

비선형 압축 영상 편집기 구현 알고리즘

박 종 준, 정 민 교, 이 진 호, 송 문 호, 김 운 경
고려대학교 전기·전자·전파공학부 통신신호처리 연구실
서울 성북구 안암동 5가 1번지 고려대학교
전화 : (02) 927-6443 / 팩스 : 927-0544

Algorithm for Realization Nonlinear Compressed Domain Video/Audio Editor

Jong-Jun Park, Min-Gyo Chung, Jin-Ho Lee, Moon-Ho Song, Woon-Kyung Kim
Communication Signal Processing Laboratory(CSP Lab)
School of Electrical Engineering, Korea University
E-mail: jjpark@davinci.korea.ac.kr

Abstract

In this paper, we report on a set of new algorithms to realize a nonlinear compressed-domain video/audio editor that overcomes various realization problems. For efficiency, the underlying algorithm, which uses a central data structure in the form of doubled linked lists, performs soft edits of cut and paste (which, in turn, involves soft implementations of frame type conversion) and addresses problems relating to video/audio synchronization and random access, and decoder buffer control.

I. 서론

사람들은 미래를 멀티미디어 시대라 말하고 있다. 이는 앞으로 비디오를 주축으로 소리, 텍스트와 같은 여러 가지 정보들이 결합된 멀티미디어 정보가 중요한 역할을 하게 될 것임을 의미하고 있다. 이러한 멀티미디어 정보의 중요성이 널리 인식되면서 각종 전자 기기 및 서비스들이 멀티미디어 정보를 취급 하고자 하는 쪽으로 발전하고 있다. 멀티미디어 정보 중 핵심이 되는 것은 비디오로서 특히 MPEG기반의 비디오는 다양한 응용을 위한 중요한 공통적인 매체로써 널리 쓰이고 있다. 따라서 MPEG기반 비디오를 어떻게 하면 쉽고 편하게 다룰 수 있느냐 하는 문제가 매우 중요하

다. 이런 관점에서 어느 비디오든지 촬영 순서에 관계 없이 임의로 접근하여 비디오를 편집할 수 있고, 프레임 단위 cut and paste, 여러 가지 특수 효과, 그 이외의 여러 가지 복잡한 기능까지 포함하는 비선형 비디오 편집기는 비디오 데이터 베이스 및 라이브러리 구축에 이용될 수 있으며 진보된 미래의 멀티미디어 서비스 제공에 기여할 수 있을 것으로 예상된다. 본 논문에서는 새로운 비선형 압축 영상 편집기 구현 알고리즘을 제시하고 기존에 나와 있는 영상 편집기에서의 여러 가지 문제점을 극복하는 세부 알고리즘을 설명하며 마지막으로 결론을 내릴 것이다.

II. 새로운 비선형 압축 영상 편집기 구현 알고리즘

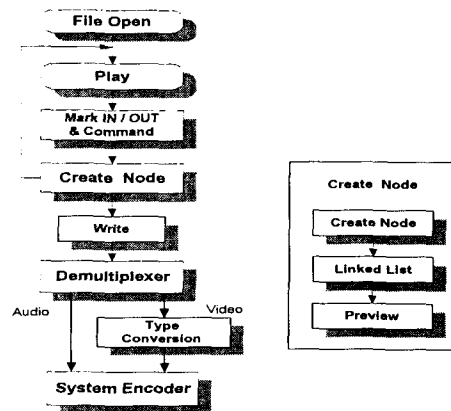


그림 1. 새로운 영상 편집기 Block Diagram

이 영상 편집기가 가지게 될 기본적인 기능으로는 우선 MPEG 디지털 영상에 대한 임의 접근 기능 (frame 단위 또는 shot 단위)과 cut and paste 기능 그리고 dissolve, fade등과 같은 각종 특수 편집 기능 등이 있다. 그림1의 block diagram에서 보듯이 편집 단위는 모두 doubled linked list 구조를 가지는 노드들로 이루어지며 사용자가 편집 명령(cut and paste 포함)을 내릴때 편집시 필요한 정보들은 각각의 노드들 속에 저장되고 최종적으로 편집된 노드들의 정보를 통해 새로운 편집된 파일을 생성하게 된다. 여기서 doubled linked list로 구성된 노드들을 따라 파일을 생성할 때 부분적인 Demux(비디오와 오디오 분리) 및 비디오 타입 변환을 수행하게 된다.

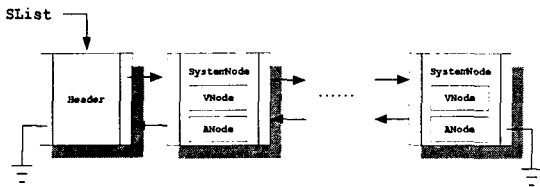


그림 2. 시스템 노드의 링크트 리스트

위의 그림 2는 영상 편집기의 기본적인 기능인 cut and paste가 동작할 때 발생하는 자료 구조로서, MPEG-1 stream으로부터 사용자가 편집할 구간에 mark in, mark out의 동작을 수행하면, 그 구간에 해당하는 비디오 스트림, 오디오 스트림의 자료를 편집의 기본이 되는 세그먼트 단위로 메모리에 저장하게 된다. 구체적으로는 다음과 같은 부가적인 정보가 추가된다. 비디오와 오디오의 동기를 맞추기 위해 필요한 시간정보, system layer로의 접근을 위한 packet에 대한 정보를 가지고 있으며, 비디오에서는 mark in, mark out 위치에 대한 파일 포인터, 화면형태에 따른 GOP 범위 내의 기준 프레임, 화면 형태 변환을 위해 필요한 정보, 비트 rate, 프레임 번호, 시간정보 등의 정보와 오디오에는 비트 rate, 프레임 번호, 샘플링 주파수 등의 정보를 갖고 있게 된다.

III. 세부 기술적인 알고리즘

구현하고자 하는 비선형 압축 영상 편집기의 세부 기술적인 알고리즘은 크게 두 가지로 분류할 수 있다.

1. MPEG 디지털 영상에 대한 임의 접근 기능
 - 가. 비디오
 - 나. 오디오
2. MPEG 디지털 영상에 대한 cut and paste 기능
 - 가. 프레임 타입의 변환
 - 나. 편집시 비디오와 오디오 동기화
 - 다. 디코더 버퍼 컨트롤

1. MPEG 디지털 영상에 대한 임의 접근 기능
 - 가. 비디오

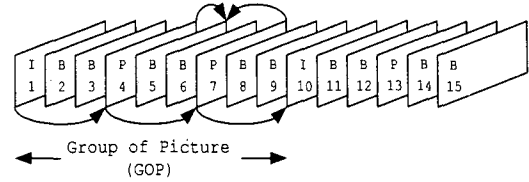


그림 3. 일반적인 MPEG 비디오 프레임 시퀀스

그림3에서 처럼 I 프레임은 그 내부에 디코딩에 필요한 모든 정보를 담고 있으므로 다른 정보 없이도 디코딩이 가능하다. 그러나 P 프레임은 선행하는 I 프레임이나 P 프레임으로부터, B 프레임은 전후의 가장 가까운 I 또는 P 프레임의 두 프레임으로부터 예측을 하므로 단독으로 디코딩을 할 수 없다. 그리고 I, P 프레임은 비트 스트림상 순서와 디스플레이 순서가 다르므로 각 프레임을 접근할 때 기술적인 알고리즘이 필요하다. 반면 B 프레임은 비트 스트림상 순서와 디스플레이 순서가 같으므로 쉽게 임의 접근이 가능하다. 여기서 새롭게 제시하는 각각의 프레임에 대한 임의 접근 방법은 표준 MPEG 스트림에서 시퀀스 헤더에 있는 프레임 rate과 비트 rate 그리고 패킷헤더에 있는 PTS (presentation time stamp) 정보를 이용하는 방법이다. 이에 대한 알고리즘은 다음과 같다.

1) 임의 접근하고자 하는 비디오 프레임에 대한 PTS를 구한다. $PTS = \frac{\text{임의 프레임수}}{\text{프레임rate}}$

2) 위에서 구한 PTS와 가장 가까운 PTS (= PTS')를 가지는 패킷을 검색하여 이동한다.

3) PTS와 PTS'의 차이에 프레임 rate을 곱하여 PTS'에서 위치에서 접근하고자 하는 비디오 프레임까지 남은 프레임 수를 계산한다.

$$(PTS - PTS') \times \text{프레임 rate} = \text{남은 프레임 수}$$

4) 위에서 구한 남은 프레임 수만큼 비트 스트림상에서 이동한 다음 I, P 프레임일 때 바로 전 I, P 프레임이 바로 찾고자 하는 프레임이고 B 프레임일 때는 그 프레임이 바로 찾고자하는 프레임이다.

나. 오디오

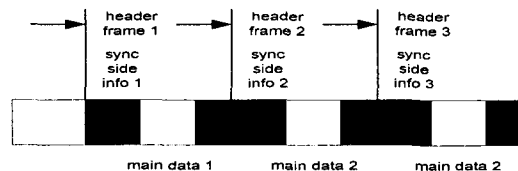


그림 4. MPEG-1 오디오 프레임 시퀀스

MPEG-1 오디오 프레임 시퀀스는 프레임 단위로 정보가 독립적으로 들어 있으므로 참조하는 프레임이 필요없이 디코딩이 가능하다. 따라서 오디오의 임의 접근 기능은 비교적 간단하다 할 수 있다. 임의 접근하고자 하는 비디오의 프레임에 대한 오디오의 프레임을 찾을 때 오디오 샘플링 주파수와 패킷 헤더의 PTS를 이용하여 구할 수 있다. 이에 대한 알고리즘은 다음과 같다.

1) 임의 접근하고자 하는 비디오 프레임에 대한 PTS를 구한다.

$$PTS = \frac{\text{임의 프레임수}}{\text{프레임rate}}$$

2) 위에서 구한 PTS와 가장 가까운 오디오의 PTS (= PTS')를 가지는 패킷을 검색하여 이동한다.

3) 오디오에서 한 프레임이 걸리는 시간을 계산

$$T_{\text{프레임}} = \frac{\text{프레임크기}}{\text{샘플링 주파수}}$$

4) PTS와 PTS'의 차이에 오디오에서 한 프레임이 걸리는 시간(= $T_{\text{프레임}}$)을 나눔으로써 PTS'에서 위치에서 접근하고자 하는 오디오 프레임까지 남은 프레임수를 계산한다.

$$\frac{(PTS - PTS')}{T_{\text{프레임}}} = \text{남은 오디오 프레임 수}$$

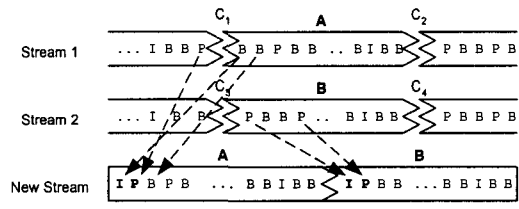
5) 위에서 구한 남은 오디오 프레임 수만큼 비트 스트림상에서 이동하면 우리가 임의 접근하고자 하는 오디오 프레임을 구할 수 있다.

2. MPEG 디지털 영상에 대한 cut and paste 기능

현재 선보이고 있는 비선형 압축 영상 편집기들은 대부분 GOP 단위의 편집 기능만을 제공하고 있는데 GOP단위의 영상 편집 기능은 기술적으로 영상의 움직임 보상에 따른 문제를 고려하지 않아도 되는 반면, 보다 정교한 편집을 원하는 사용자들에게는 만족할 만한 수준은 아닐 것이다. 따라서, 사용자가 원하는 위치에서의 편집을 가능케 하기 위해 프레임 단위의 편집 기능의 구현은 비선형 압축 영상 편집기의 필수적인 기능이 된다. 프레임 단위로 cut and paste 기능을 구현 할 때 두 가지 문제가 생길 수가 있는데, 하나는 프레임 타입 변환 문제이고, 또 한 가지는 디코더 비디오 버퍼 컨트롤 문제이다. 여기서는 완전한 디코딩 및 재인코딩 과정을 하지 않고 필요한 부분의 Demux (비디오와 오디오 분리) 및 타입 변환을 통한 재인코딩을 기본으로 한다. 이점은 첫째, 완전한 디코딩 및 재인코딩 과정을 하지 않으므로 계산량이 줄어들기 때문에 처리 속도가 빨라진다는 것이고, 둘째, 부분 Demux 및 타입변환을 통한 재인코딩을 함으로써 재인코딩시 나타나는 화질의 저하를 최소화할 수 있다.

가. 프레임 타입의 변환

MPEG 스트림은 I 프레임으로 시작되는 GOP를 단위로 이루어져 있는데, 서로 다른 영상에서 세그먼트로 자를 때 잘려지는 프레임이 I, B, P 모두 해당될 수 있다. 따라서 두 MPEG 스트림을 잘라서 이을 때에 새로 생성된 스트림이 완벽하게 디코딩이 되도록 하기 위해서는 잘려진 부분들의 시작 프레임을 I프레임으로 그 다음 프레임을 P 프레임으로 바꾸어주어야 한다. 그리고 잘려진 부분에서 바로 뒤에 나오는 I, P 프레임까지의 부분 디코딩 및 재인코딩이 필요하다. 이렇게 하면 잘려진 스트림은 완전한 하나의 GOP를 이루게 되므로 이러한 스트림들을 연결하였을 때에도 그 스트림은 완전한 디코딩이 가능하다. 그림 3은 두 스트림의 세그먼트 A(C1-C2), B(C3-C4) 부분을 연결하여 새로운 스트림(New Stream)을 생성하는 과정이다.



---▶ : 타입 변환
A, B : 편집하고자 하는 세그먼트

그림 5. MPEG 스트림에서의 cut and paste 과정
그림5에서 처럼 편집하고자 하는 세그먼트의 첫 픽처는 I 타입으로 그다음 픽처는 P 타입으로 타입 변환하면 픽처가 깨지는 것 없이 완벽한 디코딩이 가능하다.

나. 편집시 비디오와 오디오 동기화

한정된 전송률에서 비디오와 오디오에 어떤 비율로 부호량을 배분할 것인가가 시스템 품질을 좌우하는 중요한 요소가 된다. 부호화 되어 있는 비디오/오디오 비트열 배치를 보면 그림 6과 같다.

V	V	V	V	A	V	V	V	V	A	V	V
---	---	---	---	---	---	---	---	---	---	---	---

V: 비디오 팩
A: 오디오 팩

그림 6. MPEG 데이터 비디오/오디오 비트열 배치

pack start code	SCR	MUX rate	packet start code prefix	packet length	STD buffer scale	PTS	DTS	packet data
-----------------	-----	----------	--------------------------	---------------	------------------	-----	-----	-------------

그림 7. MPEG 비디오 팩

pack start code	SCR	MUX rate	packet start code prefix	packet length	STD buffer scale	PTS	packet data	0's
-----------------	-----	----------	--------------------------	---------------	------------------	-----	-------------	-----

그림 8. MPEG 오디오 팩

여기서 압축 비디오/오디오 비트열은 그림 7, 그림 8과 같이 기록된다. 비디오 신호와 오디오 신호와의 동기화는 그림 7과 그림 8에서 나타낸 세 개의 시간 정보 SCR(System Clock Reference), PTS(Presentation Time Stamp), DTS(Decoding Time Stamp)를 사용한다. 그러므로, 특정한 영상 프레임을 편집할 때 시간 정보를 이용하여 해당하는 오디오 프레임을 디코딩하면 된다.

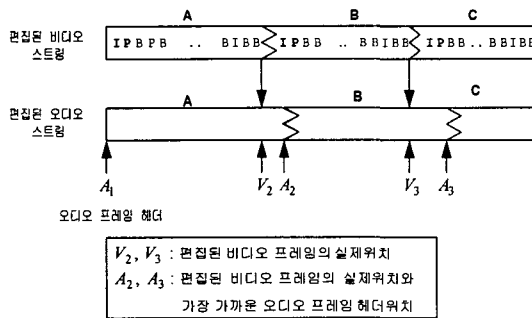


그림 9. 편집된 MPEG 비디오와 오디오 스트림

그림 9에서 보듯이 편집된 비디오 프레임의 위치와 오디오의 프레임 위치가 정확하게 일치하지 않으므로 편집된 비디오 프레임의 실제위치와 가장 가까운 오디오 프레임 헤더를 찾아서 오디오를 편집하게 된다. 이와 같이 하면 편집이 계속되면 될 수록 오디오 데이터에 대한 에러가 $V_2 - A_2$ 혹은 $V_3 - A_3$ 만큼 누적이 되므로 이에 대한 보정을 해주어야 하는데 에러 누적이 오디오 한 프레임(0.7초이내)을 넘으면 묵음 오디오 프레임을 추가하여 보정한다. 실제로 거의 정확한 비디오와 오디오 동기화가 이루어짐을 알 수 있다.

다. 디코더 버퍼 컨트롤
인코더는 MPEG 스트림을 생성할 때에 전체적인 비트 rate을 맞추기 위하여 내부에 가상 버퍼를 두고 디코더의 버퍼에서 overflow나 underflow가 일어나지 않도록 흐름을 관장한다. 즉, 생성된 MPEG 스트림은 전체적으로 비트 rate이 맞추어져있기 때문에, MPEG 스트림의 잘려진 부분들은 비트 rate을 다시 맞추어 주어야 한다. 더욱이 다른 스트림에서 잘라 이어붙인 MPEG 스트림의 경우에는 각각이 다른 비트 rate으로 인코딩되었기 때문에 생성 스트림의 비트 rate을 반드시 다시 조정을 해주어야만 한다. 낮은 비트 rate의 프레임이 연속되는 경우에는 디코더에서 충분히 빠르게 버퍼를 비우지 못하게 되어 overflow가 발생하게 되며, 반대로 높은 비트 rate의 프레임이 연속되는 경우에는 디코더에서 가정된 시간 안에 버퍼로부터 완전한 데이터를 얻을 수가 없게 되는 underflow가 발생하게 된다. overflow가 발생한 경우에는 단순히 '스터핑 0'을

채우는 것으로 해결할 수 있으며, underflow의 경우는 프레임 비트 rate을 낮추는 것이 쉽지 않다.

VI. 결론

본 논문은 비선형 압축 영상 편집기 구현에 관한 새로운 알고리즘을 제시하였다. 즉, 영상 편집기 구현에 관한 자료 구조를 제시하였으며 영상 편집 (cut and paste)시 부분 demux 및 부분 픽처 타입 변환 방법, 임의 접근 방법, 비디오 와 오디오 동기화 방법 그리고 디코더 버퍼 컨트롤 문제 해결방안을 제시하였다. 이와 같은 방법은 비선형 영상 편집기 구현에 필요한 핵심 기술이며 실제로 구현한 결과 다른 영상 편집기 보다 성능이 우수함을 알 수 있었다. 이 영상 편집기는 비디오 데이터 베이스 및 라이브러리 구축에 이용될 수 있으므로 진보된 미래의 멀티미디어 서비스를 제공하는데 상당한 기여를 할 수 있을 것으로 예상된다.

Acknowledgement

본 연구는 한국통신 멀티미디어 연구소의 지원하에 이루어졌습니다.

참고 문헌

- [1] S.-F. Chang, "Compressed-Domain Techniques for Image/Video Indexing and Manipulation", IEEE Intern. Conf. On Image Processing, ICIP 95, Special Session on Digital Image/Video Libraries and Video-on demand, Oct. 1995, Washington DC.
- [2] S.-F. Chang and D.G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video", IEEE Journal of Selected Areas in Communications, Special Issue on Intelligent Signal Processing, pp. 1-11, Jan.1995.
- [3] J.Meng and S.-F. Chang, "Tools for Compressed-Domain Video Indexing and Editing", SPIE Conference on Storage and Retrieval for Image and Video Database, Vol. 2670, San Jose, California, Feb. 1996.
- [4] S.-F. Chang, Compositing and Manipulation of Video Signals for Multimedia Networks Video Services, Ph.D. Dissertation, U.C. Berkeley, Aug. 1993.