

스마트폰에서의 실시간 영상처리를 위한 GPGPU 기반 프레임워크 구축

이만희, 강승헌, 박인규
 인하대학교 정보통신공학과

maninara@inha.ac.kr, ha00ha2@gmail.com, pik@inha.ac.kr

GPGPU Based Real-Time Image Processing Framework on a Smartphone

Man Hee Lee, Seungheon Kang, In Kyu Park

Department of Information and Communication Engineering, Inha University

요 약

본 논문에서는 스마트폰에서 해당 기기에 장착된 카메라로부터 실시간으로 입력되는 프리뷰 영상에 대하여 실시간으로 영상처리를 수행할 수 있는 프레임워크를 제안한다. 본 논문에서 제안하는 프레임워크의 경우 OpenGL ES 2.0 기반의 Shading Language 를 이용하여 모바일 GPU 에서 병렬처리를 수행함으로써 영상처리 알고리즘을 고속으로 적용할 수 있으며, 매 프레임의 입력 영상을 텍스처로 지정하고 연산 결과가 저장된 프레임 버퍼의 내용을 그대로 화면에 출력함으로써 메인 메모리와 GPU 메모리 사이의 자료 이동을 최소화 하였다. 현재 상용화 된 스마트폰에 제안하는 프레임워크를 이용하여 적용시킨 결과 필터링 기반의 여러 가지 영상처리 알고리즘의 실시간 처리가 가능함을 보여줌으로써 본 논문에서 제안하는 프레임워크의 실시간 활용을 확인할 수 있다.

1. 서론

최근 모바일 카메라의 기술적인 발전으로 인하여 스마트폰에서도 기존의 디지털 카메라와 비슷한 해상도의 영상을 취득할 수 있게 되었다. 특히 스마트폰의 특성상 항상 휴대하는 것이 필연적이고 스마트폰의 대중적인 보급으로 일상 생활을 영상으로 기록하는 도구로써 스마트폰을 활용하는 사용자가 증가하고 있는 추세이다. 또한 SNS 의 인기에 힘입어 개인의 일상 생활을 다른 사람들과 공유하는 행동이 일반화 되었고 이러한 공유 방법 중 하나의 도구로써 자연스럽게 스마트폰으로 취득한 영상을 이용하고 있다.

이와 같이 스마트폰을 이용한 영상 취득과 그 사용이 증가함에 따라 사용자들은 기존의 단순한 영상 취득 결과에 만족하지 못하고 다양한 영상처리 효과를 적용시킨 독창적인 영상을 선호하게 되었다. 그에 따라 스마트폰에서 영상에 영상처리 효과를 부여하는 어플리케이션이 꾸준히 개발되어 왔고 이러한 어플리케이션 중 대부분의 경우 미리 취득 된 영상에 몇 가지 영상처리 알고리즘을 적용하는 구조로 구성되어 있다. 그러나 위와 같은 구조의 경우 미리 취득된 영상을 이용하기 때문에 영상처리 효과가 적용된 결과를 예측하기 어렵고 그로 인해 사용자가 만족할만한 영상을 얻기 힘든 단점이 있다. 위와 같은 단점을 보완하는 방법으로 스마트폰 카메라의 프리뷰 영상을 실시간으로 처리함으로써 사용자에게 영상처리 결과를 보여줄 수 있지만 카메라 및 디스플레이의 해상도가 높고 모바일 CPU 가 높은 해상도의 영상을 실시간으로 처리할만한 연산능력을 갖고 있지 못하기 때문에 어플리케이션으로의 적용이 어려운 상태이다.

따라서 본 논문에서는 스마트폰에서의 실시간 영상처리를

위한 프레임워크를 제안한다. 복잡한 영상처리 연산의 실시간성 확보를 위하여 본 논문에서는 OpenGL ES 2.0[1] 기반의 GPU 가속을 활용하였고 GPU 에서 처리된 결과를 그대로 화면에 출력함으로써 프로세서간의 자료이동을 최소화 할 수 있다.

2. 실시간 영상처리 프레임워크의 구조

그림 1 은 본 논문에서 제안하는 프레임워크의 기본 구조를 도시화하고 있다. 그림 1 에서 보는 바와 같이 우선 하드웨어 장비에 대하여 기본 설정과 초기화를 수행한다. 그 후 입력되는 영상을 GPU 에서 처리하기 위하여 해당 영상을 텍스처로 지정함으로써 영상 정보를 GPU 로 복사한다. 텍스처로 지정된 입력 영상을 OpenGL ES 의 shading language 를 이용하여 각각의 화소에 대해 영상처리를 수행하고 그 결과를 FBO 를 이용하여 다시 연산의 입력으로 지정하거나 그대로 화면에 출력한다. 결국 위의 과정을 매 프레임 반복함으로써 실시간 영상처리의 구조를 구축할 수 있다.

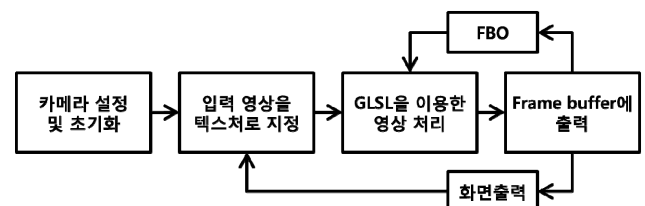


그림 1. 실시간 영상처리 프레임워크의 구조.

본 논문에서는 위에서 제안한 실시간 영상처리 프레임워크의 구조를 실제 스마트폰에 적용하기 위하여 Android[2] 기반의 스마트폰을 이용하여 실험을 진행하였다. Android OS의 경우 Java 기반의 객체지향 언어로 구성된 OS로서 스마트폰 각각의 기능을 수행하는 부분 역시 객체지향 기반의 class로 구성되어 있다. Android의 카메라 같은 경우 android.hardware.Camera class를 이용하여 카메라의 설정 및 초기화를 수행할 수 있다. 일반적으로 카메라에서 들어오는 실시간 영상을 OpenGL ES 함수를 이용하여 텍스처로 지정함으로써 GPU 메모리로 복사할 수 있다. 그러나 Android의 경우 android.graphics.SurfaceTexture 라는 카메라 프리뷰 영상의 입력 방법에 대한 특별한 class를 이용함으로써 위 과정을 생략할 수 있다. SurfaceTexture class를 Camera의 프리뷰 입력에 대한 Callback 함수로 지정하게 되면 실시간으로 입력되는 프리뷰 영상에 대하여 SurfaceTexture의 Listener를 통해 새로운 입력 영상이 사용 가능한지 확인할 수 있다. 새로운 입력 영상에 대한 사용 여부가 Listener에 의하여 통지되면 SurfaceTexture의 특정 함수를 이용하여 텍스처로 지정된 GPU 메모리의 정보를 갱신하게 된다. 위와 같은 방법으로 갱신된 텍스처는 OpenGL ES의 shading language를 이용하여 각각의 화소에서 입력 영상에 대한 접근 도구로 사용될 수 있고 C언어 스타일의 shading language를 이용하여 각각의 화소에 대해 영상처리를 수행할 수 있다. 이러한 과정으로 연산된 결과는 OpenGL ES의 frame buffer에 저장되고 그 결과를 FBO를 이용하여 다시 연산의 입력으로 지정하거나 frame buffer를 그대로 화면에 출력함으로써 실시간으로 처리되는 영상처리 알고리즘의 적용 결과를 스마트폰의 디스플레이를 통하여 확인할 수 있다. 이 때 Android에서 OpenGL ES를 활용한 화면 출력을 위하여 android.opengl.GLSurfaceView class를 사용하게 되고 실제 FBO와 같은 OpenGL ES의 부가적인 기능을 사용하기 위하여 객체지향 기반의 Java언어 대신 C언어를 이용한 개발이 가능하도록 Android NDK[3]를 활용할 수 있다.

3. 실험 결과

제안하는 프레임워크의 성능 평가를 위하여 본 논문에서는 Android 기반의 LG 전자 Optimus LTE 스마트폰을 이용하여 실험을 진행하였다. 위 장비의 경우 1.5GHz의 듀얼코어 CPU와 1GB의 메인 메모리, 800만 화소의 카메라 및 1280x720의 해상도를 갖는 디스플레이를 탑재하고 있다. 또한 OpenGL ES 2.0을 지원하는 Adreno 220 기반의 GPU를 갖고 있다. 2절에서 설명한 바와 같이 매 프레임마다 입력되는 영상은 텍스처 형태로 지정되고 OpenGL ES의 Shading Language를 이용하여 영상처리 효과가 적용된다. 실험을 위한 카메라 하드웨어 접근 및 UI 부분은 Java를 이용하여 개발되었고 Android NDK를 이용하여 OpenGL ES 설정 및 실제 영상처리 연산을 수행하였다. 실시간 영상처리 성능을 평가하기 위하여 Gaussian blur와 Sobel 필터를 이용한 영상의 gradient 연산을 실험하였다. 일반적으로 카메라의 입력 영상은 컬러 정보인 RGB 형태의 값을 갖고 있는 반면 Sobel 필터의 경우 밝기 영상에 대한 convolution 기반의 window 연산이기 때문에 색 공간의 변환이 요구되고 이 때 OpenGL ES의 shading language에서 지원하는 벡터의 내적을 이용하여 색 공간의 변환을 고속으로 수행할 수 있다.



그림 2. 스마트폰에서 Gaussian blur와 Sobel 필터를 이용한 실시간 영상처리 결과.

그림 2는 카메라의 실시간 프리뷰 입력에 대하여 Gaussian blur와 Sobel 필터를 이용한 gradient 연산 결과를 보여주고 있다. 이 때 알고리즘을 두 단계로 분리하여 컬러 정보를 밝기 정보로 변환한 결과를 FBO로 지정한 뒤 그 결과를 이용하여 위 알고리즘을 수행하였다. 실험 결과 1280x720의 해상도를 갖는 한 장의 입력 영상에 대하여 평균 61ms의 연산시간을 보여주고 결국 16.4FPS의 수행 능력으로 동영상 처리가 가능함으로써 제안하는 프레임워크의 실시간 활용을 확인할 수 있다.

4. 결론

본 논문에서는 스마트폰에 장착된 카메라로부터 입력되는 프리뷰 영상에 대하여 실시간으로 영상처리를 수행할 수 있는 프레임워크를 제안하였다. 본 논문에서 제안하는 프레임워크의 경우 OpenGL ES 2.0 기반의 shading language를 이용하여 모바일 GPU에서 병렬처리를 수행함으로써 영상처리 알고리즘을 고속으로 수행할 수 있다. 현재 상용화된 Android 스마트폰에서 Gaussian blur와 Sobel 필터 기반의 gradient 계산 알고리즘을 적용시킨 결과 본 논문에서 제안하는 프레임워크의 실시간 활용을 확인할 수 있었다. 또한 FBO를 활용하여 메인 메모리와 GPU 메모리 사이의 자료 전송을 최소화하였고 결국 여러 단계의 연산으로 이루어져 있는 복잡한 영상처리 알고리즘 또한 고속으로 처리할 수 있었다.

감사의 글

본 논문은 지식경제부 산업융합원천기술개발사업으로 지원된 연구결과입니다. [10041664, 멀티 Shader GPU 통합형 멀티 코어 퓨전 프로세서 원천 기술 개발]

참고문헌

- [1] http://www.khronos.org/opengles/2_X/
- [2] <http://www.android.com>
- [3] <http://developer.android.com/tools/sdk/ndk/>