강진욱
제머나이소프트 CTO

인코딩 중인 영상의 실시간 미디어 처리 기술

강진욱 제머나이소프트 CTO

글을 시작하며

영상의 데이터가 커지고 다양한 효과로 인하여 영상의 복잡도가 높아지는 추세는 파일을 트랜스코딩하고 처리함에 있어 더 많은 프로세싱 파워를 요구하고 있다. 특히 SD에서 HD로, 다시 Stereoscopic 3D와 UHD로 데이터의 용량은 화질과 함께 끊임없이 증가하고 있다.

현재 HD 영상의 경우 DVCPRO-HD[1], XDCAM HD[2], AVC-Intra[3] 등의 영상이 주로 사용되는데 제작 장비의 특성에 따라 변환 작업이 필요한 경우가 있다. 예를 들어 SONY 사의 XDS 시리즈 비디오서버의 경우 XDCAM HD 영상만을 지원하는 등 제작에서 송출, 아카이브에 이르기까지 다양한 영상 소스를 사용하다보면 Transcoding이 필요한 경우들이 종종 나타난다. 그 외에도 비디오 영상을 처리하면서 시간이 소요되는 기술들은 전송, 카탈로깅, 품질검사(Quality Checking) 등 다양하게 존재한다.

특히 CMS, NPS와 같은 협업을 중심으로 한 제작 과정에서는 파일을 다른 포맷으로 전환하는 Transcoding(혹은 Re-Encoding)의 과정과 Scene Detecting 등의 작업으로 인하여 대기 시간이 증가하는 비효율성이 생길 수 있다. 그렇다면 Encoding하는 도중에 영상을 처리할 수 있는 방법은 없는가? 부족하지만 이 문서는 이에 대한 해답을 제공하고자 한다.

본론

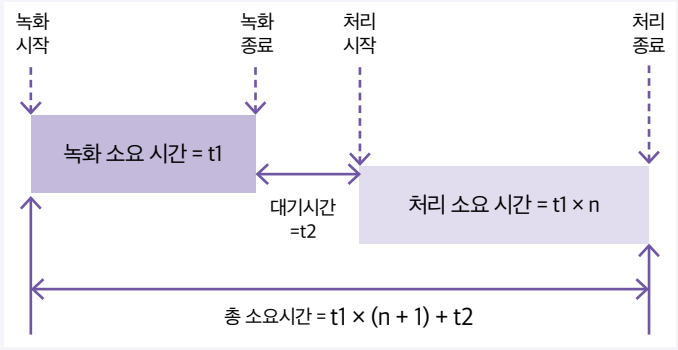
비디오의 실시간 처리 기술이란 비디오가 생성(Encoding)되는 도중에 생성되는 비디오를 이용하여 작업을 하는 경우를 의미한다. 완성되기 전 영상 파일을 이용한 작업을 수행할 수 있으므로 대기 시간을 줄일 수 있어 전체 소요 시간을 줄일 수 있다.

[그림 1]은 비디오 영상을 녹화하고 처리가 순차적 처리(Batch Processing)될 때의 총 소요 시간을 보여주고 있다. 녹화 시간이 t1이라고 하였을 때, 처리에 실시간에 대비한 처리 속도를 n이라고 하였을 때, 총 소요시간은 녹화시간(t1)과 처리 시간과 녹화완료 후 처리까지 포함되는 run-in 타임인 대기시간(t2)를 을 합한 t1 × (n + 1) + t2가 소요된다.

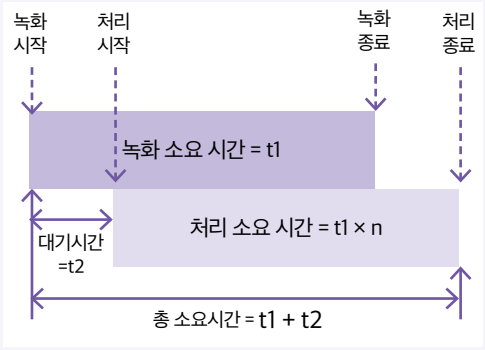
1) DVCPRO-HD는 아리랑국제방송과 현대홈쇼핑, MBC 제작에서 널리 사용되고 있음. MBC 제작은 XDCAM HD와 혼용 사용

2) XDCAM HD 422는 KBS, SBS, EBS, MBC 등에서 송출 및 제작 포맷으로 가장 널리 사용하는 포맷임

3) AVC-Intra 포맷은 Panasonic P2 카메라를 이용하는 MBC 뉴스 등에서 주요 포맷으로 사용 중



[그림 1] 순차처리 시 총 소요 시간



[그림 2] 실시간 처리 시 총 소요 시간

[그림 1]에서 비디오 영상 처리에 소요되는 시간을 줄일 수 있는 방법은 n, 즉 처리 속도를 줄이는 방법밖에는 없다. 그러나 다음 [그림 2]에서는 이를 실시간으로 처리하면서 전체 소요 시간을 줄일 수 있는 방법을 보여준다.

[그림 2]의 실시간 처리에서는 [그림 1]에 비하여 처리에 사용되는 전체 소요 시간이 줄어든 것을 확인할 수 있다. 영상을 처리하기 위해 기본적으로 필요한 run-in 시간인 대기 시간(t2)만큼 녹화가 종료된 후 작업이 진행된다는 것 외에 추가로 영상을 처리하기 위한 시간이 소요되지 않는다. 그렇다면 이러한 녹화 중 실시간 처리를 지원하는 영상에는 어떠한 것들이 있는지 방송에서 대표적인 MXF와 MOV 포맷을 통하여 살펴보도록 한다.

방송용 영상에서 트랜스코딩(Transcoding) 기술이 필요하다는 것은 영상의 압축(Coding) 방식(MPEG1, MPEG2, MPEG4, h.264, DV, DVCPRO 등) 및 저장(Container) 방식(AVI, MKV, MP4, MXF, MOV, ASF 등의 확장자로 보이는 것이 주로 저장 방식)에도 다양한 기술이 있음을 의미한다. 이 중에서 실시간 영상 처리 가능 여부를 결정하는 것은 압축(Coding) 방식이 아니라 저장(Container) 방식이다.

물론 일부 영상의 경우 기본적으로 실시간 처리가 불가능한 저장 방식임에도 불구하고 압축 방식의 특이성에 의해 실시간 처리가 가능한 경우도 존재한다.

MXF 컨테이너(Container)

방송용 MXF에서 각 프레임의 위치를 알려주는 데이터는 모두 Index Table(Time Offset을 Byte Offset으로 바꿔줄 수 있는 테이블)에 저장된다. 이 Index table에는 Intra Frame 방식의 영상에서 Inter Frame(IBP에서 LongGOP까지) 영상에 이르기까지 각 프레임의 위치를 나타내기 위한 기술들이 포함되어 있다.

Intra Frame(혹은 I Frame Only 영상) 영상의 경우, Frame의 크기가 모두 동일하다. 이 경우 Edit Unit이 하나의 프레임 크기를 의미하므로, 다음의 계산식이 Time Offset의 위치를 알려준다.

$$T_{o,n} = \frac{Position_n + Origin_n}{EditRate_n}$$

[수식 1] Time Offset의 계산식

[수식 1]에서 나타난 것처럼 I Frame만으로 이루어진 영상에서 Time Offset을 구하는 것처럼 Position(Byte Offset)을 구하는 것 역시 위 계산식에서 추출할 수 있다.

Item Name	Type	Len	Local Tag	UL Designator	Req ?	Meaning	Default
Index Table Segment	Set Key	16		Table 16	Req	An Index Table Segment set	
Length	BER Length	var			Req	Set Length(see 8.3)	
Instance ID	UUID	16	3C.0A	01.01.15.02	Req	Unique ID of this instance [RP 210 The ISO/IEC 11578 (Annex A) 16 byte Globally Unique Identifier]	
Index Edit Rate	Rational	8	3F.0B	05.30.04.05	Req	Edit Rate copied from the tracks of the Essence Container [RP 210 Specifies the indexing rate in hertz]	
Index Start Position	Position	8	3F.0C	07.02.01.03.01.0A	Req	The first editable unit indexed by this Index Table segment measured in File Package Edit Units [RP 210 Specifies the position relative to start of essence, in edit units, where indexing starts]	
Index Duration	Length	8	3F.0D	07.02.02.01.01.02	Req	Time duration of this table segment measured in Edit Units of the referenced Package [RP 210 Specifies the duration of an Index table in content units]	
Edit Unit Byte Count	UInt32	4	3F.05	04.06.02.01	D/Req	Defines the byte count of each and every Edit Unit in the Essence Container. A value of 0 defines the byte count of Edit Units is only given in the Index Entry Array [RP 210 The length of an edit unit (in Bytes) in the container]	0
IndexSID	UInt32	4	3F.06	01.03.04.05	D/Req	Stream Identifier (SID) of Index Table [RP 210 Index table stream ID]	
BodySID	UInt32	4	3F.07	01.03.04.04	Req	Stream Identifier (SID) of the indexed Essence Container [RP 210 Essence (or its container) stream ID]	
Slice Count	UInt8	1	3F.08	04.04.04.01.01	D/Req	Number of slices minus 1 (NSL) [RP 210 Number of sections indexed, per edit unit, minus one]	0
PosTableCount	UInt8	1	3F.0E	04.04.04.01.07	Opt	Number of PosTable Entries minus 1 (NPE) [RP 210 Number of position offsets indexed, per edit unit, minus one]	
Delta Entry Array	Array of DeltaEntry	var	3F.09	04.04.04.01.06	Opt	Map Elements onto Slices (see Table 20) [RP 210 Array of values used to identify elements of Essence within an edit unit]	(0,0)
Index Entry Array	Array of IndexEntry	var	3F.0A	04.04.04.02.05	D/Req	Index from Edit Unit number to stream offset (see Table 21) [RP 210 Array of values used to index elements from edit unit to edit unit]	

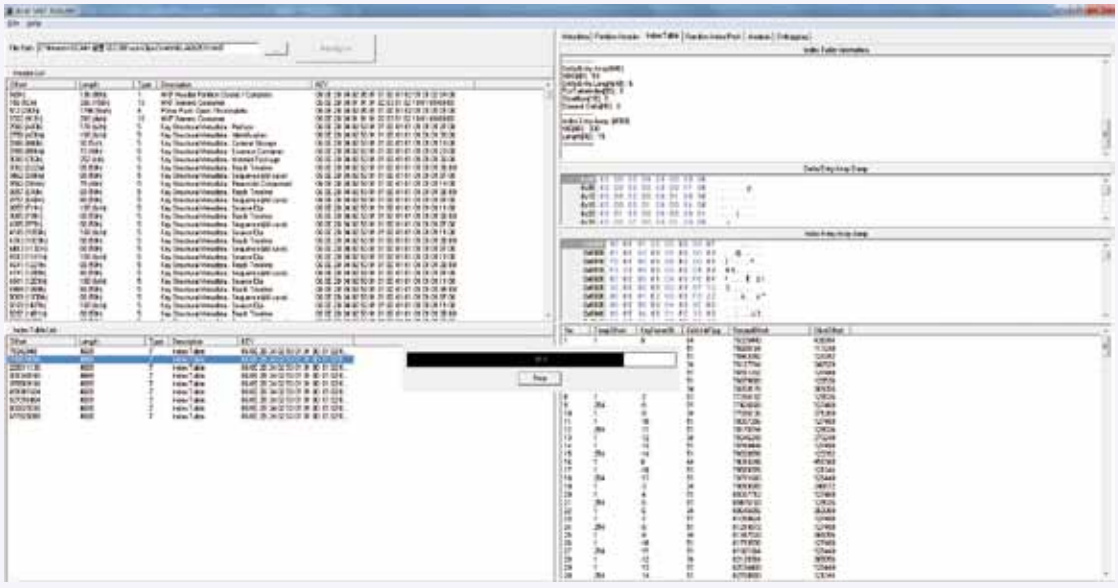
[표 1] Index Table 구조(SMPTE377M)

반면에 Edit Unit이 동일한 Intra Frame 영상에 비해 Inter Frame 영상은 각 프레임의 위치를 Index Table에서 참조해야 한다. [표 1]의 Index Table 구조에서 Index Entry Array의 배열구조가 하나의 Frame 위치를 알 수 있도록 한다. 또한 Index Entry Array에는 프레임 Position과 프레임을 Decoding 하기 위한 Key Frame과의 프레임 차이와 Flag를 통한 I, B, P 프레임 구분을 함으로써 프레임의 File Position을 획득할 수 있다.

강진욱
제머나이소프트 CTO

표만을 보고 있다면 하나의 의문점이 들 수 있다. 'Index Table이 하나만 존재한다면 모든 프레임에 대한 Index Entry를 배열로 구성해서 저장하기 전까지는 Decoding을 할 수 없고, 따라서 LongGOP 계열의 영상은 실시간 Decoding이 불가능하지 않은가?'하는 질문이 바로 그것일 것이다.

물론 이것으로 녹화 중인 영상에 대한 실시간 Decoding이 가능해지지는 않는다. MXF는 실시간 Decoding의 지원을 위해 Scattered(분산된) Index Table 구조를 지원한다. SONY XDCAM HD 영상에서 대표적으로 보이는 이 Scattered Index Table은 지정된 Interval 마다 생성됨으로써 영상의 실시간 Decoding을 가능하게 한다.(SONY XDCAM HD의 경우 10초마다 Partition과 Index Table이 생성된다)



[그림 3] XDCAM 영상의 Index Table

[그림 3]은 XDCAM 영상에서의 Index Table List를 보여주고 있다. 약 10초마다 생성되는 Index Table은 영상이 생성된 후 10초 이후부터는 이 Index Table을 이용하여 영상을 Decoding할 수 있음을 보여주고 있다. MXF의 경우 저장되는 영상이 Inter Frame 계열이건 Intra Frame 계열이건 가리지 않고 실시간 처리를 할 수 있는 것을 확인할 수 있다.

MOV 컨테이너

그렇다면 MXF와 함께 방송에서 많이 사용되는 애플(Apple)의 Quicktime, 즉 MOV 영상은 어떠한가? 일단 애플 매킨토시에서 구동되는 Final Cut Pro 7 NLE에서 보면 일부 영상의 경우 실시간 편집이 가능하다고 한다. 그럼에도 불구하고 MOV는 MXF에 비하여 실시간 처리가 어려운 포맷으로 알려져 있다.

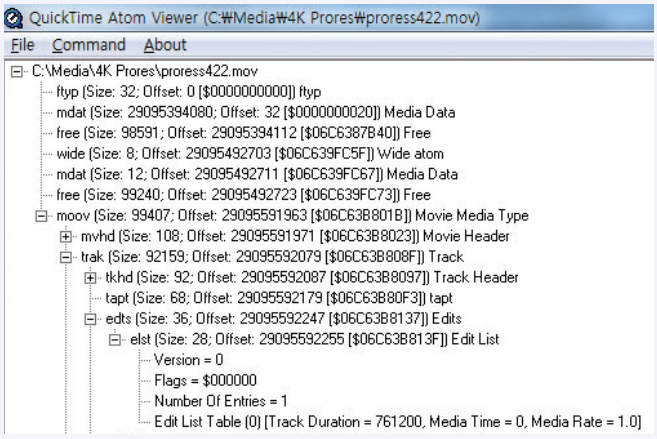
[그림 4]에서 분석된 파일은 ProRes 코덱의 영상이다. 이 영상에서 가장 큰 공간을 차지하고 있는 ATOM은 mdat ATOM임을 그림에서 확인할 수 있다(29,095,384,080 약 29GB). 즉 이 데이터가 영상과 오디오 데이터임을 확인할 수 있는데, 이 영상 데이터에서 각 Frame의 위치를 찾기 위한 Index 값은 Offset 29,095,592,079(약 29GB의 위치)에 있는 trak ATOM 내에서 찾아볼 수 있다.

이것이 의미하는 바는 그림과 같은 MOV를 녹화 중일 때에는 Index를 이용하여 영상에서 각 Frame의 Byte Offset을 확인할 수 없다는 것이다. 이렇게 Index를 이용하여 영상에서 녹화 중에 Byte Offset을 찾을 수 없다면 녹화 중 실시간 Decoding은 불가능할 수밖에 없다. 그렇다면 Final Cut Pro 7 등에서 녹화 중인 MOV 영상을 편집했던 경우가 있는 사용자들이 있을 것이다. 그 경우에 대한 설명은 어떻게 되는 것인가 궁금한 독자들이 있을 것이다.

[그림 4]의 영상은 ProRes Codec의 영상이다. 이 Prores Codec은 알다시피 Intra Frame 계열 Codec이며 매 프레임의 사이즈가 동일

하다. 즉 mdat 시작 지점부터 각 프레임에 대한 위치가 프레임마다 동일하게 증가하므로 Time Offset을 Byte Offset으로 예측할 수 있다.

그러나 이러한 설명만으로는 부족하다. Apple의 Quicktime 포맷의 경우 각 ATOM 구성에 대한 자유도가 대단히 높아 moov(Movie Media Type) ATOM을 중간중간 여러 번 배치할 수 있고, 실제 MXF의 Scattered Index Table처럼 활용할 수 있다. 이러한 기능을 이용하고 있는 Encoder가 Tools On-Air 사의 Just:In과 같은 제품이며, 이 기능을 이용하여 녹화 중인 일부 영상이 NLE에서 편집될 수 있다.



[그림 4] Prores 영상 ATOM Tree

MOV 영상의 경우에는 위에서 설명한 바와 같이 한 두 번 정도 테스트를 하고, '본 영상은 녹화 중 실시간 편집이 가능한 영상입니다'라고 할 수 없는 포맷이다. 동일한 Codec과 profile을 이용하는 경우라도, Encoding 시스템마다 다른 결과를 보일 수 있다는 것을 항상 염두에 두어야 하며, 기본적으로 녹화 중 실시간 Decoding이 쉽지 않은 포맷이라고 보수적으로 판단하는 것이 향후 큰 문제의 소지를 줄일 수 있다.

그 외의 실시간 처리가 불가능한 포맷들

Quicktime(MOV) 컨테이너가 애플의 대표적인 컨테이너라면 AVI(Audio Visual Interleave)는 마이크로소프트(Microsoft, MS)가 만든 대표적인 포맷 중 하나이다. 이 AVI 포맷 역시 index가 필요하며, index를 정리하는 작업은 모든 Encoding이 종료된 뒤에 일어나므로 Encoding과 함께 실시간 처리는 불가능한 것을 알 수 있다.

MP4 포맷 역시 MOV와 동일한 ATOM 구조를 가지고 있으며 Edit List Table이 필요한 포맷으로 대부분 Encoding 종료와 함께 프레임별 index 값을 저장함으로 실시간 처리가 불가능하다고 판단할 수 있다.

결론

얼마 전만 하더라도 실시간 미디어 처리 기술은 영상 인제스트 과정에 저해상도를 빨리 생성하는 것 외에 딱히 고민되지 않았다. 하지만 시간이 가면서 실시간 미디어 처리 기술에 대한 요구사항은 늘어나는 편이다.

최근 방송에서 회자되는 단어 중에 OVP(Online Video Platform), OTT(Over-the-top)라는 단어들이 있다. 그만큼 전통적인 방송 외의 다양한 방송 환경에 대한 고민을 하고 있다는 의미일 것이다. 그런데 이를 위해서는 방송용 포맷뿐 아니라 다양한 디바이스와 플랫폼을 위한 다수의 서비스 영상이 필요할 수밖에 없다. 그리고 이러한 Multi profile encoding 기능을 기존의 방송용 비디오서버들은 잘 제공하지 않는다. 최근 방송 시장에 판매되는 Encoder 제품 중 상당 부분을 차지하는 것이 OTT Encoder 제품들인 것도 이러한 이유일 것이다.

하지만 녹화와 동시에 실시간 트랜스코딩이 가능하다면, 이러한 OTT용 Encoder를 새롭게 구매하는 대신 가지고 있는 트랜스코더를 업그레이드하면 되지 않겠는가? 즉 현재 가지고 있는 고가의 방송 장비들에 일부 시스템을 추가함으로써 OTT용 Multi profile 실시간 transcoder를 만들 수 있다는 의미이다.

그 외에도 Encoding 완료와 동시에 편집이 가능한 NPS 시스템 구축, 외부에서 업로드를 하고 있는 영상에 대한 트랜스코딩 등 다양한 형태의 활용들이 가능할 것이다. 실제 이번 채널에이에 구축한 Edius 용 Growing 편집 기능도 이러한 실시간 Decoding 기술을 응용해서 제작된 것을 고려하였을 때 방송용 영상에 대한 실시간 처리 기술의 활용 방안은 다양하게 발전될 수 있으리라고 사료된다.