

동영상 변환 부호화를 위한 움직임벡터 재추출 기법에 관한연구

이병렬, 윤규섭, 박상희
연세대학교 전기·전자공학과

**Motion Vector Resampling Technique
for Format Transcoding and Spatial Resolution Reduction**

Byoung-Yeol Lee, Kyu-Seop Yoon, Sang-Hui Park
Dept. of Electrical and Electronic Engineering

Abstract - 많은 목적을 가진 영상 압축 표준으로 인해 원래의 목적으로 이미 부호화된 영상을 다른 목적에 맞는 영상으로 변환해야 될 경우가 발생하는데, 목적 비트율과 목적 크기, 목적 부호화 형식으로 재부호화시키는 과정을 변환 부호화라고 한다.

본 논문에서는 이미 부호화된 영상의 움직임 벡터 정보를 이용하여 변환 부호화 과정에서 새로운 움직임 벡터의 추정시간을 단축시킬 수 있고, 화질의 열화 및 계산량을 최소화 할 수 있는 새로운 방법들을 제안하고 영상의 비트율과 크기를 변환시키는 경우와 화상의 종류를 변환시키는 경우에 대해서 각각 실험을 통해 제안한 방법과 기존의 방법들을 비교, 검증하고자 한다.

1. 서 론

동영상들은 부호화시 목적에 따라 다양한 표준으로 압축된다. 이러한 경우 통신채널 상태, 저장매체 용량, 수신단 특성에 따라 기존에 부호화된 영상의 비트율의 변환 혹은 부호화되어진 압축 형식 변환이 이루어져야 할 필요가 있다. 특히 미리 부호화되어 있는 영상의 비트율과 목적 비트율의 차이가 매우 큰 경우나 수신단의 해상도에 한계가 있는 경우에는 비트율을 변환하기 전 영상의 크기를 먼저 축소해야 된다. 또한 영상이 이미 B화상을 포함한 부호화 형식으로 되어있다면 이 영상을 무선단말등의 수신단에서 화면출력하기 위해서는 B화상을 포함하지 않는 영상으로의 재부호화되어야 한다.

미리 부호화된 영상을 목적 비트율과 목적 크기, 목적 부호화 형식으로 변환 시키기 위해서 사용되는 가장 간단한 변환 부호화기는 종속 변환 부호화기로써 미리 부호화된 정보를 이용하여 완전히 복호화 한 후 다시 목적에 맞게 부호화하는 과정을 거치게 된다. 그러나 종속 변환 부호화기는 기존의 부호화된 정보를 이용하지 않으므로 전체 부호화 과정 중 60%의 시간을 소모하는 움직임 예측을 위해 많은 시간을 소비하게 된다. 따라서 이 시간을 보상하기 위해서 이미 부호화된 영상의 움직임 벡터를 재사용하는 기법들이 많이 연구되어지고 있다.[1][2]

기존의 움직임 벡터를 재사용하는데 있어서, 목적 부호화 형식과 목적 크기가 달라지게 되면 움직임 벡터의 수와 예측방향이 달라지게 되므로 새로운 움직임 벡터의 추정이 용이하지 않다. 하지만 일반적으로 움직임은 인접한 시간과 공간에 대해서 서로 큰 상관관계를 가지고 있는 특성을 이용하면 부호화 형식 혹은 크기의 변환시 기준의 움직임 벡터들을 재사용하여 변환된 영상의 움직임 벡터로 추정할 수 있다.

2. 영상 크기 및 형식 변환을 위한 움직임벡터 추정

2.1 영상 형식 변환을 위한 움직임 벡터 추정

그림 2.1과 2.2에서 화상들은 화면출력되는 순서대로 나열한 것이며, 각각의 영상에 붙어있는 첨자들은 부호화되는 순서를 나타낸다. 그림 2.1에서와 같이 미리 부호화된 영상(N=15, M=3)의 움직임 벡터 수와 방향은 화상 종류에 따라 다르다. I화상은 움직임 벡터를 수반하지 않고, P화상은 3번째 앞의 I,P화상을 기준으로 전방향 예측을 한다. 그리고 B화상은 1번째 혹은 2번째 앞, 뒤의 I,P화상을 기준으로하여 전,후 및 양방향 예측을 한다. 반면에 변환된 영상(N=∞, M=1)인 경우 그림 2.2와 같

이 I화상과 P화상만으로 구성되어있기 때문에 바로 앞의 화상을 기준으로 전방향 예측을 하므로 전방향 움직임 벡터를 추정해야 한다. 즉, 각기 다른 화상(P,B)의 움직임 벡터를 사용하여 P화상의 움직임 벡터를 추정해야만 한다.

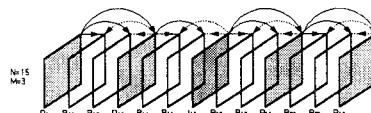


그림 2.1 변환 부호화기로 입력되는 화상

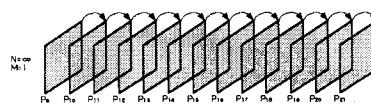


그림 2.2 변환 부호화기에서 출력되는 화상, N=∞, M=1

이를 위해서 이종 변환 부호화기에서는 각각의 화상들 사이의 움직임은 같은 속도로 움직이는 전체적인 움직임의 일부분으로 가정한다[3]. 전,후방향 움직임 벡터들을 이용하여 추정할 수 있는 후보 벡터들을 선정한 후 이렇게 추정된 여러개의 움직임 벡터 후보들 중에서 부호화 오차의 절대치의 합(sum of absolute difference, SAD)이 최소가 되게 하는 벡터를 선택하는 방법을 취한다.

이밖에 B화상을 제거시켜 I,P화상으로 이루어진 부호화 형식으로 변환하는 방법도 있다[4].

2.2 제안된 부호화 형식변환을 위한 벡터추정

이종 변환 부호화기는 8번에서 3번까지 한 화상을 재구성하게 된다. 따라서 화상을 재구성하고 재구성된 화상과 복호된 화상의 오차를 계산하는데 많은 시간이 소모되고 B화상의 움직임 벡터 추출과정에서 시간지연이 발생하게 된다[3]. 본 논문에서는 시간적 소모와 지연을 최소화 하고 새로운 움직임 벡터를 추정하는 방법을 3가지로 제안한다.

제안I. P화상 벡터 이용, 새로운 벡터 추정

소그룹내의 P화상의 움직임 벡터로 소그룹 내의 P,B화상들의 움직임 벡터를 추정하는 방법이다.

그림 2.1에서 B_{11}, B_{12}, P_{10} 화상들을 각각 그림 2.2의 P_{10}, P_{11}, P_{12} 의 화상으로 변환시키는 과정에서 새로운 움직임 벡터는 소그룹내의 가장 먼저 부호화 되어있는 P화상의 움직임 벡터를 이용한다. 따라서 상대적으로 나중에 부호화되어 있는 B화상의 움직임 벡터의 정보없이도 움직임 벡터의 추정이 가능하므로 시간적 지연이 최소화될 수 있다.

$$\textcircled{1} V_{9 \rightarrow 10}^{out}, V_{10 \rightarrow 11}^{out} \text{ 추정 : } 1/3 V_{7 \rightarrow 10}^{fwd}$$

$$\textcircled{2} V_{11 \rightarrow 12}^{out} \text{ 추정 : } V_{7 \rightarrow 10}^{fwd} - (V_{8 \rightarrow 9}^{out} + V_{9 \rightarrow 10}^{out})$$

I화상에는 움직임 벡터가 없으므로 I화상이 포함된 소그룹의 B화상과 I화상은 바로 전에 추정된 움직임 벡터를 같이 사용한다.

이러한 추정방법은 연속된 3개의 화상의 움직임 벡터를 추정하기 위하여 한번의 추정계산만 필요하므로 가장 단

순하며, 시간적 지연을 최소화할 수 있으나 매크로 블록이 인트라 부호화로 되어있는 경우에는 시간적으로 인접한 최대 6개의 매크로 블록의 움직임 벡터를 추정할 수 없는 경우가 발생한다.

제안II. P,B화상 벡터 이용, 부호화순 새로운 벡터 추정
B화상을 포함하고 있는 영상은 복호화하여 화면출력이 되는 순서와 부호화된 순서는 다르기 때문에 변환 부호화하기 위해서는 나중에 화면출력이 되는 화상의 움직임 벡터 정보를 가지고 있어야 한다.

그림 2.1에서 변환 부호화기에 입력되는 소그룹에서의 화상의 순서는 B_{11}, B_{12}, P_{10} 가 아니라 P_{10}, B_{11}, B_{12} 의 순서이다. 이들 화상에 대응되는 그림 2.2에서의 화상은 P_{10}, P_{11}, P_{12} 이 되고, 따라서 순서대로 P_{10}, B_{11}, B_{12} 의 움직임 벡터를 각각 이용하여 새로운 움직임 벡터인 $V_{9 \rightarrow 10}^{out}, V_{10 \rightarrow 11}^{out}, V_{11 \rightarrow 12}^{out}$ 를 추정한다. 그러면 구조적 복호화된 P화상을 저장할 공간이 필요하지 않고, 시간지연도 발생하지 않는다.

① $V_{9 \rightarrow 10}^{out}$ 추정 : $1/3 V_{7 \rightarrow 10}^{pred}$

② $V_{10 \rightarrow 11}^{out}$ 추정 : 소그룹 첫번째 B화상의 매크로 블록이 전후, 양방향 예측인 경우

$$V_{10 \rightarrow 11}^{pred} = -(1/2) V_{10 \rightarrow 11}^{bwd}, 1/2(V_{7 \rightarrow 11}^{pred} - (1/2) V_{10 \rightarrow 11}^{bwd})$$

③ $V_{11 \rightarrow 12}^{out}$ 추정 : 소그룹 두번째 B화상의 매크로 블록이 전후, 양방향 예측인 경우

$$1/2 V_{7 \rightarrow 12}^{pred} - V_{10 \rightarrow 12}^{bwd}, 1/2(-(1/2) V_{7 \rightarrow 12}^{pred} + V_{10 \rightarrow 12}^{bwd})$$

I화상은 움직임 벡터가 없으므로, 바로 앞 화상의 움직임 벡터를 다시 사용한다.

제안III. P,B화상 벡터 이용, 화면출력순 새로운 벡터 추정
입력되는 영상의 출력 순서대로 그림 2.1에서의 B화상의 움직임 벡터를 그림 2.2에서의 P화상의 움직임 벡터로 전환시키는 방법이다. 따라서 복호된 P영상은 위한 임시 저장공간과 M-1의 시간적 지연이 발생하게 된다. 그러나 제안II에서 제시된 한, 두화상 앞, 뒤의 움직임 벡터를 이용하는 방법보다는 동일한 화면출력순서의 화상의 움직임 벡터를 이용하기 때문에, 최적의 움직임 벡터에 더 근접한 움직임 벡터의 추출이 가능하다.

① $V_{9 \rightarrow 10}^{out}$ 추정 : 제안II ②와 동일

② $V_{10 \rightarrow 11}^{out}$ 추정 : 제안II ③와 동일

③ $V_{11 \rightarrow 12}^{out}$ 추정 : 제안II ①와 동일

2.3 부호화 크기 변환을 위한 움직임 벡터 추정

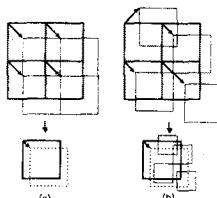


그림 2.3 1/2 크기 축소 움직임 벡터

그림 2.3과 같이 영상의 크기가 1/2로 줄게되면 4개의 변환 전 영상의 인접한 매크로블록 4개가 변환 후 영상의 1개의 매크로블록에 대응되므로 기존의 4개의 움직임 벡터로부터 축소영상의 1개의 움직임벡터를 추정해야 한다.

(1) 중간 거리 벡터 추정법 : 원영상의 4개의 움직임 벡터 중 하나와 나머지 3개의 벡터의 유클리디안 거리의 합이 최소가 되는 벡터를 선정한다.

(2) 평균벡터 추정법 : 4개의 움직임 벡터의 가로, 세로 성분의 산술평균을 새로운 움직임 벡터로 추정한다.

(3) AWW : 4개의 벡터에 각각 가중치를 두어 축소영상의 움직임 벡터를 추정한다.[1]

2.4 제안된 크기 변환을 위한 움직임 벡터 추정

기존에 제시된 수학적인 방법들은 계산량을 크게 줄일 수 있지만, 4개의 벡터의 특성의 차이가 크면 되면 최적의 움직임 벡터의 추정에는 다소 미흡하다.

(1) 최소오차 벡터 추정법

4개의 벡터 $V = (v_1, v_2, v_3, v_4)$ 를 추정 벡터의 후보로 선정하고, 각각의 벡터를 이용하여 축소된 영상들과의 오차(SAD)를 계산한 후 최소오차가 발생하는 벡터를 추정 벡터로 최종 선택한다.

$$e_i = \sum_{m=0}^{15} \sum_{n=0}^{15} |mb_o(m, n) - mb_{mc}(m, n)| \quad (3.1)$$

여기서 mb_{mc} , mb_o 는 각각 움직임 벡터 v_i 로 움직임 보상을 해준 매크로 블록과 원영상의 매크로 블록을 나타낸다. e_i 는 v_i 벡터로 예측된 매크로 블록과 원영상 매크로 블록과의 SAD이다.

$$\text{Minimum Error}(V) = v_i \in V \min e_i = e_k \quad (3.2)$$

이 추정법은 특색이 다른 4개의 벡터들을 이용하여도 최적의 움직임벡터와의 근접한 1개의 벡터를 추정할 수 있다.

3. 실험 및 결과

3.1 실험

첫 실험은 SIF(352×240) 형식, N=15, M=3, 비트율 1.15Mbps으로 이미 부호화된 영상의 움직임 벡터를 이용하여 크기의 변화없이 N=15, M=1의 영상의 움직임 벡터를 제안I, II, III 방법으로 추정한 실험이다. 두번쩨 실험은 CCIR 601(NTSC) 형식(704×480), N=15, M=1, 비트율 5Mbps로 부호화된 영상의 움직임 벡터를 재사용하여 비트율 1.5Mbps, SIF의 형식으로 비트율과 크기가 축소된 영상의 움직임 벡터를 추정한 실험이다. 기존에 제시된 축소영상의 움직임 벡터 추정 방법들과 본 논문에서 제안한 최소오차 벡터 추정법으로 실험하여 성능비교를 하였다. 비교대상으로 속도 변환부호화기로 완벽하게 재부호화된 영상과의 화질을 비교하고, 최적의 벡터와 오차를 계산한다. 또 각각의 방법에 대한 수행 계산량을 비교 평가한다.

마지막 실험은 두 실험에서 각각 가장 성능이 좋은 제안III과 최소오차벡터 추정법을 결합하여, 영상의 형식 및 크기가 동시에 변하는 경우 움직임 벡터를 추정하는 실험이다. 세실험 모두 football, Mobile & Calendar, Popple 3종의 영상 150프레임을 실험영상으로 사용하였다.

3.2 영상 형식 변환 움직임 벡터 추정 결과

(1) 화질비교

표3.1 형식변환 영상의 평균PSNR (dB)

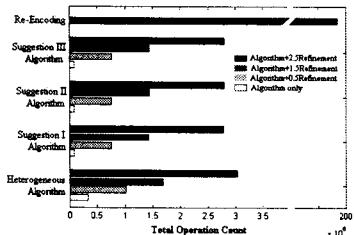
실험영상	벡터추출방법	제안 I	제안 II	제안 III
Football	전역움직임추정법	26.99		
	추정기법	25.91	26.16	26.22
	정제	26.43	26.59	26.64
	범위	26.65	26.80	26.84
Mobile & Calendar	전역움직임추정법	22.68		
	추정기법	21.44	22.14	22.14
	정제	22.43	22.65	22.65
	범위	22.56	22.67	22.67
Popple	전역움직임추정법	30.67		
	추정기법	29.82	30.07	30.13
	정제	30.11	30.34	30.39
	범위	30.24	30.48	30.50

(2) 추정 움직임 벡터 오차 비교

표3.2 형식 변환 영상의 움직임 벡터 오차 비교 (%)

실험영상	오차	제안 I	제안 II	제안 III
Football	오차	±0.5	68.30	68.75
	범위	±1.5	80.00	82.04
Mobile	오차	±0.5	89.17	88.98
	범위	±1.5	99.16	99.39
Popple	오차	±0.5	44.75	45.59
	범위	±1.5	63.56	63.71

(3) 수행 계산량 비교



3.3 크기 변환 움직임 벡터 추정 결과 (1) 화질비교

표 3.3 크기 변환 영상의 평균 PSNR (dB)

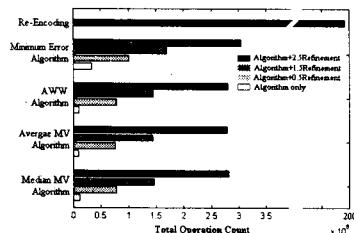
실험영상	벡터 추출방법	최소오차 벡터추정법		AWW	평균 벡터 추정법	중간거리 벡터추정법
		전역 움직임추정법	주정기법			
Football	전역 움직임추정법	26.99				
	주정기법	26.21	26.06	26.06	26.04	
	정제 범위 0.5	26.97	26.76	26.80	26.87	
Mobile & Calendar	전역 움직임추정법	22.68				
	주정기법	21.99	21.81	21.63	21.78	
	정제 범위 0.5	22.67	22.53	22.60	22.66	
Popple	전역 움직임추정법	30.67				
	주정기법	30.10	30.04	29.85	30.00	
	정제 범위 0.5	30.66	30.41	30.36	30.49	
	범위 1.5	30.66	30.55	30.58	30.58	

(2) 추정 움직임 벡터 오차 비교

표 3.4 크기 변환 영상의 움직임벡터 오차 비교 (%)

실험 영상	오차	최소오차 벡터추정법		AWW	평균 벡터 추정법	중간거리 벡터추정법
		오차 범위	정제 범위			
Football	오차	±0.5	83.37	69.67	69.36	76.99
	범위	±1.5	93.14	84.09	85.08	88.89
Mobile	오차	±0.5	96.70	90.59	94.07	95.69
	범위	±1.5	99.69	98.18	98.84	99.33
Popple	오차	±0.5	79.26	62.99	59.99	73.25
	범위	±1.5	86.38	76.16	78.03	81.35

(3) 수행 계산량 비교



3.3 형식 및 크기 변환 움직임 벡터 추정 결과 (1) 화질비교

표 3.5 형식 및 크기 변환 영상의 평균 PSNR (dB)

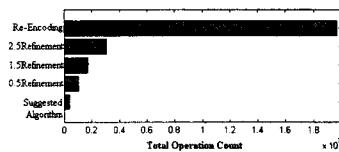
실험영상	벡터 추정방법	제안III+최소오차벡터 추정법	
		전역 움직임추정법	제안되추정법
Football	전역 움직임추정법	26.98757	
	제안되추정법	26.041271	
	정제 범위 0.5	26.573165	
Mobile & Calendar	전역 움직임추정법	22.677289	
	제안되추정법	21.994676	
	정제 범위 0.5	22.660608	
Popple	전역 움직임추정법	30.668350	
	제안되추정법	29.790981	
	정제 범위 0.5	30.274439	
	정제 범위 1.5	30.402583	

(2) 추정 움직임 벡터 오차 비교

표 5.6 형식 및 크기 변환 영상의 움직임벡터 오차 비교 (%)

실험영상	오차	제안III+중간벡터추정법
Football	오차 범위	±0.5이내 71.096970
	정제 범위	±1.5이내 83.701010
Mobile	오차 범위	±0.5이내 95.252525
	정제 범위	±1.5이내 99.656566
Popple	오차 범위	±0.5이내 57.133333
	정제 범위	±1.5이내 76.141414

(3) 수행 계산량 비교



5. 결론

본 논문에서 기존의 변환 부호화기에서 변환 영상의 움직임 벡터 추정 기법을 보완하고 새로운 방법을 제안하였다.

본 논문에서 얻어진 결론은 다음과 같다.

- 제안 I, II, III은 전역움직임 추정법에 비해 근소한 화질의 차이가 발생하지만 수행 계산량은 최소 92% 감소 한다. 또한 제안 I, II는 움직임벡터 저장을 위한 저장공간이 필요하지 않다.
- 최소오차벡터 추정법은 기존의 방법에 비해 3배정도 많은 계산량이 필요하지만, 정제과정까지 포함하면 결과적으로 평균 30%의 계산량이 감소하고, 화질은 1.5dB정도 개선된다.

최소오차벡터 추정법은 0.5범위의 정제를 통해서 전역움직임 추정법에 비해 0.03dB의 미세한 화질차이를 보이지만 수행시간은 50%이상 단축된다.

- 제안III과 최소오차벡터 추정법을 함께 사용하여 영상의 형식과 크기를 동시에 변환시킨 결과 전역움직임 추정법의 15%의 계산량으로 0.1dB 이하 작은 화질차이가 발생한다. 또한 평균 85%의 추정 움직임벡터들이 최적의 움직임벡터와의 오차범위 ±1.5이내에서 발생하여 본논문에서 제안한 움직임벡터 재사용 기법의 정확성이 검증되었다.

[참고 문헌]

[1] Bo Shen, Ishwar K. Sethi, and Bhaskaran Vasudev, "Adaptive Motion-Vector Resampling for Compressed Video Downscaling", IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.6, September 1999

[2] V.Bhaskaran and K. Konstantinides, Image and Video Compression Standards, Algorithms and Architectures. Boston, MA: Kluwer Academic, 1995

[3] Tamer Shanableh and Mohammed Ghanbari, "Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats", IEEE Transactions on Multimedia, Vol.2, No.2, June 2000

[4] Sujie J. Wee and John G. Apostolopoulos, "Field-to-Frame Transcoding with spatial and temporal downsampling", Hewlett-Packard Lab. Palo Alto, CA USA