

# SIFT 하드웨어 구현을 위한 성능 실험 및 분석

어영정<sup>○</sup> 박진홍 한탁돈 변혜란

연세대학교 컴퓨터과학과

Sheldon@yonsei.ac.kr, jhpark@msl.yonsei.ac.kr, hantack@msl.yonsei.ac.kr,

hrbyun@yonsei.ac.kr

## Performance Experiment and Analysis for SIFT on Hardware

Youngjung Uh<sup>○</sup> Jinhong Park Tackdon Han Hyeran Byun

Dept. of Computer Science, Yonsei University

### 요 약

최근 많은 컴퓨팅 작업들이 모바일로 옮겨지기 시작하면서 존재하는 알고리즘을 하드웨어에 구현하여 속도를 높이는 것이 이슈가 되고 있다. 그 중 영상의 특징 점을 기반으로 두 개 이상의 영상을 매칭하는 기술을 중심으로 하는 기술이 다양한 분야에서 연구되고 있다. 본 논문에서는 다양한 분야에서 널리 활용되는 Scale Invariant Feature Transform(SIFT)라는 특징 점 추출 알고리즘의 성능을 분석하여 모바일 디바이스를 위한 비용대비 성능이 높은 최적의 매개변수를 찾는다.

### 1. 서 론

전 세계적으로 모바일 영상획득장치가 널리 보급되고 있다. 디지털카메라는 물론이고 근래에는 핸드폰에 탑재된 카메라의 성능이 일반 디지털카메라에 준하게 되었다. 그에 따라 단순한 영상획득이 아니라 획득한 영상을 [1][2]와 같은 각종 처리과정을 거쳐 다른 이미지를 생성하거나 새로운 방법으로 이미지를 검색하는 것이 화제가 되고 있다.

모바일 환경에서 주로 이루어지는 영상처리로는 영상검색, 얼굴검출 및 인식, 간판인식, 배경대체, 파노라마생성 등이 있다. 이러한 영상처리의 기본과정은 두 영상에 공통으로 존재하는 특징 점을 추출하고 이를 비교하는 것이다. 특징 점이란 영상에 존재하는 특정 구조(선,개체 등)을 말한다. 현재 특징 점을 추출하는 많은 방법이 컴퓨터 비전 분야에서 활발히 연구 중에 있다. David G. Lowe 는 Scale Invariant Feature Transform(SIFT)[3]라는 알고리즘을 제안했는데 이는 객체의 모서리나 꼭지점 등에서 생성되는 특징 점을 벡터로 추출하여, 영상의 크기변화, 회전, 조명변화 등에 의한 변형에 대해 뛰어난 매칭 성능을 가진다. 하지만 SIFT 알고리즘은 영상을 반복적으로 연산하는 과정이 많아서 처리속도가 좋지 않다. 따라서 최근 SIFT 알고리즘을 하드웨어로 구현하여 속도를 높이고자

하는 [4][5]와 같은 다양한 시도가 이루어지고 있다. 이 작업에는 높은 성능의 하드웨어 뒷받침이 필요하지만 모바일 기기에서는 이동성을 위해 작은 크기의 하드웨어를 사용하므로 하드웨어적으로 많은 제약이 따르게 된다. 그러므로 비용대비 성능이 높도록 연산의 최적화가 필요하다.

본 논문에서는 SIFT알고리즘의 수행성능 판단 척도를 정의하고 참을만한 수행비용에 매칭 성능을 높게 하는 방법을 실험적으로 찾는다.

### 2. SIFT 알고리즘의 수행시간 및 성능을 좌우하는 요소

SIFT 알고리즘은 특징 점을 추출하기 위해 여러 가지 단계를 거치는데, 각 단계별로 수행시간 및 성능에 영향을 미치는 요인은 다음과 같다.

#### 2.1 Scale-space extrema detection

첫 번째 단계는 같은 개체가 다른 관점에서라도 반복적으로 추출될 수 있는 크기와 위치의 후보 픽셀들을 뽑는 것이다. 이 단계에서는 이미지에 다양한 가우시안 블러를 씌운 가우시안 피라미드(Gaussian Pyramid)를 생성하고 가우시안 피라미드의 차이 값을 계산한 가우시안 차이(Difference of Gaussian) 피라미드를 구축한다. 이때 각각의 피라미드를 계산하는데 옥타브와 인터벌의 곱에 비례하는 연산

이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 연구임(No. 2009-0088516)

량이 필요하다. 인터벌이란 가우시안 커널의 시그마(sigma)값을 변화시키며 적용하여 만드는 가우시안 블러가 적용된 이미지의 개수를 말한다. 옥타브 생성 단계는 여러 인터벌을 묶은 한 단계이다. 이 과정은 현재 옥타브의 이미지를 크기를 절반으로 만들어 다음 옥타브를 구한다. 이때 옥타브의 개수와 인터벌의 수는 개체의 크기변화에 얼마나 강인한지에 영향을 미친다. 그리하여 가우시안 차이 피라미드의 극값(Local Extrema)을 잠재적인 특징 점으로 택한다.

## 2.2 Keypoint localization

각각의 후보픽셀에 대하여 더 세밀한 검사를 거쳐 특징 점으로 사용할지를 결정한다. 첫 번째 검사는 낮은 대비를 가지는 픽셀을 제거한다. 두 번째 검사는 모서리에 의해 추출된 특징 점을 제거한다. 이 부분은 헤시안 매트릭스를 이용하여 테일러 전개를 하는데, 행렬연산의 속도와 정확도가 중요하다. 그리고 후보픽셀에 대해 연산을 하는 만큼 후보픽셀이 적을수록 시간이 적게 걸린다.

## 2.3 Orientation assignment & Keypoint descriptor

이미지의 그레디언트 방향에 따라 각각의 특징 점에 하나 또는 다수의 방향이 지정된다. 이 작업을 통해 회전에 대해 강인한 특징 점이 된다.

후보픽셀의 주변에 생성된 그레디언트를 종합하여 표현자를 만든다. 이 작업은 조명과 시점의 차이에 의한 변화에 대해 강인한 특징을 계산한다.

## 2.4 단계별 연산 소요 시간 분석

[표 1]을 보면 SIFT알고리즘의 연산 량은 주로 가우시안 피라미드 생성과 가우시안 차이 피라미드 구축에서 발생함을 알 수 있다. 이 연산 량에 영향을 주는 것은 Pyramid의 옥타브 수와 인터벌 수 이므로 이들을 변경하면 연산 량을 조절할 수 있다. Pyramid의 크기가 바뀌면 그만큼 추후 연산도 바뀌므로 결국 SIFT알고리즘의 전체 연산 량을 옥타브 수와 인터벌 수로 조절 할 수 있다는 뜻이다.

[표 1] SIFT알고리즘의 주요 작업의 소요시간 비율

Gaussian Pyramid	DoG Pyramid	Extrema Detection	Orientation Assignment	Descriptor Generation	Total
40.6	1.2	12.9	7.9	37.4	100

(단위:%)

## 3. 모바일 하드웨어상의 성능 판단 척도

일반적인 SIFT에서는 같은 대상을 촬영한 다른 사진 두 장을 각각 특징 추출하여 매칭시킨 쌍이 많이 나오는 것이 성능이 높은 것이다. 하지만 모바일 환경에서는 많은 계산을 통해 매치를 많이 찾아도 수행시간이 지나치게 길면 실용적이지 못하다. 그러므로 본 논문에서는 수행시간당 매칭된 특징 수를 성능판단의 척도를 삼는다.

## 4. 실험 및 결과

### 4.1 실험목표

실험을 통해 수행시간의 증가가 두드러지지 않는 하에 특징이 최대한 많이 매칭되는 매개변수를 찾는다.

### 4.2 실험방법

이미지의 크기는 모바일 환경에서 일반적으로 사용되는 VGA(640\*480)를 사용하였다. 이미지는 일반인의 휴대전화 사진첩에 주로 들어있을 만한 사진을 사용하였다. 그 종류로는 사람, 꽃, 옷, 풍경, 간판, 차량, 건물 등이 있다. 각각 같은 대상을 다른 각과 거리와 조명에서 촬영한 여러 장의 사진을 사용하였다.

같은 대상을 촬영한 두 이미지의 특징을 SIFT알고리즘을 통해 추출한 다음 KD-Tree를 이용하여 매칭하였다.

부록에 첨부한 총 9장의 사진을 사용하였는데, 사진이 포함한 물체는 꽃, 병, 표지판, 건물, 꽃, 장난감, 풍경, 새 등이다. 매치의 개수와 연산 소요시간을 종합하여 하나의 그래프로 각각 [그림 3]과 [그림 4]로 나타냈다.

Rob Hess가 구현한 SIFT Feature Detector[6]를 실험 시뮬레이터로 이용했다.

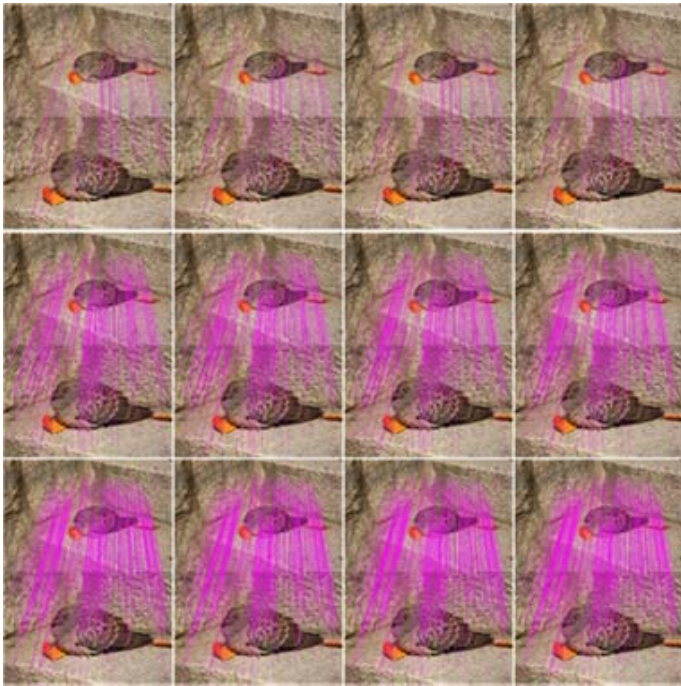
### 4.3 실험결과

매칭 결과 중 눈으로 알아보기 쉬운 이미지는 [그림 1]과 [그림 2]와 같은 결과를 보였다. 결과에서 오른쪽으로 갈수록 옥타브 수가 많고(5~8) 아래로 갈수록 인터벌이 많은(4~6) 것이다. 이를 보면 매치의 수는 옥타브 수와 약한 선형관계이고, 인터벌 수에 크게 영향을 받음을 알 수 있다.

그래서 옥타브 수와 인터벌 수에 따른 매치 수를 [표 2]와 같이 구했다. 이를 변화량에 따라 정규화하여 3차원 그래프로 나타내면 [그림 3]과 같다. 매치 수는 인터벌과 옥타브의 곱에 선형관계이다.

옥타브 수와 인터벌 수에 따른 수행시간을 [표 3]과

같이 구했다. 이를 변화량에 따라 정규화하여 3차원 그래프로 나타내면 [그림 4]와 같다. 대략적으로 매치 수와 마찬가지로 인터벌과 옥타브의 곱에 선형관계로 보이지만 그 증가 폭을 보면 인터벌5개, 옥타브6개 이후로 급격하게 증가함을 알 수 있다. 그 이상의 인터벌 수와 옥타브 수를 사용하면 PC상에서도 실시간으로 동작하기 어려울 정도의 수행시간이 걸린다. 그래서 5개의 인터벌을 사용한 매칭이면 충분히 서론에서 언급했던 모바일 환경의 실시간 작업이 가능하다.



[그림 1] 매칭결과(1)



[그림 2] 매칭결과(2)

[표 2] 인터벌과 옥타브 수에 따른 매치 수

이미지 번호	인터벌 수	옥타브 수			
		5	6	7	8
1	4	10	21	33	44
	5	61	75	89	103
	6	115	130	145	161
2	4	43	92	138	188
	5	257	336	416	492
	6	571	666	763	858
3	4	343	708	1084	1461
	5	1875	2312	2767	3223
	6	3747	4300	4871	5447
4	4	218	438	672	904
	5	1247	1612	1979	2342
	6	2765	3217	3675	4134
5	4	117	251	395	541
	5	676	817	965	1119
	6	1251	1396	1552	1709
6	4	27	55	83	112
	5	135	163	193	223
	6	263	303	345	386
7	4	32	64	96	129
	5	149	166	182	199
	6	221	248	274	297
8	4	44	93	146	199
	5	221	255	289	324
	6	348	382	422	462
9	4	52	106	160	217
	5	427	649	869	1091
	6	1389	1708	2032	2361

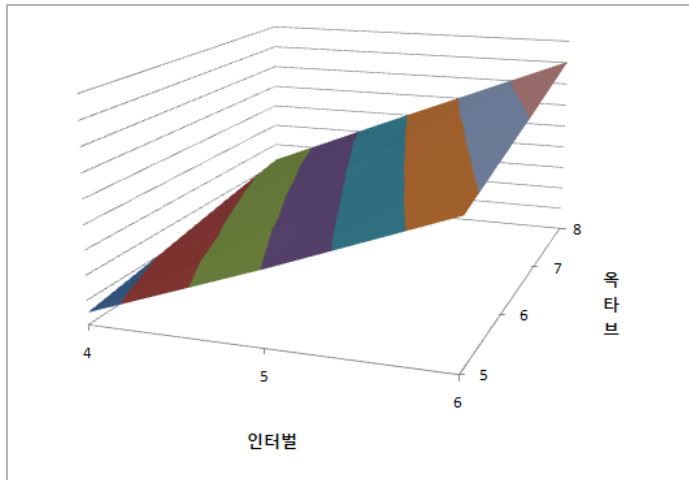
(단위 : 개)

[표 3] 인터벌과 옥타브 수에 따른 소요시간

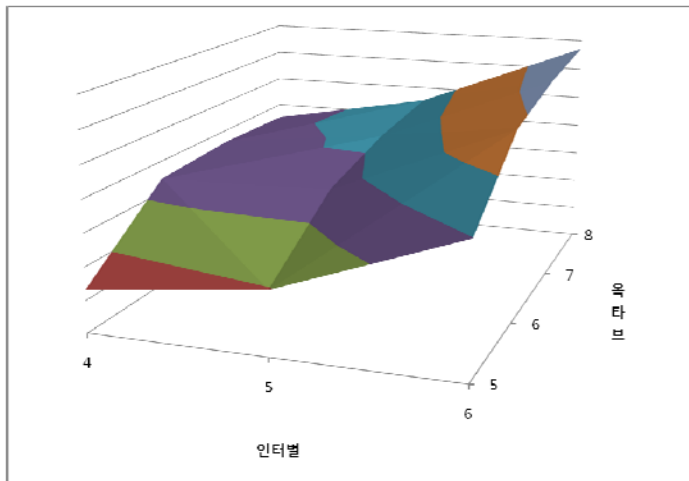
이미지 번호	인터벌 수	옥타브 수			
		5	6	7	8
1	4	4265	4687	4719	4703
	5	4391	4719	4843	4906
	6	4797	5172	5282	5343
2	4	5938	6422	6562	6609
	5	6734	7234	7485	7547
	6	7406	7984	8172	8250
3	4	5844	6250	6375	6391
	5	5735	6140	6343	6391
	6	6219	6687	6937	6968
4	4	8234	8609	8765	8750
	5	10234	10750	10891	10938
	6	12344	12984	13172	13203
5	4	5437	6000	6156	6172
	5	4687	5375	5610	5672
	6	4594	5359	5625	5688
6	4	3609	3734	3750	3766
	5	4047	4265	4297	4328
	6	4640	4860	4922	4954

7	4	4906	5172	5250	5281
	5	4875	5109	5235	5235
	6	5031	5234	5359	5375
8	4	2188	2313	2391	2406
	5	1594	1703	1765	1765
	6	1812	1937	2000	2000
9	4	8703	8906	8969	8985
	5	15469	15891	15954	15984
	6	19437	19937	20046	20079

(단위 : ms)



[그림 3] 인터벌과 옥타브 수에 따른 매치 수



[그림 4] 인터벌과 옥타브 수에 따른 소요시간

## 5. 결 론

SIFT알고리즘을 모바일 환경에서 구동시키기 위해서는 속도의 개선이 필요하다. 본 논문에서는 SIFT 알고리즘의 기본 뼈대를 형성하는 가우시안 피라미드와 가우시안 차이 피라미드의 크기를 조절하여 수행비용 대비 성능을 분석했다. 실험 결과, 5개의 인터벌과 6개의 옥타브를 사용하면 모바일 환경에서도 허용할만한 수행시간 안에 SIFT알고리즘을 충분히 사용할 수 있다.

## 6. 참고문헌

- [1] Stephen Se, David Lowe, Jim Little, Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features, IEEE International Conference on Robotics and Automation, Vol.2, pp. 2051-2058, 2006
- [2] 박종현, 이귀상, 김수형, 이명훈, Nguyen Dinh Toan, 모바일 시스템 응용을 위한 실외 한국어 간판 영상에서 텍스트 검출 및 인식, 전자공학회논문지, 46권, 2호, pp. 44~51, 2009년
- [3] David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International journal of computer vision, Vol.60, Issue 2, pp. 91-110, 2004
- [4] S. Heymann, K. Müller, A.Smolcic, B. Fröhlich, T. Wiegand, SIFT Implementation and Optimization for General-Purpose GPU, The 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 317-322, 2007
- [5] Cristina Cabani, W. James MacLean, "A Proposed Pipelined-Architecture for FPGA-Based Affine-Invariant Feature Detectors." Conference on Computer Vision and Pattern Recognition Workshop, pp. 121-121, 2006.
- [6] SIFT Feature Detector, available at <http://web.engr.oregonstate.edu/~hess/>

## 7. 부 록



이미지4



이미지5



이미지6



이미지7



이미지8



이미지9

