

재구성형 프로세서 맵핑을 위한 컴퓨터 비전 응용 분석 : SIFT

허인구*, 김용주*, 이진용*, 조영필*, 백윤홍*, 고광만**

*서울대학교 전기컴퓨터공학부

**상지대학교 컴퓨터정보공학부

e-mail: {igheo, yjkim, jylee, ypcho}@sor.snu.ac.kr, ypaek@snu.ac.kr

[**kkman@sangji.ac.kr](mailto:kkman@sangji.ac.kr)

Analysis of Computer Vision Application for CGRA Mapping : SIFT

Ingoo Heo*, Yongjoo Kim*, Jinyong Lee, Yeongpil Cho, Yunheung Paek*, Kwangman Ko**

*Department of Electrical Engineering and Computer Science, Seoul National University

**School of Computer Information Engineering, SangJi University

요 약

최근 영상이나 이미지로부터 사용자가 원하는 정보를 추출해 내고 재구성 하는 영상 인식, 증강 현실 등의 컴퓨터 비전(Computer Vision) 응용들이 각광을 받고 있다. 이러한 컴퓨터 비전 응용들은 그 동안 많은 알고리즘들의 연구를 통해 꾸준히 개선되고 향상되어 왔으나, 많은 계산량을 요구하기 때문에 임베디드 시스템에서는 널리 쓰이기 힘들었다. 하지만 최근 들어, 스마트폰 등의 모바일 기기에서의 계산 처리 능력이 향상 되고, 소비자 수요가 증가하면서, 이러한 컴퓨터 비전 응용은 점점 모바일 기기에서 널리 쓰이게 되고 있다. 하지만, 여전히 이러한 컴퓨터 응용을 수행하기 위한 계산량은 부족하기 때문에, 충분한 연산량을 제공하기 위한 방법론들이 다양하게 제시되고 있다. 본 논문에서는 이러한 컴퓨터 응용을 위한 프로세서 구조로서 재구성형 프로세서(Reconfigurable Architecture)를 제안한다. 컴퓨터 비전 응용 중 사물 인식 분야에서 널리 쓰이는 SIFT(Scale Invariant Feature Transformation)을 분석하고 이를 재구성형 프로세서에 맵핑하여 성능 향상을 꾀하였다. SIFT의 주요 커널들을 재구성형 프로세서 맵핑한 결과 최소 6.5배에서 최대 9.2배의 성능 향상을 이룰 수 있었다.

1. 서론

최근 영상이나 이미지로부터 사용자가 원하는 정보를 추출해 내고 재구성 하는 영상 인식, 증강 현실 등의 컴퓨터 비전(Computer Vision) 응용들이 각광을 받고 있다. 이러한 컴퓨터 비전 응용의 종류는 얼굴인식, 비디오 감시(surveillance), 의료 영상 분석(예: X-ray, 초음파 이미지, CT 스캔), 무인자동차나 UAV(무인항공기)를 위한 영상처리 등 굉장히 다양하고, 최근에는 모바일 기기의 포변화로 증강현실 등 새로운 응용이 계속 늘어나는 추세에 있다. 이러한 컴퓨터 비전 응용은 수많은 이미지나, 영상으로부터 필요한 정보를 추출하여 사용자에게 제공한다. 하지만 이런 유용한 정보를 추출하기 위해서는 굉장히 많은 계산량을 필요로 한다. 이 때문에 컴퓨터 비전 응용 분야는 늘 고성능의 프로세서나 응용에 특화된 전용칩을 사용해 왔다.

하지만 고성능의 범용 프로세서는 가격면에서, 전용칩은 프로그래밍 가능한 유연성에 있어서 아쉬운 점이 있다. 고성능의 범용 프로세서를 컴퓨터 비전 응용에 사용할 경우 기존의 C와 같은 범용 프로그래밍 언어로 작성된 코드들을 그대로 사용할 수 있고, 유연성을 확보할 수 있다는

측면에서 강점을 가진다. 반면, 이러한 고성능의 프로세서는 가격 면에서 부담이 되는 동시에, 성능에서도 컴퓨터 비전 응용을 처리하기에 부족한 경우가 많다. 특정 응용에 특화된 전용칩을 사용하는 경우에는 성능, 전력소모 등에서 매우 우수한 수치들을 보여준다. 하지만 이 경우 응용이 바뀌는 데에 따른 대응이 힘들어, 개발 과정의 편의성이 떨어지고 설계 재사용이 힘들다.

이러한 점들을 극복하기 위해서는 고성능의 처리 능력을 가지면서도 다양한 응용을 지원할 수 있는 프로그래밍 가능한 프로세서 기반의 솔루션이 필요하다. 본 논문에서는 이러한 프로세서 기반의 고성능 솔루션으로서 재구성형 프로세서(Reconfigurable Architecture)를 제안한다. 재구성형 프로세서는 명령어 수준 병렬성(ILP : Instruction Level Parallelism)과 데이터 수준 병렬성(DLP : Data Level Prallelism)을 모두 살릴 수 있기 때문에, 다량의 데이터를 동시에 병렬 처리하여 응용을 가속할 수 있다. 동시에, 재구성형 프로세서 상에서 수행되는 프로그램은 SW적으로 비교적 쉽게 수정이 가능하기 때문에, 전용칩에 비해 유연성이 매우 뛰어나다.

본 논문에서는 이러한 재구성형 프로세서를 이용하여

SIFT(Scale Invariant Features Transformation)[1]의 주요 커널들을 가속하여 성능 향상을 꾀하였다. SIFT는 입력 이미지 상의 특징점을 찾아주는 응용으로서 많은 사물 인식 응용의 기초로서 널리 쓰이고 있다. SIFT의 주요 커널 중 가장 많은 시간을 차지하는 3개의 커널을 재구성형 프로세서 상에서 수행한 결과 최소 6.5배에서 최대 9.3배의 성능 향상을 얻을 수 있었다.

본 논문은 다음과 같이 구성된다. 2장에서는 본 논문에서 사용된 재구성형 프로세서에 대해 간단히 논한다. 3장에서는 SIFT가 분석되어 맵핑되는 과정이 소개되고, 4장에서는 실험 결과에 대해 밝히겠다. 마지막 5장에서 본 논문을 결론짓도록 하겠다.

2. 재구성형 프로세서

그동안 명령어 수준 병렬성과 데이터 수준 병렬성을 동시에 살릴 수 있는 재구성형 프로세서에 대한 연구가 활발히 이루어져 왔다[2-4]. 이러한 재구성형 프로세서는 그동안 병렬성을 살릴 수 있는 멀티미디어 코덱, 통신 모듈의 처리 등에 주로 쓰여 왔다.

재구성형 프로세서는 아래의 그림 1과 같이 간단한 구조의 PE(Processing Element)들의 연결로 이루어져 있다. 그림 1은 예로서 16개의 PE만을 보이고 있지만 실제로는 수십개에서 100개가 넘는 PE들을 집적하는 것도 가능하다. 이러한 PE들은 ALU 수준의 operation을 할 수 있고, 필요에 따라 곱하기나 특수한 연산들을 추가할 수도 있다. 또한 내부적으로 작은 크기의 레지스터 파일을 가지고 있다. PE간의 연결 구조에 따라 조금씩 다르지만 연결이 되어 있는 PE 끼리는 data를 매 cycle마다 주고받을 수 있다. 각 PE가 해야할 작업들은 Configuration Memory를 통해 매 cycle마다 바꿀 수 있기 때문에 응용에 따라서 Configuration Memory를 적절하게 채워주면 수 십개의 PE들이 병렬적으로 원하는 연산을 하게 할 수 있다.

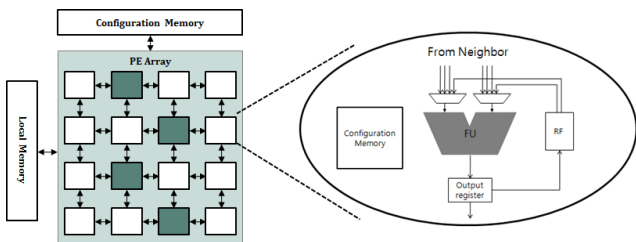


그림 1 간단한 구조의 PE들의 연결

재구성형 프로세서 상에서의 응용 맵핑은 그림2와 같은 과정을 따른다. 일단 프로그램은 DFG(Data Flow Graph)로 표현되어, 각각의 operation들 간의 입출력 관계가 명시 된다. 이렇게 만들어진 DFG는 맵핑 도구를 거쳐 2차원 배열로 구성된 재구성형 프로세서로 맵핑 된다. 맵핑은 기본적으로 소프트웨어 파이프라이닝 기법에 기반한 Modulo Scheduling[5] 기법을 사용한다. 이러한 맵핑 과정에서는 여러 가지 고려되는데, 여기에는 각 PE의 동

작 가능 operation, PE간의 연결 관계, 메모리 접근 지연 시간 등이 포함된다. 예를 들어 그림2에서 음영 처리된 PE의 경우 다른 PE와는 다르게 메모리를 접근할 수 있는 능력을 가지고 있다. 다른 PE의 경우 메모리를 직접 접근하지 못하고, 다른 PE가 읽어온 메모리 값을 전달 받아 사용해야만 한다. 이렇게 각각의 PE에 맵핑될 operation들이 결정되면, Configuration Memory에 저장될 PE들의 configuration이 정해진다.

재구성형 프로세서는 loop로 표현된 커널들 중 가장 안쪽의 loop, 즉 innermost loop만을 맵핑할 수 있는 한계를 가진다. 하지만 대부분의 커널의 경우 가장 내부의 커널들이 가장 많은 수행시간을 소모하기 때문에, 이러한 innermost loop만을 가속하는 것만으로도 뛰어난 성능 향상을 보일 수 있다.

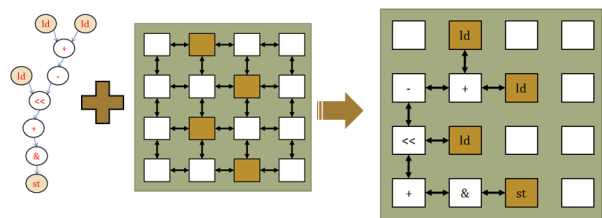


그림 2 재구성형 프로세서의 구조와 맵핑 방법

이러한 재구성형 프로세서와 SIMD(Single Instruction Multiple Data) 코어의 차이는, 예를 들어 C 언어 프로그램에서 FOR-loop의 각 iteration들이 서로 의존성이 없고 병렬화가 가능할 경우에, SIMD 또는 SIMT(Single Instruction Multiple Threads) 방식은 마치 그래픽 프로세서(GPU)에서 하듯이 각 iteration에 해당하는 연산을 서로 다른 PE에서 처리하게 함으로써, 여러 개의 iteration을 동시에 실행시키는 것이다. 반면에 재구성형 프로세서는 여러 개의 PE들을 엮어서 파이프라인을 만듦으로써 여러 iteration을 (약간의 시간적 차이는 있지만) 동시에 수행하는 것이다. 파이프라인의 경우에 파이프라인을 채우는 데에 약간의 시간이 들지만 코드가 충분히 클 경우 대부분 무시할 수 있기 때문에 SIMD와 재구성형 프로세서의 성능 면에서의 효율성은 동일하다고 볼 수 있다. 즉, N개의 PE가 있을 때 그것을 모두 사용하여 SIMD로 수행하는 것과 파이프라인을 구성하여 실행할 경우에 성능 throughput은 동일하다. 하지만, 파이프라인은 iteration 간에 데이터 의존성이 있는 경우(예: IIR filter)에도 대응이 가능하고, 파이프라인의 경우 각 PE는 하나의 연산만을 반복하므로 명령어 재해석의 오버헤드가 없고 따라서 더 에너지 효율적일 수 있으며, 메모리 포트나 곱하기 연산기와 같은 리소스의 사용이 더 균형적이라는 장점이 있다. 반면, SIMD의 경우에는 파이프라인을 채우는 오버헤드가 없고 프로그래밍이 좀 더 쉬울 수 있다.

컴퓨터 비전 응용의 경우 SIMD 방식으로 해결이 가능한 부분들이 분명 많다. 하지만 histogram과 같은 응용들은 iteration간의 의존성이 존재하기 때문에, SIMD나

치를 사용하였다.

ARM9과 재구성형 프로세서에 대해서 위에 제시된 3개의 커널의 코드를 수행하여 성능을 비교하였다. ARM은 ARM simulator를 이용해 cycle count를 얻은 뒤 1 GHz를 가정하고 수행시간을 측정하였다. 재구성형 프로세서는 본 연구팀이 개발한 맵핑 도구를 이용해 cycle count를 얻은 뒤 500 MHz를 가정하고 수행시간을 측정하였다. 맵핑 과정에서 build_gaussian_pyr은 DFG의 크기가 16개의 PE에 비해 매우 작았기 때문에 loop을 4번 unroll하여 맵핑하였다. 이를 통해 unroll하지 않았을 때에 비해 PE의 utilization이 높아지고 성능이 향상되는 효과가 있었다.

실험을 통한 성능 비교 결과는 표2와 같다. 재구성형 프로세서는 최대 16개의 PE가 동시에 연산을 수행할 수 있기 때문에 ARM9에 비해 월등한 성능 향상이 있었다. 2배의 클럭 스피드의 차이에도 불구하고 최소 6.5배에서 최대 9.27배의 성능 향상이 있었다. ori_hist, desc_hist에서 데이터 수준 병렬성이 거의 존재하지 않음에도 불구하고 명령어 수준 병렬성을 살려 상당한 성능 향상을 얻을 수 있었다.

Performance (us)	ARM9 (1GHz)	RA (500MHz)	Speedup
build_gaussian_pyr (1D 640pixels-4 times unrolled)	166.421	25.6	6.50
ori_hist	53.385	7.688	6.943
desc_hist	285.039	30.752	9.27

표 2 ARM9과 재구성형 프로세서의 성능 비교

하지만 ARM9이 single-issue 프로세서라는 점을 감안하면 성능 향상 폭이 아주 크다고는 볼 수 없다. 이는 차후 연구에서 좀 더 정확하고 잘 조직된 실험 환경을 통해 재구성형 프로세서의 효율성을 검증해야 할 것으로 본다. 그럼에도 불구하고, SIMD 프로세서에서는 성능향상이 어려운 ori_hist, desc_hist에 대해서도 재구성형 프로세서를 이용한 성능 향상 폭이 생각보다 컸다는 것은, 재구성형 프로세서를 이용한 명령어 수준 병렬성이 컴퓨터 비전 응용 분야에서 매우 유용할 수 있음을 보여준다. 또한 재구성형 프로세서는 SIMD 방식의 데이터 수준 병렬성도 살릴 수 있기 때문에 다양한 컴퓨터 비전 응용에서 사용될 수 있을 것이다.

5. 결론

본 논문에서는 최근 널리 쓰이고 있는 컴퓨터 비전 응용들의 처리를 위한 해법으로써 재구성형 프로세서를 제안하였다. 재구성형 프로세서의 효율성을 보이기 위해, 컴퓨터 비전 응용 중 널리 쓰이는 SIFT를 분석하고, 이를 재구성형 프로세서에 맵핑하여 성능 향상을 측정하였다. 실험 결과, SIFT의 세 개의 커널에 대해서 ARM9과 비교

하여, 최소 6.5배 최대 9.27배의 성능 향상을 얻었다.

재구성형 프로세서를 이용한다면, 히스토그램과 같이 SIMD 방식만으로는 해결할 수 없는 컴퓨터 비전 응용의 부분들을 명령어 수준 병렬성을 극대화 하여 해결할 수 있다. 이러한 장점을 통해 재구성형 프로세서는 기존의 SIMD 기반 컴퓨터 비전 프로세서들의 훌륭한 보완 또는 대체가 될 수 있을 것으로 본다. 재구성형 프로세서를 컴퓨터 비전 응용에서 더욱 효과적으로 사용하기 위해서는 앞서 설명한 바와 같이 메모리 접근 시간을 줄이기 위한 메모리 패턴 분석, 효과적인 메모리 아키텍처 구조, 메모리 재사용 기법에 대한 연구들을 더 진행해야 할 것이다.

Acknowledgement

본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업(과제번호 2011-0000975), 2011년도 정부(교육과학기술부)의 재원으로 한국과학재단의 국가지정연구사업(No.2011-0018609), IDEC 및 서울장학재단 하이 서울 장학금의 지원을 받아 수행되었습니다.

참고문헌

- [1] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Volume 60, Number 2, p. 91-110, 2004.
- [2] Singh, H., Lee, M.-H., Lu, G., Bagherzadeh, N., Kurdahi, F., Filho, E.: Morphosys: An integrated reconfigurable system for data-parallel and computation-intensive applications. IEEE Trans. Comput. 49(5), 465 - 481 (2000)
- [3] Kim, Y., Kiemb, M., Park, C., Jung, J., Choi, K.: Resource sharing and pipelining in coarsegrained reconfigurable architecture for domain-specific optimization. In: DATE 2005, Washington, DC, USA, pp. 12 - 17. IEEE Computer Society, Los Alamitos (2005)
- [4] Mei, B., Vernalde, S., Verkest, D., Lauwereins, R.: Design methodology for a tightly coupled vliw/reconfigurable matrix architecture: A case study. In: DATE 2004, p. 21224 (2004)
- [5] Hyunchul Park, Kevin Fan, Scott A. Mahlke, Taewook Oh, Heeseok Kim, and Hong-seok Kim. 2008. Edge-centric modulo scheduling for coarse-grained reconfigurable architectures. In Proceedings of the 17th international conference on Parallel architectures and compilation techniques (PACT '08). ACM, New York, NY, USA, 166-176. DOI=10.1145/1454115.1454140 <http://doi.acm.org/10.1145/1454115.1454140>
- [6] H. Bay et al. , "SURF : Speeded Up Robust Features", LNCS 2006, 2006.
- [7] Yan Ke, Rahul Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, pp. 506-513, 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04) - Volume 2, 2004
- [8] S.Heymann et al., "SIFT implementation and optimization for general-purpose gpu", WSCG 07, 2007
- [9] <http://kcachegrind.sourceforge.net/html/Home.html>