

낮은 해상도로의 변환 부호화에 사용되는 블록 기반의 복잡도를 이용한 벡터 추정 및 정제 기법

유 희 준, 한 두 진, 박 강 서, 박 상 회
연세대학교 전기 전자 공학과

Adaptive motion vector estimation technique for transcoder based on block complexity

Hee-Jun You, Doo-Jin Han, Kang-Seo Park and Sang-Hui Park
Dept. of Electrical and Electronics Engineering, Yonsei University

Abstract - 본 논문에서는 공간 해상도 감축이 이루어지는 디지털 동영상 변환 부호화에서의 적응 움직임 벡터 재추정 기법을 제안한다. 영상의 복잡도를 기반으로 한 에러 추정을 통해 움직임 벡터를 1차 추정하고, 이를 기반으로 정제범위를 효율적으로 조절하여 연산량을 감소시킨다. TMN5를 이용한 실험 결과 제안된 방법에 의한 추정 벡터는 기존 방법들에 의해 추정된 벡터보다 더 좋은 화질을 보였으며 적응적 정제 범위 선택에 의해 연산량도 더 적었다.

의 움직임이 항상 한 방향으로 정렬되어 있는 것은 아니다. 그림 1(b)의 그림처럼 네 개의 움직임 벡터가 일정한 방향을 나타내지 않고 서로 다른 방향을 나타내는 경우가 많다. 이러한 경우 네 개의 벡터로부터 축소된 영상에서의 한 개의 움직임 벡터를 추정하는 것은 쉽지 않다. 이를 해결하기 위해 경험적 방법에 근거한 중간값 선택법, 평균법, AWW(align to worst) 등이 제안되었다 [1][3][4].

1. 서 론

동영상의 디지털화가 확산되면서 미리 부호화되어 있는 영상을 유·무선 통신망을 통해 전송하거나 디지털 저장 매체에 저장하게 되는 경우가 많아졌다[1][2]. 이러한 경우 통신 채널의 상태나 저장 매체의 용량에 따라 기존에 부호화된 동영상의 비트율을 변화시키는 변환 부호화를 사용하게 된다. 특히, 미리 부호화 되어 있는 비트율과 목적 비트율의 차이가 매우 큰 경우나 수신단의 해상도에 한계가 있는 경우에는 비트율을 변환하기 전 영상의 크기를 먼저 다운 샘플링 하게 된다[3].

기존의 축소 변환 부호화 방법에서는 미리 부호화 되어 있는 정보를 완전히 복호화 한 후 다시 부호화 하는 과정을 거치게 된다. 이러한 경우 기존의 정보를 전혀 사용하지 않고 새로운 부호화 과정을 다시 거치게 되므로 변환 부호화에 사용되는 시간이 매우 길어진다는 단점이 있다. 부호화에서 사용되는 시간의 상당 부분은 움직임 벡터 추정과정에서 소모된다[4].

이러한 시간적 손실을 보상하기 위해 나온 방법이 모션 벡터 재사용 기법으로, 이 방법은 기존의 부호화 되어 있는 영상의 모션 벡터를 그대로 이용하여 재부호화 시에 움직임 추정과정에서 걸리는 시간을 감소시키는 방법이다.

그러나, 영상의 크기를 변화시킨 후 비트율을 변화시키는 경우에는 입력 영상의 모션 벡터와 출력 영상의 모션 벡터의 수가 달라지게 되므로 모션 벡터 재사용 기법을 직접 적용하는 것이 불가능하다. 따라서, 부호화 크기 변환을 위한 움직임 벡터 추정 방법이 필요하게 된다.

2. 낮은 공간 해상도로의 변환부호화

어떤 영상의 크기를 1/2로 줄여주게 되면, 기존 영상의 4개의 매크로 블록이 하나의 매크로 블록을 나타내게 된다. 이와 마찬가지로 각각의 매크로 블록에 할당되어 있는 4개의 움직임 벡터도 하나의 벡터로 나타내어진다. 기존의 4개의 16×16 매크로 블록의 움직임 벡터들과 다운 스케일링 후에 그 위치에 할당된 움직임 벡터의 관계를 나타낸 것이 그림 1이다.

그림 1(a)의 그림처럼 네 개의 매크로 블록의 움직임이 잘 정렬되어 있는 경우에는 네 개의 움직임 벡터 중 임의의 벡터를 선택하여도 축소된 영상의 움직임을 잘 나타낼 수 있다. 그러나, 일반적인 영상의 경우 영상에서

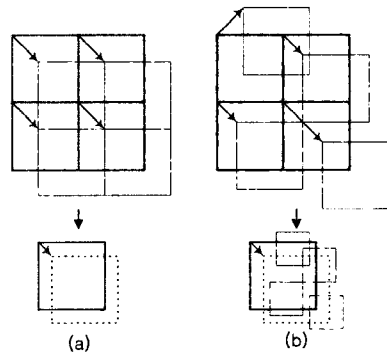


그림 1 1/2 축소 영상에서 움직임 벡터 추정

3. 제안하는 움직임 벡터 추정 기법

3.1 움직임 벡터 1차 추정

움직임 벡터 추정 에러는 영상의 복잡도 및 최적 벡터로부터의 거리와 관련이 있다는 사실이 알려져있다[5]. 이에 근거하여 다음과 같이 움직임 벡터 에러를 예측하는 추정식을 나타낼 수 있다.

$$d(p_i, cmux) = \sum_{j=1}^4 G_j \cdot \{|cmux - mux_j|\},$$

$$d(p_i, cmvy) = \sum_{j=1}^4 G_j \cdot \{|cmvy - mvy_j|\},$$

$$d(p_i, cmv) = d(p_i, cmux) + d(p_i, cmvy) \quad (1)$$

위 식에서 p_i 와 G_j 는 각각 프레임의 수평, 수직 방향 위치에 있는 매크로 블록 p 의 i, j 번째 하위 블록 및 복잡도이다. 그리고, $(cmux, cmvy)$ 및 (mux_j, mvy_j) 는 각각 i, j 번째 하위 블록의 다운 샘플링 전 매크로 블록의 움직임 벡터들을 1/2한 값들이다.

식 (1)에서 각 하위 블록의 복잡도(G_j)는 다음과 같은 식을 통해 구한다.

$$G_j = \sum_{x=1}^8 \sum_{y=1}^8 |f(x, y) - mean| \quad (2)$$

위 식에서 $f(x, y)$ 는 하위 블록 p_j 의 (x, y) 위치에 있는 화소값이고 $mean$ 은 p_j 가 포함되어 있는 매크로 블록 p 의 평균값이다.

제안한 방법에서의 복잡도를 구하는 식에서는 각 하위 블록의 평균이 아닌 매크로 블록의 평균값으로부터 복잡도를 구하게 되는데 이러한 각 하위 블록의 복잡도를 더 하게 되면 TMN5에서 intra/inter 모드 결정과정에서 그대로 다시 사용하게 되므로 부가적 연산량을 줄일 수 있게 된다. 실험의 결과 하위 블록의 평균값 대신 매크로 블록의 평균값을 사용하는 것은 성능에 크게 영향을 미치지 않음을 확인 할 수 있었다.

식 (1)의 값을 최소로 만드는 움직임 벡터가 가장 적은 에러를 가지므로 1차 추정을 통한 최적의 벡터로 선택된다. 위의 식에서 보면 p_i 의 움직임 벡터중 식을 최소로 만드는 각각의 x, y 벡터가 최적의 벡터로 선택되게 된다.

$$p \text{의 } x \text{ 벡터} = \arg \min [d(p_i, cmx)],$$

$$p \text{의 } y \text{ 벡터} = \arg \min [d(p_i, cmv)] \quad (3)$$

제안된 1차 추정 방법의 성능 평가를 위해 Football, Stefan, Suzie, Akiyo CIF (352×288) 영상들을 QCIF (176×144) 영상들로 변환 후 TMN5[6]를 이용해 기존의 방법들과 비교해 보았다. 실험에 사용된 기존의 방법은 중간값 선택법과 평균법이다. 중간값 선택법은 단순한 과정을 거치는 기존에 제안된 방법 중 가장 최고의 화질을 보이는 것으로 알려져 있다. 각 실험 영상은 고정된 양자화 계수 QP=5로 1차 부호화 된 후, 변환 부호화를 통해 QCIF 영상으로 해상도가 감축되어 QP=10으로 변환 부호화 된다. 각 영상에 대해 실험한 매크로 블록당 SAD(Sum of Absolute Difference)값의 평균은 표 1과 같다.

표 1 실험 영상들에 대한 매크로 블록당 평균 SAD

영상 추정방법	Football	Stefan	Suzie	Akiyo
평균법	3473.492	3618.213	1078.926	910.0855
중간값 선택법	3329.887	3505.493	1050.556	910.4121
제안 방법	3193.513	3472.232	1045.228	908.3825

표 1의 결과에서 알 수 있듯이 움직임이 큰 영상들(Football, Stefan)과 작은 영상들(Suzie, Akiyo) 모두에서 제안한 방법의 SAD 값이 가장 작았다.

그림 2는 Football 영상에서 각 프레임에 대한 제안한 방법과 중간값 선택법 및 평균법의 매크로 블록당 SAD 값의 차이를 나타내는 그래프이다.

그래프에서 보듯 제안한 1차 추정 방법은 기존의 방법보다 좀 더 정확한 벡터 추정을 보였으며, 향상된 성능은 움직임이 큰 시퀀스 후반부에서 더 확연하게 나타난다.

3.2 적용 움직임 벡터 정제

식 (1)에서 사용된 움직임 벡터 평가식의 결과값은 추정된 벡터의 신뢰도를 추측하는데 사용될 수 있다. 평가식의 결과는 각 하위 블록의 움직임 벡터가 서로 잘 정렬되지 않을수록, 또한 영상의 복잡도가 클수록 커지게 되며, 이러한 경우 1차 추정된 벡터의 신뢰도는 낮아지

게 된다.

목적 비트율을 200k로 고정했을 경우 각 실험 영상에 대한 전역탐색을 통해 구한 최적의 벡터와 1차 추정된 벡터와의 거리와 평가식의 결과의 관계가 표 2에 나타나 있다.

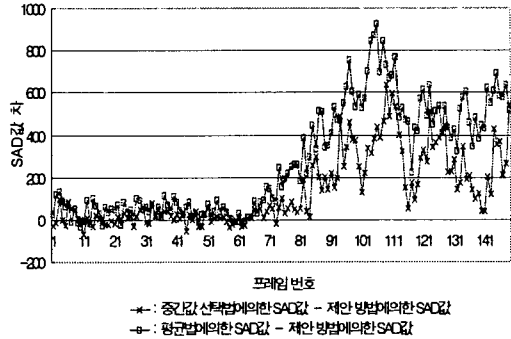
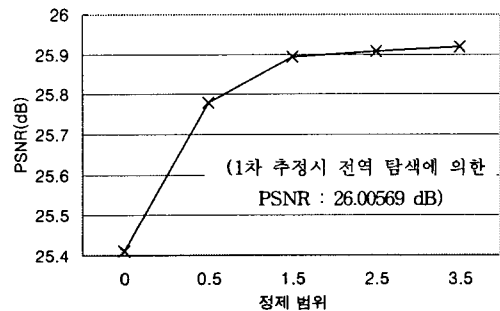


그림 2 각 프레임에 대한 제안한 방법과 기존 방법들의 매크로 블록당 SAD값의 차 그래프 (Football, QP=10)

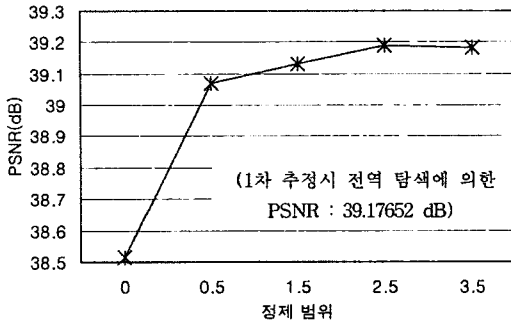
표 2 최적 벡터(전역 탐색)와 1차 추정된 벡터(제안 방법)의 거리와 평가식과의 관계 (목적 비트율 : 200k, QP=10)

영상 거리	Football	Stefan	Suzie	Akiyo
0.5 이하	12899.17	10508.52	4926.71	2338.07
1.5	33804.75	13168.66	12300.39	12153.20
2.5	46222.96	17007.79	21524.71	19004.25
3.5	52834.14	22553.89	24355.53	9431.66
3.5이상	54689.66	28766.64	21577.42	18406.40

따라서, 이를 이용해 1차 추정 벡터의 신뢰도를 구하여 벡터의 정제 범위를 적용적으로 적용하면 연산량의 감축이 가능하게 된다.



(a) Football 영상



(b) Suzie 영상

그림 3 제안한 방법에서의 정제 범위에 대한 PSNR 그래프

그림 3은 Football과 Suzie 영상의 제안한 방법에서의 정제 범위에 대한 PSNR 그래프이다. 이 그래프에서 보면 1.5 내지는 2.5정도의 정제를 거치면 전역 재탐색을 통한 최적의 영상의 화질에 근접함을 알 수 있다. 따라서, 본 논문에서는 적응적 정제 범위를 ~0.5와 ~1.5의 두 가지로 선택하였다.

if 벡터의 신뢰도 $\leq threshold$, 0.5 refinement,
 if 벡터의 신뢰도 $> threshold$, 1.5 refinement (4)

식 (4)에서 임계값 ($threshold$)은 각 실험 영상들의 1차 추정 벡터의 신뢰도를 바탕으로 한 실험치로 본 실험 결과 500에서 가장 우수한 성능을 보였다.

적용 정제 범위 선택에 의한 각 영상의 PSNR과 연산량의 감소량은 표 3에 나타나 있다.

표 3 적용 정제 범위 선택에 의한 각 영상의 PSNR 및 연산량 감소치 (목적 비트율 : 200k, QP=10)

영상	추정 방법	PSNR (dB)	정제 범위	연산량 감소치 (%)
Football	중간값 선택법	25.83061	~1.5	22.64
	제안 방법	25.87219	~0.5	
			~1.5	
Stefan	중간값 선택법	23.28499	~1.5	30.99
	제안 방법	23.32136	~0.5	
			~1.5	
Suzie	중간값 선택법	39.14560	~1.5	38.91
	제안 방법	39.17068	~0.5	
			~1.5	
Akiyo	중간값 선택법	45.36351	~1.5	92.89
	제안 방법	45.40206	~0.5	
			~1.5	

표 3에서의 연산량 감소치는 적용 정제 범위 선택에 의해 0.5 범위로 정제된 매크로블록들의 연산량을 중간값 선택법에 의해 1.5 범위로 정제된 전체 매크로 블록

의 연산량으로 나눈 백분율값이다.

위 표에서 보면 적응 정제 범위 선택에 의한 제안 방법이 고정된 정제 범위에서의 중간값 선택법에 의한 방법보다 모든 실험 영상에서 화질이 더 좋은 것을 확인할 수 있다. 또한, 적응 정제 범위 선택에 의한 제안 방법이 움직임이 작은 영상에서 정제 범위가 0.5인 매크로 블록들에게 더 많이 선택되므로 그 만큼의 연산량도 많이 감소하는 걸 알 수 있다.

4. 결 론

본 논문에서는 영상의 공간 해상도 감축에 의한 움직임 벡터 재추정 및 적응 정제 범위 선택 방법을 제안하였다.

TMN5에 의한 실험 결과 제안한 움직임 벡터 재추정 방법은 기존 방법인 중간값 선택법 및 평균법보다 좀 더 정확한 벡터 추정을 보였다. 또한, 움직임이 큰 영상과 작은 영상 모두에서 더 우수한 성능을 보였으며 특히, 움직임이 큰 영상에서 그 성능은 더 확연히 나타났다.

본 논문에서는 움직임 벡터의 추정 과정중 연산량을 줄이기 위한 방법으로 움직임 벡터 평가식에서 TMN5의 intra/inter 모드 결정 과정에서 이용되는 복잡도를 이용하였으며 그 결과값에 의해 추정된 신뢰도를 바탕으로 한 적응 정제 범위 선택에 의한 추정 방법에서 ~0.5와 ~1.5의 두 정제 범위를 적응적으로 선택하여 사용하였다. 그 결과 1.5 정제 범위에서의 기존 방법들보다 영상의 화질이 더 좋아진 반면 연산량은 훨씬 줄어들었다.

[참 고 문 헌]

- [1] Niklas Bjork and Charilaos Christopoulos, "Transcoder architectures for video coding", IEEE Transactions on Consumer Electronics, vol. 44, no. 1, pp. 101-110, 2000
- [2] Pedro A. A. Assuncao and Mohammed Ghanbari, "A Frequency Domain Video Transcoder for Dynamic Bit Rate Reduction of MPEG 2 Bit Streams", IEEE Trans. on Circuits and Systems for Video Tech., vol. 8, no. 8, pp. 953-967, 1998
- [3] Bo 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] T. Shanablesh 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
- [5] Jong-Nam Kim, Sung-Chael Byun, Yong Hoon Kim and Byung Ha 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
- [6] ITU-T Rec. H.263, "Videocodec for low bit rate communication", 1996