

고화질 비디오를 위한 효율적인 인트라 부호화 방법과 정수 이산 코사인 변환

허진, 호요성
광주과학기술원 정보통신공학과

I. 서론

시청자들에게 큰 영상과 사실감 있는 영상을 제공하는 고화질 비디오 부호화는 표준 화질 비디오 부호화에 비해 처리해야 할 데이터의 양이 많다. 기존의 H.264는 고화질 비디오의 해상도에 비해 상대적으로 작은 크기의 매크로블록을 사용하여 부호화하기 때문에 고화질 비디오를 부호화하는데 효율적이지 못하다. 따라서 고화질 비디오 부호화에 적합한 효율적인 부호화 기술이 요구된다. 본 논문에서는 효율적인 고화질 부호화를 위해 새로운 인트라 부호화 방법과 큰 크기의 정수 이산 코사인 변환을 제안한다.

II. 제안한 방법

기존의 인트라 예측 방법은 16×16 매크로블록 단위로 인트라 4×4 예측과 인트라 16×16 예측을 사용하여 부호화를 실행하였다 [1]. 그러나 기존의 방법은 고화질 비디오를 효율적으로 부호화하는데 어려움이 있다. 따라서 본 장에서는 기존의 매크로블록보다 큰 크기의 매크로블록을 제안한다. 이에 따라 새로운 인트라 예측 방법과 새로운 정수 이산 코사인 변환을 사용한다.

1. 32×32 매크로블록

현재 비디오 부호화 표준에서는 화면 내 예측을 위해 인트라 4×4 예측과 인트라 16×16 예측, 두 가지 인트라 예측을 사용한다. 기존 매크로블록의 크기는 고화질 비디오의 해상도에 비해 상대적으로 작기 때문에 고화질 비디오 부호화를 위해 적합하지 않으므로 매크로블록의 크기를 32×32로 확장한다. 이에 따라 기존의 인트라 16×16 예측과 인트라 4×4 예측은 각각 인트라 32×32 예측과 인트라 8×8 예측으로 확장한다.

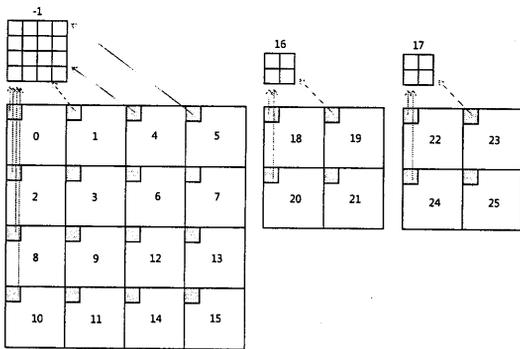


그림 1. 32×32 매크로블록의 내부 블록 스캔 순서

기존의 인트라 예측 방법은 4×4 블록을 단위로 인

트라 4×4 예측과 인트라 16×16 예측을 수행한다. 그러나 제안한 32×32 매크로블록에서는 8×8 블록 단위로 인트라 8×8 예측과 인트라 32×32 예측을 수행한다. 확장된 매크로블록을 구성하는 16개의 8×8 블록은 그림 1의 블록 스캔 순서에 따라 인트라 4×4 예측 부호화와 동일하게 부호화를 수행한다.

인트라 32×32 예측모드는 인트라 16×16 예측모드로부터 확장된 새로운 인트라 예측모드이다. 인트라 32×32 예측은 인트라 16×16 예측모드와 동일하게 4가지의 예측모드 중에서 가장 적절한 하나의 모드를 선택하여 부호화를 실행한다.

2. 8×8 정수 이산 코사인 변환

기본 블록의 크기가 4×4인 이전의 부호화 방법에서는 4×4 정수 이산 코사인 변환을 사용하였다. 하지만 기본 블록의 크기가 8×8로 확장됨에 따라 8×8 크기의 정수 이산 코사인 변환이 필요하다. 8×8 정수 이산 코사인 변환은 확장된 매크로블록 내 잔여 데이터들의 변환에 사용된다.

III 실험 결과 및 결론

표 1의 실험 결과로부터, 제안한 방법이 기존의 방법과 비교하여 고화질 비디오를 효율적으로 부호화함을 알 수 있다. 또한 제안한 방법이 저비트율에서 더 효율적임을 알 수 있다.

표 1. 실험결과 (Sunflower)

QP	PSNR (dB)		Bit rate (kbit/s)	
	H.264	제안한 방법	H.264	제안한 방법
28	40.65	40.95	10482.56	10189.76
32	38.46	38.87	7176.72	6801.62
36	36.24	36.73	4739.39	4500.45
40	34.01	34.42	3205.91	3050.41

본 논문에서는 효율적인 고화질 비디오 부호화를 위해 새로운 인트라 부호화 방법과 큰 크기의 정수 이산 코사인 변환을 제안하였다. 기존의 인트라 예측 모드의 크기를 확장하였고, 4×4 정수 이산 코사인 변환 역시 8×8 정수 이산 코사인 변환으로 확장하였다. 제안한 방법은 기존의 방법과 비교하여, PSNR의 경우 평균적으로 약 0.49dB 정도 화질을 개선하였고, 8.93% 정도 비트를 감소시켰다.

참고문헌

- [1] G.J. Sullivan and T. Wiegand, "Video Compression - From Concepts to the H.264/AVC Standard," *Proc. of the IEEE*, vol. 93, pp. 18-31, January, 2005.

고화질 비디오를 위한 효율적인 인트라 부호화 방법과 정수 이산 코사인 변환

허진, 호요성

광주과학기술원 정보통신공학과

Efficient Intra Coding Scheme and Integer Discrete Cosine Transform for High Definition Video

Jin Heo, Yo-Sung Ho

Gwangju Institute of Science and Technology

jinheo@gist.ac.kr, hoyo@gist.ac.kr

Abstract: 고화질 비디오(high definition video)는 기존의 표준 화질 비디오(standard definition video)보다 큰 크기의 영상과 더 나은 화질을 제공한다. 그러므로 고화질 비디오는 처리해야 할 데이터의 양이 표준 화질 비디오에 비해 많다. 기존의 H.264/AVC 는 고화질 비디오의 해상도에 비해 너무 작은 크기의 매크로블록을 사용하여 부호화하기 때문에 고화질 비디오를 부호화하는데 효율적이지 못하다. 따라서 고화질 비디오 부호화에 적합한 효율적인 부호화 기술이 요구된다. 본 논문에서는 고화질 비디오의 특성을 고려하여 기존의 매크로블록(16×16)보다 더 큰 매크로블록을 제안한다. 또한 기존의 4×4 정수 이산 코사인 변환(integer discrete cosine transform)보다 더 큰 크기의 정수 이산 코사인 변환을 제안한다. 제안하는 방법은 인트라 부호화 방법에서 큰 크기의 매크로블록을 고려하여 새로운 인트라 예측모드를 사용한다. 또한 확장된 블록의 크기를 고려하여 8×8 정수 이산 코사인 변환을 이용한다. 제안한 방법은 기존의 방법보다 약 0.49dB 정도 부호화 효율을 향상시켰다.

Keywords: 고화질 비디오, H.264/AVC, 인트라 부호화, 매크로블록

1. 서론

고화질(high definition)이란 ‘표준 화질(standard definition) 보다 나은 것’이라 정의한다 [1]. 고화질 비디오는 표준 화질 비디오에 비해 더 큰 영상과 더 나은 화질을 제공하므로 시청자에게 표준 화질 비디오보다 더 실제와 같은 사실감을 줄 수 있다.

최근 대형 스크린과 높은 해상도를 가진 디스플레이 장치가 개발되면서 고화질 비디오에 대한 수요가 급격히 증가하고 있다. 이러한 고화질 비디오는 비디오 통신, 산업 자동화, 엔터테인먼트뿐만 아니라 거의 모든 분야에서 주목받고 있다. 그러나 고화질 비디오는 큰 크기의 영상과 더 나은 화질을 제공하므로 처리해야 할 데이터의 양 또한 표준 화질 비디오에 비해 많다. 따라서 기존의 표준 화질 비디오 부호화 기술보다 효율적인 고화질 비디오 부호화 기술이 요구된다.

H.264는 기존의 비디오 부호화 표준들에 비해 높

은 부호화 효율을 제공한다. 이와 같은 높은 부호화 효율은 이전의 표준들과 다르게 많은 새로운 기능들을 포함하여 이들 기능을 결합한 하이브리드(hybrid) 비디오 부호화를 수행하기 때문이다. 그러나 기존의 H.264는 QCIF(176×144), CIF(352×288)와 같은 작은 해상도의 비디오 부호화에 최적화되어 있다는 문제점이 있다. 왜냐하면 매크로블록의 크기가 고화질 비디오의 해상도에 비해 상대적으로 너무 작기 때문이다 [2].

본 논문에서는 이러한 문제점을 해결하고자 기존의 매크로블록보다 큰 크기의 매크로블록을 제안한다. 제안한 큰 크기의 매크로블록 기반 부호화에서는 부호화를 수행하는 기본 블록의 크기 또한 고화질 부호화에 적합하게 확장된다. 기본 블록의 크기가 바뀌었기 때문에 기본 블록 단위로 수행되는 정수 이산 코사인 변환 역시 바뀌어야 한다. 따라서 고화질 비디오 부호화에 적합한 큰 크기의 정수 이산 코사인 변환을 제안한다.

II. H.264 인트라 부호화 방법과 정수 이산 코사인 변환

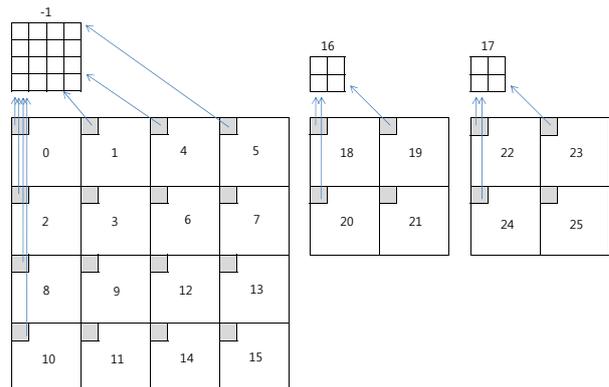


그림 1. 매크로블록의 내부 블록 스캔 순서

H.264는 블록간의 화소상관도를 사용하여 화면 내 예측 부호화를 수행한다. 이미 부호화된 화소의 정보를 이용하여 블록마다 적절한 예측방향을 선택해서 부호화하기 때문에 부호화 효율을 향상시킬 수 있다. 휘도 신호의 경우 4×4 블록과 16×16 매크로블록을 사용하여 화면 내 예측을 수행한다. 4×4 인트라 예측의 경우 총

9가지 예측모드가 있고, 16×16 인트라 예측의 경우 총 4가지 예측모드가 있다 [3]. 색차 신호의 경우 8×8 블록을 4가지 예측모드를 사용하여 부호화한다. 부호화하는 현재 블록과 각 예측모드를 통해 얻은 예측 블록 값의 차가 가장 작은 모드를 현재 블록에 대한 최적의 모드로 선택하여 부호화를 실행한다.

4×4 인트라 예측모드는 한 매크로블록을 구성하는 16개의 4×4 블록이 그림 1과 같이 0~15의 4×4 블록 순서에 따라 부호화를 수행한다. 9가지의 4×4 인트라 예측모드는 이미 부호화된 주변 화소 값들을 이용하여 4×4 블록마다 적절한 모드를 선택하고, 선택된 모드를 사용하여 4×4 블록 단위로 부호화한다. 그림 2는 9가지 4×4 인트라 예측모드를 나타낸다. 그림 2에서 A부터 M까지의 알파벳은 각 한 화소를 나타낸다.

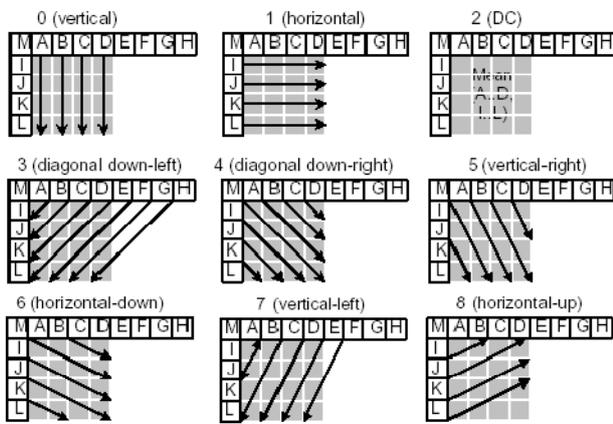


그림 2. 9가지 4×4 인트라 예측모드들 [4]

16×16 인트라 예측모드는 상단 매크로블록의 가장 아래에 있는 16개의 화소와 좌측 매크로블록의 가장 오른쪽에 있는 16개의 화소를 이용하여 평균 예측, 수평 예측, 수직 예측, 평면 예측 총 4가지의 모드 중에서 적절한 하나의 모드를 선택하여 부호화를 실행한다. 16×16 인트라 예측모드 중 평면 예측모드의 경우는 4×4 인트라 예측모드에서는 사용되지 않는 새로운 모드이다. 평면 예측모드는 상단 매크로블록 화소와 좌측 매크로블록 화소를 대각선 방향으로 보간(interpolation)하여 현재 매크로블록 값을 예측한다.

H.264는 인트라 예측을 통해 얻어진 잔여 데이터를 4×4 블록 단위로 변환 후, 양자화한다. 이때, H.264는 4×4 정수 이산 코사인 변환을 사용한다. 한 매크로블록의 데이터는 그림 1의 순서에 따라 변환된다.

H.264는 부호화할 잔여 데이터의 종류에 따라 세 가지의 서로 다른 변환을 사용한다. 그림 1에서 보듯이 매크로블록이 16×16 인트라 예측모드로 부호화되는 경우 16×16 휘도 블록의 각 4×4 블록 DC 계수들은 4×4 블록(-1)을 구성하여 4×4 Hadamard 변환을 한다. 8×8 색차 블록의 각 4×4 블록 DC 계수들은 2×2 블록(16, 17)을 형성하여 2×2 Hadamard 변환된다. 이 두 경우를 제외한 모든 다른 잔여 데이터들은 4×4 정수 이산 코사인 변환을 사용하여 변환된다.

이와 같은 매크로블록 기반의 인트라 예측 방법과 정수 이산 코사인 변환은 고화질 비디오와 같은 큰 크

기의 영상을 부호화하는데 효율적이지 못하다. 16×16 매크로블록 크기가 고화질 비디오의 크기에 비해 상대적으로 너무 작기 때문이다. 표 1은 QCIF(176×144) 비디오 영상, CIF(352×288) 비디오 영상, HD급(1280×720) 비디오 영상을 인트라 부호화 환경에서 기존의 H.264를 사용하여 부호화한 결과이다.

표 1. 영상 크기에 따른 각 인트라 모드 발생 빈도

영상	QP	각각의 모드 발생 빈도 (%)	
		인트라 4×4	인트라 16×16
FOREMAN (QCIF)	28	93.75	6.25
	32	89.74	10.26
	36	84.92	15.08
	40	77.27	22.73
FOREMAN (CIF)	28	81.12	18.88
	32	72.40	27.60
	36	62.10	37.90
	40	50.13	49.87
RUSH_HOUR (HD)	28	50.56	49.44
	32	37.78	62.22
	36	24.29	75.71
	40	13.88	86.12

표 1에서 보듯이 QCIF와 같이 영상의 크기가 작은 경우 4×4 인트라 모드가 16×16 인트라 모드보다 더 많이 발생한다. 그러나 반대로 영상의 크기가 커질수록 16×16 인트라 모드의 발생 빈도가 증가함을 알 수 있다. 이와 같은 실험결과는 H.264가 QCIF와 CIF 같은 저해상도 비디오 영상을 부호화하는데 적합하다는 것을 나타낸다. 따라서 고화질 비디오를 효율적으로 부호화하기 위해서는 기존의 인트라 모드들보다 큰 새로운 인트라 모드들이 필요하다는 것을 알 수 있다. 이에 본 논문에서는 매크로블록을 32×32 크기로 확장하고 이에 적합한 확장된 인트라 예측모드와 정수 이산 코사인 변환을 사용한다.

III. 제안한 방법

기존의 인트라 예측 방법은 16×16 매크로블록 단위로 부호화를 실행하였다. 그러나 이러한 방법은 2장에서 언급한 것처럼 고화질 비디오를 효율적으로 부호화하는데 어려움이 있다. 따라서 본 장에서는 기존의 매크로블록보다 큰 크기의 32×32 매크로블록을 제안한다. 제안한 매크로블록을 기반으로 확장된 새로운 인트라 모드를 제안한다. 또한 확장된 블록의 크기를 고려하여 기존의 4×4 정수 이산 코사인 변환보다 큰 8×8 정수 이산 코사인 변환을 사용한다.

1. 32×32 매크로블록

현재 비디오 부호화 표준에서는 인트라 예측을 위해 인트라 4×4 예측과 인트라 16×16 예측, 두 가지 인트라 예측을 사용한다. 인트라 4×4 예측은 4×4 블록 단위로, 인트라 16×16 예측은 매크로블록 단위로 부호화를 수행한다. 앞장에서 살펴본 바와 같이 기존의 16×16 매크로블록의 크기는 상대적으로 큰 고화질 비디오 부호화를 위해 적합하지 않으므로 매크로블록의 크기를 32×32로 확장한다. 이에 따라 기존의 16×16 인트라

예측과 4×4 인트라 예측은 각각 32×32 인트라 예측과 8×8 인트라 예측으로 확장한다.

기존의 인트라 예측 방법은 4×4 블록을 단위로 인트라 4×4 예측과 인트라 16×16 예측을 수행한다. 그러나 제안한 32×32 매크로블록에서는 이전의 4×4 기본 블록의 크기를 8×8로 확장한다. 따라서 확장된 8×8 인트라 예측모드는 확장된 32×32 매크로블록 내에 총 16개의 블록이 존재한다. 또한 확장된 8×8 인트라 예측모드는 기존의 4×4 인트라 예측모드와 동일한 방향의 9가지의 예측 모드를 사용하여 부호화를 수행한다. 확장된 매크로블록을 구성하는 16개의 8×8 블록은 그림 1의 4×4 인트라 예측 부호화와 동일하게 0~15의 블록 순서에 따라 부호화를 수행한다.

32×32 인트라 예측모드는 16×16 인트라 예측모드로부터 확장된 새로운 인트라 예측모드이다. 32×32 인트라 예측모드는 상단 매크로블록의 가장 아래 있는 32개의 화소와 좌측 매크로블록의 가장 오른쪽에 있는 32개의 화소를 이용하여 16×16 인트라 예측모드와 동일하게 평균 예측, 수평 예측, 수직 예측, 평면 예측 총 4가지의 예측모드 중에서 가장 적절한 하나의 모드를 선택하여 부호화를 실행한다.

인트라 예측모드에서 기본 블록의 크기가 8×8 블록으로 확장됨에 따라 색차 블록 역시 기존의 8×8 블록에서 16×16 블록으로 확장된다. 한 매크로블록을 구성하는 두 색차 정보는 그림 1과 같이 18~25의 8×8 블록 순서에 따라 부호화를 수행한다.

2. 8×8 정수 이산 코사인 변환

기본 블록의 크기가 4×4인 이전의 부호화 방법에서는 4×4 정수 이산 코사인 변환을 사용하였다. 하지만 기본 블록의 크기가 8×8로 확장됨에 따라 8×8 크기의 정수 이산 코사인 변환이 필요하다.

매크로블록이 32×32 인트라 예측모드로 부호화되는 경우 32×32 휘도 블록 내의 각 8×8 블록 DC 계수들은 이전의 16×16 인트라 예측모드와 동일하게 4×4 블록을 구성하여 4×4 Hadamard 변환을 수행한다. 두 16×16 색차 블록의 각 8×8 블록 DC 계수들 역시 이전 색차 블록 부호화와 동일하게 2×2 블록을 구성하여 2×2 Hadamard 변환을 수행한다. 이 두 경우를 제외한 모든 다른 잔여 데이터들은 8×8 정수 이산 코사인 변환을 사용하여 변환된다. H.264에서는 다음과 같은 8×8 정수 이산 코사인 변환 계수를 사용한다.

$$T_{8 \times 8} = \frac{1}{8} \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}$$

VI. 실험 결과

본 논문은 H.264 표준 소프트웨어 JM 11.0 [5]을 이용하여 실험하였다. 제안한 알고리즘의 효율성을 보이고자, 4개의 고화질(1280×720) 비디오 영상들(Sunflower, Rush_hour, Stockholm, Pedestrian_area)을 이용하여 인트라 환경에서만 부호화하였다. 실험 환경은 표 2에

나타내었다.

표 2. 실험 환경

	JM11.0	제안한 방법
인트라 예측 모드	•인트라 4×4 •인트라 16×16	•인트라 8×8 •인트라 32×32
정수 이산 코사인 변환	4×4 정수 이산 코사인 변환	8×8 정수 이산 코사인 변환
하다마드 변환	사용	사용
부호화한 프레임 개수	100	100
부호화 구조	III...	III...
양자화 파라미터	28, 32, 36, 40	28, 32, 36, 40

제안한 방법과 H.264 참조 소프트웨어 방법의 성능 비교를 위해 PSNR과 비트율 [6]을 사용했다. 표 3, 표 4, 표 5, 표 6은 성능 비교를 위해 실험한 결과를 나타낸다.

표 3. 실험결과 (Sunflower)

QP	PSNR (dB)		비트율 (kbit/s)	
	H.264	제안한 방법	H.264	제안한 방법
28	40.65	40.95	10482.56	10189.76
32	38.46	38.87	7176.72	6801.62
36	36.24	36.73	4739.39	4500.45
40	34.01	34.42	3205.91	3050.41

표 4. 실험결과 (Rush_hour)

QP	PSNR (dB)		비트율 (kbit/s)	
	H.264	제안한 방법	H.264	제안한 방법
28	40.36	40.58	5939.55	5834.80
32	38.59	38.93	4008.56	3790.97
36	36.68	37.16	2735.92	2548.46
40	34.71	35.15	1904.84	1760.19

표 5. 실험결과 (Stockholm)

QP	PSNR (dB)		비트율 (kbit/s)	
	H.264	제안한 방법	H.264	제안한 방법
28	36.17	36.17	35191.1	36283.48
32	34.00	33.98	21439.21	21836.15
36	31.97	32.05	13445.33	13518.41
40	29.92	30.06	8593.7	8176.51

표 6. 실험결과 (Pedestrian_area)

QP	PSNR (dB)		비트율 (kbit/s)	
	H.264	제안한 방법	H.264	제안한 방법
28	39.16	39.21	8209.13	8423.72
32	37.14	37.26	5338.43	5219.11
36	35.11	35.35	3508.04	3332.47
40	33.09	33.32	2315.77	2148.80

실험 결과로부터, 제안한 방법이 기존의 H.264 방법과 비교하여 고화질 비디오를 효율적으로 부호화함을 알 수 있다. 각 표를 통해 제안한 방법이 저비트율에서 더 효율적임을 알 수 있다. 또한 Stockholm 영상의 경우 물체의 텍스처 정보들이 너무 세밀하여 매크로블록의 확장으로 큰 이득을 얻을 수 없었다. 제안한 방법을 사용하여 평균적으로 약 8.93% 비트를 감소시켰다.

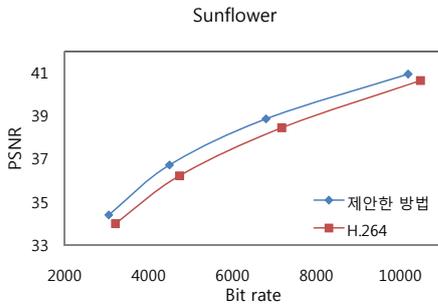


그림 3. 비트율-왜곡 곡선 (Sunflower)

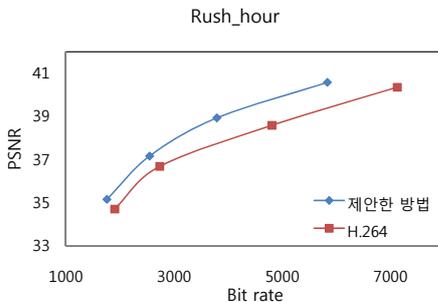


그림 4. 비트율-왜곡 곡선 (Rush_hour)

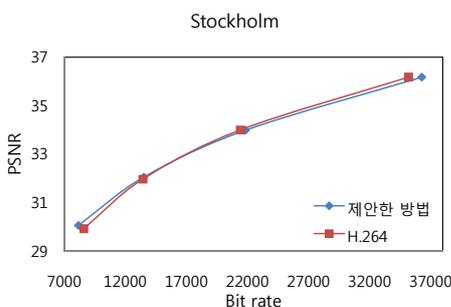


그림 5. 비트율-왜곡 곡선 (Stockholm)

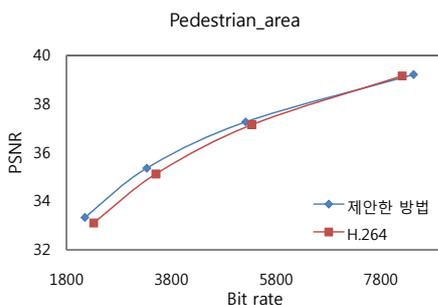


그림 6. 비트율-왜곡 곡선 (Pedestrian_area)

그림 3, 그림 4, 그림 5, 그림 6은 각각 Sunflower, Rush_hour, Stockholm, Pedestrian_area 영상에 대한 비트율-왜곡 곡선을 나타낸다. 이 그림에서 보는 바와 같이 본 논문에서 제안한 매크로블록의 확장과 8×8 정수 이산 코사인 변환 방법이 기존의 방법에 비해 고화질 비디오를 효율적으로 부호화함을 알 수 있다.

Ⅶ. 결론

본 논문에서는 H.264를 사용하여 고화질 비디오를 부호화하는 경우 기존의 매크로블록의 크기가 너무 작아 고화질 비디오를 효율적으로 부호화하지 못하는 점을 고려하여 매크로블록의 크기를 확장하였다. 이에 따라, 인트라 예측모드와 정수 이산 코사인 변환의 크기를 확장하는 방법을 제안하였다. 기존의 16×16 매크로블록의 크기를 32×32로 확장하여, 기존의 4×4 인트라 모드와 16×16 인트라 모드를 각각 8×8 인트라 모드와 32×32 인트라 모드로 확장하였다. 또한 기본 블록의 크기를 4×4 블록에서 8×8 블록으로 확장함에 따라 기존의 4×4 정수 이산 코사인 변환 역시 8×8 정수 이산 코사인 변환으로 확장하였다. 제안한 방법은 H.264의 방법과 비교하여, PSNR의 경우 평균적으로 약 0.49dB 정도 화질을 개선하였고, 약 8.93% 정도 비트를 감소시켰다.

감사의 글

본 연구는 광주과학기술원(GIST) 실감방송연구센터(RBRC)를 통한 정보통신부대학 IT연구센터(ITRC)와 교육인적자원부 두뇌한국21(BK21) 정보기술사업의 지원에 의한 것임.

참고문헌

- [1] "Understanding and Using High-Definition Video," White paper, Adobe
- [2] S. Ma, and C.-C. J. Kuo, "High-definition Video Coding with Super-macroblocks," *Proc. of SPIEIS&T Electronic Imaging*, SPIE vol, 6508, 2007.
- [3] G.J. Sullivan and T. Wiegand, "Video Compression-From Concepts to the H.264/AVC Standard," *Proc. of the IEEE*, vol.93, pp. 18-31, January, 2005
- [4] D. Storey and P. Nasiopoulos, "An Efficient Statistical Intra-mode Prediction Method for H.264," 2006 IEEE International Symposium on Signal Processing and Information Technology
- [5] JVT Reference Software JM 11.0, available online at:http://iphome.hhi.de/suehring/tml/download/old_jm
- [6] G. Bjontegaard, "Calculation of Average PSNR Difference between RD-Curve," *ITU-T Q.6/16*, Doc. VCEG-M33, April 2001.