

# 효율적인 크기조절 트랜스코딩을 위한 적응적 움직임 벡터 재산출 방법

이규찬 김성훈 오승준 박호종 안창범 \*서정일

광운대학교 VIA-Multimedia Center, \*ETRI 전파방송연구소 방송미디어연구부

{ purekc, kungfu, sjoh, hcpark, cbahn }@viame.re.kr, \*seoji@etri.re.kr

## Adaptive Motion Vector Resampling Method for Efficient Resizing Transcoding

Lee, Kyu-Chan Kim, Seong-Hoon Oh, Seung-Jun Park, Ho-Chong

Ahn, Chang-Beom \*Seo, Jeong-Il

Dept. VIA-Multimedia Center, Kwangwoon University

\*Radio and Broadcasting Laboratory, ETRI

### 요약

크기조절 트랜스코딩에서 움직임 벡터 재 예측 과정은 많은 연산량을 필요로 하기 때문에, 실시간 처리를 위해서는 이 과정의 연산량을 줄이는 것이 필요하다. 본 논문에서는 여러 영상에 대해 예측 움직임 벡터를 산출하는 방법을 적응적으로 수행함으로써, 기존 방법에 비해 화질열화 없이 연산량을 줄이는 방법을 제안한다. 전체 움직임의 크기와 움직임 벡터들의 균일성(homogeneity)을 이용하여 움직임이 작을 때는 움직임 벡터 재산출 과정 없이 예측 움직임 벡터 성분을 0으로, 움직임이 크면 움직임 벡터들의 균일성의 정도에 따라 평균값 또는 중간값을 예측 움직임 벡터 성분으로 적응적으로 선택하였다. 그리고 좀 더 효율적인 움직임 벡터 수행을 위해 제안된 과정을 수평, 수직 성분에 각각 따로 적용하였다. 가중치를 부여하여 평균값을 취하는 가중평균 방법과 비교 실험한 결과, 같은 PSNR을 유지하는 조건에서 움직임 벡터 재산출 과정의 덧셈과 곱셈 연산의 수가 평균적으로 각각 96%, 42% 정도 감소하였다.

### 1. 서론

현재, 동영상 부호화 기술은 디지털 방송, 모바일용 단말기 및 DMB 등 다양한 분야에서 서로 다른 네트워크와 포맷, 단말기에 대해 적용되고 있다. 이와 같이 복잡하고 다양한 멀티미디어 환경에서, 사진에 부호화된 비디오 데이터를 원하는 형태에 맞게 효율적으로 재가공하기 위한 트랜스코딩의 필요성은 더욱 크게 증가하고 있다. 특히, 단말의 형태가 다양해짐에 따라 사용되는 비디오 영상의 크기 또한 다양해지고, 모바일 네트워크 및 인터넷과 같이 대역폭이 제한된 네트워크에서 방대한 양의 비디오 영상을 전송하기 위해서는 크기조절 트랜스코딩을 통해 적절한 크기의 비디오 영상으로의 변환이 필요하다[1].

크기조절 트랜스코딩은 사진 부호화된 비트스트림 형태의 비디오 영상으로부터 원하는 크기의 새로운 비디오 영상을 얻는 과정을 말하며, 이 과정에서 크기 조절된 비디오 영상의 인터 매크로블록에 대한 새로운 움직임 벡터를 구하게 된다. 움직임 벡터를 구하는 가장 직관적인 방법은 기존의 비디오 영상을 완전히 복원하여 공간 영역에서 크기 조절 후, 움직임 벡터를 재 탐색하는 과정이다. 그러나 이 방법은 많은 계산량과 함께 많은 수행 시간을 필요로 하기 때문에 비효율적이다. 따라서 보다 효율적인 방법으로 기존의 비디오 영상에 대한 움직임 벡터를 재사용하여 구하는 방법이 일반적으로 사용되고 있다.

일반적으로 기존의 움직임 벡터를 재사용하여 크기조절된 비디오 영상의 움직임 벡터를 구하는 과정은 움직임 벡터 재산출(resampling)과 움직임 벡터 재보정(refinement)의 두 과정으로 나뉜다[2]. 움직임 벡터 재산출은 주어진 원래 크기의 비디오 영상에 대한 움직임 벡터들로부터 크기 조절된 움직임 벡터를 산출하는 과정으로써 일반적으로 원래 크기 비디오 영상의 움직임 벡터들의 평균값

(average)이나 각 움직임 벡터에 가중치를 두어 평균을 취한 가중 평균(weighted average) 방법, 중간값(median) 방법이 주로 사용된다. 움직임 벡터 재보정은 움직임 벡터 재산출 과정을 통해 얻어진 예측 움직임 벡터에 대해 제한된 영역 내에서 재 탐색을 수행하는 과정이다. 움직임 벡터 재산출 과정으로 얻어진 예측 움직임 벡터를 크기 조절된 영상의 움직임 벡터로 사용하기에는 화질 열화로 인한 영향이 존재하므로[3], 좋은 화질의 크기조절 트랜스코딩을 목적으로 하는 응용에 있어서는 움직임 벡터 재보정 과정을 통해 화질 열화를 최소화하게 된다.

본 논문에서는 움직임 벡터를 재산출하는 과정에 있어서 기존 방법에 비해 화질열화 없이 연산량을 줄임으로써 수행 속도를 향상시킨 방법을 제안한다. 기존의 대표적인 움직임 벡터 재산출 방법으로는 원래 크기 비디오 영상의 움직임 벡터들에 가중치를 두어 평균을 취한 가중 평균(weighted average) 방법이 사용되고 있다[4]. 그러나 이 방법은 화질열화를 줄이는 장점이 있으나, 원래 크기 비디오 영상의 움직임 벡터에 대한 가중치를 얻기 위해 네 매크로블록의 AC DCT 계수값들의 합을 구해야 하므로 계산량이 증가하고 수행시간의 지연이 발생한다.

따라서 화질은 기존 방법의 화질을 유지하면서, 각 움직임 벡터의 가중치에 대한 고려 없이 원래 크기 비디오 영상의 움직임 벡터들의 평균값을 취하는 방법과 중간값을 취하는 방법, 예측 움직임 벡터를 0으로 정하는 방법 등을 적응적으로 적용함으로써 수행 속도를 개선한 움직임 벡터 재산출 방법을 제안한다.

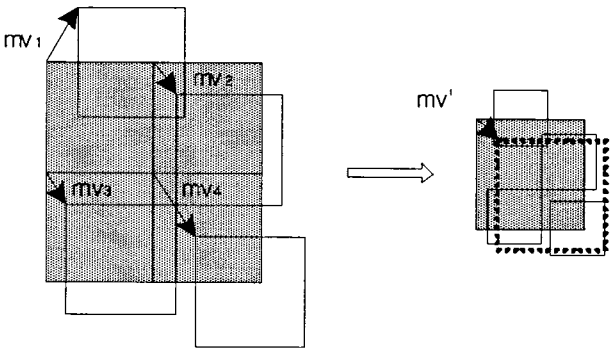
논문의 구성은 2장에서 크기조절 트랜스코딩의 구조와 기존의 움직임 벡터 재산출 방법을 설명하고, 3장에서는 본 논문에서 제안하는 움직임 벡터 재산출 방법의 구조와 움직임의 특성을 파악하기 위한 두 요소에 대해서 알아본다. 4장에서는 제안한 방법과 가중 평균 방법과

비교 실험한 연산량 및 화질 결과를 언급하고, 마지막으로 5장에서 결론을 맺는다.

## 2. 기존의 움직임 벡터 재산출 방법

움직임 벡터를 재산출하는 대표적인 방법으로 가중 평균 방법이 있다. 이 방법은 입력된 원래 크기의 비디오 영상에서 매크로블록들의 공간 활동도 (activity)를 가중치로 한 움직임 벡터들의 평균값을 크기 조절 된 비디오 영상의 움직임 벡터로 결정하는 방법이다[4]. 이때 움직임 벡터의 크기는 크기 조절의 비율만큼 조절된다. 그 한 예로, 그림 1과 같이 원래 비디오 영상의 1/4 크기만큼 줄인다면 크기조절 된 움직임 벡터를 구하는 수식은 다음과 같다.

$$mv' = \frac{1}{2} \times \frac{\sum_{k=1}^4 mv_k A_k}{\sum_{k=1}^4 A_k} \quad (1)$$



(a) 원래 영상의 매크로블록과 움직임 벡터  
(b) 크기조절된 영상의 매크로블록과 움직임 벡터  
그림 1. 1/4 크기 조절을 위한 기본적인 움직임 벡터 재산출 과정

즉, 원래 크기 비디오 영상의 움직임 벡터들  $mv_k$ 에 AC DCT 계수값들  $A_k$ 를 곱하여 각 움직임 벡터의 가중치를 두게 된다. 이 방법은 단지 평균값 또는 중간값을 취한 경우보다 좋은 화질을 얻을 수 있는 장점이 있으나, 계산 과정에서 가중치를 구하기 위해 AC DCT 계수값에 대한 합과 곱을 구해야 하기 때문에 추가적인 연산량이 증가하게 된다. 이는 곧 실시간 처리를 위한 트랜스코딩에 있어서 수행 시간의 지연을 초래하게 된다.

따라서, 연산량 증가의 문제를 해결하면서 기존 방법의 화질을 유지할 수 있는 효율적인 움직임 벡터 재산출 방법이 필요하다.

## 3. 효율적인 움직임 벡터 재산출(resampling) 방법

본 논문에서는 움직임 벡터 재산출 과정의 연산량을 줄이면서 화질을 유지하기 위한 방법으로, 움직임 벡터 재산출 과정을 움직임의 정도에 따라 적응적으로 적용하였다. 원래 크기 비디오 영상의 움직임 벡터들의 크기와 차이값으로 움직임의 정도를 판단하여, 예측 움직임 벡터 성분 값을 평균값과 중간값, 0으로 하는 방법을 움직임의 정도에 따라 적응적으로 수행하게 된다. 그리고 이 모든 과정은 좀 더 효율적인 수행을 위해 수평(horizontal), 수직(vertical) 성분에 대해 각각 따로 적용하였다.

먼저, 제안된 방법을 구현함에 있어 크기조절의 비율은 1/4 즉,

원래 크기 영상의 네 매크로블록을 하나의 매크로블록으로 크기조절 하였다. 따라서, 앞으로 본 논문에서는 네 개의 매크로블록에서 하나의 매크로블록으로 크기가 감소되는 크기조절 트랜스코딩 과정을 바탕으로 전개 할 것이다.

움직임 벡터 재산출 과정을 적응적으로 수행하기 위한 움직임 벡터의 크기와 차이값을 다음과 같이 MOTION과 VARIATION이라는 변수로 정의했다.

$$MOTION_i = \sum_{k=1}^4 |(mv_i)_k| \quad (2)$$

$$VARIATION_i = avg \sum_{k=1}^4 |(mv_i)_k - (mv_i)_{avg}| \quad (3)$$

$k$  : 원래 크기 비디오 영상의 매크로블록 인덱스

$i$  : 움직임 벡터의 수평(x) 또는 수직(y) 성분

$mv_i$  : 원래 크기 비디오 영상의 움직임 벡터 수평 또는 수직 성분 값

MOTION은 네 움직임 벡터의 수평, 수직 각각의 성분에 대한 절대값의 합으로 움직임의 크고 작음을 판단하는 기준이라 할 수 있다. 그리고 VARIATION은 네 움직임 벡터 수평, 수직 각 성분에 대한 평균적인 차이 값을 의미하며, VARIATION이 작다는 것은 네 움직임 벡터 각 성분 값이 비슷하다는 것을 의미한다.

전체적인 과정은 움직임의 정도를 나타내는 두 변수 MOTION과 VARIATION에 따라 적응적으로 예측 움직임 벡터 성분 값을 평균값과 중간값, 0으로 하는 방법을 적용하게 된다.

MOTION값이 일정 임계값보다 작으면, 움직임이 거의 없다고 판단하여 임의의 움직임 벡터 재산출 과정을 수행하지 않고 예측 움직임 벡터 각 성분 값을 0으로 정한다. 이 방법을 움직임 벡터 재산출 과정 스킵 방법이라고 한다. 그리고 MOTION값이 일정 임계값보다 크면 VARIATION값에 따라 움직임 벡터 재산출 방법을 결정하게 된다. VARIATION값이 작다는 것은 원래 크기 비디오 영상의 네 움직임 벡터 각 성분 값이 비슷하다는 것을 의미하므로 네 움직임 벡터 각 성분에 대해 평균값을 취하고, VARIATION값이 크면 네 움직임 벡터 각 성분 값의 차이가 크므로 중간값을 적용한다.

평균값과 중간값을 예측 움직임 벡터로 계산하는 방법은 다음 수식과 같다[5].

### ● 평균값

$$pmv_i = \frac{1}{4} \times \sum_{k=1}^4 (mv_i)_k \quad (4)$$

### ● 중간값

$$(d_i)_k = \sum_{\substack{k=1 \\ s \neq k}}^4 |(mv_i)_k - (mv_i)_s| \quad (5)$$

$pmv_i$ 는  $(d_i)_k$  ( $k=1 \sim 4$ ) 중 최소값을 갖는  $k$ 번째 움직임 벡터의 수평 또는 수직 성분 값

$k$  : 원래 크기 비디오 영상의 매크로블록 인덱스

$i$  : 움직임 벡터의 수평(x) 또는 수직(y) 성분

$mv_i$  : 원래 크기 영상의 움직임 벡터 수평 또는 수직 성분 값

$pmv_i$  : 크기조절 된 영상의 예측 움직임 벡터 수평 또는 수직 성분 값

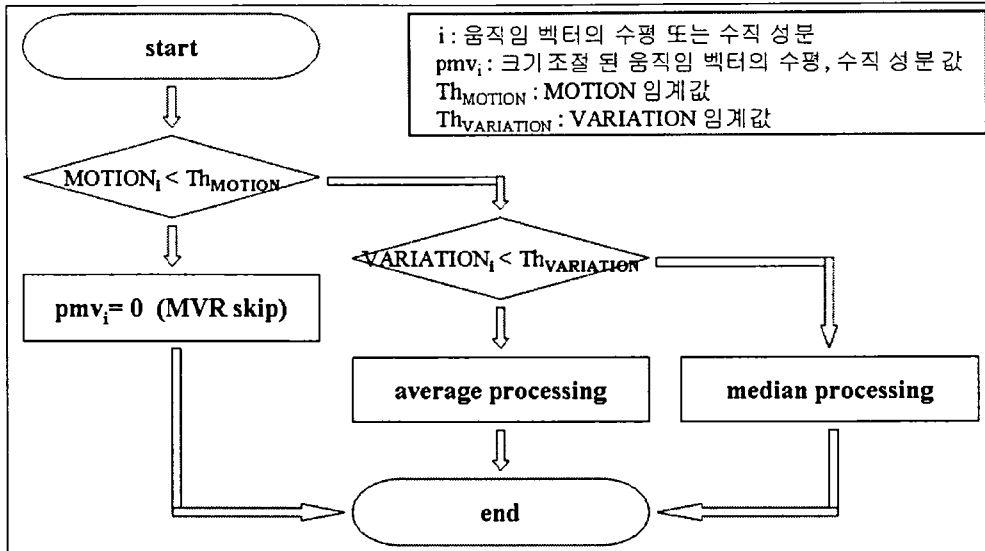


그림 2. MOTION, VARIATION에 따른 적응적 움직임 벡터 재산출 과정의 순서도

그리고 제안한 움직임 벡터 재산출 과정의 전체 순서도는 그림 2와 같다.

그림 2에서와 같이 본 제안된 방법을 적용함에 있어, 좀 더 효율적인 움직임 벡터의 수행을 위해 수평, 수평 성분을 나누어서 적용하였다. 대부분 영상의 움직임이 수직 또는 수평 방향 한쪽으로 강하게 존재한다. 따라서 수직, 수평 성분을 같이 적용하여 모든 연산을 수행하기 보다는 두 성분을 나누어 따로 적용하면 움직임이 많지 않은 성분에 대한 움직임 벡터 재산출 과정의 스킵 되는 횟수가 많아지므로 불필요한 연산을 줄일 수 있다.

#### 4. 임계값 설정 및 실험 결과

제 4장에서 제안한 움직임 벡터 재산출 과정에서 적응적 움직임 벡터 재산출을 수행하기 위한 요소 MOTION과 VARIATION의 임계값을 설정하기 위한 실험 과정과 결과를 알아본다. 그리고, 기존의 가중 평균값을 취한 움직임 벡터 재산출 방법과 제안한 방법의 실험 결과를 비교 분석한다.

본 실험을 수행한 크기조절 트랜스코딩 과정은 인트라 매크로블록은 DCT 공간상에서, 그리고 인터 매크로블록에 대해서는 화소 공간상에서 처리하는 기존의 복합 트랜스코딩 방법으로, MPEG-2 포맷의 동종 트랜스코더에서 수행되었다. 비디오 영상의 크기는 4CIF (704x576) 크기 영상에서 CIF(352x288) 크기로 1/4만큼 크기조절 하였으며, 1M CBR 모드에서 300 프레임의 crew, soccer, harbour 영상에 대해 실험하였다. 그리고, 움직임 벡터 재보정 과정은 반화소 움직임 탐색을 수행하였다.

제안한 움직임 벡터 재산출 과정에 대한 화질은 PSNR 값으로 측정하였고, 연산량은 덧셈(add)과 곱셈(multiplication) 연산의 총 수행 횟수로 산출하였다. 움직임 벡터 재산출 과정의 각 모듈에 대한 구현상의 덧셈, 곱셈, 비교연산 횟수를 표 1에서 확인할 수 있다.

먼저, 움직임의 크기와 균일성을 의미하는 MOTION, VARIATION의 임계값은, 그림 3에서 보여준 의사 코드 과정을 수행하여 최적의 MOTION, VARIATION의 임계값을 얻었다.

MOTION과 VARIATION의 임계값이 클수록, 연산량은 감소하지만 화질열화가 발생한다. 따라서 임계값 설정에 있어서 기준은, 기존 가중 평균 방법에 의한 PSNR 결과보다 작지 않는 조건에서 최소

PSNR을 유지하는 임계값을 최적 임계값으로 선택하였다. 의사코드에서 볼 수 있듯이, 우선 VARIATION의 초기 임계값을 0으로 최적의 MOTION 임계값을 구한다. 그리고 주어진 최적의 MOTION 임계값에 대한 최적의 VARIATION 임계값을 구하고, 다시 주어진 최적의 VARIATION 임계값에 대한 MOTION 임계값을 구한다. 이 과정을 반복함으로써, 각 임계값이 포화(saturation) 상태에 이를 때 즉, 같은 MOTION 임계값과 같은 VARIATION 임계값이 반복해서 얻어질 때, 그 값을 MOTION과 VARIATION의 최종 임계값으로 구하였다. 그 결과, 움직임이 많은 영상들에 대해 MOTION의 임계값은 16, VARIATION의 임계값은 5를 얻을 수 있었다.

```

VARIATION_thre = 0
While(1)
{
    for(i=0 ; i<24 ; i++)
        MOTION_thre = best[motion_i(Q)],
                       given VARIATION_thre
    for(j=0 ; j<40 ; j+=4)
        VARIATION_thre = best[variaion_j(Q)],
                          given MOTION_thre
    If same MOTION_thre repeat
    & same VARIATION_thre repeat
    END
}
i : 움직임 벡터의 수평 또는 수직 성분
Q : PSNR

```

그림 3. MOTION, VARIATION 임계값 설정 과정

이렇게 얻어진 MOTION과 VARIATION의 임계값을 적용하여, 제안한 움직임 벡터 재산출 과정을 수행한 결과, 기존의 가중 평균을 이용한 움직임 벡터 재산출 방법에 비해 PSNR이 약간 높고, 덧셈과 곱셈 연산량은 덧셈 횟수가 약 96%, 곱셈 횟수가 약 42% 씩 감소된 결과를 얻을 수 있었다.

연산	가중평균	평균값	중간값	MOTI. & VARI. 계산
덧셈	1027	3	20	26
곱셈	10	2	1	4
비교	5	0	5	0

표 1. 움직임 벡터 재산출 과정 각 모듈에 대한 덧셈, 곱셈, 비교연산 연산량

crew	PSNR	곱셈 + 덧셈
제한한방법	32.589 dB	3,366,089 a + 528,792 m
가중 평균	32.584 dB	91,282,841 a + 888,830 m

soccer	PSNR	곱셈 + 덧셈
제한한방법	31.029 dB	3,245,335 a + 533,414 m
가중 평균	31.024 dB	98,372,786 a + 909,180 m

harbour	PSNR	곱셈 + 덧셈
제한한방법	25.892 dB	3,056,088 a + 474,375 m
가중 평균	25.887 dB	112,769,735 a + 1,098,050 m

표 2. 제한한 방법과 가중평균 방법의 PSNR과 연산량 비교

## 5. 결론

본 논문에서는 비디오 영상의 크기조절 트랜스코딩 과정에서 움직임의 특성을 반영하여 움직임 벡터 재산출 과정을 적응적으로 수행함으로써, 화질열화 없이 연산량을 줄여 수행 속도를 향상시킨 효율적인 움직임 벡터 재산출 방법을 제안하였다.

움직임의 크기와 움직임 벡터들의 균일성을 이용하여 움직임이 작은 부분에 대해서는 움직임 벡터 재산출 과정의 연산 없이 예측 움직임 벡터를 0으로 정함으로써 연산량을 줄이고, 움직임이 큰 영상에 대해서는 원래 크기 영상의 움직임 벡터들의 균일성에 따라 평균값 또는 중간값을 예측 움직임 벡터로 취하였다.

그 결과, PSNR은 그대로 유지하면서 연산량에 있어서는 덧셈 약 96%, 곱셈 약 42%가 줄어 수행 속도의 향상을 얻었다.

## 감사의 글

본 연구는 2005년도 한국소프트웨어진흥원 IT-SoC 핵심설계인력양성 사업, 한국전자통신연구원 위탁과제(0201-2005-0080)로 수행되었음.

## 참고문헌

- [1] G.Keessman, et al. "Transcoding of MPEG bit streams," Signal Processing: Image Communication, vol.8, pp.481-500, Sept. 1996.
- [2] Nakajima, J., Tsuji, H., Yashima, Y., and Kobayashi, N., "Motion vector re-estimation for fast video transcoding from MPEG-2 to MPEG-4," MPEG-4, 2001 Proceedings of Workshop and Exhibition, pp.87-90, June. 2001.
- [3] 박일우, 조용국, 오승준, "영상의 크기를 변환하는 새로운 트랜스코더 구조," 신호처리합동학술대회, vol.15, no.1, pp.302-305, 2002.
- [4] B.Shen, I.K.Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video downsizing," IEEE Trans.