 && score >= 0)
            if(fabs(-D_x) < 1.5)
                if(fabs(-D_y) < 1.5)
                    if(fabs(-D_s) < 1.5)
                        {
                            Keypoint(s, x, y) = DoG(s, x, y)
                        }
                    }
                }
            }
        }
    }

```

III. 고정 소수점 모델 구현 및 성능 분석

SIFT 특징점 위치결정 알고리즘의 부동 소수점 모델은 연산시간이 펜티엄 PC 환경(Pentium4 Core2Duo 2.13GHz)에서 약 15초, 임베디드 환경에서 약 3시간의

연산속도를 가지므로 일반 펜티엄PC 환경에서 조차 실시간 객체 인식을 구현하기 힘들다. 그러므로 하드웨어 구현이 필수적이다. 따라서 본 장에서는 위치 결정 알고리즘을 하드웨어 구현을 위해 부동 소수점 모델을 고정 소수점 모델로 변환하는 과정과 그에 따른 성능을 분석한다.

본 논문에서 사용한 SIFT 특징점 위치결정 알고리즘에서 고정 소수점으로 선택한 변수는 히스토그램 평활화 모듈의 EqualizeImage 변수, 가우시안 필터 모듈의 GaussianFilterMask 변수와 GaussianImage 변수, DoG 변수, 엣지 성분 제거 모듈의 score, val 변수, 방향 설정 모듈의 Dx, Dy 변수가 있다.

부동 소수점 모델과 고정 소수점 모델의 유사성은 임의적으로 선정된 영상 10개를 기준으로 각 단계별로 고정 소수점 사용 비트수에 따른 PR(특징점 위치, 방향에 대한 정확도)와 ER(오차율) 이용하여 측정하였다. 수식 (5)는 히스토그램 평활화 영상, 가우시안 영상, DoG 영상의 부동 소수점 모델을 기준으로 고정 소수점 단계에 따른 영상의 훼손 정도를 구하기 위한 수식 (5)는 각 고정 소수점 변수에 따른 PR(특징점의 위치와 방향에 대한 정확도), 수식 (6)은 ER(오차율)을 구하기 위한 공식을 나타낸다.

- n_{Same} = 고정 소수점 모델과 부동 소수점 모델의 특징점에 대한 위치와 방향이 일치하는 개수.
- n_{Diff} = 고정 소수점 모델과 부동 소수점 모델의 특징점에 대한 위치 또는 방향이 일치하지 않는 개수.
- n_{New} = 부동 소수점 모델을 기준으로 고정 소수점 모델의 새로 생긴 특징점 개수.
- $n_{TotalFloating}$ = 부동 소수점 모델의 특징점 개수

$$PR(\%) = \frac{n_{Same}}{n_{TotalFloating}} \times 100 \quad (5)$$

$$ER(\%) = \frac{n_{Diff} + n_{New}}{n_{TotalFloating}} \times 100 \quad (6)$$

1. 히스토그램 평활 모듈

히스토그램 평활화 모듈에서는 EqualizeImage 변수를 고정 소수점 변수로 정한다. 그리고 비트별로 테스트한 후 결정된 비트수는 그림 2와 같이 화살표로 표시한다. 그림 2는 EqualizeImage 변수의 소수점 단계에 따른 PR과 ER의 테스트 결과를 보인다.

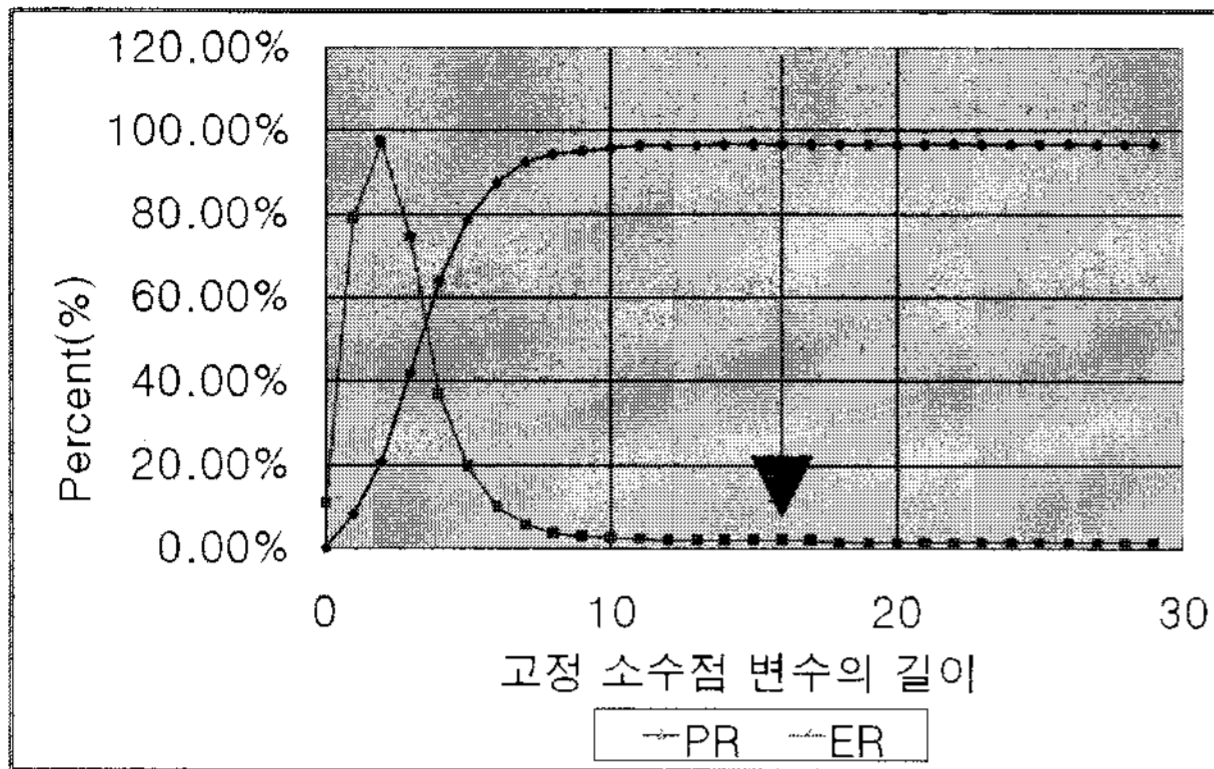


그림 2. EqualizeImage 변수의 소수점 단계에 따른 PR, ER 테스트 결과
 Fig. 2. PR, ER test result of EqualizeImage variable.

2. 가우시안 필터 모듈

가우시안 필터 모듈에서는 가우시안 필터 영상과 필터 마스크, 두 개의 고정 소수점 변수가 있다. 이 중 가우시안 필터 마스크(GaussianFilterMask)는 수식 (1)에서 생성된 1차원 마스크를 2차원으로 변환한 뒤 고정 소수점으로 변환하였다. 그리고 가우시안 필터 영상(GaussianImage)은 가우시안 필터 마스크를 적용하여 소수점 단계별로 테스트를 수행했다.

가. 가우시안 필터 마스크(GaussianFilterMask)

2차원 형식으로 변환된 가우시안 필터 마스크를 그림 3과 같이 고정 소수점 자리 수에 따른 PR 과 ER 테스트를 수행했다. 비트수가 적을수록 새로 생겨나는 특징점의 수가 약 700%까지 많아지면서 ER이 높아진다. 그림 3은 GaussianFilterMask 변수의 고정 소수점 단계에 따른 PR 과 ER 테스트 결과를 보인다.

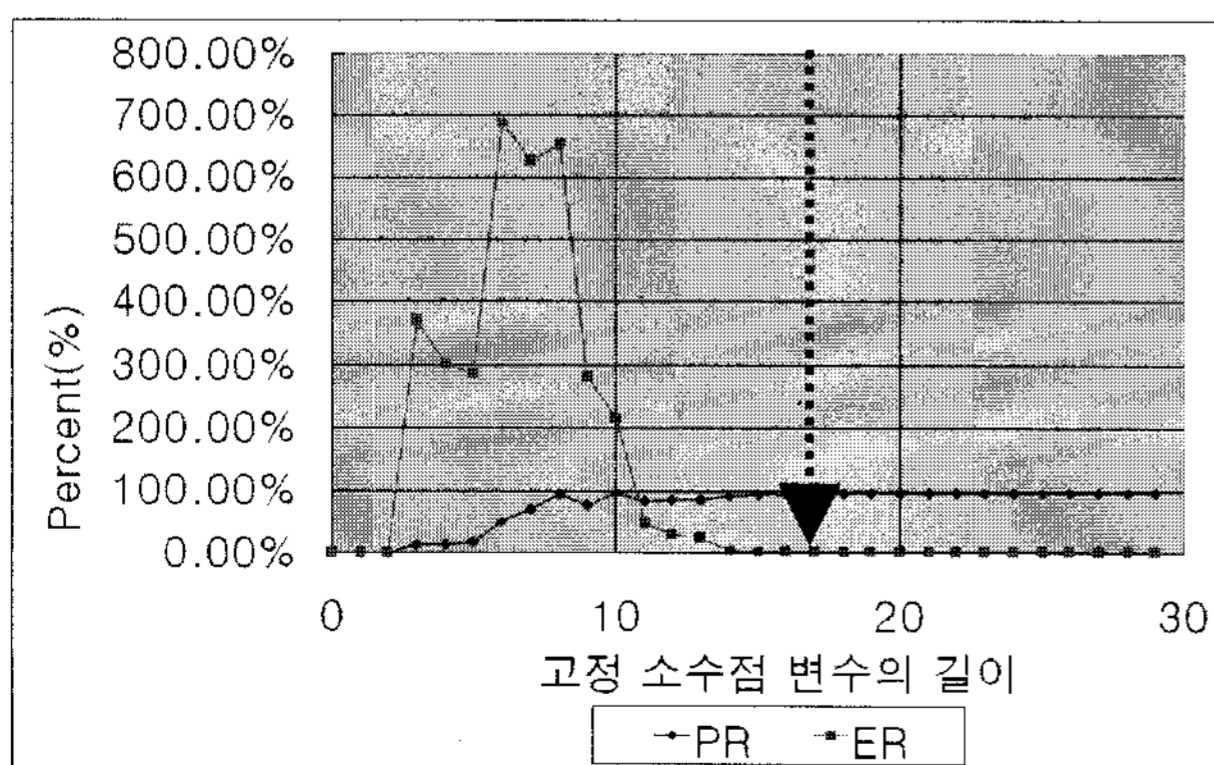


그림 3. GaussianFilterMask 변수의 고정 소수점 단계에 따른 PR 과 ER 테스트 결과
 Fig. 3. PR, ER test result of GaussianFilterMask variable.

나. 가우시안 필터 영상(GaussianImage)

GaussianImage 변수는 그림 4에서 볼수 있듯이 20비트를 사용해야 하지만, 객체의 모서리에 위치하여 매칭단계에서 불필요한 특징점이 제거되는 효과를 얻을수 있기 때문에 16비트로 고정한다. 그림 4는 GaussianImage 변수의 고정 소수점 단계에 따른 PR, ER테스트 결과를 보인다.

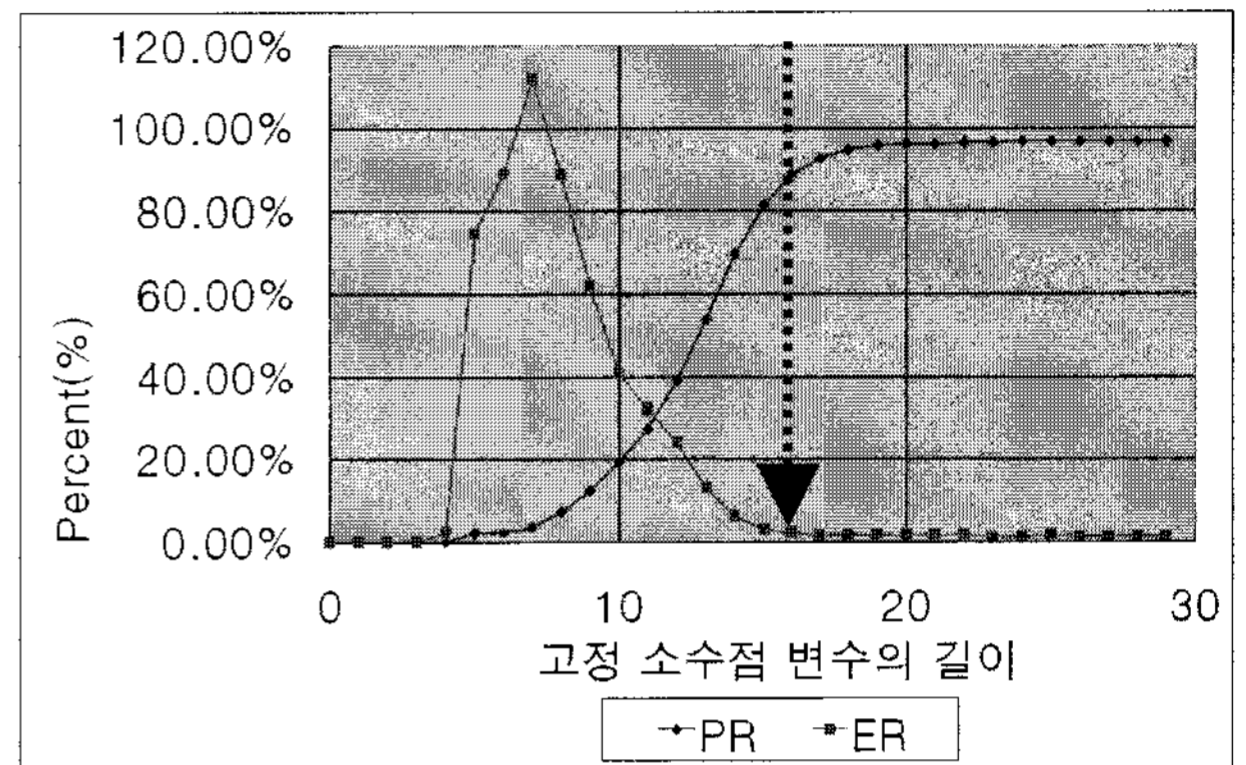


그림 4. GaussianImage변수의 고정 소수점 단계에 따른 PR, ER 테스트 결과
 Fig. 4. PR, ER test result of GaussianImage variable.

3. DoG(Difference of Gaussian) 모듈

DoG 모듈에서의 고정 소수점 변수는 DoG 변수이다. DoG 영상은 수식 (2)와 같이 가우시안 영상끼리의 차를 이용하여 생성되는 영상이기 때문에 굳이 소수점 비트별로 테스트를 할 이유가 없다. 따라서 DoG변수는 가우시안 영상의 최종 소수점 비트수를 따른다.

4. 특징점 위치 결정

특징점 후보를 수식 (3)과 같이 3x3x3 영역에서의 극값이 임계치 이상의 값을 표 4에서 설명한 엣지 성분

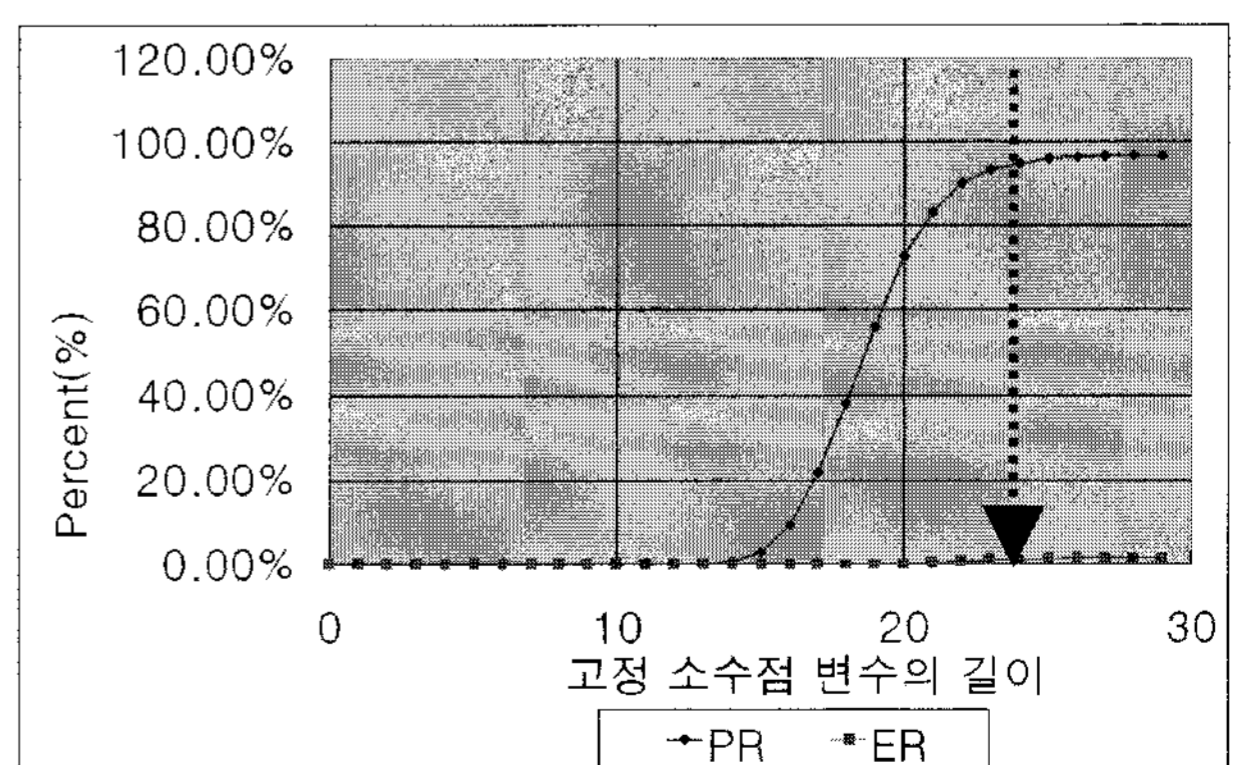


그림 5. 엣지 성분 제거 과정에 포함된 score 변수의 소수점 비트수에 따른 PR, ER 테스트 결과
 Fig. 5. PR, ER test result of score variable in the edge eliminating module.

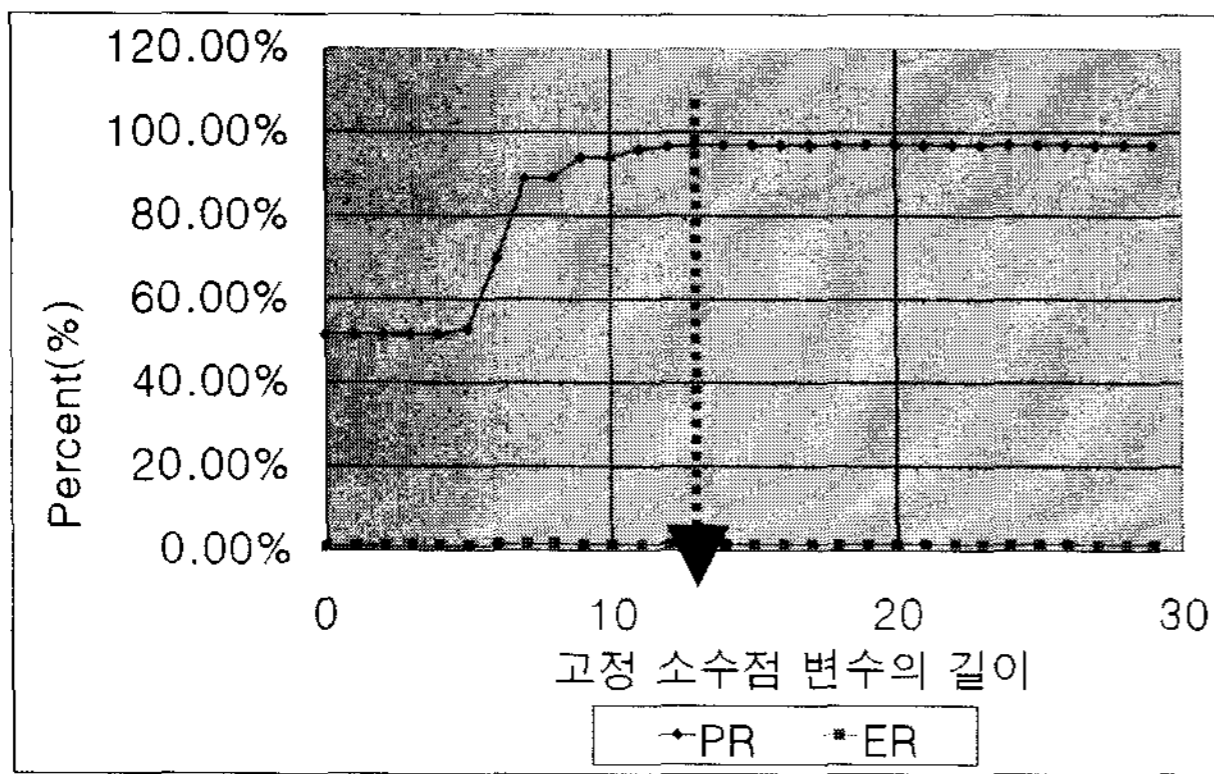


그림 6. 엣지 성분 제거 과정에 포함된 val 변수의 소수점 비트수에 따른 PR, ER 테스트 결과
 Fig. 6. PR, ER test result of val variable in the edge eliminating module.

제거 과정을 거치게 되어 특징점의 위치를 결정할 수 있는데, 테스트할 고정 소수점 변수는 score, val 이다.

그림 5는 엣지 성분 제거 모듈의 val 변수에 대한 소수점 비트수에 따른 PR과 ER 테스트의 결과를 보이고, 그림 6은 엣지 성분 제거 모듈의 score 변수의 소수점 비트수에 따른 PR과 ER 테스트의 결과를 보인다.

5. 특징점 방향 설정

수식 (4)에서 볼 수 있듯이 특징점에 대한 방향 설정 모듈의 입력 값은 Dx, Dy이고 출력 값은 bin 이다. 모듈 안에는 복잡한 연산식이 많지만, 하드웨어로 구현 시에는 복잡한 연산로직이 많을수록 시간과 로직 개수가 많아지기 때문에 모듈을 테이블로 구성하여 메모리에서 읽어 들이는 방법을 사용하는 것이 효과적이다. 따라서 Dx, Dy 변수의 소수점 비트수에 따른 bin의 정확도를 테스트한 결과 그림 7과 같은 결과를 얻었다.

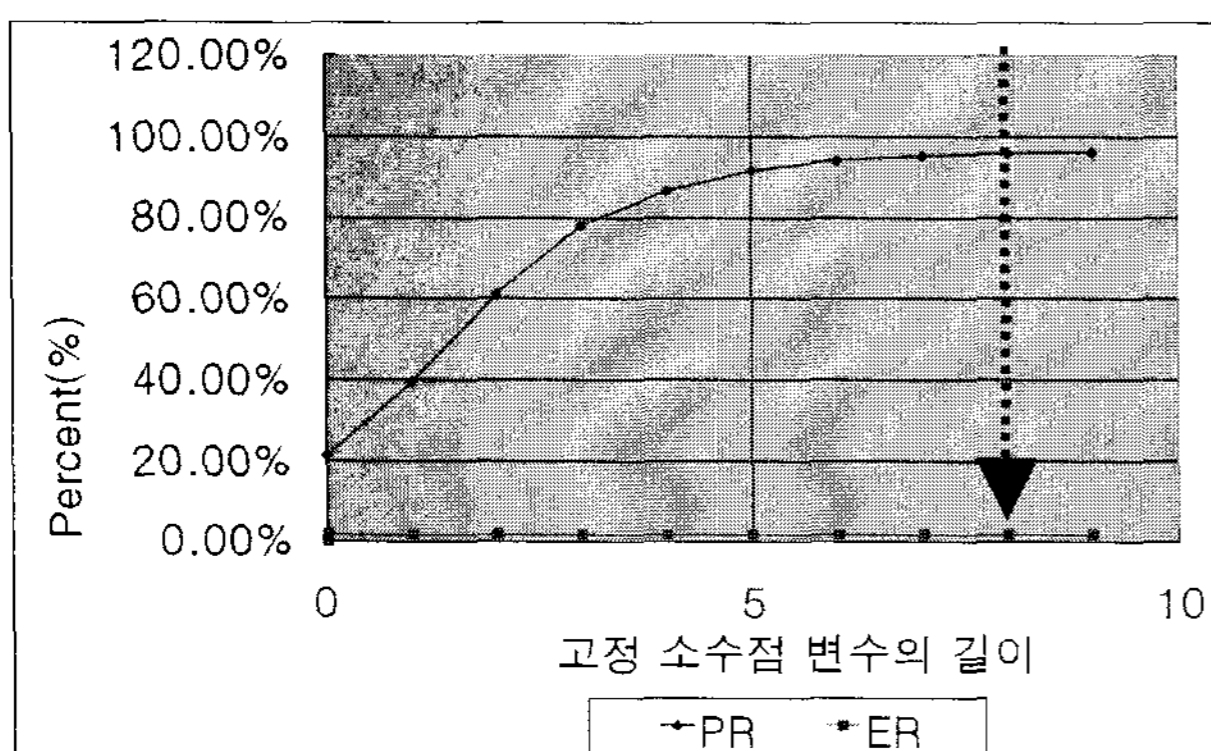


그림 7. 방향 설정 모듈의 Dx, Dy 값에 따른 bin 에 대한 정확도를 나타낸 결과
 Fig. 7. Result of accuracy about bin variable according to Dx, Dy in the orient assignment module.

그림 7은 방향 설정 모듈의 Dx, Dy 변수의 소수점 비트수에 따른 bin의 정확도에 대한 그래프를 나타낸다.

앞에서 설명한 그림 2~그림 7에 나타낸 바와 같이 고정 소수점 변수의 길이를 EqualizeImage변수는 16비트, GaussianImage변수는 16비트, GaussianFilterMask 변수는 16비트, score변수는 27비트, val변수는 13비트, Dx, Dy는 8비트로 결정한다. 그리고 결정된 각 변수에 대한 비트수를 적용한 고정 소수점 모델은 부동 소수점 모델 대비 정확도(PR) 93.57%, 오차율(ER) 2.72%의 성능을 보였으나, 고정 소수점 모델이 부동 소수점 모델에 비해 객체의 모서리 부분의 매칭단계에서 불필요한 특징점이 고정 소수점 모델에서 제거되는 효과를 얻었다. 그리고 PR이 ER보다 더 비트수 결정에 영향을 끼치므로 PR만을 이용하여 비트수를 결정할 수 있다. 그림 8은 [1]에서 발췌한 샘플영상 A와 B의 부동 소수점 모델과 고정 소수점 모델의 특징점 위치에 대하여 나타낸다. 'X'로 표시된 부분이 특징점의 위치를 나타내며, 원으로 표시한 부분이 고정 소수점 모델에서 삭제된 특징점을 육안으로 확인할 수 있는 부분이다.

연산 시간은 부동 소수점 모델의 경우 펜티엄PC에서 약 15초, 임베디드 환경에서 약 3시간에 비하여 고정 소수점 모델은 펜티엄 PC에서 약 10초, 임베디드 환경에서 약 1시간 정도로 부동 소수점 모델 대비 60~70% 정도의 향상된 처리 속도를 나타낸다. 그림 8은 부동 소수점모델과 비교되는 객체의 모서리 부분에 위치한 특징점이 제거된 고정 소수점 모델을 보인다.

IV. 하드웨어 설계 구조

본 장에서는 고정 소수점 모델을 바탕으로 하드웨어 설계를 위한 구조를 제안한다. 그림 9는 영상이 입력된 후 부터 특징점의 위치와 방향에 대한 데이터가 추출되기 까지의 하드웨어로 구현할 때의 구조를 보인다. 모듈의 구성은 테스트 플랫폼인 ARM9 프로세서의 메모리 인터페이스와 통신을 위한 Bridge, 어드레스 디코딩, 칩 셋팅, 레지스터 파일등을 포함하는 Interface, SIFT 특징점 추출연산을 처리하는 Core로 나뉘며, 아래에 SIFT연산 코어의 세부모듈을 설명한다.

(1) Resize Module

SDRAM에 저장된 원본영상을 Input Image Buffer에 순차적으로 읽어와 4단계의 확대/축소된 영상을 SDRAM에 동시에 저장한다.

그림 A 의 원본 영상

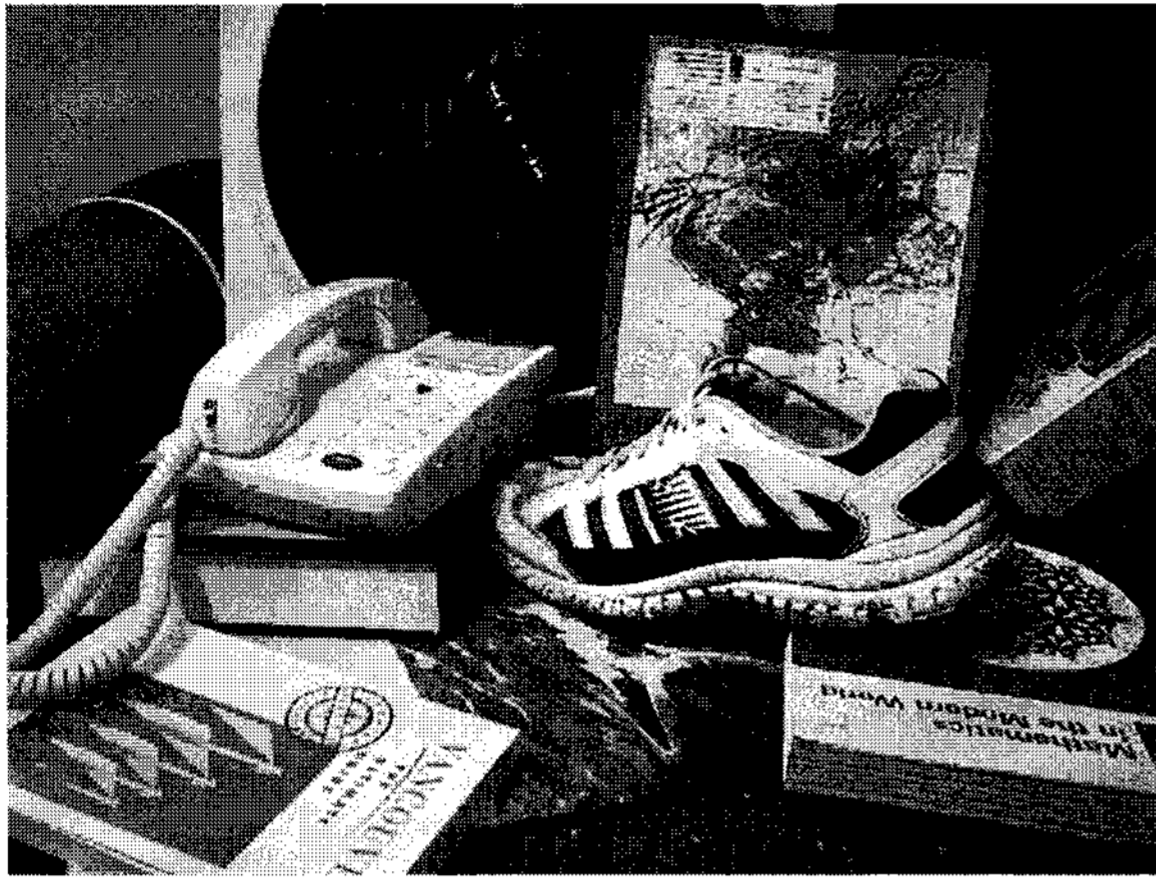


그림 B 의 원본 영상



그림 A 의 부동 소수점 모델

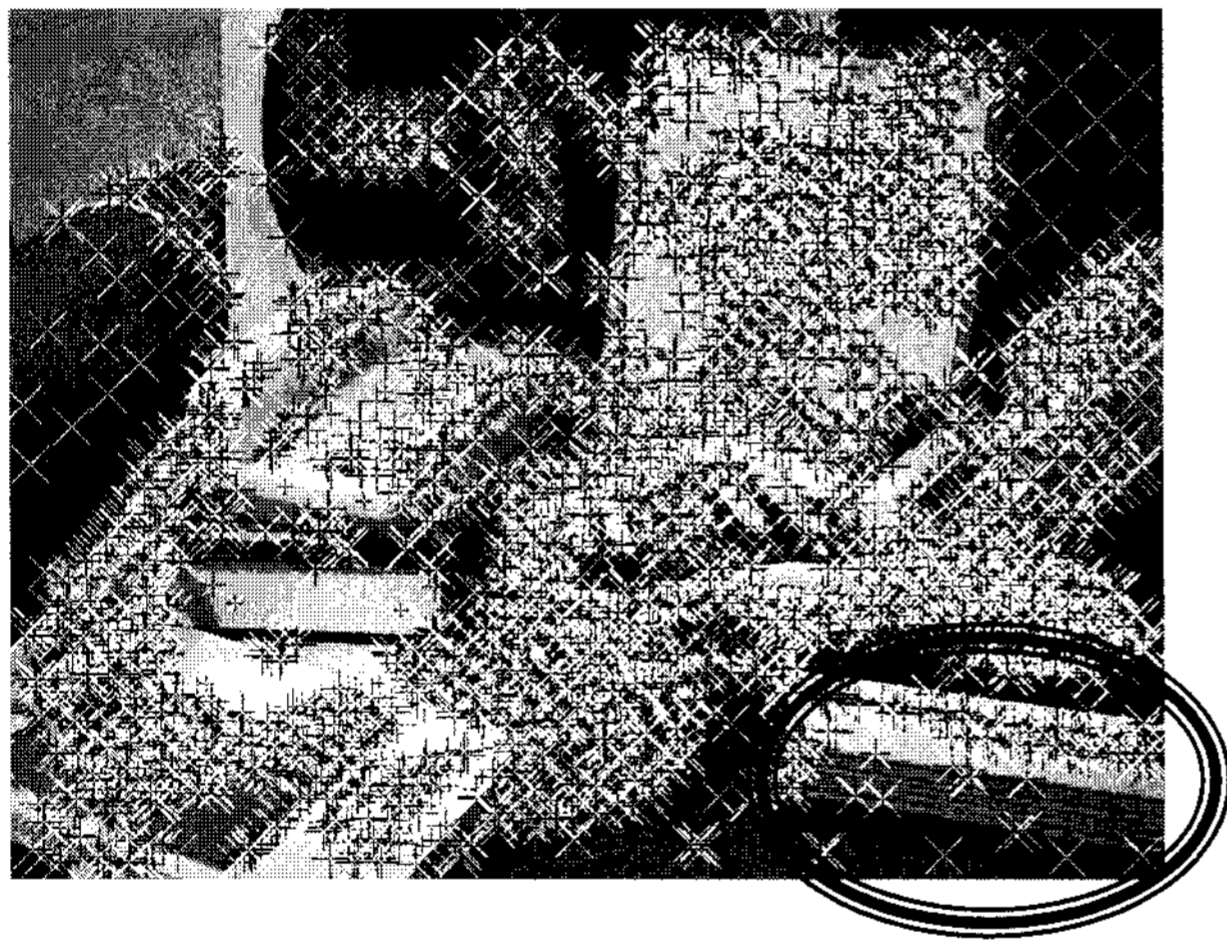


그림 B 의 부동 소수점 모델

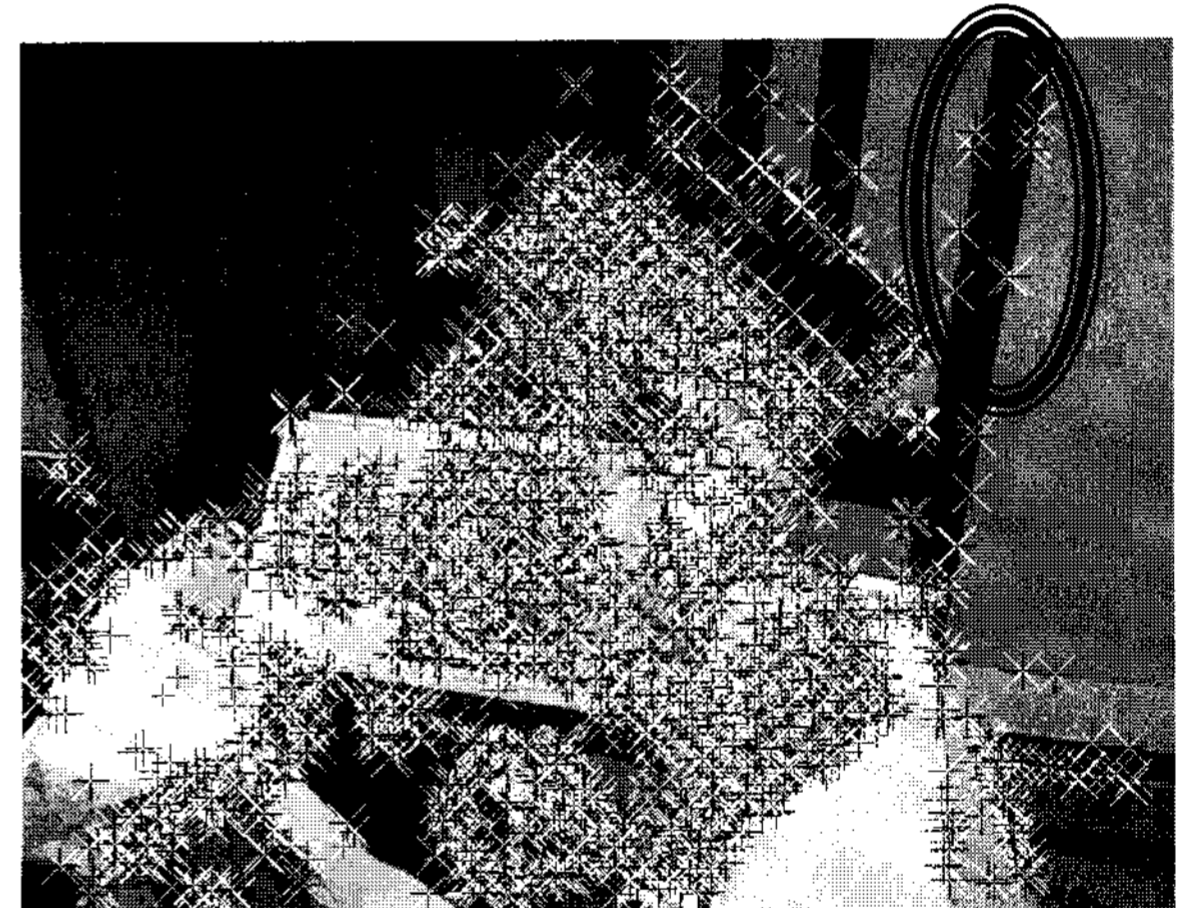


그림 A 의 고정 소수점 모델

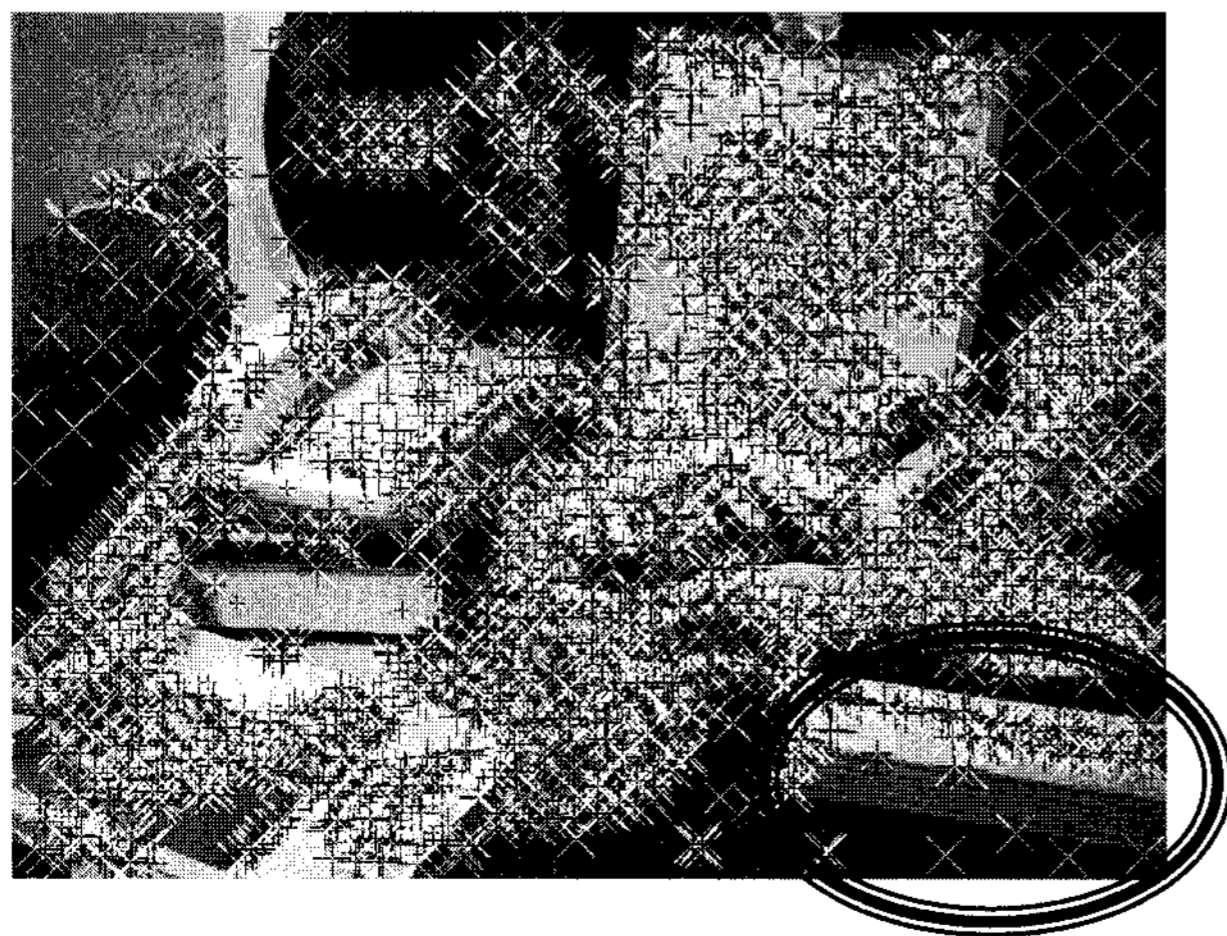


그림 B 의 고정 소수점 모델

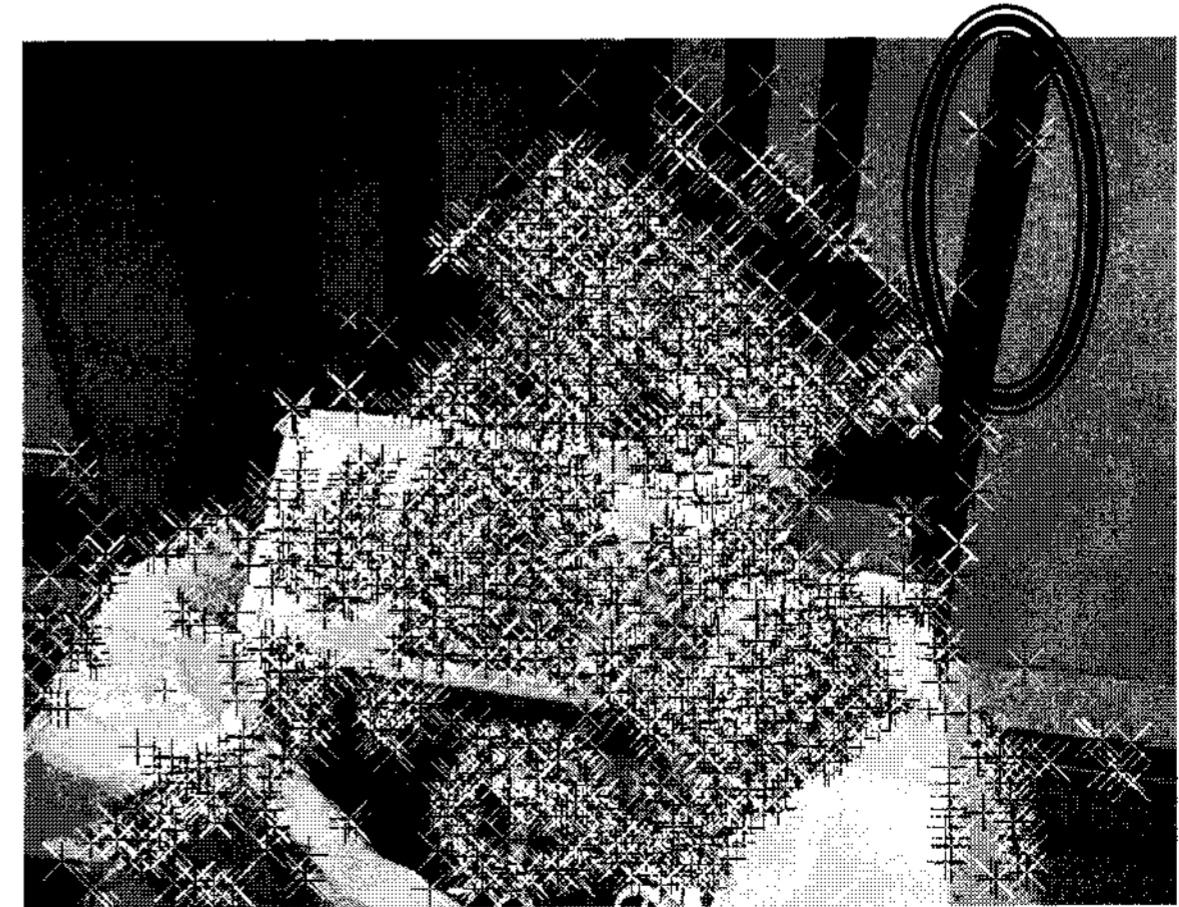


그림 8. 부동 소수점 모델과 비교되는 객체의 모서리부분에 위치한 특징점이 제거된 고정 소수점 모델
Fig. 8. Fixed point model eliminating keypoints on edge of the object compared to floating point model.

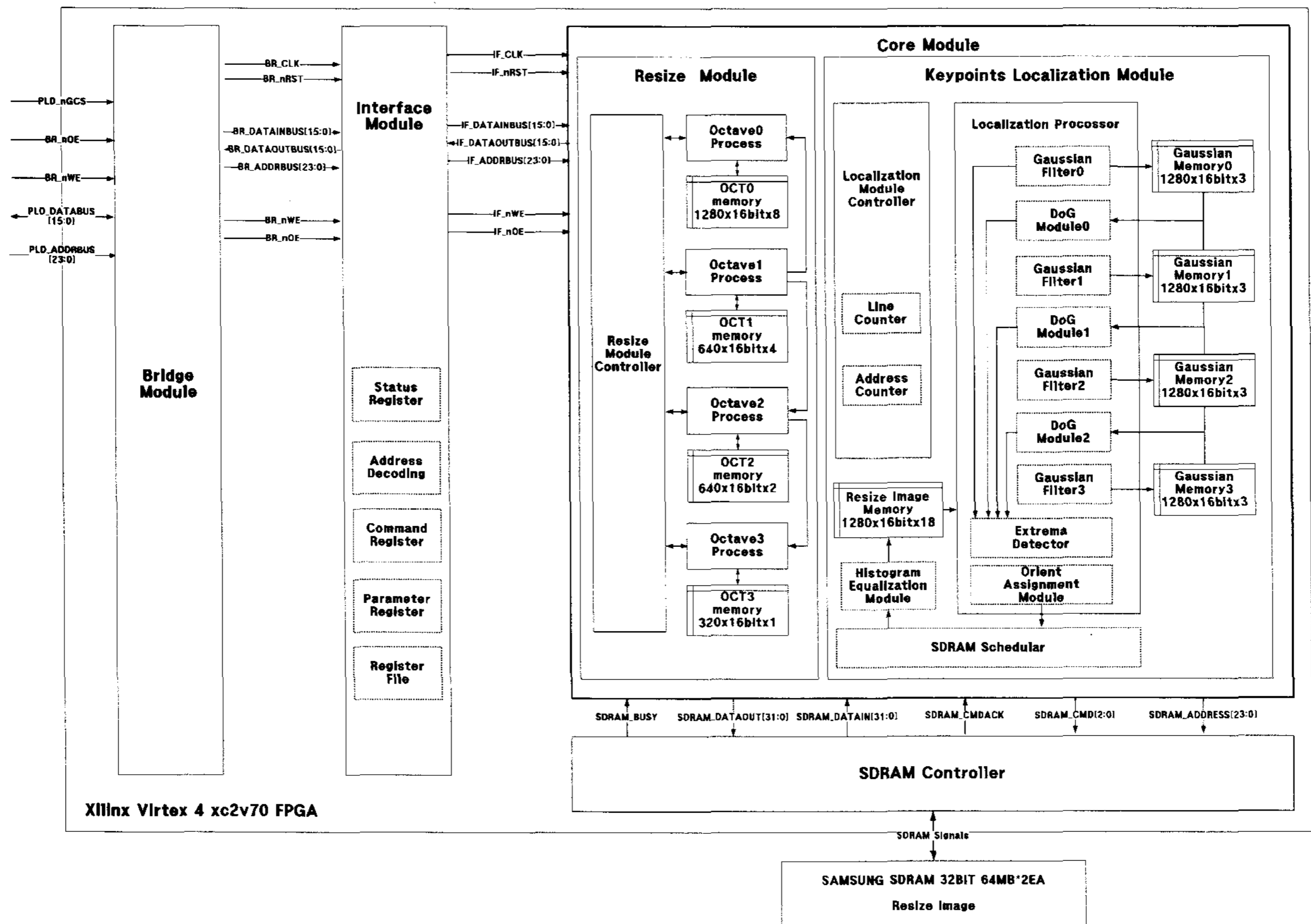


그림 9. SIFT 특징점 위치 결정 알고리즘 하드웨어 구조도
 Fig. 9. Hardware architecture of SIFT Keypoints localization algorithm.

(2) Histogram Equalize Module

SDRAM에 저장된 확대/축소된 영상을 읽어들이며 히스토그램 평활화 하여 Histogram Equalize Image Buffer에 가우시안 필터 사이즈만큼 저장한다.

(3) Gaussian Filter Module

Histogram Equalize Image Buffer에 저장된 영상을 Gaussian Mask Memory에 미리 저장된 가우시안 마스크 데이터를 읽어와, 필터 연산을 수행한 후 결과 영상 데이터를 Gaussian Image Buffer에 저장한다. 이때, 가우시안 필터는 1280x3 사이즈의 메모리를 병렬로 구성하여 Keypoint의 방향과 좌표데이터를 얻기 위한 최소한의 메모리를 사용한다.

(4) DoG(Difference of Gaussian) module

Gaussian Image Buffer에 저장된 2개의 가우시안 영상을 입력받아서 1개의 DoG영상을 생성하며, 총 가로 320, 세로 3만큼의 픽셀정보를 연속된 레벨의 DoG 3개를 읽어 들여서 극값을 찾으며, Extrema Detector

모듈에서도 동시에 픽셀 정보를 읽는다.

(5) Extrema Detector Module

DoG Image Buffer에 저장된 3개의 DoG영상 데이터를 입력받아서 특징점 후보를 결정하여 Orient Assignment Module에 좌표를 전송한다.

(6) Orient Assignment Module

Keypoint Candidate Module에서 결정된 좌표데이터를 입력받아 Gaussian Image Buffer에 저장된 가우시안 필터 영상의 해당 좌표에 대한 연산을 수행하여 특징점의 방향 성분을 계산하고 Orient Assignment Result Memory에 좌표정보, 방향정보 등을 저장한다.

(7) Keypoints Localization Top Controller

Keypoints Localization Module에 내장된 각 모듈의 동작 및 내부 메모리의 상태를 관찰하고 스케줄링 한다.

V. 결 론

본 논문에서는 실시간 객체인식을 위해 SIFT 알고리즘 중 가장 연산량이 많은 특징점 위치결정 단계를 고정 소수점 모델로 변환하고 그에 따른 성능을 분석하였으며, 하드웨어로 구현하기 위한 구조를 제안하였다.

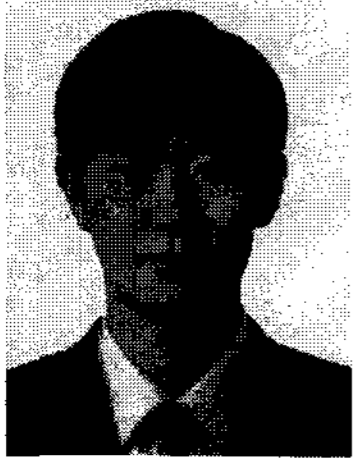
고정 소수점 모델은 부동 소수점 모델 대비 정확도(PR) 93.57%, 오차율(ER) 2.72%의 성능을 보였으나, 맷칭단계에서 불필요한 특징점이 제거되는 향상된 효과를 얻었다. 그리고 연산시간은 부동 소수점 모델의 경우 펜티엄PC에서 약 15초, 임베디드 환경에서 약 3시간에 비하여 고정 소수점 모델은 펜티엄PC에서 약 10초, 임베디드 환경에서 약 1시간 정도로 부동 소수점 모델 대비 60%~70%정도의 향상된 처리 속도를 나타냈다.

현재 IV장에서 나타낸 SIFT 특징점 위치 결정 하드웨어를 FPGA로 구현 중에 있으며 10~15 frame/sec 성능을 보일 것으로 예상된다.

참 고 문 헌

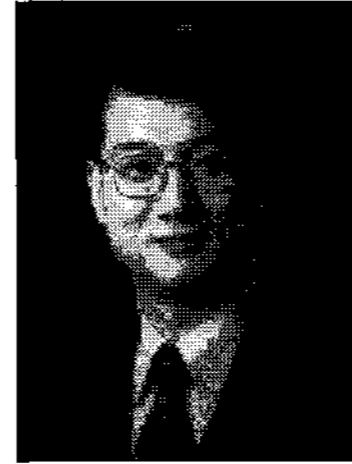
- [1] D. G. Lowe. "Object recognition from local scaleinvariant features." In Proceedings of the Seventh International Conference on Computer Vision (ICCV'99), pages 1150-1157, Kerkyra, Greece, September 1999.
- [2] D. G. Lowe, "Distinctive image features from scale invariant keypoints", International Journal of Computer Vision, vol. 60, no. 2, 2004.
- [3] 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.
- [4] Cabani C, MacLean WJ. "Implementation of an Affien-Covariant Feature Detector in Field-Programmable Gate Arrays", International Conference on Computer Vision Systems 2007, Bielefeld, Germany, March 21-24, 2007.
- [5] Stephen Se and Piotr Jasiobedzki, "Photo-realistic 3D Model Reconstruction", Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2006, pages 3076-3082, Orlando, Florida, May 2006.
- [6] Tim Barfoot, Stephen Se, Piotr Jasiobedzki, "Vision-based Localization and Terrain Modelling for Planetary Rovers", Chapter in Intelligence for Space Robotics, A. Howard and E. Tunstel, editors, TSI Press, Albuquerque, NM, 2006.
- [7] Pettersson. N, Petersson. L, "Online stereo calibration using FPGAs", Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, 6-8, June, 2005.
- [8] M. brown and D. G. Lowe. "Recognising Panoramas-slides." University of British Columbia, 2004.
- [9] M. Brown and D. G. Lowe. "Recognising Panoramas." Ninth IEEE International Conference on Computer Vision (ICCV'03) - Volume 2, 2003.
- [10] 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)
 2007년 3월~현재 광운대학교
 전자통신공학과 석사과정
 재학.

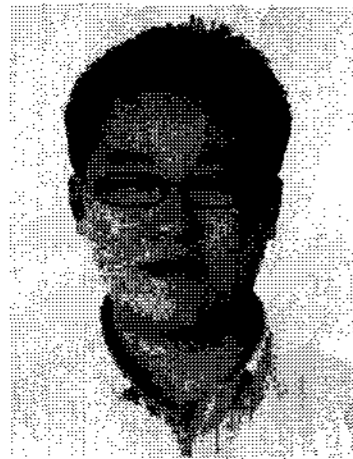
<주관심분야 : 영상처리 및 인식, 임베디드 시스
 템, 통신, 신호처리>



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>



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>