

일반논문-10-15-2-13

인접 화소를 이용한 개선된 움직임 보상

서정훈^{a)}, 김정필^{a)}, 이영렬^{a)‡}

Improved Motion Compensation Using Adjacent Pixels

Jeong-Hoon Seo^{a)}, Jeong-Pil Kim^{a)}, and Yung-Lyul Lee^{a)‡}

요 약

H.264/AVC 표준은 영상간의 시간적 중복성(Temporal Redundancy)을 줄여 부호화 효율을 높이는 인터예측 방법을 사용한다. 하지만, 영상에서 발생할 수 있는 지역적인 밝기 변화를 효율적으로 부호화 할 수 없기 때문에, 화질의 열화가 발생하여 부호화 효율이 떨어지는 문제점을 지니고 있다. 본 논문은 지역적인 밝기 변화를 인접화소 및 움직임벡터 재조정을 이용하여 효율적으로 부호화 할 수 있는 방법을 제안한다. 제안하는 방법은, H.264/AVC와 비교하여 0.01 ~ 0.21dB의 BD-PSNR (Bjontegaard Delta Peak Signal-to-noise) 이득을 얻을 수 있었다.

Abstract

The H.264/AVC standard uses efficient inter prediction technologies improving the coding efficiency by reducing temporal redundancy between images. However, since H.264/AVC does not efficiently encode a video sequence that occurs a local illumination change, the coding efficiency of H.264/AVC is decreased when a local illumination change happens in video. In this paper, we propose an improved motion compensation using adjacent pixels and motion vector refinement to efficiently encode local illumination changes. The proposed method always improves the BD-PSNR (Bjontegaard Delta Peak Signal-to-Noise Ratio) of 0.01 ~ 0.21 dB compared with H.264/AVC.

Keywords : H.264/AVC, 움직임 보상, 인접 화소, motion compensation

1. 서 론

영상 부호화 표준들 중 MPEG (Moving Picture Experts Group)과 VCEG (Video Coding Experts Group)의 공동 개발로 가장 최근에 완성된 H.264/AVC (Advanced Video Coding)^{[1][3]}에는 기존의 영상 부호화 표준들보다 높은 압

축 성능을 얻기 위해 가변 블록 움직임 예측/보상 (Variable Block-size Motion Estimation /Compensation), 다양한 방향성을 통한 화면 내 예측 (Intra Prediction), 다중 참조 프레임 (Multiple Reference Frame), 이산 역현 변환 (Discrete Cosine Transform)에 기반을 둔 4×4 정수 역현 변환 (Integer Cosine Transform), 그리고 영상 부호화 표준에 직접적으로 포함되지 않았지만 사용이 권고되고 있는 윌-왜곡 최적화 (Rate-Distortion Optimization)^[4]와 같은 기술들이 사용된다. 또한, 보다 좋은 화질, 보다 큰 영상과 같은 전문적인 영상 환경을 지원하기 위해 H.264/AVC 개정안 (Amendment) 1 : FRExt (Fidelity Range Extensions)가

a) 세종대학교 컴퓨터공학과 DMS 연구실

DMS Lab., Dept. of Computer Engineering, Sejong University

‡ 교신저자 : 이영렬(ylllee@sejong.ac.kr)

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원 사업의 연구결과로 수행되었음(ITA-2009-(C1090-0903-0011))

· 접수일(2009년12월24일), 수정일(2010년2월3일), 게재확정일(2010년2월17일)

2004년 7월에 완성되었다. FRExt에서는 다양한 컬러 형식 (Color Format), 8비트 크기 이상의 영상 표본 (Sample)을 지원하며, 부호화 성능 향상을 위해 8×8 블록 단위의 화면 내 예측, 8×8 정수 역현 변환 등을 제공하고 있다⁵⁾.

움직임 예측 / 보상은 매크로 블록을 부호화하기 위하여, 이전에 부호화를 마친 프레임에서 현재 매크로 블록과 가장 유사한 블록을 찾는 움직임 예측과정을 통하여 움직임 벡터 (Motion Vector)를 얻고, 얻어진 움직임 벡터를 이용하여 예측 블록을 생성한 뒤, 부호화 할 블록과의 움직임 보상을 통하여 부호화 할 잔여신호를 생성하는 기술이다. 이때, 16×16 크기뿐만이 아닌 최소 4×4 크기의 블록 단위까지 움직임 예측 / 보상을 수행할 수 있기 때문에, 복잡한 움직임을 보이는 영상에서도 효율적인 부호화가 가능하다.

하지만, 움직임 예측이 수행되는 프레임에서 조명변화 혹은 시점변화로 인한 지역적인 밝기 변화가 발생하고, 현재 부호화할 매크로 블록에서는 밝기 변화가 발생하지 않았다고 가정한다면, 부호화 효율이 낮아지게 될 것이다.

본 논문에서는 영상 부호화 시에 밝기 변화로 인해 부호화 효율이 저하되는 문제점을 해결하기 위해 인접 화소를 이용한 매크로블록 기반의 개선된 움직임 보상 방법을 제안한다. 이 방법을 통하여 전체적으로 발생하는 조명변화 뿐만 아니라 지역적으로도 발생할 수 있는 미세한 밝기 변화도 효율적으로 부호화가 가능하기 때문에 기존의 방법에서 발생하는 문제점을 보완할 수 있다.

제 2장에서는 H.264/AVC에서의 화면 간 예측 (Inter Prediction)과 조명 보상 기술로써 동영상 부호화 표준에 채택된 가중치 예측 (Weighted Prediction)⁶⁾을 설명하고, 제 3장에서는 제안하는 방법인 인접 화소를 이용한 움직임 보상에 대하여 설명한다. 제 4장에서는 제안하는 방법과 H.264/AVC 에서 사용되는 방법 간의 비교 실험 결과를 제시하고, 제 5장에서 결론과 함께 본 논문의 끝을 맺는다.

II. H.264/AVC의 화면 간 예측

본 장에서는 H.264/AVC에서 사용되는 화면 간 예측과

가중치 예측에 대하여 설명한다. 화면 간 예측은 이전에 부호화된 프레임을 통하여 시간적 중복성을 제거하는 방법이고, 가중치 예측은 밝기 변화를 효율적으로 부호화 할 수 있는 방법이다.

1. 가변 블록을 이용한 움직임 예측 / 보상

매크로 블록 단위로 부호화가 수행되는 H.264/AVC 화면 간 예측에서는, 16×16 크기의 매크로 블록이 최소 4×4 크기의 블록으로 분할되어 움직임 예측 / 보상을 수행할 수 있다. 즉, 그림 1과 같이 하나의 16×16 블록, 두 개의 16×8 블록, 두 개의 8×16 블록, 네 개의 8×8 블록으로 움직임 예측 / 보상이 가능하다. 여기서, 8×8 블록인 경우, 각 8×8 블록은 앞서 설명한 것과 유사한 방식으로 4×4 블록까지 분할될 수 있다.

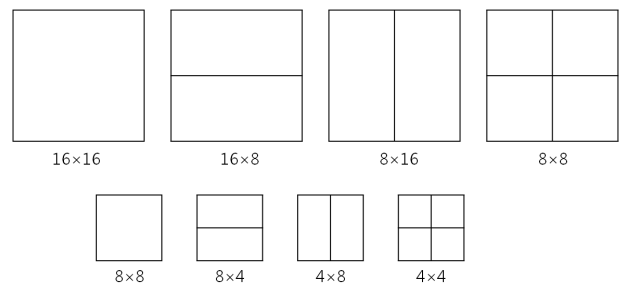


그림 1. H.264/AVC의 하위 매크로 블록 형태
Fig. 1. Feature of sub-macroblock partitions in H.264/AVC

이와 같이, 매크로 블록을 다양한 형태의 블록으로 나누어 움직임 예측/보상을 수행하는 방법을 가변 블록 움직임 예측/보상(Variable Block Motion Estimation/Compensation)이라고 한다.

각각의 하위 매크로 블록들은 독립적인 움직임 예측을 수행하여 움직임 벡터를 얻기 때문에, 현재 부호화 할 매크로 블록이 갖는 하위 매크로 블록들의 형태에 대한 정보와 하위 매크로 블록들이 갖는 움직임 벡터 및 참조프레임에 대한 정보가 부호화 된다. 일반적으로 큰 하위 매크로 블록은 프레임의 단조로운 영역에 적합하고 작은 하위 매크로 블록은 복잡한 영역에 적합하다.

2. 움직임 예측

일반적으로 동영상 부호화에서 움직임 예측은 현재 부호화 할 매크로 블록과 참조 프레임의 움직임 탐색 영역 (Search Range) 안에 존재하는 후보 블록 간의 매칭 (Block Matching)을 통하여 현재 블록과 가장 유사한 참조 블록의 위치를 가리키는 움직임 벡터를 결정한다. 블록 매칭 시에 가장 많이 사용되는 척도로 식 (1)과 같은 SAD (Sum of Absolute Differences)가 있다.

$$SAD^{(m,n)}(x,y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} |C(i,j) - R(i+x,j+y)| \quad (1)$$

$C(i,j)$ 는 현재 프레임의 (i,j) 에 위치하는 화소 값, $R(i,j)$ 는 참조 프레임의 (i,j) 에 위치하는 화소 값을 나타내며, 여기서 i 와 j 는 프레임 내 화소의 수평, 수직 위치를 나타낸다. S 와 T 는 블록 매칭에 사용되는 블록의 수평, 수직 크기, m 과 n 은 현재 프레임 블록의 수평, 수직 위치를 나타낸다. x 와 y 는 각각 후보 움직임 벡터의 수평, 수직 성분을 나타낸다.

식 (1)을 이용하여 탐색 영역의 후보 블록들 중, 최소의 SAD 값을 갖는 블록이 현재 부호화 할 블록과 가장 유사한 블록이라고 판단하고 움직임 벡터를 결정한다.

3. 움직임 보상

움직임 예측을 통해 움직임 벡터를 얻어내면 이를 이용하여 예측 블록을 생성하고 현재 부호화 할 블록과의 움직임 보상을 통하여 식 (2)와 같이 시간적 중복성을 제거한 잔차신호를 생성한다.

$$residual(i,j) = C(i,j) - R(i+x,j+y) \quad (2)$$

$C(i,j)$ 는 현재 프레임의 (i,j) 에 위치하는 화소 값, $R(i,j)$ 는 참조 프레임의 (i,j) 에 위치하는 화소 값을 나타내며, x 와 y 는 각각 후보 움직임 벡터의 수평, 수직 성분을 나타낸다.

위와 같은 일련의 과정들을 거쳐 최종적으로 생성된 잔차신호를 부호화 하는데, 가장 효율적인 매크로 블록 형태

를 결정하기 위해, 그림 1과 같은 모드들에 대하여 직접 부호화를 수행하고 블록 형태를 결정하는 율-왜곡 최적화 방법을 사용한다.

4. 가중치 예측

H.264/AVC는 가중치 예측 방법을 채택함으로써 시간에 따라 밝기가 변하는 영상에서 화질이 크게 하락하는 문제점을 해결하였다. 가중치 예측은 일반적으로 식 (3)과 같은 선형 조명 모델을 기반으로 하며, 슬라이스 단위로 휘도 (Luma)와 색차 (Chroma) 성분에 대해 조명변화를 보상한다.

$$I_t(i) \cong \alpha \cdot I_{t+\Delta t}(i) + \beta \quad (3)$$

$I_t(i)$ 는 시간 t 에 위치하는 프레임의 좌표 i 에 위치하는 화소를 의미하고, α 는 스케일링 인자 (Scaling Factor)로서 주로 영상의 대비 (Contrast)를 조정하는 목적으로 사용되며, β 는 오프셋 인자 (Offset Factor)로서 주로 영상의 밝기 (Brightness)를 조정하는 목적으로 사용된다. 이때, 스케일링 인자와 오프셋 인자의 계산 방법과 복호기로 전달하는 방식에 따라 명시적 (Explicit) 모드와 묵시적 (Implicit) 모드로 나눌 수 있다.

명시적 모드는 한 장의 프레임을 참조하는 단방향 예측 모드의 경우 한 프레임에 대한 인자 α , β 를 계산하고, 두 개의 프레임을 참조하는 양방향 예측 모드 (Bi-predictive Mode)의 경우 두 프레임에 대한 인자 α_0 , β_0 와 α_1 , β_1 을 계산하여 슬라이스 단위로 복호기에 전달된다. 인자 값들의 결정은 표준에서 정의되어 있지 않기 때문에, 부호화 효율을 높이기 위한 다양한 방법들을 사용할 수 있다. 기존에 소개된 방법들은 최소 제곱법 (Least Square Method)을 이용한⁷⁾ 방식과 계산량을 줄이기 위하여 α 와 β 가운데 한 가지 값만을 계산하여 전송하는 방법⁶⁾ 등이 있다.

이처럼 명시적 모드는 각 참조 프레임에 대한 인자를 직접 계산하여 조명변화를 보상하고, 그것을 명시적으로 복호기에 전달하는 방식이다.

묵시적 모드는 명시적 모드와는 달리 양방향 예측 모드의 경우에만 사용이 가능하며, 명시적으로 인자를 복호기

로 전달하지 않는다. 복호기는 부호기로부터 전달된 POC (Picture Order Count)를 이용하여 현재프레임과 참조프레임 간의 상대적인 거리를 계산하여, 그 거리의 비율로 스케일링 인자 α 를 부호기와 동일한 방식으로 계산한다.

그러나 이러한 방식의 조명변화 보상 방식은 슬라이스 단위로 이루어지기 때문에 지역적인 조명 변화를 효율적으로 부호화 할 수 없다는 단점이 있다.

III. 인접화소를 이용한 개선된 움직임 보상

본 장에서는 제안하는 움직임 보상 방법의 설명과 제안하는 방법과 참조 모델의 통합 구현과정을 설명한다.

1. 인접 화소를 이용한 움직임 보상

영상 내에서 조명변화가 발생하는 경우, 기존의 SAD 척도를 통하여 얻어진 움직임 벡터를 이용하여 움직임 보상을 수행한다면, 부호화 할 잔차신호가 증가하여 부호화 효율이 낮아질 것이다. 이를 방지하기 위하여, 앞선 움직임 예측 과정에서 얻어진 움직임 벡터의 인접 화소들과 현재 부호화 화소의 인접 화소들의 평균을 구하여 조명 변화의 유무를 판단한다. 움직임 벡터를 통해 생성된 예측 블록에 인접화소를 이용한 움직임 보상은 식 (4)로 정의된다.

$$residual_{adj}(i, j) = C(i, j) - clip\{R(i+x, j+y) - (m_r - m_c)\} \quad (4)$$

m_r 은 참조 블록 인접 화소들의 평균을 나타내고, m_c 는 현재 부호화 할 블록 인접 화소들의 평균을 나타내며, $clip(1)$ 함수는 예측 블록 화소 값의 범위를 제한해 주는 함수이다. 그 외의 인덱스들은 식 (2)와 동일하다. 여기서 m_r , m_c 를 인접 화소들의 평균이 아닌 블록 내 화소들의 평균이라고 정의 한다면, 보다 정확한 조명 변화를 감지할 수 있으나, 이에 대한 정보를 복호기로 전송하여야 하기 때문에 추가적인 비트를 필요로 한다^[8]. 하지만 인접화소를 이용한다면 복호기에서도 부호기에서도 동일하게 평균값을 구할 수 있으므로, 평균값을 부호화하지 않고 움직임 보상이 가

능하게 된다.

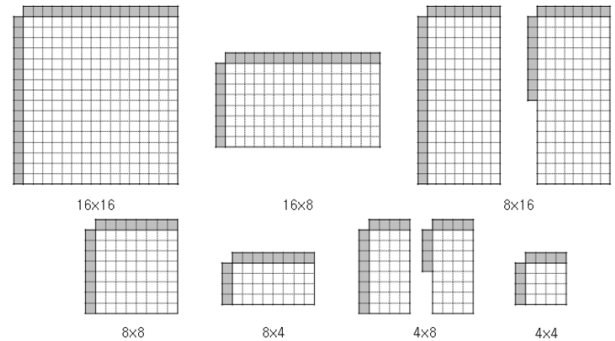


그림 2. 하위 매크로 블록에 사용되는 인접화소들
Fig. 2. Adjacent pixels used in sub-macroblock

그림 2는 매크로 블록 모드에 따라 m_r , m_c 를 구하기 위해 사용되는 화소를 보여준다. 여기서 음영이 없는 부분이 현재 부호화 할 블록의 화소들 또는 참조 블록의 화소들이고, 음영이 있는 부분이 제안하는 움직임 보상에 사용되는 인접 화소들이다.

2. 움직임 벡터 재조정

인접 화소를 이용한 움직임 보상을 수행할 경우, 움직임 예측을 통하여 얻어진 움직임 벡터가 이전 프레임의 지역적인 조명 변화에 영향을 받은 움직임 벡터일 수 있다. 따라서 보다 정확한 움직임 벡터를 얻기 위해, 얻어진 움직임 벡터를 중심으로 1/4화소 단위의 $\pm\alpha$ 범위에서 인접 화소를 이용한 움직임 벡터 재조정을 한다.

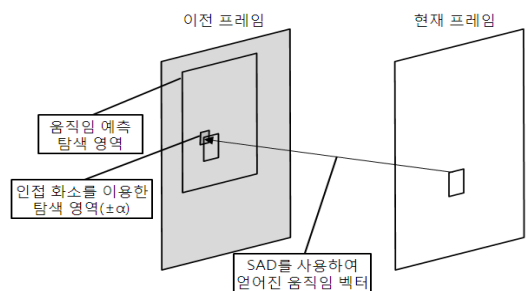


그림 3. 움직임 벡터 재조정 방법
Fig. 3. A method of motion vector refinement

그림 3은 움직임 벡터 재조정 단계의 단계를 보여주는 그림으로, 먼저 SAD를 이용하여 탐색영역 내부에서 움직임 벡터를 얻어내고, 인접 화소를 이용한 탐색 영역에서 최종 움직임 벡터를 결정한다.

3. 제안하는 방법을 위한 시그널링

일반적인 영상의 조명변화는 주로 지역적으로 발생하기 때문에 영상 내의 모든 영역, 다시 말해서 프레임 내의 모든 블록들이 조명변화 보상을 필요로 하는 것은 아니다. 오히려 일부 블록들은 조명변화 수행 시 부호화 효율이 저하될 수도 있기 때문에 블록 단위로 적응적인 조명변화 보상을 수행할 필요가 있다. 따라서 각 매크로블록에 대하여 조명변화 보상 여부를 복호기에 전달하기 위하여, 부호기는 매크로블록 단위의 시그널링 비트인 Adjacent_flag를 복호기로 전달한다. 복호기는 Adjacent_flag가 0인 경우 기존의 방식대로 복호화하며, Adjacent_flag가 1인 경우 예측블록의 인접화소와 현재 복호화 할 블록의 인접화소를 이용하여 복호화한다.

4. 제안하는 방법과 참조 모델의 통합

본 논문에서 제안하는 인접화소를 이용한 움직임 보상 방법은 참조 모델을 기반으로 하는 부/복호기와 통합된다. H.264/AVC를 기반으로 하는 참조 모델은 여러 종류의 profile을 지원하는데, 그 중에 가장 예측을 지원하는 main profile에 제안하는 방법을 적용한다. 색차 신호에서 얻을 수 있는 부호화 효율 향상은 휘도 성분에 비하여 작은 반면

에 시그널링 비트는 휘도와 색차신호에 대하여 동일하게 요구(휘도, 색차 신호 각 1비트)되기 때문에 부호화 효율 향상이 미미하거나 오히려 낮아 질 수 있다. 따라서 휘도 신호에 대해서만 제안하는 방법을 적용하였으며, 움직임 벡터 재조정을 적용할 수 없는 Skip 모드는 제안하는 방법을 이용한 성능 향상이 크지 않아서 Skip 모드를 제외한 모드들에 대해서 제안하는 방법을 적용하였다.

앞서 설명한 바와 같이, 매크로블록의 특성에 따라서 기존의 움직임 보상을 수행하는 것이 더 좋은 성능을 보일 수 있기 때문에 기존의 방법을 이용하여 결정된 매크로 블록 모드와 결정된 매크로 블록 모드에 제안하는 방법을 적용한 경우의 율-왜곡 비용 비교를 통하여 어떤 움직임 보상을 사용할지 결정한다. 움직임 보상 방법 결정을 위한 비용 (J_{MODE}) 계산식은 다음과 같다.

$$J_{MODE}(S_k, I_k | Q, \lambda_{MODE}) = D(S_k, I_k | Q) + \lambda_{MODE} R(S_k, I_k | Q) \quad (4)$$

S_k 는 k 번째 매크로블록, I_k 는 k 번째 매크로블록의 모드를 나타낸다. Q 와 λ_{MODE} 는 각각 사용된 Q_p 와 라그랑지 승수(Lagrange Multiplier)를 나타내고, 함수 D 와 R 은 매크로블록에 대한 왜곡 정도(Distortion)와 소요 비트수(Rate)를 나타낸다.

율-왜곡 최적화 과정을 통해 제안하는 방법의 성능이 높은 경우에는 Adjacent_flag를 1로 설정하여 부호화하게 되며, 반대의 경우 0으로 설정하여 부호화하게 되는데, 이러한 과정은 움직임 벡터 재조정 단계에서도 수행된다.

상기 내용들이 포함된 제안하는 부/복호기는 그림 4와 같다.

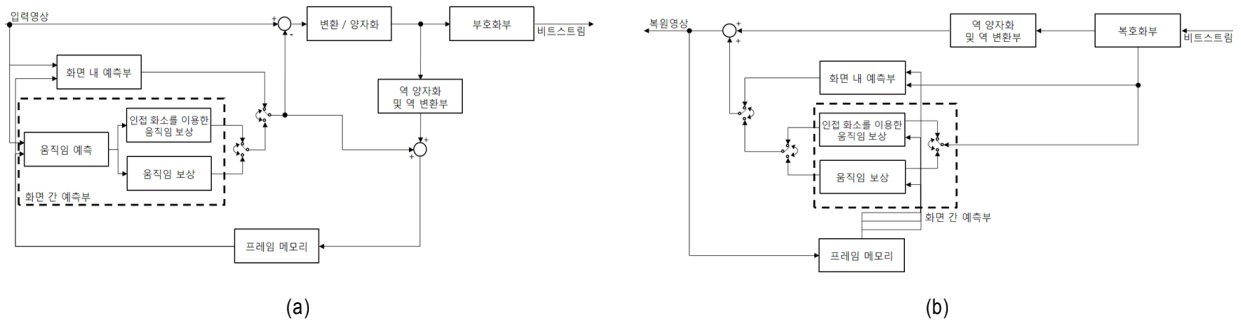


그림 4. 제안하는 방법을 이용한 부호기 및 복호기의 구조 : (a) 부호기, (b) 복호기
 Fig. 4. Structure of encoder and decoder using the proposed method : (a) encoder, (b) decoder

IV. 실험 결과

제안하는 방법은 H.264/AVC Joint Model 12.4^[9]를 기반으로 구현하였으며, main profile에서 각 4개의 720p, 1080p 영상들에 대하여 실험하였다. 자세한 실험 조건은 표 1과 같다.

표 1. 실험 조건

Table 1. Experimental conditions

GOP Structure		IPPP
Total Encoded Frames		720p : 150, 1080p : 125
Quantization Parameters		22, 27, 32, 37
Search Range		±32
Multiple Reference Frames		4

표 2. H.264/AVC와 제안하는 방법의 성능 비교

Table 2. Performance comparison of H.264/AVC and proposed method

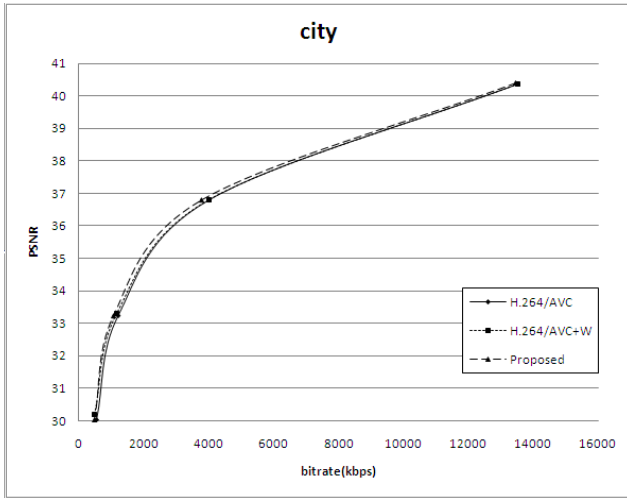
sequence	QP	H.264/AVC		H.264/AVC + W		Proposed		H.264 Vs. H.264+W		H.264 Vs. Proposed		
		PSNR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	BD-PSNR (dB)	BD-Bitrate (%)	BD-PSNR (dB)	BD-Bitrate (%)	
720P	bigships	22	40.49	8394.92	40.49	8395.63	40.51	8302.41	-0.001	0.05	0.177	-6.108
		27	37.14	2634.55	37.14	2635.02	37.13	2509.46				
		32	33.93	852.25	33.93	852.96	33.95	779.91				
		37	31.3	342.47	31.3	342.94	31.34	315.76				
	crew	22	41.98	7274.93	41.98	7106.07	41.98	7045.42	0.205	-7.69	0.15	-5.899
		27	39.41	2291.51	39.48	2188.39	39.41	2154.85				
		32	37	914.94	37.18	885.2	37.05	868.15				
		37	34.71	442.63	34.93	422.04	34.79	424.77				
	city	22	40.37	13524.86	40.37	13525.57	40.41	13450.03	0.08	-2.747	0.217	-6.875
		27	36.8	3998.59	36.8	3999.06	36.8	3782.6				
		32	33.24	1201.64	33.3	1174.17	33.23	1078.05				
		37	30.03	533.98	30.19	489.98	30.02	491.9				
raven	22	42.92	4670.18	42.92	4670.88	42.91	4535.43	0.061	-1.908	0.187	-5.172	
	27	40.14	1725.72	40.15	1720.57	40.12	1620.86					
	32	36.97	750.44	37.03	734.86	36.98	704.53					
	37	33.97	391.76	34.02	385.88	33.97	380.38					
720p Average		-	-	-	-	-	-	0.089	-3.074	0.185	-6.014	
1080P	parkjoy	22	39.38	70198.71	39.39	70426.2	39.39	70203.54	-0.008	0.163	0.013	-0.294
		27	35.76	36008.61	35.76	36087.49	35.76	35990.24				
		32	32.07	17045.47	32.07	17080.82	32.08	17023.36				
		37	28.81	7656.61	28.82	7667.33	28.82	7594.09				
	sunflower	22	43.76	5919.39	43.77	5921.55	43.72	5672.41	0.237	-6.753	0.133	-3.621
		27	41.75	2749.4	41.86	2719	41.75	2642.65				
		32	39.12	1394.04	39.39	1347.36	39.18	1356.94				
		37	36.38	764.99	36.62	740.69	36.46	756.81				
	crowdrun	22	39.19	59315.86	39.19	59316.56	39.2	59215.27	0.001	-0.038	0.042	-0.981
		27	35.74	26706.42	35.74	26706.89	35.74	26532.15				
		32	32.32	12607.85	32.32	12606.37	32.33	12501.62				
		37	29.21	6130.05	29.22	6127.05	29.23	6009.43				
toys and calendar	22	40.41	18205.59	40.41	18206.29	40.41	17944.19	0.043	-1.861	0.118	-5.9	
	27	38.27	4214.6	38.27	4212.52	38.27	3988.66					
	32	36.12	1892.85	36.19	1875.99	36.18	1779.75					
	37	33.77	1090.63	33.94	1068.68	33.87	1041.96					
1080p Average		-	-	-	-	-	-	0.068	-2.122	0.077	-2.7	
Total Average		-	-	-	-	-	-	0.078	-2.598	0.131	-4.356	

제안하는 방법의 성능 평가를 위하여,

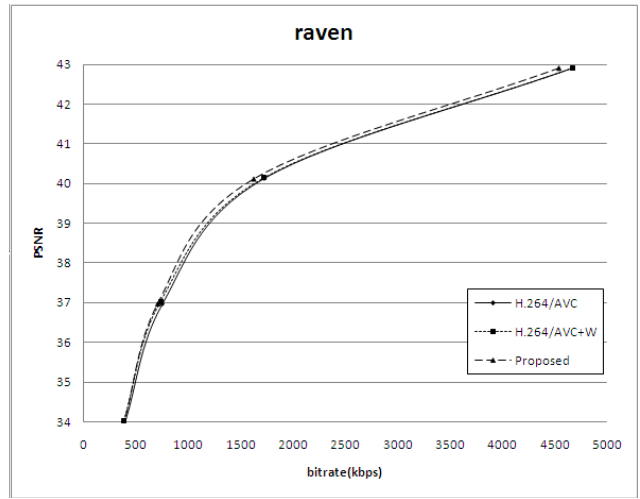
- 1) 기존의 움직임 보상(H.264/AVC) 방법
 - 2) 기존의 움직임 보상 + 가중 예측 방법
 - 3) 제안하는 방법과 기존의 방법을 선택적으로 사용하는 방법
- 세 가지 방법에 대한 실험을 수행하였다.

1. 율-왜곡 성능 비교

표 2는 H.264/AVC와 가중 예측 및 제안하는 방법 간의 BD-PSNR / BD-Bitrate^[10] 성능을 비교하여 표로 나타낸

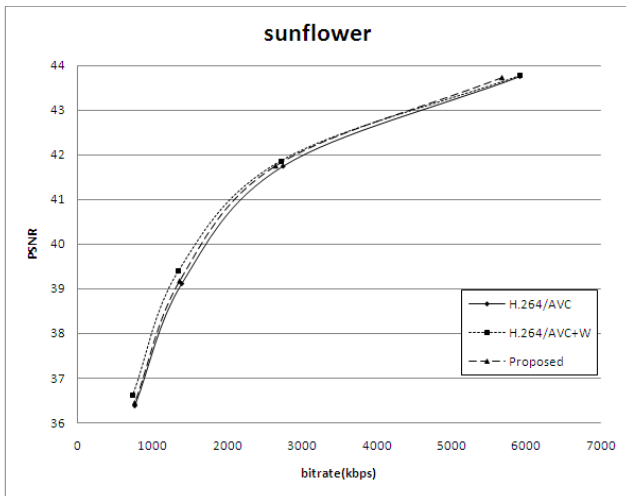


(a)

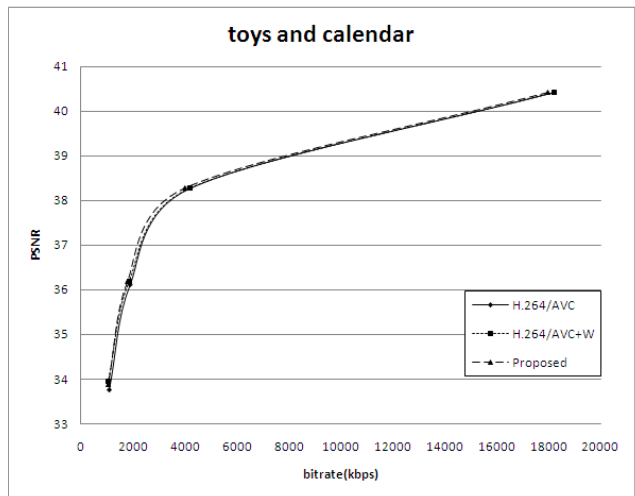


(b)

그림 5. 720p 영상의 율-왜곡 곡선 : (a) city, (b) raven
 Fig. 5. R-D curves of 720p sequence : (a) city, (b) raven



(a)



(b)

그림 6. 1080p 영상의 율-왜곡 곡선 : (a) sunflower, (b) toys and calendar
 Fig. 6. R-D curves of 1080p sequence : (a) sunflower, (b) toys and calendar

것이고 그림 5, 6은 표 2를 바탕으로 도시한 R-D curve이다. 자료에서 알 수 있듯이 전체적인 밝기 변화가 발생하는 영상을 제외한 모든 영상에서 제안하는 방법이 다른 두 가지 방법에 비하여 높은 부호화 효율을 보였다. 또한 전체적인 밝기 변화가 발생하는 경우에도 높은 bitrate 영역에서는 가중 평균보다 높은 성능을 보였다.

2. 부호기의 복잡도 분석

그림 7은 H.264/AVC와 제안하는 방법의 복잡도를 부호화 시간 측면에서 비교한 것이다. 제안하는 방법은 Skip모드가 아닌 경우에 대하여 표 3과 같이(8×8 이하의 파티션도 이와 유사하게 수행) 움직임 벡터 재조정을 위한 율-왜곡

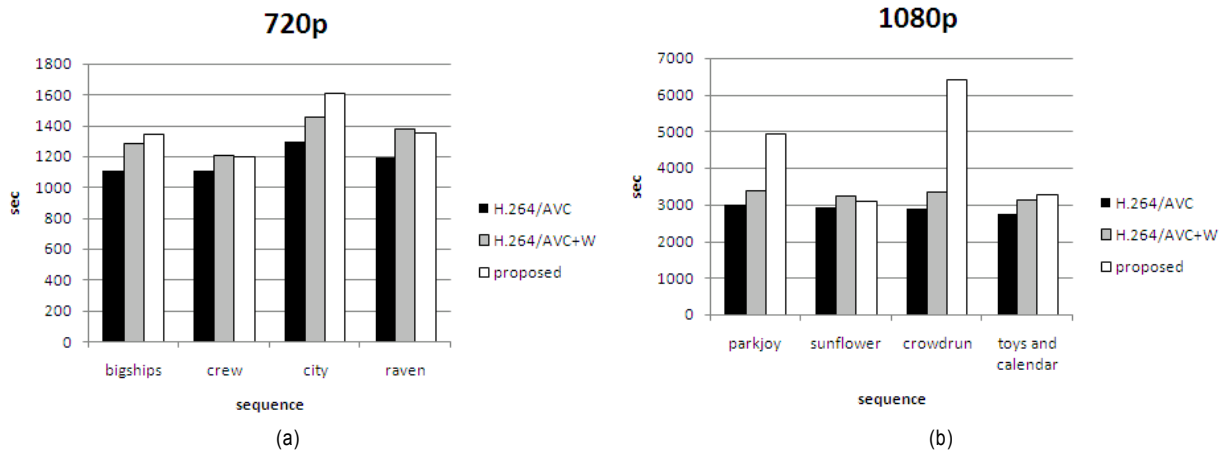


그림 7. H.264/AVC와 제안하는 방법의 부호화 시간 비교 : (a) 720p 영상 (b) 1080p 영상

Fig. 7. Encoding time comparison of H.264/AVC and proposed method : (a) 720p sequences (b) 1080p sequences

표 3. 블록 파티션 형태에 따라 수행되는 올-왜곡 최적화 횟수

Table 3. The number of RDO to the block partition

파티션 형태	16×16,	16×8	8×16
최적화 수행횟수	9	81	81

최적화가 추가로 수행되기 때문에 기존에 방법과 비교해서, 그림과 같이 복잡도가 증가함을 알 수 있다. 하지만, 복호기에서는 움직임 보상 방법에 대한 시그널링 복호화 및 인접 화소들의 평균만을 구하여 사용하므로 복호기의 복잡도는 거의 증가하지 않는다는 것을 추측할 수 있다.

3. 선택된 방법에 대한 통계 분석

그림 8은 선택된 움직임 보상 방법의 통계를 보여준다. 범례의 "00×00" 은 기존의 방법이 수행된 횟수이고 "00×00_p"는 제안하는 방법이 수행된 횟수이다. 하위 매크로 블록의 크기가 큰 경우에 제안하는 방법의 성능이 효과적인 반면에, 하위 매크로 블록의 크기가 작은 경우에는 제안하는 방법이 비교적 낮은 비율로 선택되었다. 또한 전반적으로 기존의 방법이 선택되는 비율이 높음을 알 수 있다.

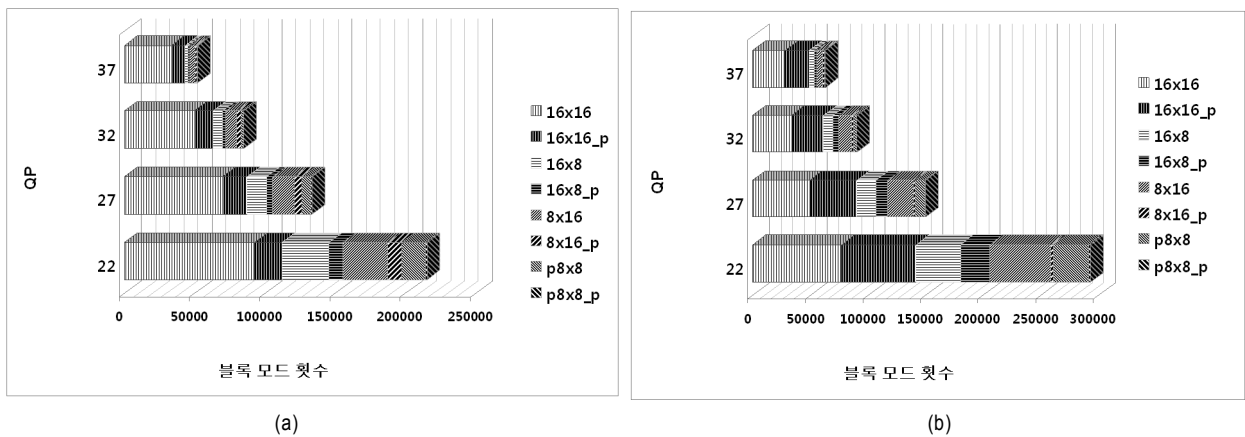


그림 8. 선택된 움직임 보상 방법의 통계 : (a) bigships (b) toys and calendar

Fig. 8. Statistic of selected motion compensation method : (a) bigships (b) toys and calendar

V. 결 론

본 논문에서 동영상 부호화 시에 밝기 변화로 발생할 수 있는 문제점에 대해 고찰하였고, 그 문제점을 해결하기 위한 매크로블록 단위의 개선된 보상 방법을 제안하였다. 인접화소를 이용한 움직임 보상을 사용함으로써 지역적인 밝기 변화 발생 시 기존의 움직임 보상 방식보다 효과적으로 예측 블록을 생성할 수 있었으며, 매크로블록 단위로 밝기 변화를 보상함으로써 보다 적응적으로 부호화를 수행할 수 있었다.

앞으로 시그널링에 대한 정보를 효율적으로 전송할 수 있는 방법에 대한 연구와 인접 화소를 더욱 효율적으로 적용하는 방법에 대한 연구를 진행할 수 있을 것이다.

제안하는 방법은 시간적인 예측 부호화 수행 시 영상의 밝기 변화가 많이 발생하고, 하위 매크로 블록이 큰 경우에 높은 부호화 효율을 보였으며, 부호화 시에 소요되는 시간은 비교적 많아졌지만, 복호화 시에는 기존과 거의 유사한 시간에 복호화가 가능하다는 것을 예상할 수 있다. 제안하는 방법은 동영상 부호화에서 보다 높은 압축 효율을 얻기 위해서 반드시 필요한 기술이며, 앞으로 동영상 부호화에서 유용하게 활용될 것으로 전망된다.

참 고 문 헌

- [1] ITU-T Recommendation H.264 and ISO/IEC 14496-10 "Advanced video coding for generic audiovisual services", May 2003.
- [2] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard", IEEE Trans. Circuits Sys. Video Technol., pp. 560-576, July 2003.
- [3] Thomas Wedi, Hans Georg Musmann "Motion-and Aliasing-Compensated Prediction for Hybrid Video Coding", IEEE Trans. Circuits Syst. Video Technol., pp.577-586, July 2003.
- [4] G. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," IEEE Signal Processing Magazine, Vol.15, No.6, pp.74-90, Nov.1998.
- [5] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas," IEEE International Conference on Image Processing 2005, Genova, Italy, Sept.11-14, 2005.
- [6] J. M. Boyce, "Weighted Prediction in the H.264/MPEG AVC Video Coding Standard," Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium, vol. 3, pp. III-789-92, May 2004.
- [7] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose, "Global Brightness-Variation Compensation for Video Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 988-1000, Dec. 1998.
- [8] Jae-Ho Hur, Sukhee Cho, and Yung-Lyul Lee, "Adaptive Local Illumination Change Compensation Method for H.264/AVC Based Multiview Video Coding", IEEE Trans. on CSVT, Vol. 17, No 11, pp. 1496-1505, Nov. 2007.
- [9] H.264/AVC Reference Software Version 12.4, <http://iphome.hhi.de/~suehring/tml/download/>
- [10] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves", ITU-T SG16 Q.6 VCEG, Doc. VCEG-M33, March 2001

저 자 소 개



서 정 훈

- 2008년 2월 : 세종대학교 인터넷학과 (학사)
- 2010년 2월 : 세종대학교 컴퓨터공학과 (석사)
- 현재 : (주)필링크 사원
- 주관심분야 : 영상압축, H.264/AVC

저 자 소 개



김 정 필

- 2010년 2월 : 세종대학교 컴퓨터공학과 (학사)
- 현재 : 세종대학교 컴퓨터공학과 (석사과정)
- 주관심분야 : 영상압축, H.264/AVC



이 영 렬

- 1985년 2월 : 서강대학교 전자공학과 (학사)
- 1987년 2월 : 서강대학교 전자공학과 (석사)
- 1999년 2월 : 한국과학기술원 전기 및 전자공학과 (박사)
- 1987년 1월 ~ 1994년 2월 : 삼성전자 중앙연구소 DMS Lab. 선임연구원
- 1999년 3월 ~ 2001년 8월 : 삼성전자 중앙연구소 DMS Lab. 수석연구원
- 2001년 9월 ~ 현재 : 세종대학교 컴퓨터공학부 컴퓨터공학과 부교수
- 주관심분야 : 영상처리(압축, 복원), 영상전송, 멀티미디어 시스템, 3차원 비디오코딩, 트랜스코딩, Scalable 비디오 코딩