

효율적인 프레임 제거를 위한 트랜스코더 시스템 구조

김현희^o 김성민 박시용 정기동
 부산대학교 컴퓨터공학과
 h2k@pusan.ac.kr

The Transcoder System Architectures for an efficient Frame-skipping

Hyunhee Kim^o Sungmin Kim Siyong Park Kidong Jung
 Dept. of Computer Science & Engineering, Pusan National University

요 약

트랜스코딩은 이미 압축된 비디오를 이질적인 클라이언트에게 적응적으로 전달하기 위한 해결책이다. 일반적으로, 계산량을 줄이기 위해서 제안된 트랜스코딩 기법들은 비디오 화질의 열화를 발생시키고 그와 반대의 경우는 많은 계산량을 초래한다. 이와 같은 계산량과 화질 사이의 문제를 해결하기 위해서 여러 가지 측면이 연구되어 왔다. 하지만, 대부분의 연구가 트랜스코더 내부에 한정되어 있었고, 서버 측과의 상호작용을 통한 성능 향상에 대한 연구는 적었다. 멀티미디어 데이터를 전력과 성능이 낮은 단말기 또는 낮은 대역폭의 네트워크에 속한 이질적인 클라이언트에 서비스할 때 트랜스코더 자체의 해결 방안에서 서버 측의 특정 작업을 추가할 경우 트랜스코더에서 실제 처리해야 하는 프레임의 개수를 줄일 수 있는 서비스 효율의 향상을 기대할 수 있다. 따라서 본 논문에서는 효율적인 트랜스코더와 서버 측 기반의 알고리즘을 함께 고려하여 계산 처리 과정을 줄일 수 있는 트랜스코더 시스템 구조를 제안한다.

1. 서 론

인터넷을 통한 비디오 서비스는 이제 익숙한 소재가 되었고 더불어 효과적이고 적응적인 서비스를 위한 트랜스코딩도 많은 연구가 거듭되어 여러 응용에 사용되고 있다. 동일한 멀티미디어 콘텐츠를 LAN, VDSL과 같은 높은 대역폭의 네트워크에 속한 클라이언트에 서비스할 경우와 무선 네트워크 또는 이동 통신 네트워크에 속한 클라이언트에 서비스할 경우, 특정한 네트워크의 클라이언트가 만족할 만한 서비스를 받지 못하거나 반대로 서비스를 전혀 받지 못할 경우가 발생한다. 이를 위해서 동일한 멀티미디어 콘텐츠를 서로 다른 네트워크와 클라이언트 단말을 고려한 여러 가지의 버전으로 준비하는 것은 시스템의 비용을 높게 된다. 트랜스코딩은 클라이언트가 포함된 네트워크 입구 또는 코어 네트워크의 출구에서 해당 클라이언트에 맞는 형태의 버전으로 변환할 수 있는 적응성을 제공한다. 이때, 트랜스코딩은 멀티미디어 데이터의 변환을 위해 복호화와 부호화 과정이 동시에 필요하게 되는데, 이 과정은 많은 계산량을 필요로 한다. 또한, 복호화와 부호화를 통한 화질의 열화가 한 번 더 발생하는 특징을 가진다. 이에 따라, 지금까지 적은 계산량과 높은 비디오 화질을 위한 효율적인 트랜스코딩 기법과 구조가 많이 연구되어 왔다. 하지만, 이질적인 클라이언트를 지원하기 위해서 트랜스코더에서의 복호화 과정의 처리량을 최소화할 수 있는 서버 측 기반의 알고리즘을 함께 고려한다면 트랜스코더 시스템의 성능을 더욱 더 향상시킬 수 있다.

지금까지의 연구들은 하나의 시퀀스 내에서 프레임 사이의 시간적 연관성을 고려한 움직임 벡터를 예측하거나 보정하는 기법이 적용된 트랜스코더에 한정된 내부적인 구조가 대부분이다. 본 논문에서는 기존의 우수한 트랜스코더를 이용하고 서버 측의 추가 작업을 통해 효율적으로 계산량

을 줄일 수 있는 트랜스코더 시스템 구조를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 소개하고, 제안한 기법인 효율적인 프레임 제거를 위한 트랜스코더 시스템 구조는 3장에서 설명한다. 마지막으로, 4장에서는 결론 및 향후 과제를 제시한다.

2. 관련 연구

스트리밍 서버가 효율적으로 비디오를 서비스하기 위해서는 클라이언트의 환경을 고려해야 한다. 서버 측 기반의 알고리즘의 예로, VOD 서버가 VCR 기능을 제공하기 위해서 여러 가지의 서로 다른 버전의 같은 비디오를 저장하여 서비스하는 기법이 있다. 제공되는 비디오는 속도, 방향에 구애받지 않고 서비스 될 수 있으므로, 클라이언트는 디코더의 성능을 제한받지 않고 다양한 서비스를 받을 수 있다. 그러나 서버의 저장 공간의 문제를 해결하기 위해 두 가지 스트리밍으로 Fast-Forward, Fast-Reverse 등의 기능을 제공할 수 있는 기법도 연구되었다[1].

비트율을 줄이기 위해서 가장 일반적인 복호기와 부호기를 직렬로 배치하는 픽셀-도메인 기반의 방법이 있으나 계산량이 너무 많으므로 실시간 응용에 부적합하다[2]. 최근에는 이를 보완하기 위해 입력된 비트스트림을 DCT-도메인에서 양자화된 DCT 계수를 부분적으로 복호화 하는 기법이 제안되었다[3]-[5]. 그러나 양자화 에러가 축적되어서 연이은 프레임에 영향을 미치는 "drift" 문제가 발생되고 이를 해결하기 위해서는 부가적인 계산량을 필요로 한다. 그러나 재-양자화만으로는 원하는 비트율을 얻을 수 없기 때문에 프레임을 제거하는 트랜스코더에서 이전 프레임의 주요한 움직임 벡터를 선택하거나 보정하는 기법이 제안되었다[6]-[7]. 그러나 이 기법들은 실시간에 적합하고 원하는 비트율을 얻을 수 있었지만, 부정확한 움직임 예측으로

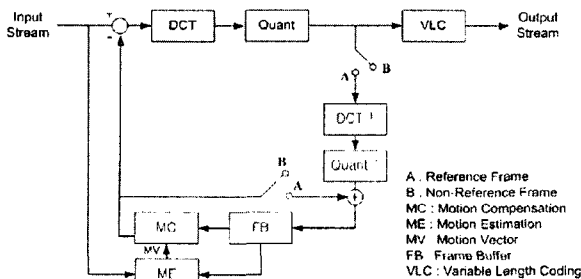
인한 화질의 열화를 피할 수 없었다.

3. 효율적인 프레임 제거를 위한 트랜스코더 시스템 구조

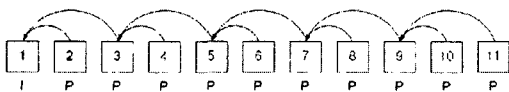
일반적으로, 프레임을 제거하는 트랜스코더에서 가장 큰 이슈는 시간적 연관성을 가지는 프레임들 사이의 예측 관계의 재구성이다. 예를 들어, 하나의 시퀀스가 I-프레임만으로 구성되었다면 이웃한 프레임간의 참조 관계는 불필요할 것이고, 프레임 제거 과정을 통해서 원하는 비트율을 얻을 수 있을 것이다. 그러나 가장 인기 있는 비디오 압축 표준인 MPEG-1/2/4 또는 H.26X로 부호화된 시퀀스의 I-프레임뿐만 아니라 P-/B-프레임들은 보다 높은 압축을 위해 시간적인 예측 관계를 가지게 된다. P-/B-프레임의 경우, 현재 프레임은 이전 프레임과의 차이 값만을 가지고 부호화 되므로 I-프레임에 비해 크기가 훨씬 작다. 프레임-제거 트랜스코더에서 삭제되는 프레임은 연이어 오는 프레임-삭제되거나 혹은 출력될-에 의해 참조되므로, 트랜스코더 내부에서는 처리되는 프레임에 해당된다. 그러므로 대부분의 삭제되는 프레임은 그것이 포함하고 있는 중요한 정보가 출력될 프레임에 활용되기 위해 단순히 제거되지 않고 부가적인 처리 과정을 겪는다.

본 논문에서는, "예측주기"를 사용하여 삭제될 프레임의 부가적인 처리를 줄일 수 있다. 여기서, 예측주기는 참조 프레임이 나타나는 프레임 간격을 말한다. 예측주기의 시작 프레임은 "예측주기-1" 개의 프레임이 모두 참조하고 있으므로, 구간 내의 하나 혹은 여러 개의 프레임이 삭제되더라도 참조되는 프레임이 버퍼에 저장되어 반복적으로 재-사용될 수 있다.

아래 나열된 그림은 본 논문에서 제안하는 시스템 구조이다. [그림 1]은 트랜스코더에 입력 스트림을 제공하는 비디오 서버의 구조이고, 부호기의 내부에 포함된 두 개의 스위치를 통해서 [그림 2]에서 설명되는 시퀀스 구조를 출력시킨다. [그림 2]는 예측주기를 통해서 생성된 프레임의 구조를 보인다.



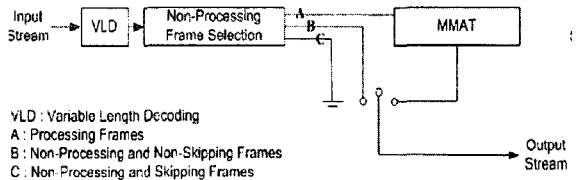
[그림 1] 서버의 부호기



[그림 2] 예측주기를 통한 프레임 구조

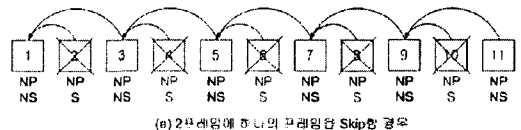
I-/B-/P-프레임으로 구성된 대부분의 시퀀스에서 참조되는 프레임간의 거리가 멀수록 참조 프레임들을 통해 예측한 차이값이 커진다. 이 점을 고려하여 본 논문에서는 프레임간의 예측주기 2를 기반으로 연속적으로 프레임을 제거하는 트랜스코더를 제안한다.

[그림 1]에서 출력되는 시퀀스는 [그림 3]의 트랜스코더에 전달된다. [그림 3]은 입력된 스트림의 프레임이 NP, P 혹은 NS, S 프레임인지에 따라 선택적으로 처리하는 트랜스코더의 구조를 설명하고 있다. 각각의 프레임이 가지게 되는 속성으로 NP는 Non-Processing, P는 Processing, NS는 Non-Skipping, 그리고 S는 Skipping을 의미한다. 대부분의 트랜스코더는 움직임 벡터를 재-사용하여 계산 복잡도를 낮추는데, 보다 정확한 움직임 예측을 위해서 움직임 보정을 수행한다. [8]에 제안된 기법은 움직임 벡터의 특징에 따라 기존의 불필요한 계산량을 줄임으로써 보다 효율적인 트랜스코딩을 수행한다. 본 논문에서는 [8]에서 제안된 기법의 움직임 재사용 기법을 개선시켜 적용한 MMAT(Motion-compensation Mode Adaptive Transcoding)를 사용한다.

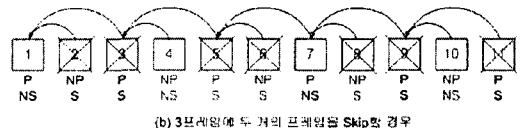


[그림 3] 트랜스코더 구조

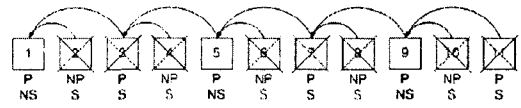
[그림 3]의 트랜스코더가 제거되는 프레임 개수의 동적인 특성을 제공하기 위해서 본 논문에서는 제거되는 프레임의 개수가 서로 다른 세 가지의 경우를 [그림 4]에 설명하고 있다.



(a) 2프레임에 하나의 프레임만큼 Skip한 경우



(b) 3프레임에 두 개의 프레임만큼 Skip한 경우



(c) 4프레임에 세 개의 프레임만큼 Skip한 경우

[그림 4] 프레임 제거의 예

(a)의 경우는 삭제되는 프레임이 출력될 프레임에 참조되지 않으므로, 부가적인 처리 과정이 전혀 필요하지 않다. (b)에서 2번 프레임이 삭제되어도 이후의 어떤 프레임도

그것을 참조하지 않지만, 3번 프레임의 경우에는 4번과 5번이 참조하고 있고 3번 또한 1번을 참조하고 있다. 결과적으로, 1번은 1-프레임이고 3번은 제거될 프레임이지만 두 프레임 모두 부가적인 처리 과정이 필요하다. 세 개의 프레임이 제거되는 (c)의 경우도 마찬가지로 삭제되는 프레임에 부가적인 처리 과정이 필요할 것이므로, (a)에서 설명된 것처럼 예측주기를 2로 정함이 더 효율적임을 알 수 있다. (b)와 (c)에 부가적인 처리 과정이 추가될지라도 기존의 제시된 프레임간의 참조 관계를 이용한 기법에 비하면 상당히 적은 계산량으로 처리가 가능하다. [그림 3]에서 Processing Frame으로 처리되어 MMAT 내부로 옮겨질 경우, 연속적으로 제거되는 프레임 수는 (식 1)과 같다.

$$\{2와 n의 최소공배수 / 2\} - 1 \dots\dots\dots (식 1)$$

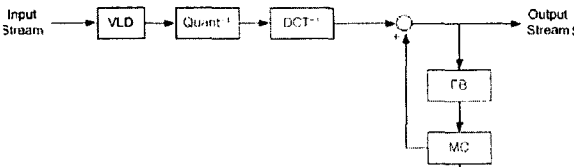
(단, n은 연속적으로 제거되는 프레임 수)

[그림 4]에서 설명된 연속적인 프레임 제거의 경우를 수식으로 정리하면 다음과 같다.

$$실제 프로세싱에 필요한 프레임 수 = \left\lfloor \frac{n}{2} \right\rfloor \dots\dots\dots (식 2)$$

(단, n은 연속적으로 제거되는 프레임 수)

[그림 5]는 보편적인 클라이언트의 복호기로서, 본 논문에서는 어떠한 기법도 적용하지 않았다. 이미 서버 측 기반의 프레임 참조 관계의 재-정의와 경량화된 트랜스코딩으로 기존의 기법보다 많은 계산량을 줄일 수 있으므로, 성능 또는 네트워크 대역폭이 낮은 환경에 속한 클라이언트도 쉽게 복호화 할 수 있다.



[그림 5] 클라이언트의 복호기

4. 결론 및 향후 과제

전통적인 복호기-부호기의 형태를 가지는 트랜스코더 구조의 너무 많은 계산량으로 인해 압축된 형태의 DCT-도메인에서 부분적인 DCT 계수만을 복호화하여 이용하는 기법이 제안되었다. 그러나 여러 축적의 문제점 때문에 연구된 재-양자화 기법도 원하는 비트율을 얻기에는 부족했다. 그래서 이전 프레임과의 연관성이 가장 높은 매크로블록의 움직임 벡터를 재-사용하거나 보정하여 사용하는 기법이 제안되어 원하는 비트율을 얻을 수는 있으나, 부정확한 시간적 연관성으로 화질의 열화가 발생하는 한계를 가진다.

본 논문에서는 시스템의 관점에서 효율적인 프레임-제거 트랜스코딩의 경량화된 구조를 제안하였다. 서버의 부호기에서는 스위치를 통해서 미리 프레임간의 참조 관계를 제

안된 기법에 따라 재구성하고, 트랜스코더에서는 처리 과정이 필요한 프레임과 그렇지 않은 프레임을 구분한다. 이후 처리 과정이 필요한 프레임에 대해서 트랜스코딩을 하게 되므로 계산 과정을 효율적으로 감소시킬 수 있다. 클라이언트의 복호기는 낮은 비트율의 트랜스코딩된 비디오를 복호화하고 출력하므로, 부가적인 처리 과정이 필요하지 않다. 따라서 클라이언트의 환경적인 요소에 의한 간섭을 최소화할 수 있으며 대부분의 단말기에 지원될 수 있다.

제안된 시스템 구조는 서버측이 트랜스코더의 경우 없이는 직접적으로 클라이언트에 서비스를 할 수 없다. 따라서 앞으로 서버와 클라이언트 간의 직접적인 서비스가 가능한 트랜스코더와 클라이언트의 구조에 대한 연구가 필요하다.

참고 문헌

[1]S. Huang, "Improved Techniques for Dual-Bitstream MPEG Video Streaming with VCR Functionalities," IEEE Trans. on Consumer Electronics, Vol. 49, No. 4, NOVEMBER 2003.
 [2]A. Vetro and C. Christopoulos and H. Sun, "Video Transcoding Architectures and Techniques: An Overview," IEEE signal Processing Magazine, ISSN: 1053-5888, Vol. 20, Issue 2, pp. 18-29, March 2003.
 [3]H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling," IEEE Trans. Circuits, Syst., Video Technol., vol. 6, pp. 191-199, Apr. 1996. 900 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 11, NO. 8, AUGUST 2002
 [4]Y. Nakajima, H. Hori, and T. Kanoh, "Rate conversion of MPEG coded video by re-quantization process," in Proc. IEEE Int. Conf. Image Processing'95, vol. 3, Washington, DC, Oct. 1995, ICIP95, pp. 408-411.
 [5]Kai-Tat Fung and Wan-Chi Siu, "DCT-based Video Frame-skipping Transcoder," Proceedings, IEEE International Symposium on Circuits and Systems (ISCAS'03), Vol. II pp. 656-659, Bangkok Thailand, May 2003.
 [6]J. Youn, M. T. Sun and C. W. Lin, "Motion vector refinement for high performance transcoding," IEEE Trans. Multimedia, vol. 1, pp. 30-40, Mar. 1999.
 [7]Mei-Juan Chen; Ming-Chung Chu; Chih-Wei Pan, "Efficient motion-estimation algorithm for reduced frame-rate video transcoder," Circuits and Systems for Video Technology, IEEE Transactions on , Volume: 12 , Issue: 4 , April 2002 Pages:269 - 275
 [8]K. Fung, Y. Chan, and W. Siu "Low-Complexity and High-Quality Frame-Skipping Transcoder for Continuous Presence Multipoint Video Conferencing," IEEE Trans. Multimedia, vol. 6, no. 1, February 2004.