

정수 DCT를 이용한 H.263 부호기에 관한 연구

김 용 옥*, 허 도 근*
원광대학교 전자공학과

A Study on the H.263 Encoder using Integer DCT

Kim Yong-Wook*, Huh Do-Geun*
Dept. of Electronic Eng. Wonkwang Univ.

Abstract

This paper is studied the high speed processing moving picture encodec to compress and encode a moving picture by real time. This is used the new motion vector search algorithm with smallest search point in H.263 encodec, and is applied the integer DCT for the encodec by converting a moving picture.

The integer DCT behaves DCT by the addition operation of the integer using WHT and a integer lifting than conventional DCT that needs the multiplication operation of a floating point number. Therefore, the integer DCT can reduce the operation amount than basis DCT with having an equal PSNR because the multiplication operation of a floating point number does not need.

I. 서 론

대표적인 변환 부호화 방식인 DCT는 연산이 간단하고 변환 비용이 적게들며 KLT와 근접한 성능을 보이므로 DCT는 JPEG과 같은 정지 영상 압축[1], MPEG1, 2와 같은 높은 비트율의 영상압축, H.263이나 MPEG-4와 같은 낮은 비트율의 영상압축에 대부분 사용된다[2].

하지만 DCT 커널은 부동소수점 수이므로 변환 과정에서 부동소수점 수의 곱셈으로 인한 연산량 증가로 부호화 비용과 연산 시간이 증가한다. 연산 시간의 증가는 DCT를 사용하는 동영상 부호기에서 동영상 정보의 실시간 처리를 어렵게 한다. 따라서 부동소수점 수의 연산을 줄이거나 제거할 수 있는 DCT에 대한 새로운 알고리즘이 필요하다.

상기의 이유로 본 논문에서는 동영상을 실시간으로 압축하여 부호화하기 위하여 H.263 부호기에 기존의 부동소수점 DCT대신 정수형 DCT를 적용시켜 고속처리 동영상 부호기를 연구한다.

정수형 DCT는 정수만을 연산의 대상으로 하여 DCT 변환을 수행하는 방식으로 WHT와 lifting scheme에 의해 이루어진다[15]. 부동소수점 수의 곱셈 연산을 필요로 하는 기존 DCT와 달리 정수형 DCT는 덧셈 연산과 shift 연산만을 이용하므로 연산 시간과 연산량을 기존 DCT보다 줄일 수 있다.

본 논문에서는 먼저 기존 DCT의 연산량을 줄이기 위해 사용한 정수형 DCT의 알고리즘을 서술한다. 둘째 정수형 DCT를 H.263 부호기의 동작을 설명하고 마지막으로 기존 DCT와 정수형 DCT의 성능, H.263에 기존 DCT와 정수형 DCT를 적용시킨 동영상 부호기의 성능과 기존 DCT와 정수 DCT의 호환성을 모의 실험을 통하여 비교, 평가한다.

정수형 DCT는 정수의 덧셈 연산만으로 변환을 수행하면서도 기존 직교 변환 알고리즘과 대등한 성능을 보이며 정수형 DCT를 사용한 H.263 부호기는 기존 DCT를 사용한 H.263 부호기와 대등한 성능을 보인다. 또한 정수형 DCT와 기존 DCT는 사용된 시스템내에서 서로 호환된다.

2. 정수형 DCT

본 논문에서는 직교 변환의 특성과 정수간 변환을 보장하며, 기존 DCT 표준들과 호환될 수 있는 정수형 DCT를 사용한다. 1차원 정수형 DCT 커널을 만드는 기능도는 그림 1과 같다.

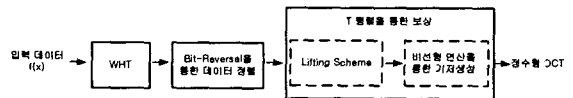


그림 1. 정수형DCT알고리즘의 블록도

그림 1에서 입력 데이터 $f(x)$ 를 먼저 WHT 하여 Bit-Reversal을 통하여 홀수 번째 데이터와 짝수 번째 데이터로 정렬한다. 기존 DCT 커널과의 보상을 위하여 T 행렬을 곱한 후 이때 발생하는 부동소수점 수의 기저를 ± 1 또는 0으로 만들기 위하여 비선형 연산을 행하여 정수형 DCT를 수행한다.

1) WHT를 통한 DCT의 분해(decomposition)

(1) WHT 행렬

WHT는 알쉬 변환과 하다마드 변환에 기초하므로 WHT 커널 행렬 $H_w(x, u)$ 는 $H_w^{-1} = H_w^T$ 를 만족시키며 행렬의 기저로 ± 1 만을 갖는다. 따라서 WHT는 덧셈 연산만으로 직교 변환을 수행한다. 본 논문에서는 식 (1)의 8-point WHT 커널 $H_{w8}(x, u)$ 을 사용한다[3].

$$H_{W_8}(x,u) = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (1)$$

식 (1)의 WHT 커널을 이용한 WHT는 식 (2)와 같다.

$$F = H_{W_8} f_{BRO} \quad (2)$$

여기서 f_{BRO} 는 입력 $f = [f(0) f(1) f(2) f(3) f(4) f(5) f(6) f(7)]^T$ 를 식 (3)의 bit-reversal 행렬 B 를 이용하여 재정렬 시킨 데이터로 $f_{BRO} = [f(0) f(4) f(2) f(6) f(1) f(5) f(3) f(7)]^T$ 이다.

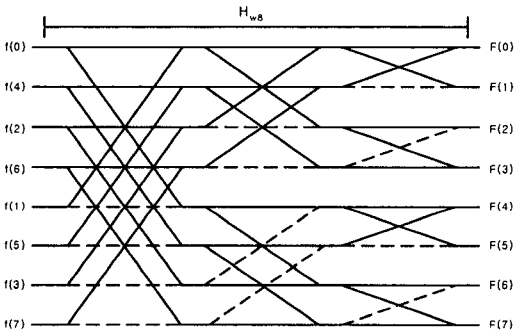


그림 2. WHT의 lattice structure

그림 2는 식 (2)의 WHT 과정을 나타낸다. 그림 2에서 실선은 식 (1)의 기저 중 +1을 나타내고 점선은 -1을 나타내며 24번의 덧셈연산으로 8-point WHT를 수행한다[4][5].

(2) Bit-Reversal 행렬

bit-reversal 행렬은 순차적으로 입력되는 데이터를 홀수 번째 데이터와 짝수 번째 데이터로 구분하여 그림 2에서 보인 입력 데이터의 순서로 재정렬한다. 8-point 입력에 대한 bit-reversal 행렬 B 는 식 (3)과 같다.

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

여기서 $B = B^{-1} = B^T$ 이고 $B^2 = I$ 이다.

(3) WHT를 통한 DCT

입력 f 의 DCT는 식 (4)와 같은 행렬식으로 표현할 수 있다.

$$F = A f \quad (4)$$

여기서 A 는 DCT 커널이다.

H_w 를 이용하여 입력 데이터 f 에 대한 DCT를 표현하면 식 (5)와 같다.

$$F_{BRO} = B F = T B H_w f_{BRO} \quad (5)$$

식 (5)에서 F_{BRO} 는 WHT가 bit-reversal을 통해 재정렬된 입력 데이터를 사용하므로 기존 DCT된 결과 F 의 데이터 순서를 이와 일치시키기 위하여 사용한다. 또한 행렬 T 는 기존 DCT의 순방향 커널과 식 (5)의 WHT를 이용한 정수형 DCT 커널의 계수값을 일치시키기 위한 행렬로써 $T = B(A H_w^T) B$ 이다. 여기서 A 는 기존 DCT의 커널이다. 행렬 T 의 사용으로 인해 정수형 DCT가 기존 DCT와 동일한 동작이 가능해진다. $N=8$ 에 대한 T 의 행렬식은 식 (6)과 같다.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9239 & 0.3827 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.3827 & 0.9239 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9061 & -0.0747 & 0.3753 & 0.1802 \\ 0 & 0 & 0 & 0 & 0.2126 & 0.7682 & -0.5133 & 0.3182 \\ 0 & 0 & 0 & 0 & -0.3182 & 0.5133 & 0.7682 & 0.2126 \\ 0 & 0 & 0 & 0 & -0.1802 & -0.3753 & -0.0747 & 0.9061 \end{bmatrix} \quad (6)$$

식 (6)으로부터 행렬 T 의 항으로 정수형 DCT를 표현하면 식 (7)을 얻을 수 있다.

$$F = B F_{BRO} = B T B H_w f_{BRO} \quad (7)$$

행렬 T 는 부동소수점 수를 포함하므로 이는 정수형 DCT의 동작을 제한하는 요소로 작용한다. 따라서 이러한 부동소수점 수를 "lifting scheme"와 비선형 연산을 이용하여 정수로 변환하는 과정이 필요하다.

행렬 T 의 부동소수점 수를 정수로 변환하기 위해서는 행렬 T 에 간단한 비선형 연산을 사용하여 정수를 제외한 나머지 부분을 제거하면 된다.

행렬 T 의 부동소수점 수를 정수로 변환하기 위해서는 행렬 T 에 간단한 비선형 연산을 사용하여 정수를 제외한 나머지 부분을 제거하면 된다. 그림 3은 행렬 T 를 통해 표현된 식 (7)의 정수형 DCT를 나타낸다.

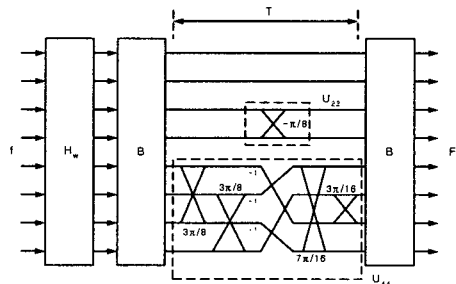


그림 3. WHT와 행렬 T 를 이용한 정수형 DCT의 구조

2) 정수형 DCT와 Lifting scheme

(1) Givens Rotation

식 (6)의 행렬을 block-diagonal의 형태로 다시 쓰면 식 (8)과 같다.

$$T = \begin{bmatrix} [I] & & & \\ & [I] & & 0 \\ & & [U_{22}] & \\ 0 & & & [U_{44}] \end{bmatrix} \quad (8)$$

여기서 $U_{22} = R_{(-\pi/8)}$ 은 2×2 행렬이고 U_{44} 는 4×4 행렬이다. R_{θ} 는 Givens rotation이고 θ 는 rotation 각이다. WHT를 통하여 구성되는 정수형 DCT는 기존 DCT 커널과의 보상을 위해 사용된 행렬 T 를 Givens rotation으로 분해함으로써 이루어진다. 일반적으로 $N \times N$ 직교 행렬은 $N(N-1)/2$ 개의 Givens rotation으로 분해되며 $N=8$ 인 경우 28개의 rotation 각이 존재한다[19]. 그러나 기존 DCT 커널은 변환 계수가 대칭성을 가지므로 그림 3에서 보인 바와 같이 5개의 rotation 각 $-\pi/8, 3\pi/8, 3\pi/8, 7\pi/16, 3\pi/16$ 으로 감소된다.

2×2 Givens rotation U_{22} 은 식 (9)와 같이 표현되고 분해(decomposition)된다.

$$R_{\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} 1 & \frac{\cos(\theta)-1}{\sin(\theta)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin(\theta) & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\cos(\theta)-1}{\sin(\theta)} \\ 0 & 1 \end{bmatrix} \quad (9)$$

여기서 θ 는 2π 의 정수배가 되어서는 안된다. 식 (9)와같은 분해를 "lifting scheme"라 하고 대각선 방향의 성분은 모두 1이다. 대각선 밖의 0이 아닌 행렬의 요소를 lifting 곱이라 하고 일반적으로 부동소수점 수이다. lifting 곱은 one-wing 버터플라이 구조에서 가중계수가 된다. "lifting scheme"는 부동소수점 수만으로 구성되는 행렬을 하나의 부동소수점 수만을 갖는 행렬들로 분해하여 연산의 부담을 줄인다.

식 (9)는 lifting 곱을 정수로 변환하기 위해 비선형 연산을 butterfly 구조의 wing에 사용하더라도 가역성(reversibility)을 보장한다[6]. U_{44} 는 그림 3에서 보인 바와 같이 4개의 rotation 각에 대한 U_{22} 의 확장으로 해석할 수 있으므로 U_{44} 내의 $3\pi/8$ 을 rotation 각으로 갖는 2×2 Givens rotation의 출력은 다른 2×2 Givens rotation의 입력이 되어 식 (9)를 통해 "lifting scheme"를 수행한다.

식 (7)에서 행렬 T 의 계수는 Givens rotation의 rotation 각과 그림 2.4로부터 유도할 수 있다. 식 (7)에

서 행렬 T 의 입, 출력 관계는

$$F_{BRO} = T B H_w f_{BRO}$$

이다.

그림 4는 행렬 T 의 부동소수점 lifting 곱을 정수로 변환하기 위한 임의의 비선형 시스템 S_0 와 S_1 을 포함하는 lifting scheme를 나타낸다. S_0 와 S_1 은 가산된 값은 반드시 감소되므로 전체 변환은 가역성을 보장한다.

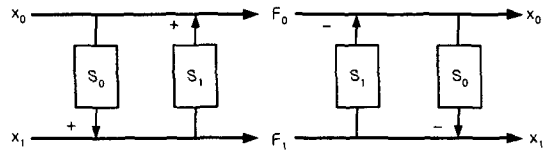


그림 4. lifting scheme

식 (9)의 Givens rotation 분해를 통한 DCT는 등가적인 DCT 구현 방법중 한가지일 뿐이지만 이는 간단한 비선형 연산의 포함만으로 정수형 DCT를 기존 DCT와 거의 동등한 성능을 갖도록 한다. 식 (9)를 통해 표현되는 U_{22} 와 U_{44} 의 lifting 곱은 부동소수점 수를 포함하므로 그림 4 처럼 반올림이나 $[\cdot]$ 와 같은 비선형 연산을 lifting step에 사용하여 행렬 T 의 계수 중 부동소수점 수를 정수로 치환하여 정수간 변환을 얻을 수 있다. 이를 "정수 lifting"이라 하고 정수 lifting을 사용한 변환의 결과가 정수형 DCT이다. 정수형 DCT는 기존 DCT의 근사화 방법 중 하나로 가역성을 만족하면서 정수간 변환을 제공한다.

3. 고속 처리 H.263 부호기

본 논문에서는 그림 5와 같이 기존 DCT의 연산 속도를 향상시키기 위하여 정수형 DCT를 사용하는 H.263의 부호기를 구성한다. 부호기는 프레임을 16×16 크기로 나눈 매크로 블록(Macro Block)단위로 움직임 벡터를 탐색하고 8×8 크기인 블록 단위로 정수형 DCT하여 부호화한다.

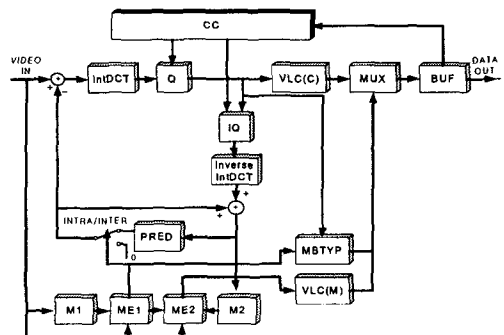


그림 5. 정수형 DCT를 사용한 H.263 부호기

4. 모의 실험 및 결과

본 논문에서는 정수형 DCT를 사용한 고속 처리 H.263 부호기의 성능을 다음의 관점에서 평가한다. 첫째로 정수형 DCT와 기존 DCT의 영상 압축 변환의 성능을 비교한다. 둘째로 H.263 부호기에 정수형 DCT를 적용시킨 H.263 부호기와 기존 DCT를 적용한 H.263 부호기의 성능을 비교하고, 마지막으로 H.263 부호기에서 기존 DCT와 정수형 DCT의 호환성을 실험한다.

표 1 각 변환 알고리즘에 따른 재구성 영상의 PSNR

변환알고리즘	DCT	정수형 DCT
PSNR[dB]	29.93	29.87

표 1은 정수형 DCT는 기존 변환 알고리즘과 비교하여 PSNR의 차이가 매우 적음을 보인다.

정수형 DCT와 기존 DCT의 8×8 블록에 대한 연산량을 표 2에서 비교하였다.

표 2 각 변환 알고리즘에 대한 8×8 블록의 연산량

연산량	변환알고리즘	DCT	정수형 DCT
	곱셈		128
덧셈		112	90

표 2는 정수형 DCT는 덧셈 연산만을 사용하므로 기존 변환 방식들보다 변환에 필요한 연산량을 효율적으로 감소시킴을 보인다.

H.263 부호기에서 기존 DCT와 정수형 DCT산의 호환성을 실험한다. 실험에 사용된 동영상은 100프레임의 "Susie" QCIF 영상을 사용하였으며 전송 속도는 64Kbps로 하였다.

표 3은 부호화와 복호화에서 정수형 DCT나 기존 DCT만을 사용한 경우와 각각의 방법을 부호화와 복호화에서 번갈아 사용한 경우의 Y/Cb/Cr 성분에 대한 PSNR이다. 표 3는 정수형 DCT를 사용한 H.263 부호기는 기존 DCT를 사용한 H.263 부호기에 비하여 PSNR이 저하되지 않으며 또한 부호화와 복호화에서 기존 DCT와 정수형 DCT가 호환됨을 보인다.

표 3 H.263 부호기에서 정수형 DCT와 DCT의 비교

부호화	복호화	
	역 DCT	역 정수형 DCT
DCT	35.13/41.65/43.80	35.08/41.53/41.65
정수형 DCT	35.07/41.58/41.75	35.18/41.53/41.77

5. 결 론

H.263 부호기에서 정수형 DCT와 기존 DCT의 호환성을 실험하기 위하여 부호기와 복호기에 정수형 DCT나 기존 DCT 중 한 방법만을 사용한 경우와 두 방법을 부호기와 복호기에서 번갈아 사용한 경우를 비교하였다. 그 결과 두 방법을 번갈아 사용한 경우는 동일한 방법만을

사용한 경우에 비해 PSNR을 저하시키지 않으면서 시각적으로 서로 호환됨을 보였다.

따라서 새로운 움직임 벡터 탐색 알고리즘과 정수형 DCT를 적용한 H.263 부호기는 효율적인 동영상의 실시간 처리를 가능하게 하므로 PSTN이나 이동 통신망과 같은 초저속망에서 동영상의 저장 및 전송에 적합한 방식이다. 또한 정수형 DCT는 기존 DCT와 호환을 이루므로 정수형 DCT를 사용하는 시스템은 기존 DCT를 사용하는 시스템의 교체없이 사용할 수 있으므로 시스템 개발에 대한 부담을 줄일 수 있다.

참 고 문 헌

- [1]William B. Pennebaker and Joan L. Mitchell, "JPEG Still Image Data Compression Standard," Van Nostrand Reinhold, 1993
- [2]Joan L, Mitchell, William B. Pennebaker, Chad E. Fogg, "Mpeg Video : Compression Standard(Digital Multimedia Standards Series)," Chapman & Hall, Oct. 1996.
- [3]K. G. Beauchamp, "Applications of Walsh and related function," Academic Press, 1984.
- [4]Ying-Jui Chen, Soontoro Oraintara and Truong Nguyen, "Integer Discrete Cosine Transform", IEEE Trans., Feb. 2000.
- [5]S. A. Dyer, N. Ahmed, and D. R. Hummels, "Classification of vectorcardiograms using Walsh and cosine orthogonal transforms," IEEE Trans., Electromag., Computa., vol. EMC-27, pp. 35-40, Feb. 1985.
- [6]Ricardo A. D. Zanardini, "Givens Rotation," Apr. 2002.