

GPU 기반 콘텐츠 품질검사 실시간 고속화 시스템 개발

*이문식 *최성우 **안기욱 **김민기 *정병희

*한국방송공사 **(주)이타기술

celestia@kbs.co.kr

Development of GPU Based High-speed Contents Quality Check System

*Lee, Moonsik *Choi, Sungwoo **An, Kiok **Kim, Mingi *Jung, Byunghee

*KBS **ITA technology

요약

방송 제작 환경은 고품질의 콘텐츠를 빠르고 효율적으로 서비스하기 위하여 IT 기반 시스템으로의 전환을 진행하여 완성 단계에 이르렀으며, 대부분의 방송 콘텐츠는 파일 기반으로 제작 및 보관되고 있다. 과거 테이프 기반에서 파일 기반 콘텐츠로 전환되면서 신호 레벨로 진행되던 전통적인 품질 관리에 대한 새로운 방안이 요구되었으며, 이를 위하여 파일 기반 콘텐츠에 최적화된 콘텐츠 품질검사 시스템 개발이 진행되어 왔다. 이미지 처리에 기반하는 오류 검출 알고리즘의 복잡성으로 인하여 실시간 검사를 지원하지 못하여 HD 실시간 시스템에 적용에 어려움이 있었으며, 대용량의 아카이브 시스템에서는 품질검사 시간에 대한 단축이 지속적으로 요구되고 있다.

이에 본 논문에서는 방송 환경에서 발생하는 블록 오류 등 다양한 A/V 오류를 고속으로 검출하기 위하여 최근에 급부상하고 있는 GPU 기반의 병렬처리를 이용하는 품질검사 실시간 고속화 시스템의 구현에 대하여 기술하고자 한다.

1. 서론

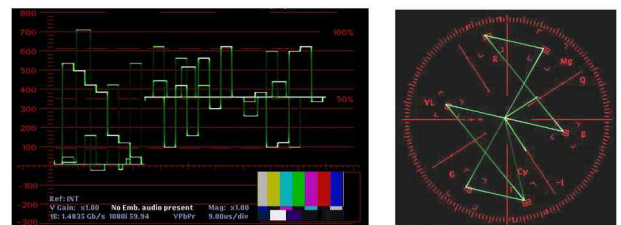
방송 제작 환경은 10여 년 전부터 IT 기반 제작 장비의 보급에 맞추어 파일 기반 제작 환경으로 점진적으로 변화하여 최근 완성 단계에 이르렀다. 제작/송출 워크플로우 전반적으로 파일 형태의 콘텐츠가 기본 포맷으로 사용되고 있으며, 이는 고품질의 방송 콘텐츠를 더욱 빠르고 효율적으로 서비스할 수 있는 기반이 되었다.

그러나, 방송 제작 프로세스 상에서 VCR, 인코더 등의 장비 요인 또는 장시간 사용에 의한 열 발생, 불안정한 전원, 먼지 등의 환경 요인에 의하여 다양한 A/V 오류가 발생하고 있다. 방송사에서는 방송 품질을 높이기 위하여 이러한 오류를 찾기 위한 노력을 지속적으로 기울여 왔다. 과거 테이프 기반의 방송 제작 환경에서는 <그림 1>과 같이 비디오 분석기, 컬러바/테스트 톤 등을 이용하여 신호 레벨의 품질검사를 수행하였으나, 오디오/비디오 콘텐츠 내부에 포함되어 있는 오류는 품질 관리자가 직접 재생해서 눈으로 보며 오류를 찾을 수밖에 없었다. 파일 기반의 제작 환경에서는 이러한 방식은 비효율적이기에 새로운 품질검사 방식이 요구되었다.

파일 기반의 콘텐츠는 미디어 파일로 존재하므로 파일의 장점을 살려 대용량의 콘텐츠를 자동화된 방식으로 처리할 수 있으며, 이미지 처리 기법을 적용하여 콘텐츠 내에 포함된 A/V 오류에 대해서도 주관적인 판단을 배제하고 일관성 있게 품질검사를 수행할 수 있다. 이를 실현하기 위하여 자동화된 콘텐츠 품질검사 시스템 개발을 진행하고 있으나, 오류를 검출하기 위한 알고리즘은 이미지 처리에 기반하므로 각 화소별로 필터링하고 특징점을 추출하여 주변 정보들과 비교 분석

하는 등 상당히 복잡한 과정을 거쳐야하기에 처리 시간이 많이 소요되었다. 이를 해결하기 위하여 CPU 병렬 처리 기법을 도입하였지만, HD 콘텐츠에 대한 실시간 검사에는 여전히 어려움이 있다. 또한 비디오 아카이브 등 수십만 시간의 대용량 콘텐츠를 보관하고 있는 시스템에서는 전체 검사 시간이 상당 기간 걸릴 수 있기에 이에 대한 단축을 지속적으로 요구하고 있다.

이에 본 논문에서는 검사 시간이 오래 걸리는 복잡한 오류 검출 알고리즘에 대하여 더욱 빠르게 수행할 수 있도록 최신의 GPU 병렬 처리 기법을 적용하여 HD 실시간 검사를 지원하고자 한다.



a. 컬러바 신호 검사



b. 테스트 톤 신호 검사

그림 1. 테이프 기반 제작 환경에서의 품질검사

2. 콘텐츠 품질검사 시스템

방송 제작 시스템에서의 품질검사는 크게 두 가지 범주로 나눌 수 있다. 첫 번째는 재생 가능성을 확인할 수 있는 형식에 대한 적합성 검사이며, 두 번째는 콘텐츠 내부에 포함된 A/V 오류에 대한 검사이다. 파일 기반에서의 콘텐츠는 미디어 파일로 존재하므로 파일 포맷과 압축 포맷에 대한 메타데이터를 검사하여 형식에 대한 적합성 검사를 수행할 수 있으며, 콘텐츠 내부 A/V 오류는 이미지/오디오 내부 분석을 통해 오류 검출이 가능하다.

방송에서 사용되는 콘텐츠 포맷은 MPEG-2, H.264 등의 압축 포맷과 MXF, MOV 등의 파일 포맷이 주로 사용되고 있으며, 현재 표준화 중인 HEVC 포맷도 UHD TV용으로 준비하고 있다. 콘텐츠 형식에서 검사하는 주요 항목으로는 사용 코덱, 화면 비율, 해상도, 컬러 샘플링, 오디오 채널수, 전송률 등의 설정 값이 있으며, 이러한 항목들에서 의도되지 않은 값을 갖는 콘텐츠를 검출하게 된다.

콘텐츠 내부의 A/V 오류는 <그림 2>에서 대표적으로 많이 발생하는 오류의 예시를 보여준다. 디지털 콘텐츠에서는 블록 단위의 압축을 사용하기에 압축 정보 유실 또는 비동기화에 의하여 네모난 형태의 블록 오류가 가장 많이 발생하고 있으며, 오래된 아날로그 콘텐츠에서는 자기 신호가 약해져서 색/휘도 신호가 열화되는 색열화와 노이즈에 의해 흰색 점들이 나타나는 드롭아웃 형태의 오류가 많이 발생한다. 이외에도 의도되지 않은 컬러바, 단색, 화면 정지, 음량 크기, 음량 좌우 밸런스 이상 등이 많이 발생한다.



그림 2. A/V 오류 예시

방송 제작 환경에서 주로 사용되는 포맷에 대한 검사와 자주 발생하는 A/V 오류를 검사할 수 있는 콘텐츠 품질검사 시스템에 대한 개발을 진행하고 있다. 대량의 콘텐츠 검사에 적합하도록 검사 서버와 관리 서버를 분리하였으며, 제작 시스템과 유기적인 연계가 가능하도록 HTTP 기반의 API를 지원하고 있다. 시스템 구성은 <그림 3>과 같이 관리 서버와 다수의 검사 서버로 이루어져 있으며, 공유저장장치에 저장되어 있는 콘텐츠에 대하여 관리 서버로부터 할당받아 개별 검사 서버에서 직접 접근하거나 FTP 전송 받아 검사하도록 되어있다. 검사 결과와 진행 상황은 클라이언트 PC에서 웹페이지를 통해 점검할 수 있다.

검사 서버는 다중 작업이 가능하도록 여러 대의 검사 서버를 운용할 수 있으며, 검사 알고리즘을 컴포넌트화하여 개별 기능 개선이 빠르게 이루어질 수 있도록 개발하였다. 검사 가능한 주요 오류 유형은 비디오의 경우 블록, 블록 스트라이프, 컬러바, 단색 프레임, 프리즈 프레임,

임, 인터레이스, 색열화 등이며, 오디오의 경우 뮤트, 테스트 톤, 고주파 노이즈, 볼륨 레벨 등이다. 지원 포맷으로 파일 포맷은 MXF, MOV, MP4, WMV, WAV, MP3 등이며 압축 포맷으로는 비압축, MPEG2, DVC PRO, H.264, ProRes, WMV, PCM, AES-3, WMA 등 방송 환경에서 사용되는 포맷을 지원한다.

관리 서버는 검사 서버, 작업 스케줄과 작업 결과를 관리하며, 와이 퍼포더 또는 API를 통한 검사 목록 입력이 가능하고 검사 프로파일, 작업 우선순위 변경 기능과 검사 결과에 대한 통계를 제공한다.

실시간 인코딩과 대용량의 파일을 다루는 시스템에서는 검사시간 이내로 끝나는 실시간 고속 검사가 중요한데, 현재 검사 속도는 이에 미치지 못하고 있다. 따라서 실시간 검사를 위하여 병렬처리 등 고속화를 위한 다양한 기법을 적용하여 검사 속도에 대한 요구를 만족시키기 위한 노력을 기울이고 있다.

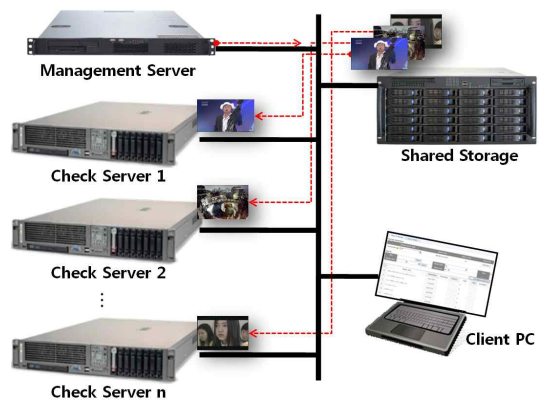


그림 3. 콘텐츠 품질검사 시스템 구성

3. 품질검사 실시간 고속화

품질검사에 대한 요구사항에서 고속화는 검사 효율성을 높일 수 있는 주요 성능 중 하나이다. SD 콘텐츠의 경우에는 해상도가 낮아 실시간의 1/2배 시간 정도의 고속 검사가 가능하지만, HD 콘텐츠 검사 속도는 해상도가 높아 처리해야 할 데이터가 많고 알고리즘이 복잡하여 처음 개발 시에는 실시간의 3~5배 정도 시간이 걸리게 되었기에 HD 콘텐츠에 대한 품질검사 적용에 어려움을 겪었다. 그래서 이를 해결하고자 CPU 및 GPU 병렬처리 기법을 도입하였다.

CPU 병렬처리에서는 <그림 4>와 같이 먼저 공통적으로 여러 검사 알고리즘에서 사용되는 흑백 변환, 컬러 히스토그램 등을 공통 서브 알고리즘으로 도출하여 전처리하였고, 이후 이들을 이용하는 검사 알고리즘을 배치하여 효율적으로 구성하였다. 개별 검사 알고리즘에서는 각각의 검사 시간에 따라 우선순위를 두어 각기 다른 코어에 최적화 배치하여 최종 검사 시간이 최소가 될 수 있도록 병렬처리를 하였다. 또한, <그림 4> 하단처럼 흑백 변환에서 이미지 영역을 여러 영역으로 나누어 영역별로 병렬처리를 하였는데, 다른 검사 알고리즘에서도 영상의 영역을 여러 영역으로 나누어 각각을 다른 유휴 코어에 할당하는 병렬처리 이중화를 통해 효율을 더욱 향상시켰다. 이러한 병렬처리를 통해 50% 이상 성능을 향상시켜 실시간의 1.5~2배 정도로 검사 시간을 단축시켰지만, 여전히 실시간 검사 성능을 만족시키지 못하였기에 최근 병렬처리 분야에서 새로이 각광받고 있는 GPU 병렬처리 기법을 도입하였다.

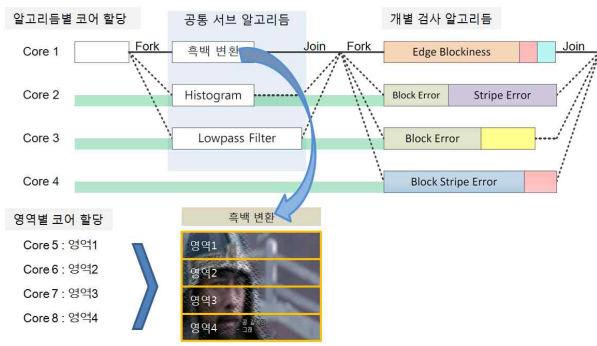


그림 4. CPU 병렬처리 구성

GPU 병렬처리는 <그림 5>와 같이 기본적으로 CPU와는 다른 병렬처리 개념을 갖는다. CPU는 다중 코어에 작업을 분할 할당하거나, 하나의 코어에 다양한 작업의 스레드를 생성하여 시분할로 다중 작업을 진행하게 되지만, GPU에서는 동일한 기능의 스레드를 무수히 많이 생성하여 동시에 수행하는 방식의 병렬처리를 사용하므로 병렬처리의 효율을 극대화하게 된다.



그림 5. CPU와 GPU 병렬처리 방식 비교

GPU는 동일한 연산에 대한 병렬화이기에 화소 단위 연산, 필터 연산 등에서 매우 유리한 구조를 갖는 반면 스택 연산 등에서는 불리한 결과를 얻게 된다. 또한 데이터를 GPU로 전송하거나 받는 시간이 추가되어야 하므로 메모리 전송 모듈 설계가 중요하다. 모든 알고리즘을 GPU로 전환하기보다는 GPU에 유리한 모듈만 적용하고 나머지 부분은 CPU에서 처리를 하여 <그림 6>과 같이 효율적으로 연동되는 모델을 구성하였다.

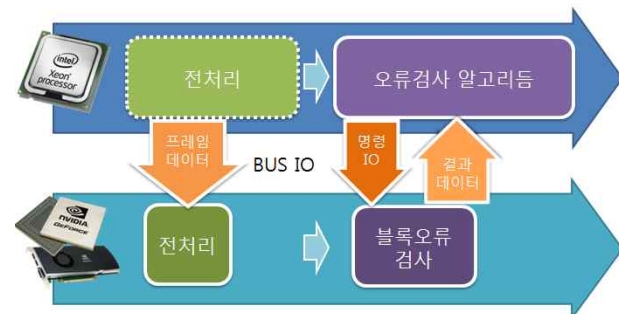


그림 6. CPU와 GPU 간의 연동

GPU 메모리 전송 모듈은 프레임 데이터 전송과 결과 데이터 전송으로 이루어져 있는데, 전송 시간은 3~4ms 정도 소요가 된다. 프레임 데이터 전송과 GPU 검사 알고리즘을 비동기로 구현하여 메모리 버퍼가 비게 되면 즉시 전송을 수행하도록 하여 프레임 데이터가 사전에

미리 전송되도록 하여, 전송시간이 검사시간에 영향을 주지 않도록 구현하였다.

GPU 처리는 전처리 기능(흑백 변환, 컬러 히스토그램)과 검사 간을 주로 소요하는 블록 등의 검사 알고리즘에 대하여 적용하였다. 흑백 변환은 아래 수식(1)과 같이 부동소수점 연산을 포함하고 있는데, 이를 수식(2)와 같이 근사치를 갖는 정수 연산으로 변환하여 GPU 적용에도 무리가 없으면서 수행 속도도 빠르게 하였다.

$$Y = 0.299R + 0.587G + 0.114B \quad \dots (1)$$

$$Y' = (4897R + 9617G + 1868B) \gg 14 \quad \dots (2)$$

컬러 히스토그램은 R, G, B 화소값에서 상위 4비트를 얻어서 직렬 배열하여 12비트 데이터를 생성한 후, 같은 값을 갖는 히스토그램 개수를 구하였다. GPU에서는 화소에 대한 처리가 동시에 이루어지기에 결과값을 같은 변수에 동시에 쓰는 경우 메모리 경쟁 상태가 되어 문제가 발생하게 된다. 메모리 경쟁 상태를 해결하기 위하여 CUDA에서 제공되는 atomicAdd 함수를 사용하였다. 전처리 모듈은 화소 단위의 단순 연산으로 이루어져 있기에 GPU 적용에 많은 효과를 볼 수 있는 기능이다.

블록 오류 검출 알고리즘은 필터 연산이 주 연산이기에 GPU 적용이 용이하였다. 각 화소값에 1x3 최대/최소 필터를 적용한 후, 최대/최소 간의 차이를 구하여 세부 라인이 들어가 있는 블록 패턴만을 극대화한 후, 센서스 변환(Census Transform) 필터를 적용하여 주변 화소로부터 지역 패턴을 생성하여 세부 라인이 포함된 블록 패턴을 검출하였다. CPU 소요 시간이 가장 긴 알고리즘이기에 GPU 검사 시간 단축이 가장 중요한 모듈이다.

4. 결과

개발된 품질검사 기능 중 CPU 처리만을 사용한 경우와 GPU 모듈을 적용한 경우에 대하여 각각 수행 시간을 측정하여 <표 1>에서 정리하였다. 영상 한 프레임의 재생 시간은 33.3ms이므로, 이 시간 이내로 검사가 가능해야 실시간 검사를 지원할 수 있다. 실험 서버는 AMD FX-8350 8코어 CPU, 8GB RAM, NVidia GTX 780 ti GPU 2개, windows 7 64비트를 사용하였다.

비교 항목	CPU 처리	GPU 처리
전처리 모듈 (1프레임) (흑백변환, 히스토그램)	8~10ms	2ms
블록 오류 (1프레임)	30~42ms	15~16ms
CPU 사용량	70~80%	30~40%
전체 검사 수행 시간 (5분 길이, 1프레임)	9분28초, 63ms	4분55초, 33ms

표 1. 단일 프레임 이미지 처리 속도 비교

<표 1>의 실험 결과를 보면 병렬화 요소에 따라서 모듈당 GPU 처리 시간이 50~80% 정도 향상된 것을 알 수 있다. 전처리 모듈의 경우 병렬화 효율이 좋아 80% 정도의 성능 향상이 되었으며, 조금 더 복잡한 블록 오류 검사의 경우에도 50% 이상 성능이 향상되었다. CPU 사용량에서도 GPU 처리에 의해 40% 정도 CPU 사용량이 줄어

든 것을 확인할 수 있다. 블록, 프리즈 프레임, 블록 스트라이프, 컬러 바, 단색, 볼륨 레벨, 뮤트 등을 검사하도록 설정한 5분 길이의 콘텐츠에 대한 전체 검사 수행 시간을 비교해보면 CPU 처리의 경우 실시간의 2배 가까이 수행 시간이 소요되는데, 전처리와 블록 오류 검사를 GPU로 구현한 전체 검사의 경우 수행 시간은 실시간 검사 속도 이내에 검사가 완료되었다.

CPU 처리의 경우 조건 연산에 의한 분기가 달라져 모듈별 실행 시간이 이미지 형태에 따라서 2배 가까이 차이가 나는 경우가 종종 발생하였다. 반면 GPU 처리의 경우 이미지 형태와 관계없이 각 화소마다 동기화된 처리를 하게 되므로 거의 일정한 실행 시간이 걸리는 것을 확인할 수 있었다. 따라서 GPU 처리가 일관된 실행 시간을 보장해 주기에 더욱 예측 가능한 수행 결과를 제공한다는 장점이 있다.

이 외에도 블록 스트라이프, 프리즈 프레임 등의 다른 검사 알고리즘에도 GPU 병렬처리를 적용할 경우 검사 시간은 더욱 단축될 것으로 기대된다.

5. 결론

방송사에서는 고품질의 콘텐츠를 시청자에게 서비스하여 서비스 만족도를 높여야 한다. 이를 위하여 콘텐츠 품질검사를 세부적으로 수행할 필요가 있으며, 이러한 목적으로 자주 발생하는 A/V 오류를 자동으로 검출할 수 있는 품질검사 시스템을 개발하여 방송 시스템에 적용하여 왔다. 실시간, 대용량 시스템에서는 검사 시간 단축에 대한 요구 사항이 중요하게 도출 되었기에, 검사 속도를 높이기 위하여 CPU 병렬처리 기법과 최근 떠오르고 있는 GPU 병렬처리 기법을 적용하여 실시간 고속 검사를 구현하였다. 현재는 전처리 모듈과 가장 복잡한 블록 오류 검출에 GPU 모듈을 적용하였는데, 추후 다른 검사 알고리즘에도 GPU 병렬처리를 적용한다면 검사 속도는 더욱 향상될 것으로 기대된다.

방송 콘텐츠에서 검출된 블록, 블록 스트라이프 등의 A/V 오류에 대하여 검출 이후의 대처 방안에 대한 요구가 있다. 현재는 깨끗한 원본을 찾아 재인코딩 등의 과정을 거쳐 해결하고 있지만, 원본에 오류가 포함되어 있는 경우 또는 한 프레임 오류 등 작은 오류의 경우에는 여전히 콘텐츠에 남아있게 된다. 이러한 오류들을 복원하여 콘텐츠 품질을 높일 수 있도록 오류 복원 연구도 수행하고 있으며, 오류 검출과 복원이 일원화된 콘텐츠 품질관리 워크플로우의 완성을 목표로 연구를 진행할 계획이다.

6. 참고문헌

1. Thomas Dove, "The missing link in file-based video: Automated QC", NAB 2006
2. Dave Guerrero, "Use of automated file based quality control tools in a broadcast environment", IBC 2007
3. 정영훈, "CUDA 병렬 프로그래밍", 프리렉, 2011