

# 복호화기 측의 예측을 이용한 움직임 벡터 부호화

\*원광현 \*\*양정엽 \*\*\*전병우

성균관대학교 정보통신공학부

\*wkh12345@skku.edu, \*\*binbak1@ece.skku.ac.kr, \*\*\*bjeon@skku.edu

## Motion Vector Coding using Decoder-side Estimation

\*Won, Kwanghyun \*\*Yang, Jungyoup \*\*\*Jeon, Byeungwoo

Sungkyunkwan University, School of Information and Communication Engineering

### 요약

H.264/AVC 부호화 표준은 움직임 벡터를 부호화하기 위해 인접 블록이 가지는 다수의 움직임 벡터 중에서 확률적으로 해당 움직임 벡터와 가장 유사한 중간값을 예측 움직임 벡터로 사용한다. 이러한 방법은 다수의 움직임 벡터 중에서 어떤 움직임 벡터가 예측값으로 사용되었는지에 대한 추가 정보 없이 비트량을 효과적으로 감소시킬 수 있는 장점이 있으나, 중간값을 이용한 예측 움직임 벡터는 해당 움직임 벡터를 부호화하는데 소요되는 비트량을 항상 최소로 만드는 최적 예측값이 아니라는 단점이 있다. 이러한 문제를 해결하기 위해 다수의 인접 블록이 가지는 움직임 벡터 중에서 특정 움직임 벡터가 예측값으로 사용되었는지 표현하는 정보를 복호화기에 알려주도록 하여 항상 최적의 예측 움직임 벡터를 선택함으로써 부호화 효율을 향상시킬 수 있으나, 이에 대한 추가 정보를 부호화해야 하는 문제점이 발생하게 된다. 본 논문에서는 부호화기가 부호화 효율 측면에서 가장 우수한 움직임 벡터를 예측값으로 선택하고, 이를 부호화기가 스스로 예측함으로써 인접 블록이 가지는 다수의 움직임 벡터 중에서 특정 움직임 벡터가 예측값으로 사용되었는지에 대한 정보없이 움직임 벡터 부호화에 소요되는 비트량을 효과적으로 감소시키는 움직임 벡터 부호화 방법을 제안한다. 제안한 부호화기는 윌-왜곡 측면에서 가장 우수한 예측 움직임 벡터를 선택하고, 복호화기는 부호화기가 선택한 예측 움직임 벡터를 정합 기술을 사용하여 스스로 예측한다. 실험 결과는 제안 방법이 QCIF 및 CIF 영상에서 약 2.2%의 전체 비트량을 감소시킬 수 있음을 보여준다.

## 1. 서론

H.264/AVC 부호화 표준에서 부호화 된 정보들은 크게 모드 부호화(Mode Coding), 영상 부호화(Textual Coding), 움직임 벡터 부호화(Motion Vector Coding)의 세 종류로 구분될 수 있다. 특히 부호화된 비트스트림 내의 움직임 벡터 정보는 낮은 비트율로 갈수록 영상 정보에 비해 상대적으로 많은 부분을 차지하게 된다. 이는 움직임 벡터 부호화 기술이 전체적인 압축율을 향상시킬 수 있는 중요한 요소임을 의미한다. H.264/AVC 부호화 표준에서는 움직임 벡터를 부호화 하는데 소요되는 비트량을 감소시키기 위해 주변 블록이 가지는 움직임 벡터들의 중간값(Median)으로 얻어진 예측 움직임 벡터와 현재 블록의 움직임 벡터와의 차분값을 부호화하는 방법을 사용한다[1]. 이러한 이유는 주변 블록의 움직임 벡터들의 중간값으로 얻어진 예측 움직임 벡터가 현재 블록의 움직임 벡터와 가장 유사하다는 확률적 근거에 의한 것이다. 그러나 중간값을 이용한 예측 움직임 벡터는 현재 블록의 움직임 벡터와의 차분값을 항상 최소로 만드는 최적의 예측값은 아니다. 따라서 다수의 인접 블록이 가지는 움직임 벡터 중에서 중간값을 사용하여 선택한 예측 움직임 벡터보다 윌-왜곡(Rate-Distortion) 측면에서 보다 효과적인 예측 움직임 벡터가 존재하고 부호화기와 복호화기가 모두 이를 선택하는 것이 가능하다면, 항상 최적의 예측 움직임 벡터를 사용함으로써 보다 높은 압축 효율을 달성할 수 있음을 의미한다.

이러한 문제를 해결하기 위해 부호화기에서 윌-왜곡 측면에서 항상 최적인 예측 움직임 벡터를 선택하여 현재 움직임 벡터와의 차분값을 복호화기에 전송하고, 전송한 움직임 벡터가 어떤 예측값을 사용하였는지에 대한 정보를 추가적으로 전송하는 MVC(Motion Vector Competition) 방법에 제안되었다[2]. 이러한 방법은 다수의 인접 움직임 벡터 중에서 부호화 효율 측면에서 항상 최적인 움직임 벡터를 예측값으로 선택하고 어떤 움직임 벡터가 예측값으로 사용되었는지를 표현하는 정보를 복호화기에 알려주도록 하여 부호화 효율을 향상시킬 수 있다. 그러나, 어떤 움직임 벡터가 예측값으로 사용되었는지에 대한 추가적인 정보를 부호화해야 한다는 문제점이 발생하게 된다. 예를 들어, 최적 예측값을 선택함으로써 얻을 수 있는 비트 감소량보다 추가적인 정보를 부호화하기 위한 비트 증가량이 더 많을 경우에는 부호화 효율의 향상을 보장할 수 없다.

따라서 본 논문에서는 부호화기에서 어떤 움직임 벡터가 예측값으로 사용되었는지에 대한 추가적인 정보의 전송없이 복호화기에서 이를 선택하는 방법을 움직임 벡터 부호화 방법을 제안한다. 제안한 부호화기는 임의의 예측 움직임 벡터 후보 중에서 윌-왜곡 측면에서 가장 우수한 예측 움직임 벡터를 선택하고, 복호화기가 이를 스스로 예측할 수 있는 경우에는 최적 예측 움직임 벡터를, 그렇지 않은 경우에는 중간값을 이용한 예측 움직임 벡터를 사용하여 움직임 벡터를 부호화

※ 이 논문은 과학기술부의 재원으로 한국과학재단 국가지정 연구실사업으로 수행된 연구임(R0A-2006-10286-0(2008))

한다. 복호화기는 부호화기의 판단에 따라 정합 기술을 사용하여 스스로 예측 움직임 벡터를 예측하거나 중간값을 사용한 예측 움직임 벡터를 사용해 움직임 벡터를 복원한다. 제안 방법은 QCIF 및 CIF 영상에서 약 2.2%의 전체 비트량을 감소시킬 수 있음을 보여준다.

## 2. 움직임 벡터 부호화 기술

본 장에서는 H.264/AVC 부호화 표준에서 채택한 움직임 벡터 부호화 기술 및 MVC 방법에서 제안한 움직임 벡터 부호화 기술을 설명하고 이러한 방법의 한계점에 대한 문제를 제기한다.

### 가. H.264.AVC 부호화 표준의 움직임 벡터 부호화 기술

H.264/AVC 부호화 표준에서는 보다 향상된 압축 효율을 달성하기 위해 이미 복원된 주변 블록의 움직임 벡터를 통해 예측 움직임 벡터를 생성하고 이를 이용해 해당 움직임 벡터와의 차분 움직임 벡터(Differential Motion Vector)  $\overrightarrow{dmv}^D$ 를 식 (1)과 같이 계산하고, 이를 엔트로피 부호화(Entropy Coding)하여 복호화기에 전송하는 기술을 채택하고 있다. 이러한 이유는 주변 블록이 가지는 움직임 벡터가 현재 블록의 움직임 벡터와 매우 유사하여 차분값을 0에 가깝게 만들어 엔트로피 부호화 시에 압축 효율을 향상시킬 수 있기 때문이다.

$$\overrightarrow{dmv}^D = \overrightarrow{mv}^D - \overrightarrow{pmv}^D = \begin{cases} mv_x^D - pmv_x^D \\ mv_y^D - pmv_y^D \end{cases} \quad \text{식(1)}$$

그림 1은 H.264/AVC 부호화 표준에서 해당 움직임 벡터를 위한 예측값을 생성하는데 이용되는 공간적으로 인접한 블록의 움직임 벡터를 보여준다. 그림 1에서  $\overrightarrow{mv}^D$ 는 부호화할 현재 블록이 가지고 있는 움직임 벡터를 의미하며, 이는 식 (1)에서와 같이 x축 성분  $mv_x^D$ 와 y축 성분  $mv_y^D$ 로 구성된다.  $\overrightarrow{pmv}^D$ 는 주변 블록의 움직임 벡터를 통해 얻어지는 예측 움직임 벡터(Predictive Motion Vector)를 의미하며, 동일하게 x축 성분과 y축 성분으로 구성된다. 예측 움직임 벡터는 식 (2)와 같이 인접한 움직임 벡터의 중간값으로 계산된다.

$$\overrightarrow{pmv}^D = \begin{cases} pmv_x^D \\ pmv_y^D \end{cases} = \begin{cases} \text{median}(mv_x^A, mv_x^B, mv_x^C) \\ \text{median}(mv_y^A, mv_y^B, mv_y^C) \end{cases} \quad \text{식(2)}$$

그림 1에서와 같이  $\overrightarrow{mv}^A, \overrightarrow{mv}^B, \overrightarrow{mv}^C$ 는 주변 블록의 움직임 벡터를 의미하고, *median* 함수는 입력된 움직임 벡터들의 x축 성분과 y축

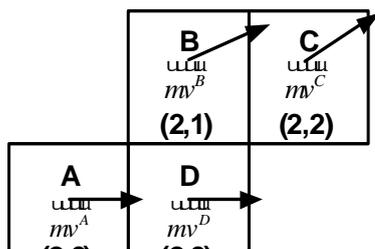


그림 1. 현재 블록과 주변 블록의 움직임 벡터

성분 각각의 중간값을 정의하는 함수이다. 이러한 방법은 중간값을 통해 얻어진 예측 움직임 벡터가 현재 움직임 벡터와 상당히 유사하기 때문에 엔트로피 부호화하는데 소요되는 비트량을 확률적으로 최소화할 수 있고, 부호화기와 복호화기가 동일한 *median* 함수를 사용할 수 있으므로 예측 움직임 벡터를 생성하기 위한 추가적인 정보가 불필요하다는 장점이 있다. 중간값을 통해 얻어진 예측 움직임 벡터는 확률적으로 해당 움직임 벡터와 가장 유사한 예측값이지만, 항상 움직임 벡터의 차분값을 최소로 만드는 최적의 예측값은 아니다. 예를 들어, 그림 1에서와 같이, 현재 움직임 벡터가 (2,0)이고, 중간값을 통해 얻어진 예측 움직임 벡터가 (2,1)이라고 가정하면, 차분 움직임 벡터는 (0,1)이 된다. 만약 중간값을 사용하지 않고 블록 A의 움직임 벡터를 예측 움직임 벡터로 선택할 경우, 차분 움직임 벡터는 (0,0)이 된다. 즉, 중간값을 이용한 예측 움직임 벡터는 항상 최적의 예측 움직임 벡터를 선택할 수 없다는 한계점을 지니게 될 뿐만 아니라, 움직임이 빠르거나 랜덤하게 발생할 경우 인접한 블록의 움직임 벡터  $\overrightarrow{mv}^A, \overrightarrow{mv}^B, \overrightarrow{mv}^C$ 와 해당 블록의 움직임 벡터  $\overrightarrow{mv}^D$  사이의 유사성이 더욱 감소하기 때문에 이러한 문제가 더 심각하게 발생할 수 있다.

### 나. MVC 방법의 움직임 벡터 부호화 기술

ITU-T 산하 VCEG(Video Coding Experts Group) 국제 표준화 기구의 차세대 비디오 부호화 기술 후보인 KTA(Key Technology Area) 소프트웨어에 채택된 MVC 방법은 주변 블록의 움직임 벡터 뿐 아니라 그림 2와 같이 참조 영상에 속해 있는 동일 위치의 움직임 벡터까지 예측 움직임 벡터의 후보로 사용한다[3]. 즉, 그림 1과 같이 공간적으로 인접한 블록의 움직임 벡터 및 그림 2와 같이 시간적으로 인접한 블록의 움직임 벡터 각각에 대해 식 (3)의 윌-웨곡 함수를 계산하고, 그 중에서 가장 부호화 효율이 우수한 예측 움직임 벡터를 선택하여 현재 움직임 벡터와의 차분값을 계산하여 복호화기에 전송함으로써 부호화 효율 측면에서 최적인 예측 움직임 벡터를 선택하는 것이 가능하다.

$$J = D + \lambda(R_r + \lambda_m R_m + \lambda_{mv} R_{mv}) \quad \text{식(3)}$$

그러나 식 (3)과 같이, 어떤 움직임 벡터가 예측값으로 사용되었는지를 표현하는 추가 정보를 전송하고 이에 대한 비트량  $R_{mv}$ 를 고려한 윌-웨곡 함수를 사용하기 때문에 항상 최적의 예측 움직임 벡터를 선택할 수 없다는 문제점이 있다. 예를 들어, 윌-웨곡 측면에서 보다 효과적인 예측 움직임 벡터가 존재하여 움직임 벡터를 부호화하는데 필요한 비트량을 감소시키더라도, 이를 표현하기 위한 추가 정보가 증가

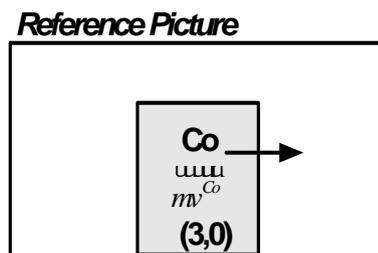


그림 2. 참조 영상의 움직임 벡터

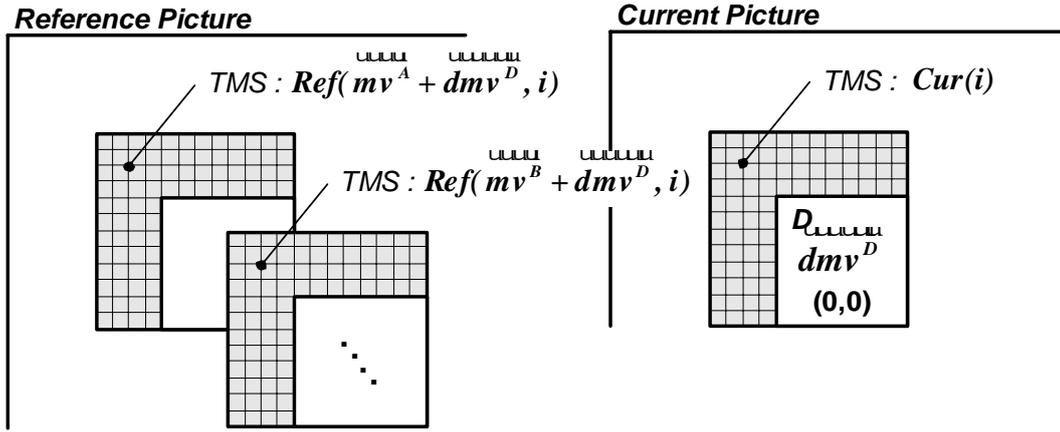


그림 3. 제안하는 복호화기 측의 예측을 이용한 움직임 벡터 부호화 기술

하게 되면 식 (3)의 율-왜곡 함수를 통해 최적의 예측 움직임 벡터를 선택하는 것이 불가능하게 된다. 즉, 예측 움직임 벡터 후보의 개수가 증가하면, 보다 효과적인 예측 움직임 벡터를 선택하여 부호화 효율이 증가할 가능성이 있지만, 추가 정보 역시 증가하게 되므로 항상 부호화 효율을 향상시킬 수 없음을 의미한다.

### 3. 제안한 움직임 벡터 부호화 기술

#### 가. 부호화기에서 최적의 예측 움직임 벡터 선택 기술

제안한 움직임 벡터 부호화 기술은 부호화기에서 최적의 예측 움직임 벡터를 선택함으로써 움직임 벡터의 차분값을 부호화하는데 소요되는 비트량을 항상 최소로 발생한다. 최적의 움직임 벡터를 선택하기 위한 예측 움직임 벡터 후보 집합은 현재 영상의 복원된 주변 블록에 속해 있는 움직임 벡터들과 참조 영상에 속해 있는 움직임 벡터들 간의 조합으로 선정한다. 선정된 예측 움직임 벡터 후보 중에서 현재 블록의 움직임 벡터와의 차분값을 비트량 측면에서 최소가 되게 하고, 또한 복호화기가 스스로 예측 가능한 최적의 예측 움직임 벡터를 부호화기에서 최종적으로 선택한다. 그러나 복호화기가 최적의 예측 움직임 벡터를 스스로 예측하는 것이 불가능할 경우가 발생하게 되므로, 부호화기는 이후 설명할 복호화기의 예측 움직임 벡터 예측 과정을 추가적으로 수행하여 복호화기가 스스로 예측 가능한지 여부를 판단할 수 있는 정보만을 추가로 전송한다.

#### 나. 복호화기에서 예측 움직임 벡터 예측 기술

제안한 움직임 벡터 부호화 기술은 복호화기에서 예측 움직임 벡터를 스스로 예측하기 위해 전송받은 차분값을 통해 움직임 벡터를 복원한다. 즉, 복호화기는 예측 움직임 벡터 후보들을 사용하여 현재 움직임 벡터 후보들을 선정한다. 현재 움직임 벡터 후보들을 선정 후, 이를 이용한 참조 영상의 복원된 주변 블록의 화소값과 현재 영상의 복원된 주변 블록의 화소값의 차분값을 계산하여 최소의 정합오류(Matching Error)를 가지게 되는 움직임 벡터 후보를 현재 부호화 할 움직임 벡터로 결정한다[4]. 그림 3은 복호화기 측에서 스스로 예측 움직임 벡터를 예측하기 위한 정합방법을 보다 상세히 나타낸 것이다. 그림 3에서  $TMS$ 는 현재 영상과 참조 영상의 복원된 인접 블록의 화소 집합이고,  $i$ 는  $TMS$ 에 포함되어 있는 화소들의 위치를 나타낸다. 그리고  $Ref$ 는 각각의 움직임 벡터 후보가 지정하는 참조 영상의  $TMS$  집

합의 화소값이고,  $Cur$ 은 현재 영상의 인접 블록에 있는  $TMS$  집합의 화소값이다.

전술한 바와 같이, 예측 움직임 벡터 예측 과정은 복호화기가 스스로 가능한지 여부를 판단하기 위해 부호화기에서 추가적으로 수행된다. 즉, 부호화기가 선택한 최적의 예측 움직임 벡터와 복호화기가 예측한 최적의 예측 움직임 벡터가 동일한 경우에만 복호화기가 스스로 예측 움직임 벡터를 예측 가능한 것이고, 그렇지 않으면 예측이 불가능한 것이다. 복호화기는 이를 표현하는 정보를 부호화기로부터 전송받아 예측 여부를 판단하여 부호화를 수행한다.

### 4. 실험 결과

제안한 움직임 벡터 부호화 방법에 대한 성능을 평가하기 위하여

표 1. 부호화 조건

프로파일	Baseline Profile
프레임율	QCIF: 15Hz CIF: 30Hz
탐색기법	Full Search
탐색범위	$\pm 32$
움직임 벡터	1/4-pel ME
참조 영상	4
하다마드(Hadamard) 변환	Used
RD 최적화	Used
심볼 모드	CAVLC
양자화 매개변수	I-Picture: 22,27,32,37 P-Picture: 23,28,33,38
GOP 구조	I-P-P-P Structure

표 2. 율-왜곡 실험결과 (비교대상 : H.264/AVC Baseline Profile)

Sequence	MVC Method		Proposed Method		
	BDPSNR [dB]	BDBR [%]	BDPSNR [dB]	BDBR [%]	
QCIF	Foreman	0.009	-0.157	0.125	-2.196
	Stefan	0.026	-0.373	0.141	-2.080
	Table	0.046	-0.856	0.101	-1.904
CIF	Foreman	0.039	-0.902	0.109	-2.525
	Stefan	0.072	-1.625	0.091	-2.030
	Table	0.066	-1.207	0.155	-2.848
Average	0.043	-0.853	0.120	-2.264	

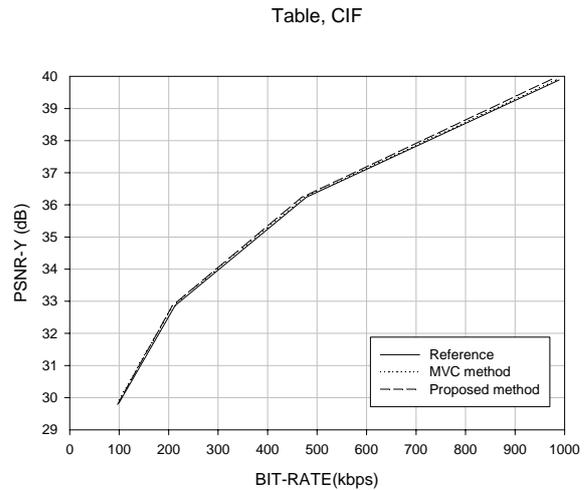
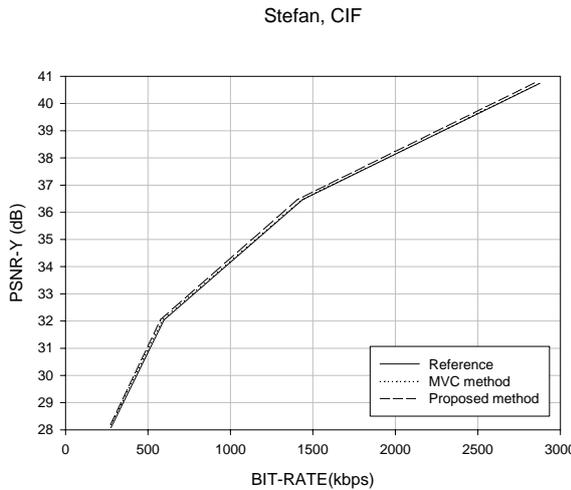
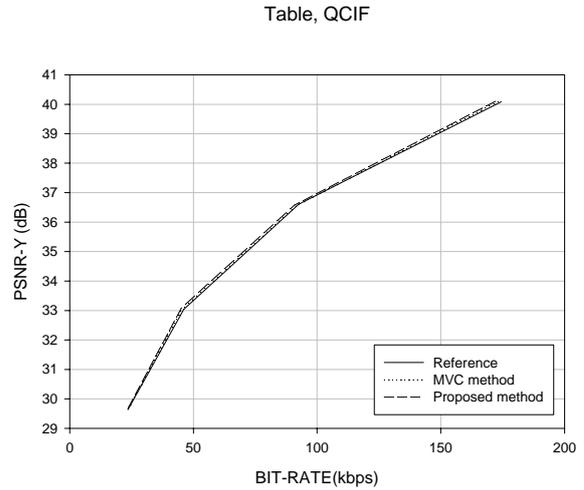
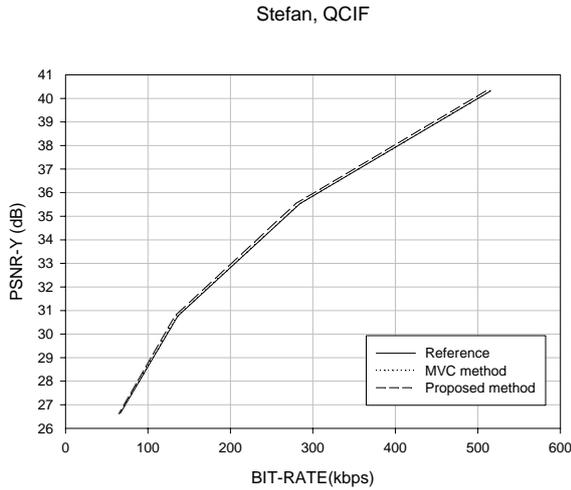


그림 4. 제안방법의 울-왜곡 성능의 비교

KTA 2.0 소프트웨어를 수정하여 사용하였다. 실험 영상은 QCIF와 CIF 포맷을 사용하였고, 자세한 부호화 조건은 표 1과 같다. MVC 방법과 제안 방법은 16x16 매크로블록 모드부터 분할된 4x4 매크로블록 모드까지 적용되었으나, 예외적으로 SKIP 모드는 H.264/AVC 부호화 표준에서 사용하는 방법을 적용하였다. MVC 방법과 제안방법의 울-왜곡 성능을 평가하기 위해 각각 H.264/AVC 표준과의 BDPSNR (Bjontegaard PSNR), BDBR(Bjontegaard BR)을 측정하였다[5].

표 2의 결과를 살펴보면, MVC 기술은 평균적으로 약 0.8%의 전체 비트량 감소 및 약 0.04dB의 화질 개선 효과를 얻을 수 있지만, 제안 기술은 약 2.2%의 전체 비트량 감소 및 약 0.12dB의 화질 개선 효과를 얻을 수 있었다. 즉, 제안 기술이 MVC의 기술보다 움직임 벡터 차별값을 효과적으로 감소시키는 한편, 추가 정보의 전송이 불필요하므로 울-왜곡 관점에서 보다 향상된 결과를 보여줄 수 있음을 의미한다. 특히 Stefan 및 Table 영상에서 가장 좋은 성능을 관찰할 수 있는데, 이는 제안방법이 움직임이 빠르고 랜덤할 경우에 보다 효과적임을 의미한다.

## 5. 결론

본 논문은 H.264/AVC 부호화 표준 및 MVC 방법에서 제안된 움직임 벡터를 부호화 방법의 한계점을 극복하기 위한 새로운 움직임 벡터 부호화 방법을 제안하였다. 제안한 움직임 벡터 부호화 방법은 복호

화기가 예측 할 수 있는 최적의 움직임 벡터를 선택하고, 이를 스스로 예측하는 방법이다. 실험 결과는 제안한 움직임 벡터 부호화 방법이 기존 H.264/AVC 부호화 표준 및 MVC 방법에 비해 보다 효과적으로 움직임 벡터를 부호화할 수 있음을 보여준다.

## 참고 문헌

- [1] JVT of ISO/IEC MPEG and ITU-T VCEG, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264, ISO/IEC 14496-10 AVC, 2003.
- [2] J. Jung and G. Laroche, "Competition-Based Scheme for Motion Vector Selection and Coding", Doc. of ITU-T SG16/Q6 VCEG, VCEG-AC06, July. 2006.
- [3] KTA Reference Software, <http://iphome.hhi.de/suehring/tml/download/KTA/jm11.0kta2.0.zip>
- [4] Y. Suzuki, C. S. Boon, and T. K. Tan, "Inter frame coding with template matching averaging," Proc. IEEE ICIP, vol. 3, Sep. 2007, pp. 409-412.
- [5] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Doc. of ITU-T SG16/Q6 VCEG, VCEG-M13, Apr. 2001.