Journal of Korea Multimedia Society Vol. 16, No. 4, April 2013(pp. 483-492)
http://dx.doi.org/10.9717/kmms.2013.16.4.483

SURF 기반 특징점 추출 및 서술자 생성의 FPGA 구현

나은수[†], 정용진[††]

요 약

SURF는 영상의 특징점을 추출하고 서술자를 생성하는 알고리즘으로 객체인식 및 추적, 파노라마 이미지 생성 등 여러 영상처리 시스템에 응용되고 있다. SURF 알고리즘은 영상의 크기, 회전, 시점 등의 변화에 강인한 특징을 갖지만 복잡하고 반복적인 연산이 많아 실시간 처리가 어렵다. 실제 PC(Pentium, 3.3GHz) 환경에서 1000개 정도의 특징점이 추출되는 VGA(640×480) 해상도의 영상을 이용하여 실험한 결과 특징점 추출 및 서술자 생성에 총 240ms 이상이 걸려 약 4frame/sec로 실시간 처리가 불가능한 것을 확인하였다. 본 논문에서는 SURF 알고리즘의 메모리 접근 패턴을 분석하여 라인 메모리를 효율적으로 구성해 메모리 사용을 최소화하고 반복적으로 수행되는 연산을 병렬처리 하는 방법으로 하드웨어를 설계하였다. 하드웨어 설계 검증 결과 Xilinx사의 Virtex5LX330 FPGA를 타겟으로 합성 시 101,348LUTs(66%)와 1,367KB의 내부 메모리를 사용하고, 100MHz 동작 클록에서 30 frame/sec로 실시간 처리가 가능함을 볼 수 있었다.

FPGA Implementation of SURF-based Feature extraction and Descriptor generation

Eun-Soo Na[†], Yong-Jin Jeong[††]

ABSTRACT

SURF is an algorithm which extracts feature points and generates their descriptors from input images, and it is being used for many applications such as object recognition, tracking, and constructing panorama pictures. Although SURF is known to be robust to changes of scale, rotation, and view points, it is hard to implement it in real time due to its complex and repetitive computations. Using 3.3 GHz Pentium, in our experiment, it takes 240ms to extract feature points and create descriptors in a VGA image containing about 1,000 feature points, which means that software implementation cannot meet the real time requirement, especially in embedded systems. In this paper, we present a hardware architecture that can compute the SURF algorithm very fast while consuming minimum hardware resources. Two key concepts of our architecture are parallelism (for repetitive computations) and efficient line memory usage (obtained by analyzing memory access patterns). As a result of FPGA synthesis using Xilinx Virtex5LX330, it occupies 101,348 LUTs and 1,367 KB on-chip memory, giving performance of 30 frames per second at 100 MHz clock.

Key words: SURF, Feature extraction(특징점), Descriptor(서술자), Integral image(적분 영상), FPGA (하드웨어 구현)

※ 교신저자(Corresponding Author) : 정용진, 주소 : 서울특별시 노원구 월계1동 광운대학교 참빛관 810호 실시간구조 연구실(139-701), 전화 : 02) 940-5551, FAX : 02) 942-5517, E-mail : yjjeong@kw.ac.kr
접수일 : 2012년 5월 7일, 수정일 : 2012년 11월 12일
완료일 : 2013년 2월 12일
[†] 준회원, 광운대학교 전자통신공학과
(E-mail : kiss_medarling@kw.ac.kr)
[††] 준회원, 광운대학교 전자통신공학과
※ 본 연구는 2012년 정부(교육과학기술부)의 재원으로 한국연구재단(NRF-2010-0014557)과 지식경제부가 지원하는 산업융합원천기술개발사업(10039188, 스마트 자동차용 프로그래머블 융복합 멀티미디어 SoC 플랫폼 개발)을 통해 개발된 결과임을 밝힙니다.

1. 서 론

영상처리 분야에서 영상인식이나 분석을 위해 영상의 고유 특징점(feature point)을 추출하고 추출된 특징점을 표현하는 서술자(descriptor)를 생성하여 다른 영상과 비교하는 방법을 많이 이용한다. 이와 같이 특징점과 서술자는 서로 다른 영상에서 공통되는 영역을 찾는 방법 중에 하나로 객체인식 및 추적, 파노라마 이미지 생성 그리고 3차원 영상복원과 같은 응용프로그램에 많이 활용되어 지고 있다. 영상에서 특징점을 추출하고 서술자를 생성하는 방법은 여러 논문들에서 소개되었다. 대표적인 것으로 SIFT (Scale Invariant Feature Transform)[1], MSER (Maximally Stable Extremal Regions)[2] 그리고 SURF(Speeded Up Robust Feature)[3] 등이 있다.

SIFT와 SURF 두 알고리즘은 모두 흑백 영상을 이용하여 크기, 회전, 시점 등의 변화에 불변하는 특징점과 서술자를 생성하는 공통적인 특성을 갖고 있지만 연산처리 속도 면에서는 SURF가 더욱 빠른 장점을 갖고 있다. 연산 속도가 빠르기 때문에 하드웨어 자원이 제한적인 임베디드 환경에서 개발하는 응용프로그램에는 SURF 알고리즘을 적용하는 것이 용이하다. 하지만 SURF 알고리즘을 PC(Pentium, 3.3GHz) 환경에서 1000개 정도의 특징점이 추출되는 VGA(640×480) 해상도의 영상으로 실험한 결과 특징점 추출 및 서술자 생성에 총 240ms(특징점 추출 30ms, 서술자 생성 210ms) 이상의 시간이 소요됨으로써 약 4 frame/sec로 실시간 처리가 불가능하였다. 그리고 임베디드 환경인 ARM11(806MHz) 프로세서를 이용하여 실험한 결과 총 4,314ms로 초당 1 프레임조차 처리하지 못하였다. 따라서 SURF 알고리즘의 실시간 처리를 위해서는 하드웨어 구현이 필수적이다.

본 논문에서는 기존의 SURF 기반 특징점 추출기의 하드웨어 설계 연구결과인 논문[4]를 이용하여 특징점 추출기를 구성하고 추가적으로 서술자 생성기 설계 및 전체를 통합하여 실시간 처리가 가능한 SURF 알고리즘의 하드웨어 구조를 제안하고 검증한다. 논문의 구성은 2장에서는 SURF 알고리즘에 대해 설명하고, 3장 에서는 하드웨어 설계, 4장은 결과 및 분석, 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 알고리즘

SURF 알고리즘은 크기, 회전, 시점과 같은 영상의 변화에도 불변하는 특징점과 서술자를 생성하는 특성을 갖는 알고리즘이다. SURF 알고리즘은 그림 1과 같이 적분 영상 생성, 특징점 추출, 서술자 생성 이렇게 3단계로 구분된다. 세부적으로 특징점 추출은 고속 헤이시안 검출기(Fast-Hessian Detector)[5]와 비 최대치 억제(Non Maxima Suppression)[6]로 구성 되고, 서술자 생성은 특징점 주방향 검출, 영상회전 및 크기 변환과 서술자 벡터 추출로 구성되어 진다.

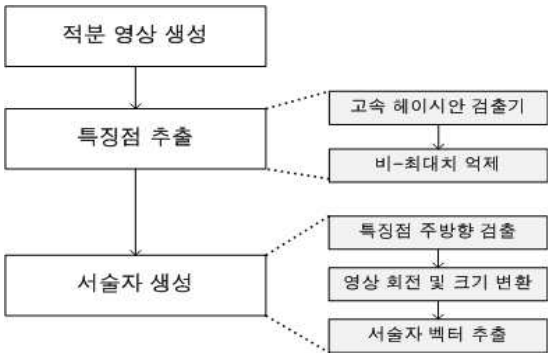


그림 1. SURF 알고리즘 구성도

2.1 적분 영상 생성

적분 영상은 “Summed-area tables for texture mapping”[7]에서 소개 되었고 입력 영상의 화소 값을 누적하여 생성한다. 적분 영상을 이용하여 박스 모양의 필터를 적용 시에 필터의 크기에 관계없이 일정 연산만으로 필터 내의 화소 값의 합을 얻을 수 있다. SURF 알고리즘에서 많이 사용하는 박스 모양의 필터 연산의 효율성을 위해 적분 영상을 사용하게 된다. 아래 식 (1)은 적분 영상 생성 수식으로 I은 원본 영상인 8비트 흑백 영상이고, II는 적분 영상 화소 값이다.

$$II(x,y)=\sum_{i=0}^{x}\sum_{j=0}^{y}I(i,j) \tag{1}$$

그림 2는 적분 영상을 이용하여 특정 사각형 영역 내의 화소 값의 합을 구하는 방법을 표현하고 있다. 최종적으로 식 (2)과 같이 사각형의 영역의 크기와 관계없이 두 번의 뺄셈 연산과 한 번의 덧셈 연산만으로 특정 사각형 내의 화소 값의 합을 구할 수 있게 된다.

$$Sum\,of\,pixel\,values = D + A - B - C \quad (2)$$

2.2 특징점 추출

SURF는 코너와 Blob과 같은 부분을 특징으로 추출하는 고속 헤이시안 검출기를 이용하고 크기 변화에 강인한 특징점을 추출하기 위해 스케일 공간(Scale space)[8]을 적용한다. 그리고 마지막으로 비 최대치 억제를 통해 스케일별로 특징점을 추출한다.

2.2.1 고속 헤이시안 검출기

고속 헤이시안 검출기는 그림 3과 같이 σ의 분산을 취하는 가우시안 2차 미분 필터를 근사화한 박스필터를 적용하여 컨볼루션 연산을 취한다. 박스 필터 연산을 효율적으로 하기 위해 적분 영상을 이용하게 된다.

$$H(x,y,\sigma) = \begin{bmatrix} Dxx(x,y,\sigma) & Dxy(x,y,\sigma) \\ Dxy(x,y,\sigma) & Dyy(x,y,\sigma) \end{bmatrix} \quad (3)$$

박스필터를 적용하여 위의 식 (3)과 같이 헤이시안 행렬을 구한다. 이때 Dxx, Dyy, Dxy는 각각 x, y, xy방향으로의 박스필터 연산 결과이다. 최종적으로 식 (4)와 같이 헤이시안 행렬식을 구할 수 있다.

$$determinant(H) = DxxDyy - Dxy^2 \quad (4)$$

2.2.2 비 최대치 억제

SURF는 크기 변화에 강인한 특징점을 얻기 위해 그림 4와 같이 박스 필터의 크기를 일정 간격으로 변화 시키며 스케일별로 헤이시안 행렬식을 구한다.

SURF 알고리즘은 위와 같이 생성된 스케일 공간에서 그림 5와 같이 × 표시된 지점의 헤이시안 행렬식 값이 임계값 보다 크고 인접 3×3×3 영역 각각의 헤이시안 행렬식 값보다 큰 경우에만 특징점으로 판단하는 비 최대치 억제 방법을 통하여 특징점을 추출한다.

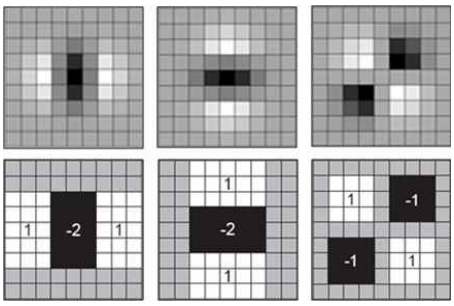


그림 3. 가우시안 2차 미분필터와 근사화한 박스필터

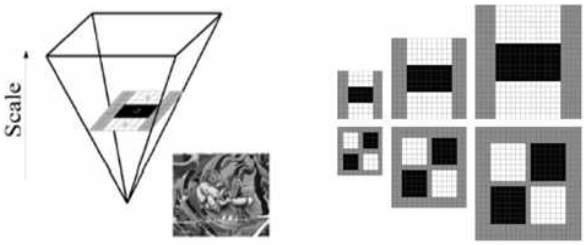


그림 4. 스케일 공간(좌), 박스필터 크기 변화(우)

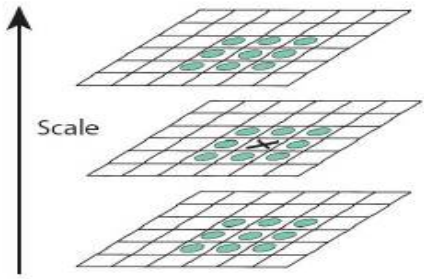


그림 5. 비 최대치 억제

2.3 서술자 생성

SURF 알고리즘의 서술자는 특징점 주변 영상을 이용하여 부분 영상을 구성하고, 부분 영상에서 화소값의 변화를 이용하여 총 64차원의 벡터를 구성한다.

2.3.1 특징점 주방향 검출

서술자 벡터를 추출하기에 앞서 회전 변화에 강한 특성을 갖기 위해 특징점의 주방향을 검출한다. 주방향을 검출하기 위해 그림 6의 좌측과 같은 하알 웨이블릿(Haar-wavelet) 필터를 적용하여 x축과 y축 방향의 응답 d_x와 d_y을 구하고 식 (5)과 같이 아크탄젠트 함수를 이용하여 기울기 G_{haar}을 구한다. 이렇게 구해진 d_x, d_y, G_{haar}을 이용해 그림 6의 우측과 같은 크기와 방향을 가지는 벡터 공간을 생성한다. 벡터

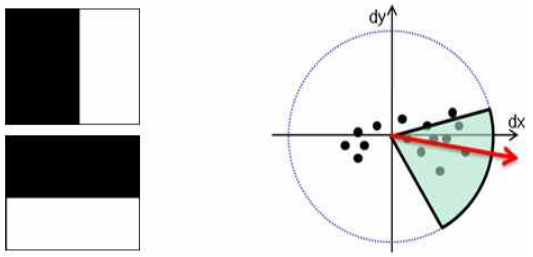


그림 6. 하알 웨이블릿 필터 (좌), 하알 웨이블릿 응답 결과 벡터 공간 (우)

공간에서 60도 윈도우를 5도씩 이동하면서 윈도우내에 벡터 성분들의 합을 구하고 그 크기를 비교하여 가장 큰 윈도위의 방향을 주방향으로 결정한다.

$$G_{haar} = atan\left(\frac{d_y}{d_x}\right) \quad (5)$$

2.3.2 영상 회전 및 크기 변환

영상의 회전이나 크기에 변화가 생겨도 서술자는 변하지 않고 고유의 값을 얻기 위해 특징점의 주방향과 스케일 정보를 이용하여 특징점 주변의 영상을 회전하고 크기를 변화시킨 부분 영상을 구성한다.

그림 7과 같이 특징점 중심의 위치에서 주방향을 기준으로 스케일 정보에 따라 일정 크기 영역의 영상을 수직이 되도록 회전 시키고 회전된 영상을 20×20 픽셀 크기로 변환시켜 서술자 벡터를 추출할 부분 영상을 구성한다.

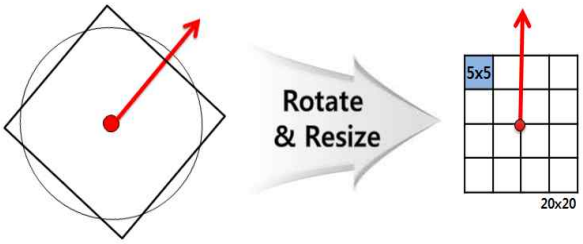


그림 7. 영상 회전 및 크기 변환

2.3.3 서술자 벡터 추출

영상 회전 및 크기 변환을 통해 구성된 부분 영상을 그림 8과 같이 16개 영역으로 분할하여 각 영역마다 하알 웨이블릿 응답을 계산하여 $\sum dx$, $\sum |dx|$, $\sum dy$, $\sum |dy|$ 을 추출한다. 이렇게 4개의 벡터를 16개 영역에서 추출하여 총 64차원의 서술자를 생성하게 된다. 마지막으로 외부의 빛이나 조명과 같은 주위 환경에 영향을 줄이기 위한 방법으로 각 영역별로 서술자 벡터를 Unit vector로 정규화 하여 저장한다.

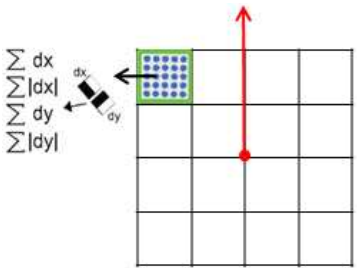


그림 8. 하알 웨이블릿 응답을 이용한 서술자 벡터 추출

3, 하드웨어 설계

본 논문에서는 SURF 기반 특징점 추출과 서술자 생성을 하드웨어로 설계하기 위해 고정소수점 모델의 구현과 성능 분석[9,10]을 통하여 데이터의 크기를 정하고 하드웨어 구조를 설계하였다. 특징점 추출 및 서술자 생성기의 하드웨어는 그림 9와 같이 입력 영상으로부터 특징점을 추출하고 추출된 특징점의 서술자를 생성하여 저장하는 단계로 구성된다. 이번 장에서는 SURF 알고리즘의 효율적인 하드웨어 설계 방법과 하드웨어 내부 모듈의 동작에 대해서 설명한다.

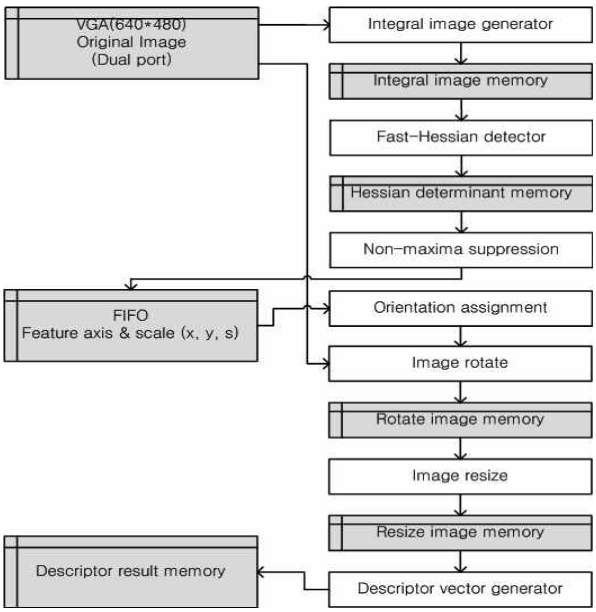


그림 9. 특징점 추출 및 서술자 생성기 하드웨어 흐름도

3.1 효율적인 하드웨어 설계

하드웨어의 효율적인 설계에 있어 중요한 요소로 연산기의 병렬처리와 메모리의 효율적 사용 등이 있다. SURF 알고리즘의 효율적인 하드웨어 설계를 위해 적분 영상의 메모리 사용 패턴을 분석하여 최소 크기의 메모리를 구성하고 라인 메모리를 이용해 메모리 접근 횟수를 줄일 수 있도록 설계하였다. 그리고 박스 필터 연산이나 하알 웨이블릿 응답 연산과 같이 반복적으로 수행되는 연산은 병렬처리를 통해 연산 시간을 최소화하여 실시간 처리가 가능하도록

설계하였다.

3.3.1 메모리 사용량 최적화

SURF 알고리즘의 효율적인 처리를 위해서는 적분 영상의 활용은 필수적이다. VGA 해상도(8비트, Gray scale) 영상 전체에 대한 적분 영상은 한 픽셀을 27비트로 표현해야 전체 적분 영상을 모두 나타 낼 수 있다. 그러므로 VGA 해상도 영상의 적분 영상 크기는 약 1MB ($640\times480\times27bit$)의 크기로 전체를 내부 메모리에 저장하여 사용하기에는 무리가 있다. 이와 같은 문제를 해결하기 위해 각 스케일 공간에서 메모리 접근 패턴을 분석하여 최소로 필요한 크기의 적분 영상을 구성하도록 하였다.

고속 헤이시안 검출기에서 최소로 필요한 적분 영상 메모리 크기는 각 스케일 정보에 따라서 표 1과 같이 서로 다르다. 그리고 주방향 검출기에서 사용되는 적분 영상 메모리 최대 크기는 표 2와 같다. 고속 헤이시안 검출기에서는 스케일별로 박스 필터 크기와 샘플링 간격에 따라서 적분 영상의 최소 크기가 결정되어 지고 주방향 검출기에서는 추출된 특징점의 스케일 정보에 관계없이 최대로 필요한 적분 영상의 크기가 결정된다. 위와 같은 분석 결과를 통해서 동작에 필요한 최소 메모리의 크기를 결정하여 설계하였다. 결과적으로 1MB의 적분 영상 메모리 사용을 고속 헤이시안 검출기 연산에 142KB($640\times65\times27bit$)와 주방향 검출기 연산에 123KB($640\times57\times27bit$)의 메모리를 사용하여 총 265KB로 구현함으로써 약 73.5%의 메모리 사용량 감소효과를 얻을 수 있었다.

표 1. 고속 헤이시안 검출기 적분 영상 메모리 사용량

구 분	박스 필터 크기 ($row\times column$)	패턴 분석 결과 적분 영상 메모리 크기 ($width\times depth\times databits$)
스케일 0	7×7	640×9×27
스케일 1	9×9	640×11×27
스케일 2	15×15	640×17×27
스케일 3	21×21	458×17×27
스케일 4	27×27	214×11×27

표 2. 주방향 검출기 적분 영상 메모리 사용량

구 분	패턴 분석 결과 적분 영상 메모리 크기 ($width\times depth\times databits$)
스케일 통합	640×57×27

그리고 적분 영상을 저장한 메모리에서 한 개의 버퍼 메모리를 추가적으로 구성하여 그림 10과 같이 버퍼 메모리는 적분 영상을 저장하는 쓰기 동작을 수행하고 나머지는 적분 영상 데이터를 읽어가는 읽기 동작을 수행한다. 결과적으로 다음 연산에 필요한 메모리를 미리 준비함으로써 메모리 읽기 동작에서 병목현상이 발생하지 않도록 설계하였다.

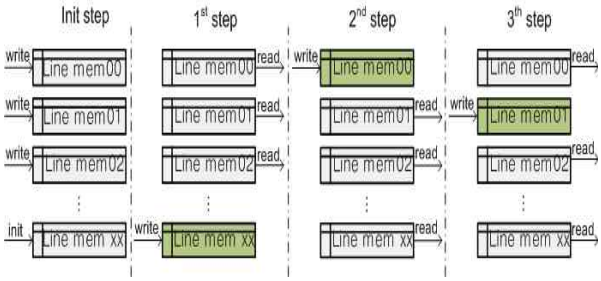
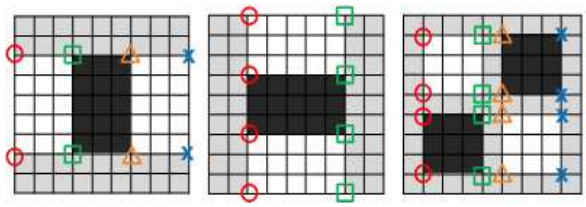


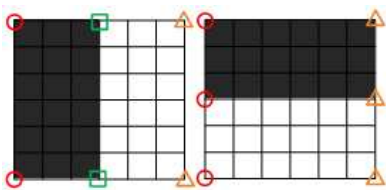
그림 10. 라인 버퍼 메모리를 이용한 적분 영상 읽기 및 쓰기 동작

3.3.2 메모리 접근의 효율적인 방법

고속 헤이시안 검출기에 사용되는 박스 필터와 특징점 주방향 검출 단계에서 사용되는 하알 웨이블릿 필터 연산에는 적분 영상 메모리 접근이 많이 이루어진다. 그림 11에서 보이듯이 박스 필터는 최소 16번, 하알 웨이블릿 필터는 6번 메모리 접근이 필요하게 된다. 이때 메모리의 접근을 효율적으로 하기 위해서 그림 11(a)에서 박스 필터와 그림 11(b)의 하알 웨이블릿 필터의 메모리 접근 패턴을 분석하였다.



(a) 박스 필터 메모리 접근 패턴



(b) 하알 웨이블릿 필터 메모리 접근 패턴

그림 11. 필터 연산의 적분 영상 접근 패턴 [4].

분석 결과 그림 11에서 같은 모양으로 표시된 부분은 동시에 읽을 수 있도록 하기 위한 방법으로 그림 12와 같이 라인 메모리를 구성하였다. 적분 영상을 행 단위로 구분하여 라인 메모리 형태로 저장해 동일한 주소(address)에 해당하는 적분 영상 값을 모든 라인에서 동시에 읽을 수 있게 되면서 박스 필터는 최대 4번, 그리고 하알 웨이블릿 필터는 최대 3번의 적분 영상 메모리 접근으로 연산을 수행할 수 있도록 설계하였다. 결과적으로 박스 필터 연산과 하알 웨이블릿 필터 연산에서 메모리 접근 횟수는 각각 75%, 50% 줄어드는 효과를 얻을 수 있었다.

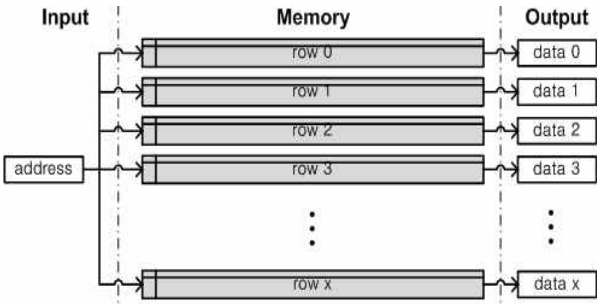


그림 12. 라인 메모리 구조 [4].

3.3.3 병렬 처리 연산 구조

SURF 특징점 추출기는 스케일 공간을 적용하여 연산 한다. 이때 각각의 스케일 공간에서 특징점을 추출하기 위한 연산을 순차적으로 처리하게 되면 연산시간이 오래 걸리게 된다. 이러한 점을 극복하기 위해 그림 13과 같이 고속 헤이시안 검출기와 비 최대치 억제 연산을 스케일별로 병렬 처리 하여 통해 실시간 동작이 가능한 구조로 설계하였다.

서술자 생성기에서는 최종적으로 서술자 벡터를 추출하기 위해 부분영상을 16개 영역으로 나누어 각각에 하알 웨이블릿 연산을 통하여 서술자를 생성한다. 이때 16개 영역을 순차적으로 처리하지 않고 그림 14와 같이 하알 웨이블릿 연산기를 병렬로 두어

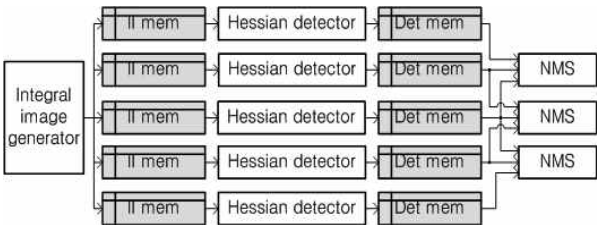


그림 13. 병렬 처리 연산 구조 [4].

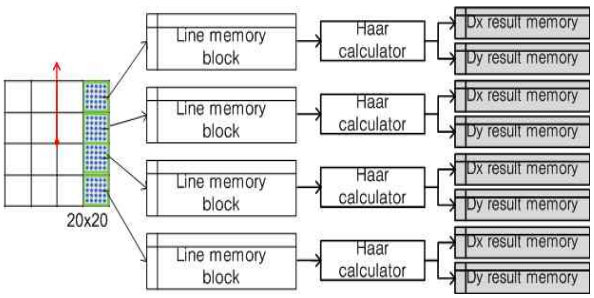


그림 14. 병렬 처리 연산 구조

4개 영역씩 동시에 처리가 가능 하도록 하였다.

3.2 하드웨어 내부 구조

SURF 알고리즘의 하드웨어 구조는 그림 15와 같이 적분 영상 생성, 특징점 추출 그리고 서술자 생성으로 구성되어 지고 세부적으로 고속 헤이시안 검출, 비 최대치 억제, 주방향 검출, 영상 회전, 크기 변환 그리고 서술자 벡터 추출 이렇게 7개의 주요 모듈을 중심으로 구성되어있다.

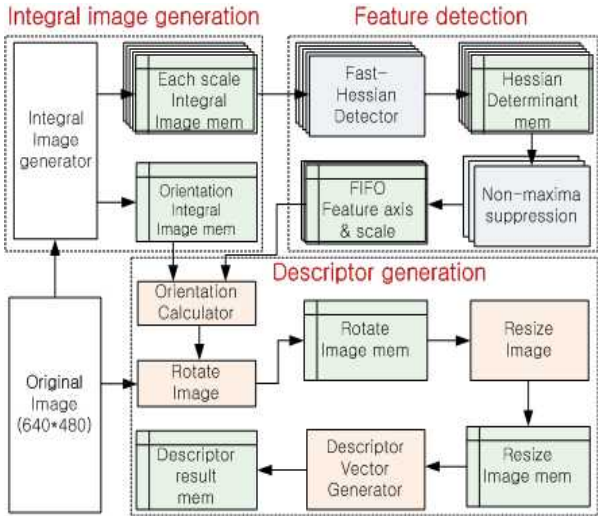


그림 15. SURF 기반 특징점 추출 및 서술자 생성기 하드웨어 내부 구조

3.2.1 적분 영상 생성 모듈

적분 영상 생성 모듈의 연산 구조는 입력 영상 메모리로부터 8비트 흑백 영상 데이터를 먼저 가로줄로 합산하여 적분 영상 메모리에 저장하고 다시 이전 라인의 적분 영상 메모리의 픽셀을 읽어 합하여 출력한다. 이때 영상 가장자리는 픽셀의 값이 없으므로 0을 입력하도록 하였다.

3.2.2 특징점 추출 모듈

특징점 추출은 고속 헤이시안 검출 모듈과 비 최대치 억제 모듈로 구성하였다. 고속 헤이시안 검출 모듈은 적분 영상과 박스 필터를 이용해 헤이시안 행렬식 값을 구하고, 스케일 별로 구해진 행렬식 값을 비 최대치 억제 방법에 적용해 최종적으로 특징점을 추출한다. 추출된 특징점 정보는 FIFO에 저장하여 서술자 생성 모듈에서 특징점 좌표와 스케일 정보를 읽어갈 수 있도록 설계하였다. 특징점 추출 모듈에서는 연산 시간을 감소시키기 위해 고속 헤이시안 검출 모듈과 비 최대치 억제 모듈 모두 병렬로 처리하도록 하였다.

3.2.3 주방향 검출 모듈

특징점의 주방향을 검출하기 위해 특징점 좌표와 스케일 정보 그리고 특징점 주변 영상을 이용한다. 주방향 검출 모듈의 내부 구조는 하알 웨이블릿 응답 모듈, 아크탄젠트 연산 모듈, 슬라이딩 윈도우 그리고 사인 및 코사인 연산 모듈로 구성되어져있다. 특징적으로는 고정 소수점 모델링을 통해 미리 최적의 삼각함수 롬 테이블을 만들어 연산에 이용하였다.

3.2.4 영상 회전 및 크기 변환 모듈

영상 회전 및 크기 변환은 특징점의 주방향을 이용하여 서술자 벡터를 추출할 부분 영상을 얻는 모듈이다. 먼저 특징점 주변 영상을 주방향이 수직이 되도록 회전하고 20×20 픽셀 크기로 크기 변환을 시켜 부분 영상을 저장한다. 회전된 영상을 저장하는 메모리와 크기 변환된 영상을 저장하는 메모리는 모두 라인 메모리를 사용하여 동시에 여러 개의 데이터를 쓰고 읽을 수 있도록 구성하였다.

32.5 서술자 벡터 추출 모듈

서술자 벡터 추출 모듈은 회전 및 크기 변환된 부분 영상을 이용하여 서술자 벡터를 추출한다. 이때 연산 처리의 효율을 위해 부분 영상이 저장된 메모리는 라인 메모리로 구성하여 하알 웨이블릿 응답 모듈과 정규화 연산 모듈 모두 병렬 처리가 가능하도록 설계하였다. 빛과 조명등의 민감한 영상의 특성상 하알 웨이블릿 응답 결과를 그대로 서술자 벡터로 사용하지 않고 각 영역별 정규화 과정을 통해 최종 서술자 정보를 저장하도록 하였다.

4. 결과 및 분석

하드웨어 설계에 대한 검증환경은 그림 16과 같이 삼성의 Cortex-A8 Core(1GHz)를 탑재한 마이크로비전의 MV210 보드와 Xilinx의 Virtex5LX330 FPGA를 탑재한 EDA-Mod 5VLX330 보드를 EBI (External Bus Interface)를 이용하여 연결하였고 주변 장치로 로지텍의 USB2.0 인터페이스의 웹캠 그리고 7인치 LCD를 사용하였다. Xilinx ISE 12.4 환경에서 Verilog HDL을 이용하여 RTL(Register Transfer Level)로 설계하고 Mentor사의 Modelsim 6.5d를 이용하여 기능 및 타이밍 시뮬레이션을 검증한 결과 소프트웨어를 이용한 고정 소수점 모델링 결과와 시뮬레이션 결과가 동일한 것을 확인하였다. 최종적으로 FPGA를 타겟으로 합성 결과 101,348LUTs(66%)와 1,367KB의 내부 메모리를 사용하였다. 최대 동작 클록은 128MHz까지 가능하지만 실제 FPGA 동작에는 100MHz 클록을 적용하였다.

그림 17은 전체 SURF 하드웨어의 타이밍 분석도이다. 특징점 추출기는 적분영상 생성, 고속 헤이시안 검출기, 비 최대치 억제 3개의 모듈이 파이프라인 구조로 동작되도록 설계하였고 서술자 생성기의 주방향 검출, 영상회전, 크기변환, 서술자 벡터 추출 모듈은 순차적으로 처리 되도록 설계하였다.

특징점 추출기는 전처리 단계로 최소 크기의 적분영상 생성에 32,640 클록이 소요되고 특징점 추출에는 1,536,000 클록이 소요된다. 그리고 서술자 생성기는 특징점 한 개의 서술자 벡터를 추출하는데 1,734 클록이 소요되고 1000개의 특징점의 서술자를 생성하기 위해서는 1,734,000 클록이 소요된다.

1000개의 특징점이 추출되는 VGA 해상도의 입력 영상을 이용한 SURF 하드웨어는 총 3,302,640 클록이 소요된다. 그러므로 100MHz 동작 클록에서 특징점 추출에는 15.7ms가 소요되고 서술자를 생성하는데 약 17.4ms가 걸리므로 총 33.1ms 정도가 걸리며

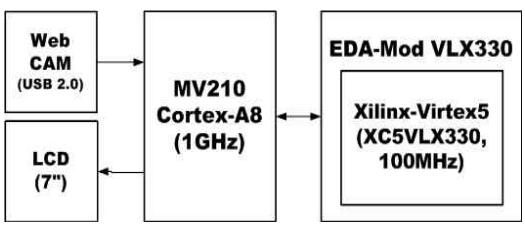


그림 16. 검증 환경

이를 초당 프레임으로 계산하면 약 30 frame/sec로 동작이 가능하였다. 단, 이때 특징점을 1000개 정도로 예상하여 계산한 이유는 검증 환경에서 구성된 VGA 해상도의 웹캠을 이용하였을 때 특징점의 개수가 최대 1000개를 넘는 경우가 거의 없었기 때문이다.

구현된 하드웨어는 입력 영상, 적분 영상, 특징점 및 서술자 정보를 저장하기 위해 약 1,367KB의 SRAM을 사용한다. 메모리 사용 세부 내용은 표 3을 통해서 확인할 수 있듯이 입력 영상과 적분 영상에서 많은 메모리를 사용하고 있다. 특히 입력 영상의 경우는 특징점 추출과 서술자 생성 두 단계에서 모두 사용되는 것을 고려하여 듀얼 포트 메모리를 사용하였다.

표 4에 PC와 임베디드 환경에서 SURF 알고리즘을 소프트웨어로 구현하여 실험한 결과와 본 논문에서 기술한 하드웨어로 구현된 결과를 비교하였다. 수행시간을 보면 하드웨어 구현 결과 실시간 동작이 가능한 것을 확인할 수 있다.

본 논문에서 구현한 특징점 추출 및 서술자 생성기 하드웨어와 비슷한 사례로 [11]과 [12]이 있으며, 표 5는 [11]과 [12]를 본 논문에서 설계한 하드웨어와 비교한 결과이다. [11]은 SURF 특징점 추출기를 FPGA에 구현한 논문이고, [12]은 SURF 기반 특징

표 3. 세부 메모리 사용량

구 분	메모리 사용량 (Byte)
입력 영상 메모리(듀얼)	614,400
적분 영상 메모리	281,200
행렬식 저장 메모리	194,000
특징점 저장 메모리	8,192
서술자 저장 메모리	262,144
기 타	6,864
Total	1,366,800

표 4. 연산시간 비교

구 분	PC	임베디드	FPGA (본 논문)
Device	Pentium i5	Marvell PXA320	Virtex5LX330
동작클록	3.3 GHz	806 MHz	100 MHz
메모리	DDR3 2 GB	M-DDR 128 MB	1,367 KB (SRAM)
수행시간	240 ms	4,314 ms	33 ms

표 5. 성능 비교 분석

구 분		[11]	[12]	본 논문
입력영상		1024×768	640×480	640×480
Device		Xilinx Spartan6	Xilinx Virtex5 XC5VFX130T	Xilinx Virtex5 LX330
동작클록		200 MHz	200 MHz	100 MHz
수행시간	특징점 추출	100 ms	7.55 ms	15.7 ms
	주방향 검출	×	10.45 ms	17.4 ms
	서술자 생성	×	×	

점 추출과 특징점 주방향 검출까지 FPGA로 구현한 논문이다. 입력 영상, 동작 클록 그리고 테스트 영상에 따라서 연산시간에 차이가 크게 나타나고 비교 논문들은 서술자 벡터 추출 연산을 수행하지 않으므로 정확한 비교가 어렵다. 하지만 본 논문에서는 서술자 생성기까지 구현하여 실시간 처리가 가능함을 보이고 있다.

구현된 하드웨어의 응용프로그램 적용을 위해 그림 18과 같이 객체인식시스템을 구축하였다. 시스템의 동작은 웹캠으로부터 입력된 영상에 대해서 특징점을 추출하고 서술자를 생성한다. 그리고 미리 저장된 3개의 데이터베이스 영상과 서술자 비교를 통하여 데이터베이스에 저장된 객체가 있는지 판별한다.

인식 결과는 소켓 통신을 통하여 PC로 전송하고 UI 프로그램을 통하여 인식된 객체를 출력한다. 또한 실시간으로 LCD 화면에 인식된 객체의 위치를 사각형으로 표시하도록 하였다.

시스템의 전체 동작 중에서 본 논문에서 제안한 하드웨어의 연산 시간만을 측정해보았다. 측정 방법은 일반 사무실 환경에서 VGA 해상도의 웹캠을 사용하여 연속적으로 1000장의 영상을 입력하고 입력된 시간부터 연산이 끝나는 시간을 연속적으로 확인하여 평균을 내었다. 측정 결과 1프레임의 영상을 처리하는데 약 35.6ms가 소요되어 약 28frame/sec로 동작하는 것을 확인하였다. 이를 통해서 객체 인식 시스템에 충분히 적용되어질 수 있음을 확인할 수 있었다. 그리고 100여장 테스트 영상을 이용하여 하드웨어 시뮬레이션에서 처리한 결과와 FPGA에서 동작시킨 결과가 동일한지를 비교하여 실제 FPGA

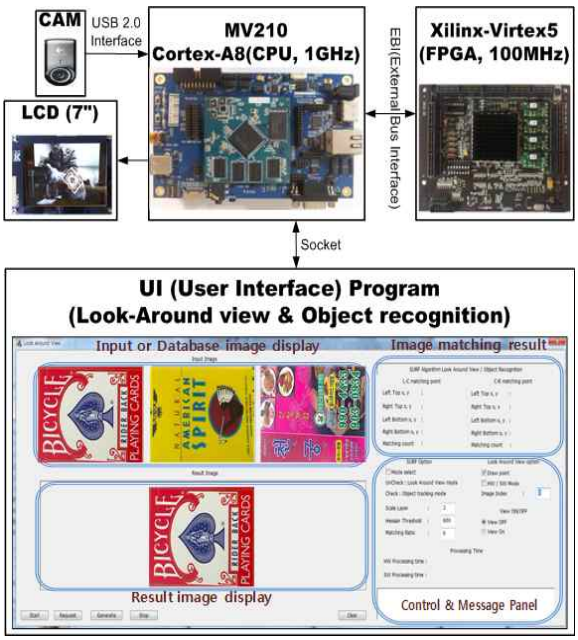


그림 18. SURF 하드웨어를 이용한 객체인식시스템

환경에서도 정상적으로 동작하는 것을 확인하였다.

5. 결론 및 향후 연구방향

SURF 알고리즘을 임베디드 환경에서 실시간으로 구동하기 위해서는 하드웨어 구현이 필수적이다. 본 논문에서는 SURF 알고리즘 기반 특징점 추출 및 서술자 생성기의 실시간 처리를 위해 하드웨어로 구현하고 FPGA를 이용하여 검증 및 분석하였다.

특징점 추출기의 연산의 실시간 처리를 위해 각 스케일 공간은 병렬로 처리하여 연산 시간을 최소화하였고 효율적인 하드웨어 설계를 위해 메모리 접근 횟수를 줄이는 라인 메모리 구성, 그리고 적분 영상 접근 패턴을 분석하여 최소 크기의 적분 영상을 메모리에 저장하여 사용하도록 하였다. 또한 삼각함수와 같이 복잡한 연산을 미리 저장해둔 롬 테이블을 이용하여 쉽게 해결할 수 있었다.

VGA 해상도의 웹캠 영상을 이용하고 본 논문에서 설명한 하드웨어 구조를 사용하여 SURF 기반 특징점 추출 및 서술자 생성기를 FPGA에 구현한 결과 특징점이 1000개 정도 추출되는 영상은 100MHz의 동작 속도에서 33ms 정도 시간이 걸려 30 frame/sec로 실시간 처리가 가능한 것을 확인하였다.

하지만 본 논문에서 제안한 하드웨어에서 사용되는 1,367KB의 내부메모리는 하드웨어 자원이 제한적인 임베디드 환경에서 사용하기에 크다는 단점을 갖고 있다. 향후 연구에는 이러한 단점을 극복하기 위한 메모리 사용량 감소에 노력할 필요성이 있다. 또한 연산 처리의 효율과 속도 향상을 위해 특징점 추출, 서술자 생성 그리고 특징점 정합 3단계를 파이프라인 처리가 가능하도록 설계할 필요성이 있다.

최종적으로 본 논문에서 설계한 SURF 기반 특징점 추출 및 서술자 생성기는 임베디드 환경에서 객체인식 및 추적 그리고 파노라마 이미지 생성과 같은 응용프로그램의 실시간 처리를 위해 많이 활용 되어질 것으로 기대된다.

참 고 문 헌

[1] D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints, Cascade Filtering Approach,” *IJCV,* Vol. 60, No. 2, pp. 91-110, 2004.

[2] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions,” *BMVC*, pp. 384-396, 2002.

[3] H. Bay, T. Tuytelaars and L. Van Gool. “Surf: Speeded up robust features,” *CVIU*, Vol. 110, No. 3, pp. 346-359, 2006.

[4] 류재경, 이수현, 정용진 “SURF 알고리즘 기반 특징점 추출기의 FPGA 설계,” 멀티미디어학회 논문지, 제14권, 제3호, pp. 368-377, 2011.

[5] K. Mikolajczyk and C. Schmid, “An Affine Invariant Interest Point Detector,” *ICCV,* Vol. 1, pp. 525-531, 2001.

[6] Neubeck. A, “Efficient Non-Maxium Suppression,” *ICPR,* Vol. 3, pp. 850-855, 2006.

[7] Crow. F, “Summed-Area Tables for Texture Mappint,” *Proc. of the 11th Annual Confrence on Computer Graphics and Interactive Techniques*, pp. 207-212, 1984.

[8] Tony. Lindeberg, “Feature Detection with Automatic Scale Selection,” *IJCV,* Vol. 30, No. 2, pp. 79-116, 1998.

[9] 나은수, 강철호, 정용진 “SoC 하드웨어 설계를 위한 SURF 알고리즘의 고정 소수점 모델 구현

및 성능분석," 대한전자공학회 추계학술대회, pp. 249-252, 2011.

[10] Krystian. Mikolajczyk and Cordelia. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615-1630, 2005.

[11] Jan. Svab, Tomas. Krajnnik, Jan. Faigl, and Libor Preucil. "FPGA Based Speeded Up Robust Features," *IEEE International Conference on TePRA*, pp. 35-41, 2009.

[12] Bouris. D, Nikitakis. A, and Papaefstathiou. I, "Fast and Efficient FPGA-based Feature Detection Employing the SURF Algorithm," *FCCM*, pp. 3-10, 2010.

나 은 수

2012년 광운대학교 전자통신공학과 학사 졸업
2012년 광운대학교 전자통신공학과 석사 졸업
관심분야 : SoC 설계, 영상처리 및 인식, 임베디드 시스템 설계

정 용 진

1983년 서울대학교 제어계측 공학과 학사 졸업.
1983년 3월~1989년 8월 : 한국전자통신연구원(ETRI).
1995년 미국 UMASS 전자전산공학과 박사 졸업.
1995년 4월~1999년 2월 삼성전자 반도체 수석 연구원.
1999년 3월~현재 광운대학교 전자통신공학과 교수.
관심분야 : 무선통신, 정보보호, SoC 설계, 영상처리 및 인식, 임베디드 시스템