논문 2015-10-07

SURF알고리듬에서의 고속 특징점 검출 방식

(A Fast Interest Point Detection Method in SURF Algorithm)

황 인 소, 엄 일 규, 문 용 호, 하 석 운*
(In-So Hwang, Il-Kyu Eom, Yong-Ho Moon, Seok-Wun Ha)

Abstract : In this paper, we propose a fast interest point detection method using SURF algorithm. Since the SURF algorithm needs a great computations to detect the interest points and obtain the corresponding descriptors, it is not suitable for real-time based applications. In order to overcome this problem, the interest point detection step is parallelized by OpenMP and SIMD based on analysis of the scale space representation process and localization one in the step. The simulation results demonstrate that processing speed is enhanced about 55% by applying the proposed method.

Keywords : SURF, Interest point, Parallel processing, OpenMP, SIMD

Ⅰ. 서 론

CPU 클록의 동작 속도를 고속화하여 연산 처리 성능을 개선하고 자 하던 시기에 소프트웨어를 통한 최적화 및 가속화는 크게 관심을 받지 못하는 연구 영역중의 하나였다. 그런데 고속 발전해 오던 CPU가 발열, 파워 소모 등의 문제로 성장의 한계에 다다르게 되면서 이를 극복하기 위해 CPU의 멀티코어(Multi-Core)화가 대두되었다. 더불어 이 같은 하드웨어의 변화에 발맞춰 소프트웨어에서도 각각의 코어들을 충분히 활용할 수 있도록 하는 병렬처리 기법들이 등장하게 되었다. 오늘날 고성능 시스템 개발에 있어서 병렬처리 기법들을 효과적으로 활용하는 연구[1, 2]가 많은 주목을 받고 있다.

최근에 지능형 영상 분석시스템, 보행자 인식 시스템, 생체 인식 시스템 등 다양한 산업 분야에 객체인식 기법의 적용이 활발히 시도되고 있다. 지금까지 고성능 객체인식 알고리듬이 지속적으로 제시되어 왔지만 방대한 계산량으로 인하여 실시간 처리가 요구되는 응용 분야에 직접 적용하는 것은 쉽지 않은 상황이다. 따라서 실시간 처리기반의 응용분야에 효과적으로 활용되기 위해서는 처리 속도의 개선이 병행되어야만 한다.

객체인식 기법 중 하나인 SIFT(Scale Invariant Feature Transform)[3]는 영상의 크기(Scale)와 회전(Orientation)에 대하여 불변하는 특징점(interest point)을 추출하고 이를 기반으로 객체 인식을 수행하는 대표적인 알고리듬으로 알려져 있다. 그러나 우수한 인식 성능에도 불구하고 객체인식에 소요되는 방대한 연산량은 고속 처리에 문제를 초래하기 때문에 실시간 처리에 적합하지 않아 그 활용 범위가 제한되는 단점을 지니고 있다. 이 같은 SIFT의 문제점을 극복하기 위해 SURF(Speed Up Robust Features)[4] 알고리듬이 개발되었다. 그러나 이 알고리듬은 SIFT보다 향상된 연산 처리 속도를 보이지만 실시간으로 객체를 인식하고 활용해야 하는 응용분야에 적용하기에는 여전히 연산처리 속도의 개선이 요구되어진다.

지금까지 SURF알고리듬을 실시간 응용 분야에 활용하기 위하여 하드웨어적인 측면에서 활발한 연구[5-8]가 수행되었다. 반면에 소프트웨어적인 관점에서는 상대적으로 많은 연구가 진행되지 못하였다. 소프트웨어 측면에서의 기존 연구[9, 10]에서는 특징점 추출단계의 스케일 공간 구성 과정에 OpenMP를 적용하여 병렬화를 시도하였다. 따라서

*Corresponding Author(swha@gnu.ac.kr)
Received: 3 July 2014, Revised: 13 Aug. 2014, 11 Sep. 2014, Accepted: 30 Sep. 2014.
I.S. Hwang, Y.H. Moon, S.W. Ha: Gyeongsang National University / ERI
I.K. Eom: Pusan National University
※ 본 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임.(No.2014R1A1A2056434).

ⓒ IEMEK J. Embed. Sys. Appl. 2015 Feb.: 10(1) 49-55
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2015.10.1.49

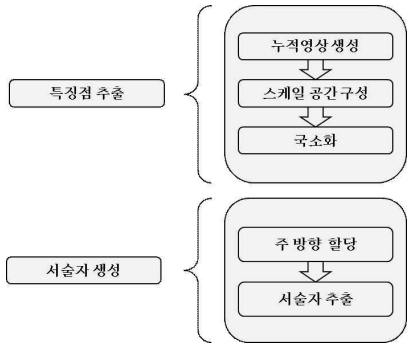

그림 1. SURF 알고리듬 순서도
Fig. 1 Flow chart of SURF algorithm

특징점 추출 단계에 대한 완전한 병렬화는 연구되지 못하였다. 이에 본 논문에서는 SURF 알고리듬에서 수행되는 특징점 추출 단계를 병렬처리 기법을 이용하여 고속화하는 방법을 제안한다. 제안방식에서는 특징점 추출 단계를 구성하는 스케일 공간 구성(Scale Space Representation)과정의 가속화를 위하여 OpenMP와 SIMD 기법을 적용한다. 또한 국소화(Localization)과정의 고속화를 위해서는 기존 순차처리 부분을 병렬처리에 적합하도록 변경한 후 OpenMP를 적용함으로써 실행속도를 향상시킨다. 다양한 영상들을 이용한 모의실험을 통하여 제안방식이 특징점 추출 단계의 연산 속도를 약 55% 개선시킴을 알 수 있다.

II. SURF 개요

SURF 알고리듬은 그림 1과 같이 특징점 추출과 서술자(Descriptors) 생성의 2단계로 나누어진다.

첫 번째 단계인 특징점 추출은 크게 누적영상(Integral Images) 생성, 스케일 공간구성, 국소화의 세 과정으로 세분화 된다. 먼저 Box type의 Convolution 필터가 효과적으로 적용되는 누적 영상을 생성한다. 그리고 스케일 공간 구성 과정에서는 스케일에 불변하는 특징점 추출을 위해 이미 생성된 누적 영상에 Hessian Matrix에 기초한 필터를 적용하여 그림 2와 같이 스케일 영상들을 생성한다[4]. 마지막으로 국소화 과정에서는 생성된 스케일 영상들 간의 비교를 통하여 극점(Extrema)을 판별하고 특징점들을 검출한다.

두 번째 단계인 서술자 생성단계는 주방향 할당(Orientation of Assignment)과정과 서술자 추출과정으로 구성된다. 주 방향할당 과정에서는 검출된

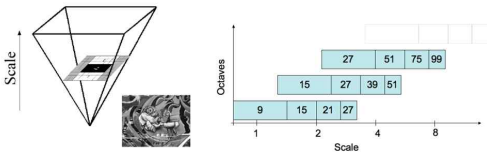

그림 2. SURF에서의 스케일 공간 구성[4]
Fig. 2 Scale space representation in SURF

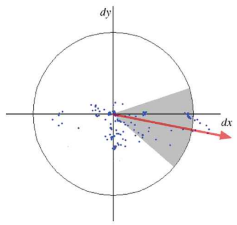

그림 3. 주 방향 할당[4]
Fig. 3 Orientation Assignment

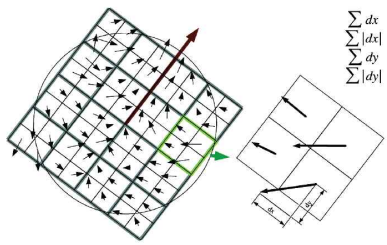

그림 4. 서술자 생성[4]
Fig. 4 Computation of descriptors

각 특징점을 중심으로 반경 6s 값에 해당하는 영역 화소들에 대하여 Haar wavelet Filter를 x, y 방향으로 적용한다. 그리고 그림 3과 같이 $\frac{1}{3}\pi$영역내의 각 결과 값들을 누적한 후 최대 누적값을 주 방향으로 할당한다[4].

서술자 추출 과정에서는 특징점 주변에 20s 크기의 사각 영역을 설정 후 그림 4와 같이 4x4의 부분 영역으로 분할한다. 그리고 각 부분 영역을 다시 5x5로 분할 한 후 Haar wavelet 필터를 적용하고 Gaussian 가중치를 부여한다. 그리고 각 부분 영역별 dx, dy, $|dx|$, $|dy|$의 총합을 구하여 총 4개의 서술자들을 생성한다. 각 특징점은 4x4 부분 영역별 4개씩 총 64개로 구성되는 서술자를 가지게 된다[4].

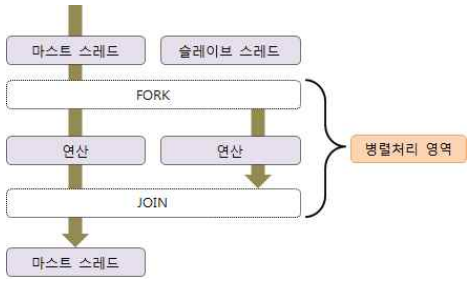

그림 5. OpenMP에서의 Fork-Join 과정
Fig. 5 Fork-Join procedure in OpenMP

Ⅲ. 병렬처리 기법

멀티코어 환경에서 최대의 성능을 얻기 위하여 병렬처리 기법의 활용에 관한 연구가 많은 관심을 받고 있다. 하나의 명령어로 여러 개의 데이터를 동시에 처리할 수 있는 SIMD[11], 공유메모리 기반의 병렬처리 기법인 OpenMP[12], GPU등 이기종 장치를 이용한 병렬화 기법인 CUDA[13] 기법들이 활발하게 개발되어 왔다. 본 논문에서는 위에서 언급한 병렬처리들 중 OpenMP와 SIMD를 적용한 고속 처리 방식을 개발한다.

1. OpenMP(Open Multi Processing)

OpenMP[12]는 공유메모리를 기반으로 다중 스레드를 이용하여 병렬프로그램을 작성하기 위한 표준 API로서 지시어(Directives), 보조지시어, 실시간 라이브러리(Runtime Library), 환경변수(Environment Variables)의 3부분으로 구성된다. DEC, IBM, Intel등으로 이루어진 OpenMP ARB에 의해 1997년 최초 Fortran용 1.0 버전이 배포된 이후 2013년 4.0 버전까지 배포되어 사용되고 있으며 현재 C/C++도 지원되고 있다.

OpenMP는 그림 5와 같이 Fork-Join 모델로 구동된다. OpenMP 지시어가 수행되는 순간 마스터 스레드는 미리 설정되어 있는 스레드 수에 따라 슬레이브 스레드를 Fork하여 스레드 팀을 구성한다. 생성된 스레드팀은 병렬처리를 수행한 후 Join에 의하여 동기화되는 시점에 소멸되고 마스터 스레드만 남게 된다.

OpenMP는 기존의 순차 프로그램 코드에 간단한 OpenMP지시어를 삽입함으로써 매우 간편하고 쉽게 프로그램을 병렬화 할 수 있는 장점을 지니고 있다. 그런데 이것은 데이터 의존성이 없는 데이터

표 1. MMX, SSE, AVX의 환경 비교
Table 1. Configuration of MMX, SSE and AVX

구분 \ 지원방식	MMX(1999)	SSE(2008)	AVX(2011)
주소명칭	mm0 ~ mm7	xmm0 ~ xmm7	ymm0 ~ ymm15
레지스터 크기	64bit	128bit	256bit
레지스터 수	8개	8개	16개

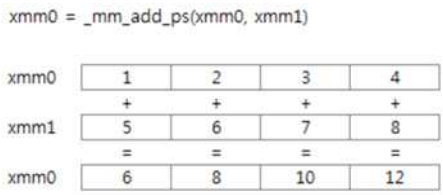

그림 6. SSE에서의 덧셈 연산 사례
Fig. 6 Example of addition computation based on SSE

간의 연산처리에 효과적인 반면에 데이터 의존성이 있는 데이터의 연산에는 데이터 경합 현상을 유발하여 도리어 프로그램 수행 시 오류를 발생시킬 수 있다. 따라서 OpenMP를 효과적으로 적용하기 위해서는 데이터간의 의존 관계를 명확하게 분석하여 적절한 지시어를 사용하는 것이 중요하다.

2. SIMD(Single Instruction Multiple Data)

Single Instruction Multiple Data 라는 약어에서 알 수 있듯이 SIMD[11]는 하나의 명령어로 여러 개의 데이터를 동시에 계산하는 병렬처리 방식이다. CPU 내부에는 SIMD를 지원하기 위한 부가적인 레지스터 및 명령어가 별도로 존재한다. SIMD는 1997년경에 제시된 MMX를 시작으로 SSE, SSE2, SSE3, SSE4, AVX, AVX2로 확장 발전되어 왔다. 이들은 내장하고 있는 레지스터의 수와 레지스터의 크기에 따라 분류된다. 현재 생산되는 대부분의 CPU에서는 SIMD가 지원되고 있다.

표 1에서 MMX의 경우 64bit 크기의 레지스터가 8개 존재함을 알 수 있다. 따라서 4Byte 데이터 2개가 동시에 처리 될 수 있다. SSE의 경우 128bit의 레지스터가 8개 존재하며 4Byte 데이터 4개가 동시에 처리 된다. 그림 6은 SSE에서의 덧셈 명령어를 이용한 병렬 연산을 나타낸 것이다.

그림 6에서 xmm0 레지스터에 4Byte 크기의 데이터 4개(1, 2, 3, 4)가 패킹되어 저장되어 있다. xmm1 레지스터에는 더하고자 하는 데이터 4개(5,

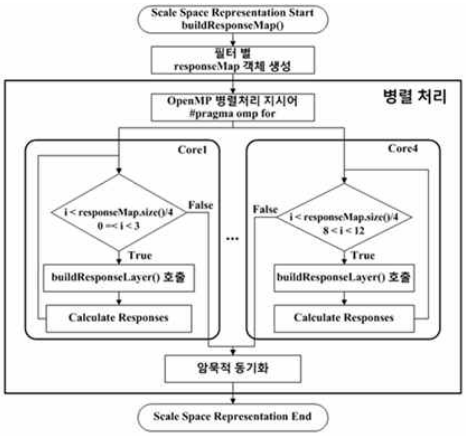

그림 7. 스케일 공간 구성에 대한 병렬처리 과정
Fig. 7 Parallel processing procedure for scale space representation

6, 7, 8)가 패킹되어 저장되어 있다. 그리고 SIMD명령어 _mm_add_ps(xmm0, xmm1)의 실행에 의하여 동시에 4개의 데이터가 더해져서 그 결과가 xmm0 레지스터에 저장되게 된다. 기존 방식에서는 4번의 덧셈이 개별적으로 이루어지지만 SIMD에서는 하나의 명령어로 4개의 데이터가 동시에 처리됨으로써 수행시간의 단축이 이루어진다.

Ⅳ. 제안하는 고속화 방식

본 논문에서는 오픈소스로 공개된 OpenSURF Library[14]를 기반으로 하여 구현된 순차 프로그램 코드에 대하여 병렬처리 기법들을 적용한다.

1. 스케일 공간 구성 과정에 대한 병렬화

SURF 알고리듬은 영상의 스케일을 변경하지 않고 필터의 크기를 변화시켜 가면서 스케일 공간을 구성한다. 이때, 필터가 적용되어 얻어진 결과 값들은 다른 크기의 필터에 전혀 영향을 주지 않고 개별적으로 저장되므로 데이터의 독립성이 보장된다. 따라서 병렬처리를 적용하여도 데이터 경합과 같은 오류가 발생하지 않는다.

OpenMP 지시어의 적용에 의하여 기존 싱글 코어에서 이루어지던 스케일 공간 구성 작업은 그림 7과 같이 코어별로 균등하게 나뉘어져서 수행되게 된다. 9x9 크기의 필터부터 387x387크기의 필터까지 총 12개 종류의 필터를 적용하여 스케일 공간을 구성하게 되는데 그림 7과 같이 코어를 4개로 설정

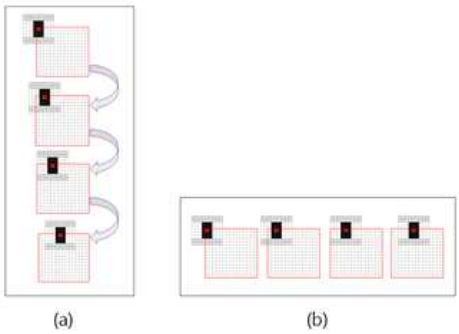

그림 8. 스케일 공간 구성에서 SIMD 병렬화를 위한 데이터 패킹 사례 (a) 기존방식 (b) 병렬처리
Fig. 8 Examples of data packing for SIMD based parallel processing in scale space representation. (a) Serial, (b) Parallel

할 경우 코어별 3가지 종류의 필터가 배분되어 스케일 공간 구성 작업이 진행되게 된다. 코어별로 3가지 종류의 필터링 작업을 진행하기 때문에 이론상 싱글코어 대비 4배의 개선효과가 있어야 한다. 그러나 스레드 팀이 구성되고 소멸되는데 오버헤드가 존재하기 때문에 이론상의 효과가 얻어지지 않는다. 그리고 이와 같은 오버헤드로 인해 연산양이 작은 작업의 병렬화는 그 효과가 미미하기 때문에 병렬화가 무의미해지는 경우도 있다.

스케일 공간 구성 작업 중 Box type의 convolution 필터 적용은 그림 8 (a)와 같이 동일한 행에서 화소별로 일정 간격만큼 옆으로 이동하면서 수행되고 한 행의 필터처리가 완료되면 다음 행으로 넘어가는 방식으로 이루어진다. 따라서 이러한 규칙성은 SIMD 병렬화 기법의 적용을 가능케 한다. 그림 8의 (b)는 4개 화소에 대한 필터 처리를 병렬화하기 위한 SIMD 명령어 적용에 있어서 데이터 패킹 원리를 보여 준다. 각 코어별 필터링에 있어서 SIMD 병렬화 기법을 결합할 경우 필터링 작업을 보다 더 고속화 할 수 있다.

2. 국소화 과정에 대한 병렬화

스케일 공간 구성이 완료된 이후에 이루어지는 국소화 과정에서는 극점을 구하고 극점일 경우 보간(Interpolation)을 적용하는 순차적인 처리를 통하여 특징점들을 추출하게 된다. 이때 보간에 소요되는 계산량은 국소화 과정에서 소요되는 계산량의 대부분을 차지한다.

국소화 과정을 고속화하기 위하여 기존 국소화

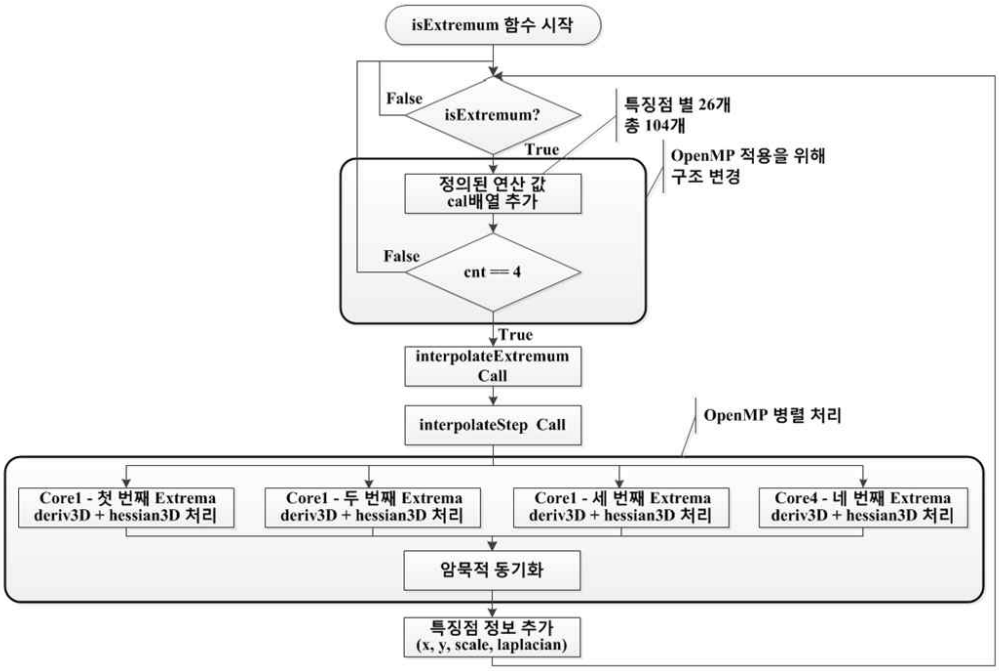

그림 9. 국소화에 대한 병렬처리 과정
Fig. 9 Parallel processing procedure for the localization

과정의 순차처리를 그림 9와 같이 병렬처리로 변경한다. 보간 과정의 OpenMP 적용을 위하여 극점이 추출되면 바로 보간을 적용하던 기존 처리와는 달리 제안 방식에서는 그림 9와 같이 4개의 극점이 추출될 때까지 대기한다. 이때 추출된 극점에 대하여 보간에 필요한 정보들이 그림 9의 cal 배열에 임시 저장된다. 그리고 극점 4개가 모두 추출된 이후에 보간 과정에 진입하여 보간 과정이 병렬처리 된다. 즉, 코어1에는 첫 번째 극점, 코어2에는 두 번째 극점, 코어3에는 세 번째 극점, 코어4에는 네 번째 극점이 할당되고 cal 배열에 저장된 대응하는 정보를 이용하여 각각의 극점에 대한 보간 과정이 동시에 수행되게 된다.

V. 모의실험 결과

본 논문에서는 제안 방식의 우수성을 검증하기 위하여 다양한 해상도의 영상들에 대하여 모의실험을 수행하였다. 모의실험은 Windows7 OS, Microsoft Visual Studio 2010, Intel C++ Composer XE 2011 환경 하에서 OpenMP 지시어, SIMD 명령어를 적용하고 그 성능을 비교, 확인하였다.

표 2. 특징점 추출 단계에서의 성능비교
Table 2. Performance comparisons for the interest point detection step

방식 \ 해상도		512x512 (499)	1024x1024 (3021)	2048x2048 (10328)	4096x4096 (36244)
기존 순차 방식 [14]	누적영상	0.004	0.012	0.047	0.183
	스케일 공간	0.036	0.146	0.591	2.381
	국소화	0.010	0.053	0.180	0.636
	총 실행시간 (sec)	0.0498	0.2112	0.8184	3.1996
기존 병렬 방식 [9]	스케일 공간	0.024	0.097	0.384	1.544
	국소화	0.009	0.054	0.183	0.638
	총 실행시간 (sec)	0.0366	0.1630	0.6138	2.3327
제안 방식	스케일 공간	0.012	0.051	0.199	0.790
	국소화	0.007	0.034	0.119	0.426
	총 실행시간 (sec)	0.0230	0.0966	0.3646	1.3994
개선효과(%)		53.82%	54.26%	55.45%	56.26%

표 2는 특징점 추출 단계에서의 실행시간을 비교한 것이다. 표 2에 기존 병렬방식과 제안방식의 총 실행시간에는 누적 영상에 대한 처리 시간이 포함되어 있다. 그리고 표 2의 개선효과는 식(1)과 같이 정의된다.

표 3. 전체 실행시간 비교
Table 3. Entire execution time comparisons

방식 \ 해상도	512x512 (499)	1024x1024 (3021)	2048x2048 (10328)	4096x4096 (36244)
순차방식[14] (sec)	0.1502	0.8236	2.9280	10.6626
제안 방식 (sec)	0.1252	0.7064	2.4698	8.8788
개선효과(%)	16.64%	14.23%	15.65%	16.73%

$$\text{개선효과}(\%) = \left(1 - \frac{\text{제안방식 실행시간}}{\text{순차프로그램 실행시간}}\right) \times 100 \quad (1)$$

표 2는 OpenMP, SIMD, 구조변경을 통한 병렬화가 수행된 제안 방식이 기존 병렬화 방식[9] 보다 우수함을 보여준다. 또한 표 2로부터 기존 순차방식에 비하여 제안 방식이 특징점 추출 단계의 실행 속도를 53.82%~56.26% 개선함을 알 수 있다.

표 2에서 제시된 제안 방식의 성능 개선을 전체 실행시간 관점에서 살펴보는 것은 의미있는 일이다. 표 3은 제안 방식이 전체 실행시간을 14.23% ~ 16.73% 개선함을 보여준다.

Ⅵ. 결 론

본 논문에서는 SURF 알고리듬의 특징점 추출 단계의 스케일 공간 구성, 국소화 과정에서 구조변경 및 병렬화를 통해 고속화를 수행하였고 성능 측정 및 평가를 진행하였다. 모의 실험 결과 제안 방식은 해상도와 특징점의 수에 관계없이 기존 순차방식의 특징점 추출 단계를 약 50% 향상 시켰다. 또한 이것은 전체 실행 시간을 약 15% 개선시키는 효과를 가져왔다. SURF 알고리듬의 고속화를 완성하기 위해서는 서술자생성 단계 및 특징점 정합 단계에 대한 추가적인 병렬처리 연구가 요구된다.

References

[1] Y.M. Kim, J.M. Kim, “Design and Verification of High-Performance Parallel Processor Hardware for JPEG Encoder,” IEMEK J. Embed. Sys. Appl., Vol. 6, No. 2, pp. 100-107, 2011 (in Korea).

[2] Y.H. Lee, J.H. Kim, “ePRO-OMP: A Tool for Performance/Energy PRofiler and Analyzer for OpenMP Applications,” IEMEK J. Embed. Sys. Appl., Vol. 6, No. 5, pp. 287-293, 2011 (in Korean).

[3] D.G. Lowe, “Distinctive Image Features from Scale - Invariant Keypoints,” International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.

[4] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. “Speeded-Up Robust Features (SURF),” Computer Vision and Image Understanding, Vol. 110, No. 3, pp. 346- 359, 2008.

[5] E.S. Na, Y.J. Jeong, “FPGA Implementation of SURF-based Feature extraction and Descriptor generation,” Journal of Korea Multimedia Society, Vol. 16, No. 4, pp. 483-492, 2013 (in Korean).

[6] Y.S. Do, Y.J. Jeong, “Hardware Design of SURF-based Feature Extraction and Description for Object Tracking,” Journal of The Institute of Electronics Engineers of Korea, Vol. 50, No. 5, pp. 83-93, 2013 (in Korean).

[7] J. Svab, T. Krajnik, J. Faigl, L. Preucil, “FPGA based Speeded Up Robust Features,” Proceedings of IEEE International Conference on Technologies for Practical Robot Applications, pp. 35-41, 2009.

[8] T. Sledevic, A. Serackis, “SURF Algorithm Implementation on FPGA,” Proceedings of the 13th Biennial Baltic Electronics Conference, pp. 291- 294, 2012.

[9] J.C. Kim, Y.H. Jung, E.S. Park, X. Cui, H.I. Kim, “Development of Fast Feature Detector Using Parallel Processing,” Proceedings of Control, Automation, and Systems Symposium, pp. 613-618, 2008 (in Korean).

[10] J.C. Kim, Y.H. Jung, E.S. Park, X. Cui, H.I. Kim, U.Y. Huh, “The Implementation of Fast Object Recognition Using Parallel Processing on CPU and GPU,” Journal of Institute of Control, Robotics and Systems, Vol. 15, No. 5, pp. 488-495, 2009.

[11] Y.H. Jung, SIMD Parallel Programming, FREELEC. 2012.

[12] Y.H. Jung, OpenMP Parallel Programming , FREELEC. 2011.

[13] D.B. Kirk, W.W. Hwu, Programming

Massively Parallel Processors: A Hand-on-Approach, NVIDIA. 2010
[14] OpenSURP Library. Available: http://www.chrisevansdev.com/

저 자 소 개

황 인 소

2004년 2월 경상대학교 컴퓨터과학과 학사.
2013년 3월~현재, 경상대학교 융합SW학과 석사과정.
관심분야: 임베디드 시스템 소프트웨어, 영상처리, 병렬화, 최적화.
Email: inso77@gnu.ac.kr

문 용 호

1998년 부산대학교 일반대학원 전자공학과 박사.
1998년~2001년 삼성전자 DM연구소 책임연구원.
2012년 미국 캘리포니아 주립대학교(UCSB) 방문교수.
2007년~현재, 경상대학교 정보과학과 부교수.
관심분야: 영상처리, 항공전자, 임베디드 시스템, SoC, 병렬프로그래밍, 동영상 부호화 표준.
Email: yhmoon5@gnu.ac.kr

엄 일 규

1998년 부산대학교 일반대학원 전자공학과 박사.
2007년~현재, 부산대학교 전자공학과 교수.
관심분야: 통계신호처리, 영상처리, 영상포렌식, 스테그분석.
Email: ikeom@pusan.ac.kr

하 석 운

1995년 부산대학교 전자공학과 박사.
2002년 미국 캘리포니아 대학교(UCR) 방문연구.
1993년~현재, 경상대학교 정보과학과 교수.
관심분야: 영상처리, 임베디드 소프트웨어, 컴퓨터비전, 신경회로망
Email: swha@gnu.ac.kr