모바일 이기종 컴퓨팅 시스템에서 영상처리 고속화를 위한 CPU측 병렬처리 방법

백아람, 최해철
국립한밭대학교 정보통신전문대학원 멀티미디어공학과
aram98123@naver.com, choihc@hanbat.ac.kr

Parallel Processing Method on CPU for Image Processing on Mobile Heterogeneous Computing System

Aram Beak Haechul Choi
Hatbat National University, Graduate School of Information and Communications Multimedia Engineering

요약

모바일 기기의 보급률과 성능이 급속도로 성장하면서 모바일 기기에서의 비디오 소비 또한 크게 증가하였다. 하지만, 전력과 공간을 줄이기 위해 설계된 모바일 플랫폼은 데스크톱 플랫폼과 비교하여 성능의 한계가 존재한다. 따라서 대용량 비디오 처리를 위해 SIMD 아키텍처를 이용하는 임베디드 GPU를 활용하여 이와 같은 한계를 극복하기 위한 고속화 연구가 많이 진행되고 있다. 저장된 데이터를 활용하는 영상처리는 GPU 뿐만 아니라 CPU가 반드시 함께 이용되어야 하며, 모바일 환경에서의 이기종 컴퓨팅 시스템은 프로세서 사이의 낮은 전송속도와 이로 인한 대기시간, 모바일 운영체제가 지원하는 데이터 형태의 필수적인 사용 등의 구조적 단점이 존재한다. 본 논문에서는 임베디드 GPU를 활용한 영상처리 고속화를 위해 임베디드 CPU측에서 병렬처리를 이용하여 앞서 설명한 단점들을 극복하고 실험결과로 모바일 이기종 컴퓨팅 구조에서 임베디드 CPU 활용이 전체적인 연산 효율을 증가시키는 결과를 보였다.

1. 서론

많은 데이터 연산을 요구하는 영상처리 알고리즘은 작은 크기와 낮은 전력을 목적으로 설계된 모바일 플랫폼에서 고속으로 처리하는데 성능의 한계가 존재한다. 이러한 문제를 극복하기 위해 SIMD(Single Instruction Multiple Data) 아키텍처를 이용하는 임베디드 GPU 연산 능력을 일반적인 연산에 수행하는 기술인 GPGPU(General Purpose Graphics Processing Unit)를 활용하여 모바일 환경에서도 영상처리를 고속화하는 연구가 많이 진행되고 있다. 하지만 모바일 환경에서 CPU와 GPU를 모두 활용하는 이기종 컴퓨팅 시스템은 프로세서 사이의 낮은 전송속도와 이로 발생하는 대기시간, 모바일 운영체제가 지원하는 데이터 형태의 필수적 사용, 부족한 주기억장치 용량 등의 구조적 단점들이 존재한다.

따라서 본 논문에서는 영상처리를 위한 모바일 이기종 컴퓨팅 처리 구조의 효율성을 높이고자 임베디드 CPU의 처리 과정을 병렬로 구현하고 성능을 비교 및 분석한다. 본 논문의 2장에서는 임베디드 CPU와 GPU를 활용한 영상처리 구조에 대해 설명하고, 3장에서는 구현한 모바일 영상처리 구조에서 CPU측 병렬처리 방법에 대해 설명한다. 4장에서는 동일 환경의 순차 처리 방법과 구현한 방법의 실험 결과를 비교하고 5장에서 결론을 맺는다.

2. 임베디드 CPU와 GPU를 활용한 영상처리 구조

모바일 어플리케이션은 디스플레이를 위해 CPU와 GPU를 모두 이용해야 한다. GPU는 병렬처리를 이용하는 연산능력이 뛰어난 하드웨어이며, 그래픽 데이터를 출력하는 것이 주 목적이다. GPU 드라이버는 GPU와 직접적으로 통신이 가능한 코드이며, 모바일 그래픽스 API인 OpenGL은 운영체제의 프레임워크와 함께 렌더링을 위해 비트맵 데이터에 접근하게 된다. 게임과 같이 그래픽을 위해 일부 특화된 어플리케이션의 경우에 직접적으로 OpenGL과 소통이 가능하다.

GPU를 이용하여 영상처리를 구현하기 위해서는 반드시 CPU 작업이 필요하다. CPU와 GPU는 시스템 버스를 통해 연결되며, OpenGL과 프레임워크를 통해 CPU와 GPU 사이의 데이터 전송이 가능하다. GPU에서 데이터에 접근하기 위해서는 RAM(Random Acees Memory)에서 VRAM(Virtual RAM)으로 데이터 이동이 필요하다. GPU에서 병렬처리를 위해 텍스쳐 형태로 접근이 필요하며, CPU 이미지 배열로 직접적인 접근은 불가능하다. 텍스쳐 데이터 용량이 큰 경우 프로세서간 데이터 이동 시간이 증가하며, 텍스쳐 변환을 위해 CPU에서도 모바일 운영체제에서 지원하는 영상 데이터로 변환 과정이 필요하다. 즉, 영상처리를 위한 이기종 컴퓨팅에 요구되는 조건은 CPU 메모리 영역에 저장된 기존 데이터를 불러와 GPU를 함께 이용하는 것이 필수이며, 영상처리를 위한 영상 데이터 준비 과정이 필요하고 GPU에서는 가속 가능한 영상처리 알고리즘만 연산을 수행하게 된다.

3. 임베디드 CPU측 병렬처리 구조

그림 1은 이기종 컴퓨팅을 이용한 가장 기본적인 영상처리 순차구조이다. CPU로 집중되는 연산을 분산시키기 위해 GPU를 활용하고 SIMD 아키텍처의 병렬처리를 이용하여 영상처리 알고리즘을 GPU에서 고속화한다. 또한 GPU로 데이터 전송을 위해 CPU와 GPU 사이에 데이터 변환 버퍼를 구현하고, 버퍼는 CPU 영상 배열을 GPU의 텍스쳐 형태로 변환하여 데이터를 전송한다. 데이터의 크기가 증가할수록

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원(B0126-15-1013, 퍼즐형 Ultra-wide viewing 공간 미디어 생성 및 소비 기술 개발)과 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원(NRF-2013R1A1A1010344, SHVC 기반 고효율 스케일러블 비디오 부호화 및 고속화 알고리즘 연구)을 받아 수행된 연구임.

데이터 전송시간 증가로 인한 전체적 시스템 성능 효율이 낮아지므로 CPU와 GPU 사이의 데이터 이동이 최소화 되었으며, 결과적으로 한 프레임마다 데이터 이동이 한 번 발생하게 된다.

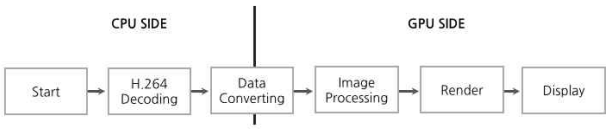


그림 1. 모바일 CPU-GPU 영상처리 순차구조

위 영상처리 구조에서 CPU 측에 필요한 과정으로 저장된 영상을 호출하기 위한 영상 복호화 단계와 모바일 운영체제에서 영상처리를 위해 필요한 영상 데이터 타입 변환 단계, 그리고 GPU로 데이터 전송을 위한 텍스쳐 변환 과정이 있다. CPU 측의 문제점으로는 설명한 모듈 사이의 대기시간이 발생하며, 영상 해상도가 커질수록 임베디드 CPU 연산량이 기하급수적으로 증가하여 영상처리 준비과정으로 발생하는 CPU 측의 연산 부담이 크게 증가한다. 소프트웨어 단계에서는 병렬처리를 활용하여 CPU 구조 모듈 사이의 대기시간을 최소화하고 전체적인 프로그램 처리 성능 효율을 높일 수 있다. 그림 2는 영상 복호화 과정과 데이터 타입 변환 과정을 모듈화하고 두 모듈을 CPU 측에서 스레드를 이용하여 병렬로 처리하는 과정을 나타낸다. 이러한 병렬처리 방법으로 각 처리 과정 사이의 대기시간 최소화가 가능하다.

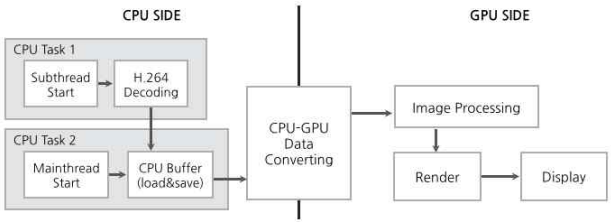
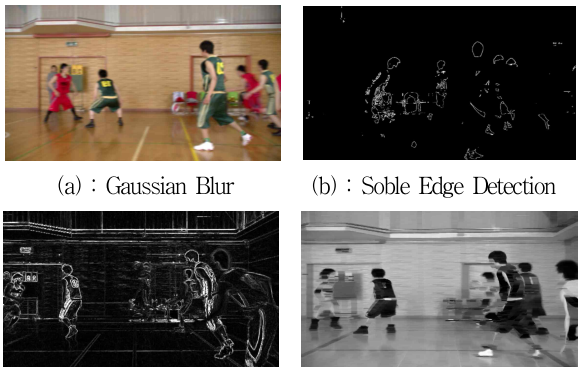


그림 2. 모바일 CPU-GPU 영상처리 순차구조에서 CPU측 병렬처리

4. 실험 결과 및 분석

구현된 구조는 ARM v7 기반의 듀얼코어 1.2Ghz CPU와 트리플 코어로 구성된 SGX543MP3 GPU, 그리고 1G 메모리로 구성된 모바일 기기에서 실험되었다. 참조영상은 50fps로 H.264 디코더에 의해 추출되었으며, 3가지 해상도로 실험에 사용되었다. 영상처리 알고리즘은 Gaussian Blur, Bilateral Filter, Sobel Edge Detection, Canny Edge Detection, Dilation, Erosion이 사용되었다. 실험은 모바일 CPU-GPU 영상처리 순차구조에서 CPU측 순차처리와 병렬처리를 처리했을 때 성능을 비교하였다.



(a) : Gaussian Blur (b) : Soble Edge Detection

(c) : Canny Edge Detection (b) : Erosion

그림 3. 모바일 환경에서 이기종 컴퓨팅 구조를 이용한 영상처리 결과

표 1은 모바일 CPU-GPU 영상처리 순차구조에서 CPU측 순차처리와 병렬처리 방법을 이용한 영상처리 알고리즘 수행시간을 비교한 성능이다. 전체적으로 1.05~1.31배 빠른 성능을 보였으며, Gaussian Blur, Bilateral Filter, Sobel Edge Detection 알고리즘은 단일 알고리즘 처리 구조로서 GPU 사용량이 다중 알고리즘 처리에 비해 적기 때문에 CPU측 모듈의 대기시간이 감소하여 더 높은 성능 향상을 보였다. 반대로 Canny Edge Detection, Dilation, Erosion의 경우 다중 알고리즘 처리 구조로서 CPU측 모듈의 대기시간이 증가하여 고속화 성능이 단일 알고리즘 처리 구조보다 낮은 것을 알 수 있다.

표 1. 모바일 CPU-GPU 영상처리 순차구조에서 CPU측 순차처리와 병렬처리를 이용한 영상처리 알고리즘 수행시간 및 고속화율

영상처리 알고리즘	해상도	ⓐ: CPU 순차처리(초)	ⓑ: CPU 병렬처리(초)	고속화율 (ⓐ/ⓑ)
Gaussian Blur	854×480	15.12	12.67	1.19x
	1280×720	25.06	20.43	1.23x
	1920×1080	48.38	36.82	1.31x
Bilateral Filter	854×480	15.30	13.16	1.16x
	1280×720	24.97	20.38	1.23x
	1920×1080	44.95	35.53	1.27x
Sobel Edge Detection	854×480	15.02	12.33	1.22x
	1280×720	25.01	20.21	1.24x
	1920×1080	47.43	36.21	1.31x
Canny Edge Detection	854×480	16.84	15.19	1.11x
	1280×720	32.68	29.06	1.12x
	1920×1080	69.32	61.18	1.13x
Dilation	854×480	15.30	14.56	1.05x
	1280×720	28.33	24.07	1.18x
	1920×1080	58.97	49.49	1.19x
Erosion	854×480	15.21	14.36	1.06x
	1280×720	28.34	24.14	1.17x
	1920×1080	59.18	49.37	1.20x

또한 모든 알고리즘 실험에서 영상의 해상도가 높아져 처리해야 하는 데이터가 증가할수록 고속화 효율이 상승하는 것으로 나타났다. GPU의 사용량이 증가하면 고속화 효율이 낮아지는 것과 다르게 CPU측에서 처리해야 하는 데이터가 증가할 경우에 병렬처리 구조는 순차처리 구조에 비해 고속화 효율이 더 증가하는 것을 알 수 있다.

5. 결론

본 논문에서는 임베디드 CPU와 GPU를 함께 활용하는 이기종 컴퓨팅 환경을 이용해 영상처리 순차구조에서 CPU측 순차처리 방법과 병렬처리 방법을 구현하고 실험 결과를 비교하였다. 구현 방법은 영상의 해상도가 높아지거나 단일 영상처리 알고리즘 구조와 같이 GPU 사용량이 적을수록 활용 효율이 증가하는 결과를 보였다. 이러한 실험은 모바일 플랫폼에서 영상처리 최적화에 있어서 CPU측 병렬처리가 효율적이라는 결과를 보였으며 임베디드 프로세서 사이의 구조적 최적화 또한 필요하다는 결론을 내릴 수 있다.

참 고 문 헌

[1] A-Ram Baek, Kangwoon Lee, and Haechul Choi "Speed-up Image Processing on Mobile CPU and GPU," Asia Pacific Conference on Multimedia and Broadcasting(APMediaCast 2015), pp. 79-81, Apr, 2015

[2] http://www.objc.io/issue-3/moving-pixels-onto-the-screen.html