

트랜스 코딩에서의 효율적인 움직임 벡터 재추정 기법 연구

論 文

53D-8-8

Efficient Technique of Motion Vector Re-estimation in Transcoding

韓 科 辰* · 朴 康 恕** · 劉 晞 峻*** · 金 鋒 坤§ · 朴 相 暎§§

(Doo-Jin Han · Kang-Seo Park · Hui-Joon Yu · Bong-Gon Kim · Sang-Hui Park)

Abstract - A novel motion vector re-estimation technique for transcoding into lower spatial resolution is proposed. This technique is based on the fact that the block matching error is proportional to the complexity of the reference block with Taylor series expansion. It is shown that the motion vectors re-estimated by the proposed method are closer to optimal ones and offer better quality than those of previous techniques.

Key Words : Transcoding, Motion Vector Re-estimation, Taylor Series Expansion

1. 서 론

멀티미디어 기술의 성장에 따라 멀티미디어 통신의 환경은 더 다양하고 더 복잡하게 발전하였다. 주문형 비디오(VoD)나 모바일 멀티미디어 통신의 경우 미리 부호화된 비디오가 사용자들에게 전송되고 재생된다. 이때 미리 부호화된 비디오는 그 자체로는 다양한 환경의 전송 채널과 사용자 기기에 적용될 수 없기 때문에 각각의 전송 채널과 사용자 기기의 특징에 알맞도록 변환되어야 할 필요가 있다. 이것은 미리 부호화된 일련의 비디오 시퀀스를 주어진 환경에 맞는 다양한 형식, 비트율, 공간적 해상도, 시간적 해상도로 재구성한 새로운 비디오 시퀀스로 변환해주는 트랜스 코딩을 통하여 이루어질 수 있다[1][2].

효율적인 트랜스 코딩 알고리즘은 비디오의 화질을 저하시키지 않으면서 변환에 필요한 연산량을 최소화 해야만 한다. 이를 위해서 대부분의 트랜스 코더는 원본 비디오 스트림의 움직임 벡터를 재사용함으로써 움직임 추정에 걸리는 시간을 줄인다. 움직임 추정에 걸리는 시간은 전체 스트림의 트랜스 코딩에 걸리는 시간의 약 60%를 차지하기 때문에 움직임 벡터를 재사용하는 것은 트랜스 코더에 커다란 시간적 이득을 가져오게 된다[3]. 원본 비디오 스트림의 움직임 벡터로부터 새로운 움직임 벡터를 추정하는 방법은 공간적 시간적 해상도를 낮추는 트랜스 코더의 효율성을 결정하는 중요한 요소이다. 이러한 움직임 벡터의 추정 방법으로서 이전에 몇가지 방법이 제시되었다. 이 제안된 움직임

벡터 재추정 방법들은 대부분 실험적인 결과에 의해 얻어진 방법들이다[1][2][3].

본 논문에서는 블록 매칭 오차는 참조 블록의 Taylor 급수 전개에 의해 구해진 복잡도에 비례한다는 사실로부터 트랜스 코딩에 사용되는 움직임 벡터 평가식을 수학적으로 도출하였다. 그리고 이 제안한 방법의 성능을 이전의 방법과 비교하였다.

2. 본 론

2.1 공간 해상도를 낮추는 트랜스 코딩

미리 부호화된 비디오의 비트율에 비해서 전송 채널의 용량이 심하게 부족하거나 사용자 기기가 지원하는 공간 해상도가 제한되어 있는 경우에는 공간 해상도를 낮추는 트랜스 코딩이 필요하다. 공간 해상도를 낮추는 트랜스 코딩 시에는 입력으로 들어오는 비디오의 움직임 벡터를 바로 재사용하고 정제하는 것이 불가능하다. 왜냐하면 트랜스 코딩된 비디오의 움직임 벡터는 입력으로 들어오는 부호화 된 비디오의 움직임 벡터와 공간적 규모가 다르기 때문이다[2]. 공간 해상도를 낮추는 트랜스 코딩에선 일반적으로 공간 해상도를 1/2로 축소하는 경우가 가장 많다. 이 경우 트랜스 코딩된 비디오의 움직임 벡터는 그림 1에서처럼 입력으로 들어오는 4개의 움직임 벡터로부터 결정된다. 4개의 움직임 벡터 중 중간값을 가지는 벡터를 트랜스 코딩된 매크로블록의 움직임 벡터로 사용하는 중간 벡터 선정(MVS)[1] 방법, 4개의 움직임 벡터에서 예측 오차가 크게 발생한 매크로블록에 해당하는 움직임 벡터에 가중치를 크게 주어 움직임 벡터를 구하는 AWW(Align-to-Worst Weight)[3] 방법, 4개의 움직임 벡터들의 평균값을 취해서 움직임 벡터로 사용하는 방법[1] 등의 실험을 통해 도출된 몇몇 방법들이 이러한 경우의 트랜스 코딩에 사용되어 왔다.

* 正 會 員 : 延世大 工大 電氣電子工學科 博士課程

** 正 會 員 : 三星電子

*** 學生會員 : 三星電子

§ 學生會員 : 延世大 工大 電氣電子工學科 碩士課程

§§ 終身會員 : 延世大 工大 電氣電子工學科 教授 · 工博

接受日字 : 2004年 3月 25日

最終完了 : 2004年 7月 5日

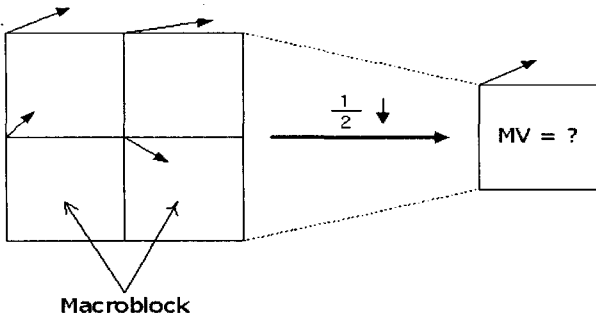


그림 1 공간 해상도를 낮추는 경우의 움직임 벡터 재구성
Fig. 1 MV recomposition in spatial resolution reduction

2.2 제안한 알고리즘

[4]에서 Taylor 급수 전개를 이용하여 참조 블록의 절대값의 증가량과 매칭 오차와의 관계를 보였다. $p = (x, y)$ 일 때의 영상의 휘도를 $f_i(p)$ 라 하고 위치 p 에서의 최적의 움직임 벡터는 $mv = (mvx, mvy)$ 라 하자. Taylor 급수 전개의 변형된 형태를 이용하여 주어진 후보 움직임 벡터 $cmv = (cmvx, cmvy)$ 의 블록 매칭 오차를 식 (1)에 서처럼 참조 블록의 기울기의 절대값에 대한 식으로서 나타낼 수 있다.

$$d_t(p, cmv) = |f_t(p) - f_t(p + cmv)| \approx \left| \frac{\partial f_t(p)}{\partial x} (cmvx - mvx) \right| + \left| \frac{\partial f_t(p)}{\partial y} (cmvy - mvy) \right| \quad (1)$$

식 (1)은 위치 p 에서의 블록 매칭 오차는 영상의 복잡도에 대응하는 블록의 기울기의 절대값에 비례한다는 것을 보여준다.

위치 p 에서의 가로 방향의 블록의 기울기의 절대값은 다음의 식 (2)와 같이 정의된다.

$$|G_x| = \left| \frac{\partial f_t(p)}{\partial x} \right| = \sum_x \sum_y |f_t(x, y) - f_t(x+1, y)| \quad (2)$$

여기서 x 와 y 는 위치 p 에서의 블록의 가로방향 세로방향 픽셀의 위치를 뜻한다.

영상의 공간 해상도를 반으로 줄이는 트랜스 코딩의 경우, 2차 부호화에서의 새로운 움직임 벡터는 공간 해상도가 두배로 큰 원본 비디오 시퀀스의 1차 부호화에서 사용된 4개의 매크로블록에 대한 움직임 벡터들로부터 재추정된다. 만일 2차 부호화에서의 낮은 공간 해상도의 비디오 시퀀스의 매크로블록이 4개의 서브 블록으로 이루어져있다고 가정하면, 매크로블록의 블록 매칭 오차는 다음과 같이 4개의 서브 블록의 블록 매칭 오차의 합으로서 나타낼 수 있다.

$$d_t(p) = d_t(p_1) + d_t(p_2) + d_t(p_3) + d_t(p_4) \quad (3)$$

여기서 $p_1 \sim p_4$ 는 매크로블록 p 의 서브블록을 의미한다. 각각의 서브블록의 최적의 움직임 벡터는 입력으로 들어오는 공간 해상도가 두배인 비디오 시퀀스의 매크로블록의 움직임 벡터를 반으로 나눈것이라고 할 수 있다. 이렇게 구해진 서브블록의 움직임 벡터를 입력 움직임 벡터라고 하자. 그렇다면 서브블록의 블록 매칭 오차는 식 (1)에서 최적의 움직임 벡터를 입력 움직임 벡터로 대체함으로써 추정할 수 있다. 같은 매크로블록 안에서는 각각의 서브블록은 같은 cmv 를 갖는다. 이것은 하나의 매크로블록에는 오로지 하나의 움직임 벡터만을 가지기 때문이다. 그러면 cmv 에 대한 위치 p 에서의 매크로블록의 블록 매칭 오차 $d_t(p, cmv)$ 는 다음과 같이 나타낼 수 있다.

$$d_t(p, cmv) = \sum_{i=1}^4 \left\{ \left| \frac{\partial f_t(p_i)}{\partial x} (cmvx - mvx_i) \right| + \left| \frac{\partial f_t(p_i)}{\partial y} (cmvy - mvy_i) \right| \right\} \quad (4)$$

여기서 (mvx_i, mvy_i) 는 각각의 서브블록에 대한 입력 움직임 벡터이다.

식 (5)에서처럼 낮은 공간 해상도로 트랜스 코딩되는 경우의 매크로블록에 대한 재추정된 움직임 벡터는 4개의 서브블록의 cmv 중에서 $d_t(p, cmv)$ 를 최소화하는 cmv 로 선택할 수 있다.

$$\text{estimated } mv \text{ of } p = \arg \min [d_t(p, cmv)] \quad (1 \leq t \leq 4) \quad (5)$$

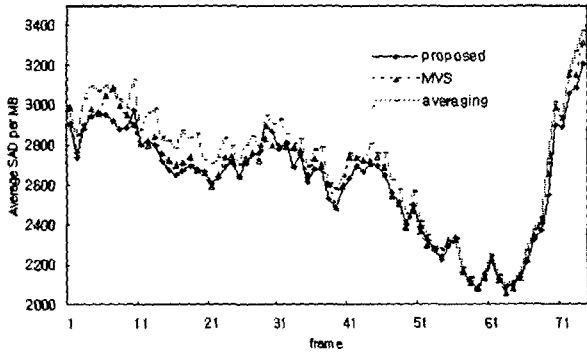
식 (4)는 입력 움직임 벡터중의 하나에서 블록 오차가 최소가 되고 $cmvx$ 와 $cmvy$ 는 독립적으로 구해진다는 것을 보여준다. 또한 식 (4)는 매크로블록의 최적의 움직임 벡터는 입력 움직임 벡터간의 거리뿐만 아니라 각 서브 블록의 복잡도에도 관계가 있음을 보여준다. 만일 각각의 서브 블록이 동일한 기울기를 가진다고 가정하면, 재추정된 움직임 벡터는 입력 벡터들의 중간값이 된다.

2.3 실험 결과

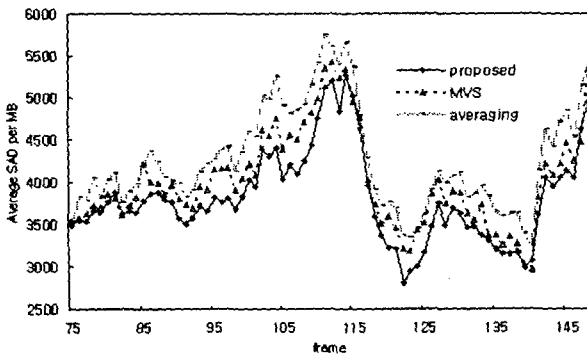
제안한 방법의 성능을 알아보기 위해 CIF 크기의 Football 영상 시퀀스 150 프레임을 H.263 비트스트림으로 부호화했다. 여기서 사용된 H.263 부호화기는 TMN 5[5][6]를 사용하였으며 양자화 계수(QP)는 5로 설정하였다. 그리고 나서 부호화된 영상을 QCIF 크기의 영상으로 트랜스 코딩 하였으며 트랜스 코딩시 움직임 벡터의 선정 방법으로서 평균값, MVS, 그리고 제안한 방법을 사용하였다. 이때의 양자화 계수(QP)는 10으로 고정하였다. 여기서는 제안한 방법과 MVS를 비교하였다. 왜냐하면 MVS는 이전에 제안된 알고리즘 중 간단하면서도 가장 좋은 성능을 보여주기 때문이다.

그림 2(a)와 (b)는 주어진 영상 시퀀스에 대해 ± 0.5 픽셀만큼 정련을 했을 때의 매크로블록당 절대 차이의 합(SAD)의 평균을 나타낸다. 낮은 공간 해상도로의 트랜스 코딩시에, ± 0.5 픽셀만큼의 정련은 필수불가결하다. 이것은 큰 공

간 해상도에서 들어오는 영상 시퀀스의 움직임 벡터를 낮은 공간 해상도에서 사용시 반으로 나누는데서 발생하는 반올림 오차를 보정해주어야 하기 때문이다.



(a) 1~74 프레임

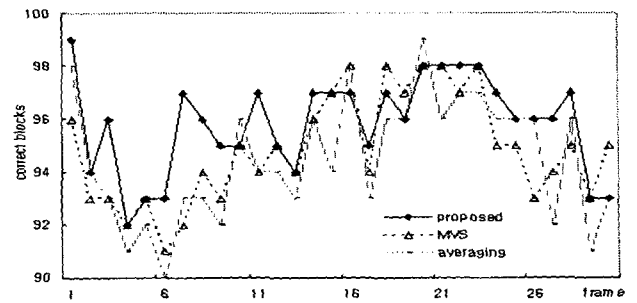


(b) 75~150 프레임

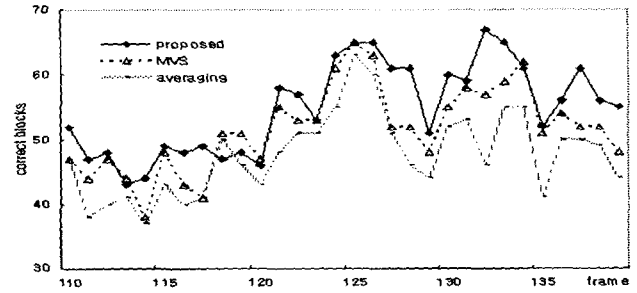
그림 2 각각의 프레임별 매크로블록당 평균 SAD
Fig. 2 Average SAD per MB for each frame

그림 2(a)는 Football 영상 시퀀스에서 상대적으로 움직임이 작은 앞부분의 75 프레임의 SAD를 나타낸다. 그리고 그림 2(b)는 움직임이 빠른 후반의 75 프레임의 SAD를 나타낸다. 제안한 방법은 앞부분의 75프레임의 경우 약간의 성능 향상이 있으며 후반의 75 프레임의 경우에는 성능 향상이 두드러진다. 제안한 방법의 평균 SAD는 MVS 보다는 199가, 평균값 보다는 357이 작았다. 기울기 계산에 사용된 계산량은 두 블록의 SAD를 계산할때의 계산량과 거의 동일하였다. 그리고 ± 0.5 픽셀만큼 정련을 하지 않은 경우의 제안한 알고리즘의 SAD는 ± 0.5 픽셀만큼 정련한 MVS의 SAD보다 단지 12.7 만큼 컸다. ± 0.5 픽셀만큼 움직임 벡터를 정련하는에는 여덟 블록의 SAD 계산이 필요하다.

각각의 알고리즘의 정확도를 평가하기 위해서 각각의 알고리즘에서 추정된 움직임 벡터를 전방위 검색 움직임 추정 방식에 의해 구해진 최적의 움직임 벡터와 비교해 보았다. 그림 3은 각 알고리즘의 최적의 움직임 벡터와 일치하는 움직임 벡터의 개수를 나타낸다. 이 결과 역시 제안한 방법이 움직임이 빠른 후반의 75 프레임의 경우 그 성능이 두드러짐을 보여준다.



(a) 1~30 프레임



(b) 111~140 프레임

그림 3 올바르게 재추정된 움직임 벡터의 개수
Fig. 3 Number of correctly re-estimated MV's

비트율이 일정한 경우에서 제안한 알고리즘과 다른 알고리즘을 비교해 보았다. 제안한 방법의 성능을 알아보기 위해 CIF 크기의 Football 영상 시퀀스 150 프레임을 부호화 하였다. 양자화 계수(QP)는 5로 설정하였다. 그리고나서 이 부호화된 영상을 QCIF 크기의 영상으로 트랜스 코딩 하였으며 비트율은 200 Kbit/s로 고정하였고 몇몇 범위에서 움직임 벡터의 정련을 수행하였다. 그림 4는 각각의 알고리즘에 다양한 정련 범위를 가지고 트랜스 코딩된 영상 시퀀스의 평균 PSNR을 나타낸다. 트랜스 코더를 거치지 않은 완전히 다시 부호화된 비디오의 경우 평균 PSNR은 25.99dB이다. 그림 4를 보면 제안한 방법에 의해 재부호화된 비디오가 가장 좋은 성능을 보여줌을 알 수 있다. 제안한 방법에 ± 0.5 픽셀만큼 정련을 한 경우의 PSNR은 다른 알고리즘에서 ± 1.5 픽셀만큼 정련을 한 경우와 같은 정도의 PSNR을 보여줌을 그림을 통해 확인할 수 있다.

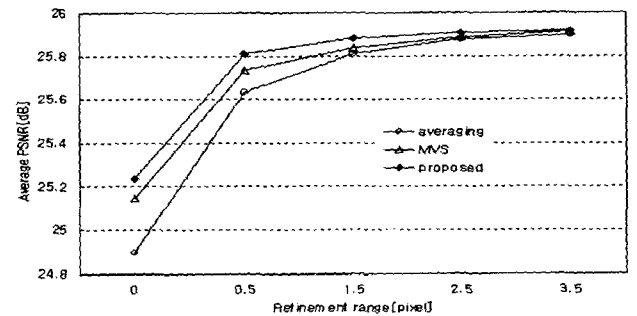


그림 4 정련 범위에 따른 평균 PSNR
Fig. 4 Average PSNR for refinement range

3. 결 론

본 논문에서는 낮은 공간 해상도로의 디지털 비디오 트랜스 코딩시의 효율적인 움직임 벡터 재추정 기법을 제안하였다. 움직임 벡터 평가식은 Taylor 급수 전개로부터 도출되었으며 이 평가식에 의해서 트랜스 코딩된 비디오의 새로운 움직임 벡터가 추정되었다. 제안된 방법에 의해 재추정된 움직임 벡터는 이전의 방법에 의해 구해진 움직임 벡터 보다 좋은 성능을 보였으며 최적의 움직임 벡터에 보다 가까웠다. 이전 방법에 의한 알고리즘에 비하여 제안한 방법의 알고리즘의 우수성은 움직임이 큰 비디오의 경우 더 두드러진다.

참 고 문 헌

- [1] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," IEEE Transactions on Multimedia, vol. 2, no. 2, pp. 101-110, 2000.
- [2] N. Bjrk and C. Christopoulos, "Transcoder architectures for video coding," IEEE Transactions on Consumer Electronics, vol. 44, no. 1, pp. 88-98, 1998.
- [3] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video downscaling," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 6, pp. 929-936, 1999.
- [4] J. N. Kim, S. C. Byun, Y. H. Kim, and B. H. Ahn, "Fast full search motion estimation algorithm using early detection of impossible candidate vectors," IEEE Transactions on Signal Processing, vol. 50, no. 9, pp. 2355-2365, 2002.
- [5] ITU-T Rec. H.263, "Video codec for low bit rate communication," 1996.
- [6] Telnor Codec, ITU-T/GS-15, "Video codec test model, TMN5," Telnor research, 1995.

저 자 소 개



한 두 진 (韓 杜 辰)

1976년 10월 25일생. 1999년 연세대 전기 전자공학과 졸업, 2001년 동 대학원 전기 전자공학과 졸업(공석). 2001년~현재 동 대학원 전기전자공학과 박사과정



박 강 서 (朴 康 恕)

1972년 5월 27일생. 1996년 연세대 전기공학과 졸업. 2003년 동 대학원 전기전자공학과 졸업(공박). 2003년~현재 삼성전자



유 희 준 (劉 晞 峻)

1975년 6월 17일생. 2002년 광운대 전파공학과 졸업, 2003년 연세대 대학원 전기전자공학과 졸업(공석). 2003년~현재 삼성전자



김 봉 곤 (金 鋒 坤)

1976년 8월 1일생. 2002년 광운대 전파공학과 졸업. 2002년~현재 동 대학원 석사과정



박 상 희 (朴 相 暻)

1939년 8월 25일생. 1962년 연세대학교 전기공학과 졸업. 1971년 동 대학원 전기공학과 졸업(공박). 1970년~현재 연세대학교 전기전자공학과 교수