

효율적인 크기조절 비디오 트랜스코딩을 위한 움직임 벡터 재 산출에 관한 연구

이규찬[○], 손채봉, 오승준, 안창범
광운대학교, VIA 멀티미디어 센터
{purekc[○], bongbong, sjoh}@media.kw.ac.kr, cbahn@daisy.kw.ac.kr

Motion Vector Reestimation Method for Efficient Resizing Video Transcoding

Kyuchan Lee[○], Chaebong Sohn, Seung-Jun Oh, and Changbeom Ahn
VIA-Multimedia Center, Kwangwoon Univ.

요 약

본 논문은 움직임 벡터 재 산출 방법에 있어서 제한된 영역 내의 움직임 벡터 재 탐색 과정을 보다 효율적으로 운용함으로써 기존의 방법보다 빠르게 동작하는 크기조절 트랜스코딩 방법을 제안한다. 기존의 크기조절 비디오 트랜스코딩은 크기 조절된 비디오의 정형화된 재 산출 과정을 통해 움직임 벡터를 구하고, 화질 열화의 최소화를 위해 재 산출된 움직임 벡터를 제한된 탐색 영역 내 재 탐색을 통해 좀더 정확한 움직임 벡터를 구하는 방법을 사용한다. 그러나 움직임 벡터 재 탐색 과정은 많은 계산이 요구되어 속도 저하의 원인이 되며, 비디오의 특성에 따라 움직임 벡터 재 탐색 과정이 불필요한 경우가 발생한다. 따라서, 움직임이 없는 비디오에 대해서는 움직임 벡터 재 탐색 과정을 수행하지 않음으로써 트랜스코딩 과정의 수행 속도를 개선하고, 움직임이 많은 프레임에 대해서는 움직임 벡터 재 탐색을 수행하여 화질 열화를 최소화하는 방법을 제안한다. 제안된 방법을 적용하였을 때, 움직임이 적은 비디오일수록 화질 열화 없이 트랜스코딩 속도가 약 15%의 향상된 성능을 얻었다.

1. 서 론

다양한 멀티미디어 환경에서의 서비스를 위해서는 기존의 방대한 량의 비디오 데이터를 효율적으로 재가공할 트랜스코딩 과정이 필요하다. 그 과정에는 비디오 포맷, 비트율, 비디오 크기 등에 대한 변환으로 구성된다. 본 논문에서는 이러한 트랜스코딩 방법 중 동일한 포맷에서 비디오의 크기를 조절하는 트랜스코딩 방법을 다룬다. 크기 조절 트랜스코딩은 기존의 사전 부호화된 비트스트림 형태의 비디오 데이터로부터 원하는 크기의 새로운 비디오 데이터를 얻는 과정을 말하며, 이 과정에서 크기 조절된 비디오의 인터 매크로블록에 대한 새로운 움직임 벡터를 구하게 된다. 움직임 벡터를 구하는 가장 직관적인 방법은 기존의 비디오 데이터를 완전히 복원하여 공간 영역에서 크기 조절 후, 움직임 벡터를 재 탐색하는 과정이다. 그러나 이 방법은 많은 계산량과 함께 많은 시간을 필요로 하기 때문에 비효율적이다. 따라서 보다 효율적인 방법으로, 기존의 비디오 데이터에 대한 움직임 벡터를 재 사용하여 구하는 방법이 제안되었다. 즉, 원래 비디오 데이터의 움직임 벡터들의 평균값이나 각 움직임 벡터에 가중치를 주어 새로운 움직임 벡터를 구하는 방법이다[1, 2]. 이 방법은 계산량 면에서 개선된 결과를 얻을 수 있으나, 정확한 움직임 벡터를 찾지 못해 화질 열화를 일으키기 때문에 제안된 영역 내에서 재 탐색함으로써

화질 열화를 최소화하는 과정을 수행한다. 하지만 이 방법은 비디오의 모든 프레임에 대해 동일한 움직임 벡터 재 산출 과정을 수행하기 때문에 비디오 특성에 따라 불필요한 과정을 수행하게 된다. 즉, 비디오에서 움직임이 거의 없는 프레임에 대하여 재 산출된 움직임 벡터에 대한 재 탐색 과정은 화질 열화 방지와 트랜스코딩의 속도에 관하여 효율적이지 못하다. 움직임이 거의 존재하지 않는 경우에 움직임 벡터 재 탐색 과정을 수행하지 않고 기존의 정형화된 방법만으로 움직임 벡터를 재 산출함으로써 화질 열화를 적게 하면서 트랜스코딩의 속도를 개선할 수 있다[1].

따라서 본 논문에서는 비효율적인 움직임 벡터 재 산출 과정을 개선하기 위해, 움직임이 매우 적은 프레임을 예측하여 그 프레임에 대한 움직임 벡터 재 산출과정에서 수행되는 재 탐색 과정을 제거하고 움직임이 많은 프레임만 재 탐색 과정을 수행함으로써 화질 열화를 최소화하여 트랜스코딩 속도를 개선하는 방법을 제안한다.

본 논문의 구성은 2장에서 기존의 정형화된 과정을 통한 움직임 벡터 재 산출 방법을 소개한다. 3장에서는 기존 방법의 비효율성을 지적하고 프레임의 움직임을 판별함에 있어서 기준을 세우는 문제에 대해 알아본다. 4장에서는 제안된 방법을 통해 실험한 결과에 대하여 언급하고 마지막으로 5장에서 결론을 맺는다.

2. 움직임 벡터 재 산출 방법

제 2장에서는 기존의 움직임 벡터 재 산출에 대해 알아 본다. 움직임 벡터를 재 산출하는 대표적인 방법은 입력된 원래 크기의 비디오에서 매크로블록들의 공간 활동도를 가중치로 하여 기존 움직임 벡터의 평균을 새로운 움직임 벡터로 결정하는 것이다[1, 2]. 이때 움직임 벡터의 크기는 크기 조절의 비율만큼 조절된다. 그 한 예로, 그림 1과 같이 원래 비디오의 1/4 크기만큼 줄인다면 새로운 움직임 벡터의 계산은 식(1)과 같다.

$$mv' = \frac{1}{2} \times \frac{\sum_{i=1}^4 mv_i A_i}{\sum_{i=1}^4 A_i} \quad (1)$$

여기서, mv_i 는 원래 크기의 비디오의 매크로블록에 대한 움직임 벡터이고, 크기 조절된 결과의 움직임 벡터는 mv' 이다. 그리고, A_i 는 가중치로써 0이 아닌 AC 계수값의 합으로 구해진다.

그러나 이들 산출된 움직임 벡터만을 사용하여 새로운 영상을 부호화할 경우 정확한 최소 에러값을 찾을 수 없어 출력 비트량 증가와 화질 저하를 일으킨다[1]. 따라서 좀더 정확한 움직임 벡터 예측을 위해 제한된 영역 내에서의 움직임 벡터 재 탐색 과정이 수행된다[3].

3. 기존 움직임 벡터 재 산출 방법의 문제점과 효율적인 재 산출 방법

위에서 알아본 기존의 움직임 벡터 재 산출 과정은 비디오의 특성을 고려하지 않고 모든 영상에 대하여 동일하게 적용된다. 그러나 비디오의 특성에 따라 움직임 벡터 재 산출 과정에서 화질 저하를 막기 위한 움직임 벡터 재 탐색 과정의 필요성에 대해 생각해 볼 필요가 있다.

움직임이 적거나 거의 존재하지 않는 비디오의 경우, 원래 크기 비디오의 움직임 벡터 값들은 대부분 0에 가까운 값이고 이들의 평균값으로 산출된 새로운 움직임 벡터 또한 마찬가지로 0에 가까운 값을 갖게 된다. 그러므로 움직임 벡터 재 탐색 없이도 매우 정확한 움직임 벡터를 구할 수 있으며, 그림 2의 (a) foreman 영상 중 255~299번째 프레임에서 볼 수 있듯이 오히려 잘못된 움직임 벡터

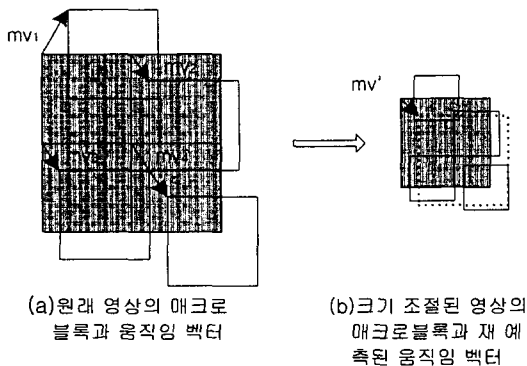


그림 1. 1/4 크기 조절을 위한 움직임 벡터 재 산출

를 재 탐색함으로써 화질 열화의 결과를 초래할 수도 있다.

그림 2의 (b) akiyo 영상과 같이 움직임이 매우 작은 비디오에 대한 움직임 벡터 재 탐색과정의 수행 여부에 따른 화질을 보면, 약간의 차이는 존재하지만 두 경우 모두 전체적인 화질이 PSNR 측면에서 35dB 이상으로 매우 높다는 것을 알 수 있다. 따라서 전체적으로 움직임이 적은 비디오에 대한 화질 열화 방지를 위한 움직임 벡터 재 탐색 수행은 불필요하다.

이와 같은 기존의 움직임 벡터 재 산출 과정의 비효율적인 문제를 해결하기 위해, 움직임 벡터 재 탐색 과정 수행함에 있어서 비디오의 움직임을 먼저 검사하고 움직임이 적은 프레임에 대해서는 재 탐색 과정을 제거하여 트랜스코딩 속도를 높이는 방법을 제안한다.

먼저 고려해야 할 사항은 비디오의 움직임이 크고 적음을 판별하기 위한 움직임의 수치와 기준 설정의 문제이다. 비디오의 움직임은 비디오 데이터 중 움직임 벡터와 직접적인 관련이 있다. 즉, 한 프레임에 대한 움직임이 많다면 대부분의 매크로블록의 움직임 벡터가 존재하게 되고, 움직임이 클수록 움직임 벡터의 값이 크다. 따라서 식(2)로 정의된 바와 같이, 입력되는 원래 크기의 비디오 한 프레임에 대한 움직임 벡터들의 절대치들을 모두 더해 크기 조절된 프레임의 움직임을 수치화한다.

$$MOTION = \sum_{i=1}^N (|mv_h(i)| + |mv_v(i)|) \quad (2)$$

여기서 N은 원래 크기의 비디오 한 프레임에 대한 총 매크로블록 개수이며, mv_h , mv_v 는 각 매크로블록에 대한 수평, 수직 방향의 움직임 벡터이다.

위 수식을 통해 수치화된 움직임의 크기 MOTION 값을 정해진 임계값에 대해 비교하여 움직임의 많고 적음을 판별한다. 판별된 움직임이 적은 경우에는 움직임 벡터 재 탐색 과정을 수행하지 않고, 움직임이 많은 경우에 화질 열화를 막기 위해 재 탐색 과정을 수행한다. 이 때, 움직임의 많고 적음의 기준이 되는 임계값은 특성에 따른 다양한 비디오에 대한 실험을 통해 얻을 수 있는 값으로 다음 장에서 임계값 설정의 실험에 대해 자세히 설명한다.

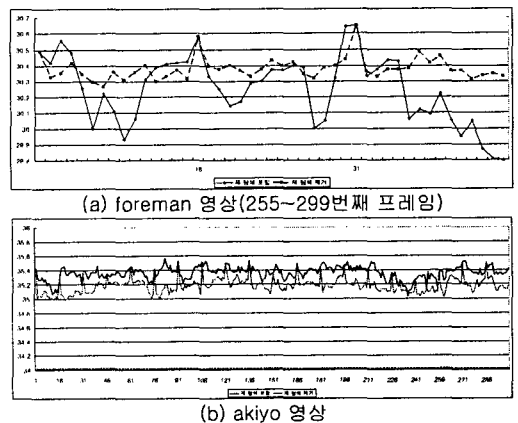


그림2. 움직임이 적은 영상의 움직임 벡터 재 탐색 수행 여부에 따른 PSNR 비교

4. 실험 과정 및 결과

제 4장에서는 비디오의 움직임이 많고 적응을 판별하기 위한 임계값을 설정하는 과정과 움직임 벡터 재 탐색 과정 수행 여부에 따른 실험 결과에 대하여 기술한다.

실험은 팬티엄3 1GHz의 시스템에서 수행되었다. 크기 조절 트랜스코딩 방법에서 인트라 매크로블록은 DCT 공간상에서 그리고, 인트라 매크로블록에 대해서는 화소 공간상에서 처리하는 기존의 복합 트랜스코딩 방법[3, 4]으로써 MPEG-2 포맷에서 MPEG-2 포맷으로의 동종 트랜스코더이며, VBR 모드에서 동작한다.

모의 실험을 위한 비디오는 움직임의 특성을 고려하여 움직임이 적은 akiyo 영상과 중간 움직임을 갖는 foreman 영상, 움직임이 많은 stefan 영상을 선택하였으며, 비디오의 크기는 352x288 CIF 크기에서 176x144 QCIF 크기로 크기조절 하였다. 움직임 벡터 재 탐색 과정은 ± 2.5의 제한된 탐색 영역 내에서 수행하였다.

3장에서 비디오의 각 프레임에 대한 움직임을 정의한 MOTION 값을 분석해 보면, 움직임이 매우 적은 akiyo 영상은 모든 프레임에서 MOTION 값이 200을 넘지 않고, 움직임이 많은 stefan 영상의 경우 부분적으로 2,000이하의 값을 갖지만 최고 7,300까지 대부분 MOTION 값이 크고 변화가 심하다. 비교적 움직임의 상태를 잘 파악할 수 있었던 foreman 영상의 경우, 그림3에서 볼 수 있듯이 화면전환이 발생하는 약 170~230번째 프레임의 경우 MOTION 값이 최고 12,000까지 매우 높았고, 완만한 움직임을 보인 화면전환 이전 프레임에서는 대략 2,000 정도의 값을 가졌다. 그리고, 화면전환 이후 거의 정지 영상과 같이 움직임이 거의 없었던 프레임의 경우 그 값이 대부분 1,000 이하로 나타났다. 따라서, 이러한 비디오의 특성에 따른 분석에 의해 움직임의 많고 적음을 판별하기 위한 임계값을 1,000, 2,000으로 설정하여 실험을 수행하였다. 실험 결과, 표를 통해 알 수 있듯이 움직임이 많은 stefan 영상의 경우, 그 효율은 작지만 임계값 2,000 일 때 PSNR이 0.07dB 감소하였고 약 0.5fps 정도의 속도

표1. PSNR

	akiyo	foreman	stefan
모두 수행	35.363	31.598	29.384
모두 제거	35.185	31.621	29.091
임계값 1000	35.185	31.590	29.377
임계값 2000	35.185	31.553	29.313

표2. speed (fps)

	akiyo	foreman	stefan
모두 수행	20.268	20.219	20.038
모두 제거	23.559	22.051	21.759
임계값 1000	23.421	20.643	20.076
임계값 2000	23.367	21.179	20.590

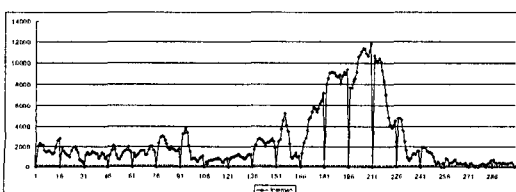


그림3. foreman 영상의 프레임당 MOTION 값

가 향상되었다. 움직임이 적은 akiyo 영상은 임계값이 1,000, 2,000 일 때, 또는 재 탐색 과정을 전혀 수행하지 않은 경우 모두 재 탐색 과정을 전부 수행했을 때보다 PSNR은 약 0.2dB정도 낮아지지만, 수행속도가 모두 23fps 정도로 약 3fps 이상 빨라지는 것을 볼 수 있다. 즉, MOTION 값이 대부분의 프레임에서 1,000을 넘지 않으므로 임계값이 1,000 이상이면 재 탐색 과정 수행이 거의 이루어지지 않아 높은 속도 향상을 얻을 수 있었으며, 이 경우 PSNR은 모두 35dB 이상으로 0.2dB 차이는 사실상 크지 않음을 알 수 있다. foreman 영상의 경우, PSNR이 0.04dB 감소할 때, 임계값이 1,000인 경우에는 화면 전환 이후 움직임이 거의 없기 때문에 약간의 속도 향상을 얻었고, 임계값 2,000인 경우에는 화면전환 이전의 프레임 대부분에서 또한 재 탐색 과정을 수행하지 않아 약 1fps 속도가 향상 될 수 있었다.

이상의 결과를 통해, 제안된 방법은 움직임이 적은 비디오일수록 기본적으로 화질이 높아서 화질 저하의 우려가 거의 없고 트랜스코딩 속도는 기존의 방법보다 빠른 향상된 성능을 얻을 수 있음을 알 수 있었다.

5. 결론

본 논문에서는 영상의 크기조절 트랜스코딩 과정에서 비디오의 특성을 고려하여, 화질 열화를 최소화하면서 속도 향상을 위한 움직임 벡터 재 산출 방법을 제안하였다. 움직임 벡터 재 산출 과정에서 수행되는 움직임 벡터 재 탐색 과정은 움직임이 적은 비디오일수록 그 필요성이 거의 없고, 따라서 움직임의 많고 적음을 판별하여 움직임이 적은 프레임에 대해 움직임 벡터 재 탐색 과정을 제거하였다. 그 결과, 움직임 적은 비디오에 대한 트랜스코딩 속도가 약 15% 향상된 성능을 얻었으며, 비디오의 움직임이 적을수록 더욱 효율적인 성능을 보였다.

감사의 글

본 연구는 한국과학재단 목적이초연구 (R01-2002-000-00179-0) 과제로 수행되었음.

참고 문헌

- [1] B.Shen, I.K.Sethi, and B. Vasudev, " Adaptive motion-vector resampling for compressed video downsizing," IEEE Trans. Circuits Syst. Video Technol., vol.9, pp.929-936, Sept. 1999.
- [2] Y.Q.Liang, L.P.Chau, and Y.P.Tan, " Arbitrary downsizing video Transcoding using fast motion vector reestimation," IEEE Signal Processing Lett., vol.9, pp.352-355, Nov. 2002
- [3] 박일우, 조용국, 오승준, " 영상의 크기를 변환하는 새로운 트랜스코더 구조," 신호처리합동학술대회, vol.15, no.1, pp.302-305, 2002
- [4] R.Dugad and N.Ahuja, " A Fast scheme for image size change in the compressed domain," IEEE Trans.CSVT, vol.11, no.4, pp.461-474, April. 2001