

논문 2009-46CI-2-1

# H.264/AVC에서 고속 I Slice 부호화/복호화 방법

## (Fast I Slice Encoding/Decoding Method in H.264/AVC)

오 형 석\*, 신 동 인\*, 김 원 하\*\*

(Hyungsuk Oh, Dongin Shin, and Wonha Kim)

### 요 약

본 논문에서는 H.264/AVC의 I-slice의 모든 블록들을 복원하지 않고 블록의 경계 부분만을 복원하여 intra prediction을 고속으로 수행하는 방법을 개발한다. 이를 위하여 intra prediction의 참조 화소들로 구성될 수 있는 차이 블록의 경계를 고속으로 복원하는 고속 역 정수 DCT를 개발한다. 고속으로 복원된 차이 경계 화소들과 각 예측 모드에 알맞게 구한 예측 화소들을 더하여 경계 화소들을 update하며, intra prediction에 필요한 참조 화소들로 구성한다. 개발된 기법은 H.264/AVC의 정수 DCT와 호환성을 유지하고, 고화질 영상 부호화시 사용되는 대표적인 HD 시퀀스에 적용 가능함을 실험으로 검증하였다.

### Abstract

This paper develops a fast method performing intra prediction which only restores block boundary pixels without decoding all blocks in an I slice of H.264/AVC. To accomplish this, we develop a fast integer inverse DCT scheme that quickly decodes residual block boundary which can be consisted of references pixels. we add the restored block boundary pixels and appropriate calculated prediction pixels for each intra prediction mode and consist of needed reference pixels. The experiments showed that the proposed method produces the reliable performance with reducing the computational complexity, compared to conventional method when applied to H.264/AVC integer DCT.

**Keywords :** H.264/AVC, integer DCT, intra prediction

### I. 서 론

광대역통신망의 급속한 발전과 보급은 고화질의 디지털 방송 및 IPTV와 같은 멀티미디어 서비스를 활성화시키고 있다. 이러한 추세에 따라 고화질과 고효율의 압축 부호화가 가능한 H.264/AVC는 멀티미디어 서비스의 중심으로 평가받으며 고화질 방송의 새로운 contents 시대를 열어가고 있다. 또한 H.264/AVC는 방

송뿐만 아니라 H.264/AVC의 I slice만을 사용하여 방송에 대한 간단한 정보 및 contents를 작은 화면에 제공하는 Visual Radio와 H.264/AVC의 I slice만을 부호화하고 디코더에서 Motion Estimation(ME) 과정을 수행하는 Distributed Video Coding(DVC)에서도 사용될 것이 확실시 되고 있다<sup>[1]</sup>. 따라서 다양한 응용부분에 H.264/AVC를 사용하기 위해서는 서비스 device에 맞게 H.264/AVC를 고속으로 처리할 수 있는 효율적인 기술이 필요하다.

H.264/AVC는 고효율과 고화질을 보장하기 위하여 복잡도가 높은 기술들을 채택하였다. 특히 기존의 MPEG-2,-4와 달리 H.264/AVC에서는 I-slice에서도 공간 영역의 prediction인 intra prediction 기술을 사용한다<sup>[2-3]</sup>. Intra prediction은 주변 블록들을 이용하여 예측하는 기술이기 때문에 주변 블록들이 반드시 복원되어야 한다<sup>[3]</sup>. 따라서 H.264/AVC의 temporal prediction인

\* 학생회원, 경희대학교 전자전파공학과  
(College of Electronic and Radio Engr. Kyunghee Univeristy)

\*\* 정회원, 경희대학교 전자정보대학  
(College of Electronic and Information Engr. Kyunghee Univeristy)

※ 본 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임.  
(KRF-2007-511-D00193)

접수일자: 2009년2월20일, 수정완료일: 2009년3월6일

P 또는 B slice를 생성하지 않고 I slice만을 사용하는 Visual Radio 또는 DVC와 같은 응용부분에서는 intra prediction만을 위하여 I slice의 모든 블록들을 복원하는 것이 비효율적이다. 본 논문에서는 I slice의 블록들을 모두 복원하지 않고 intra prediction을 고속으로 수행하는 방법을 개발한다.

H.264/AVC의 Intra prediction은 I slice에서 이전의 복호화된 주변 블록들을 이용하여 현재 블록을 예측하는 방법으로 블록의 크기에 따라 서로 다른 예측 모드들로 구성된다<sup>[3~4]</sup>. 또한 서로 다른 예측 부호화 모드들 중에 가장 좋은 모드를 선택하여 생성된 예측 블록과 원 영상 블록의 차이 블록을 부호화한 후 다음 블록의 예측을 위해 다시 부호화된 차이 블록을 복원하고 예측 블록과 더하여 부호화된 블록을 복원시킨다. 따라서 H.264/AVC의 intra prediction은 특정위치의 블록을 부호화 또는 복호화시키기 위해서는 이전 블록들이 모두 복원되어 참조 블록을 생성하는 recursive 구조를 가진다<sup>[3~4]</sup>. 그러므로 intra prediction이 수행되기 위한 I-slice의 recursive 구조는 복잡도가 높으며, I-slice의 모든 블록들이 복원되어 저장되어 있어야 하기 때문에 많은 메모리를 요구한다.

본 논문에서 제안하는 방법은 이전 블록들을 모두 복원하여 참조 블록(reference block)을 생성하는 구조 대신에 intra prediction에 필요한 참조 픽셀(reference pixel)들만 구하여 정수 DCT와 호환성을 유지하는 새로운 고속 역 정수 IDCT을 개발한다. 제안하는 방법은 블록 경계에 있는 7개의 화소만을 복원하여 intra prediction에 필요한 참조 픽셀들을 구성하기 때문에 residual 블록과 estimated 블록을 생성하지 않는다. 따라서 제안하는 방법은 H.264/AVC와 동일한 I slice를 생성하며 복잡도와 메모리를 40% 이상 줄인다.

논문의 구성은 다음과 같다. II장에서는 H.264/AVC의 Intra prediction을 자세히 설명하고, III장에서는 Intra prediction을 수행하기 위하여 estimate 블록 생성을 위한 고속 역 정수 DCT 기법을 개발한다. IV장에서는 실험을 통하여 제안하는 기법의 화질과 복잡도를 비교, 분석하여 성능을 검증한다. V장에서는 본 논문의 결론을 도출한다.

## II. H.264/AVC의 Intra prediction

Intra-prediction은 원 영상블록과 인접한 이전에 복

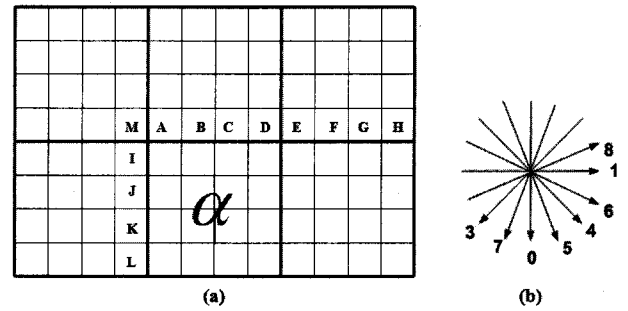


그림 1. 4x4 intra prediction을 위한 참조화소들과 예측 모드의 방향. (a) 4x4블록 intra prediction에 대한 13개의 참조화소들과 예측 블록 (b) 4x4 intra prediction에 대한 9가지 예측 모드의 방향

Fig. 1. Reference pixels and 9 direction of prediction for 4x4 intra prediction. (a) 13 reference pixels and estimated block for 4x4 intra prediction (b) 9 direction of prediction.

원한 블록의 경계에 있는 화소들을 이용하여 예측한 블록과의 차이 블록을 생성한다. 생성된 차이 블록은 정수 DCT와 양자화를 거쳐서 부호화된다. 복원 블록은 차이 블록을 역양자화하고 역 정수 DCT하고나서 예측블록과 더하여 생성된다. 다음 번 예측 블록은 복원된 블록들의 화소 중 부호화할 블록과 가장 인접한 최대 32개의 화소들을 참조하여 예측 모드에 따라서 생성한다<sup>[3~4]</sup>. 4x4블록에 대한 intra prediction은 인접한 13개의 화소들을 참조하며, 16x16블록에 대한 intra prediction은 인접한 32개의 화소들을 참조한다. 휘도성분의 4x4 블록의 예측 모드는 인접한 13개의 화소들을 참조하여 서로 다른 방향으로 예측하는 8가지 모드와 평균값을 취하는 모드로 총 9가지 모드로 구성되고, 16x16 블록의 예측 모드는 가장 인접한 최대 32개의 화소들을 참조하여 서로 다른 방향으로 예측하는 3가지 모드와 평균값을 취하는 모드를 합하여 4가지 모드로 구성되어 있다. 색차성분의 8x8블록의 예측 모드는 4가지로 구성되어 있으며 휘도성분의 16x16 블록의 intra prediction 모드와 크기 차이를 제외하고 예측 부호화 방법은 동일하다.

그림 1은 4x4 intra prediction을 위한 인접한 13개의 참조화소들과 예측 모드의 방향을 나타낸다. 그림 1.(a)에서  $\alpha$ 는 현재 예측할 블록이며, A~M은 복원된 주위 블록들의 참조 화소들을 나타낸다. 그림 1.(b)는 8가지의 예측 모드에 따른 예측 방향을 나타내며, 평균값을 취하는 모드는 방향을 표시하지 않는다. 또한 휘도 16x16 블록의 모드 또는 8x8 블록의 모드는 그림 1.(b)에서 예측 모드가 0인 vertical 방향, 예측 모드가 1인

horizontal 방향, 평균값을 취하는 모드를 사용하고 마지막 모드인 plane 모드의 경우에는 위쪽 블록의 화소들과 왼쪽 블록의 화소들을 대각선방향으로 interpolation한다. 따라서 그림 1에서 보듯이 예측 블록은 현재 부호화될 블록  $\alpha$ 의 이웃하는 위쪽 또는 왼쪽 블록의 복원된 화소로 구성된다. 즉  $f_{mode}$ 를 prediction mode에 따라서 예측 블록을 생성하는 함수라 하면 다음과 같이 예측 블록  $EstBL_{\alpha}$ 을 나타낼 수 있다.

$$EstBL_{\alpha} = f_{mode}(A, B, C, D, E, F, G, H, I, J, K, L, M) \quad (1)$$

그러므로 intra prediction의 참조 화소들은 부호화하고자 하는 블록의 주위 블록들이 반드시 복원되어야 하고, 부호화하고자 하는 블록의 예측 블록은 복원된 주위 참조화소들로부터 예측 모드의 방향에 따라 산술적인 연산을 이용하여 구해진다<sup>[3~4]</sup>.

### III. Estimated block 생성을 위한 고속 역 정수 DCT 기법

제II장에서 논하였듯이, intra prediction 수행 과정에서는 참조할 영상 블록을 복원시켜야 한다. 참조할 블록을 복원하기 위해서는 차이 영상 블록의 정수 DCT 계수를 역 정수 DCT를 하여야 한다<sup>[5]</sup>. 본 절에서는 기존의 4x4 역 정수 DCT를 적용한 방법과 동일한 결과를 얻으면서 복잡도를 현저히 감소시키는 고속 역 정수 DCT 기법을 제시한다.

참조 화소들이 이전에 복원된 영상 블록들의 경계에 있다는 것을 이용하면 최대 32개의 참조 화소들은 복원 블록의 경계에 있는 7개의 화소로 구성할 수 있다. 다음 그림 2.(a)는 4x4블록에 대한 참조화소들과 예측 블록의 관계를 보여 준다. 현재 예측하려는 블록을  $\alpha$ 라 하면 그림에서 보듯이 이전에 복원한 인접 블록들은 Q, P, R, S, T가 된다. A~M는 블록  $\alpha$ 을 예측하는데 사용되는 인접한 13개의 참조 화소들이다. 이러한 참조 화소들을 블록에서의 위치로 표현하면 다음과 같다.

$$\begin{aligned} A &= P(3,0), B = P(3,1), C = P(3,2), D = P(3,3) \\ E &= Q(3,0), F = Q(3,1), G = Q(3,2), H = Q(3,3) \\ I &= S(0,3), J = S(1,3), K = S(2,3), L = S(3,3), M = R(3,3) \end{aligned}$$

즉  $f_{mode}$ 를 prediction mode에 따라서 예측 블록을 생성하는 함수라 하면 식(1)과 같이 예측 블록

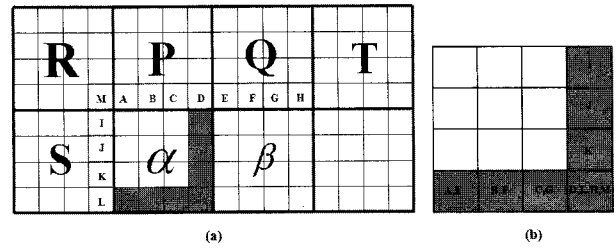


그림 2. Intra prediction을 위한 참조화소들과 블록과의 관계. (a) 4x4블록 intra prediction에 대한 13개의 참조화소들과 예측 블록 (b) 하나의 블록 내에서 복원되어야 할 7개의 경계 화소들

Fig. 2. The relation between reference pixels and block for intra prediction. (a) 13 reference pixels and estimated block for 4x4 intra prediction (b) 7 boundary pixels to be restored in an one block.

$EstBL_{\alpha}$ 을 나타낼 수 있다.

$$\begin{aligned} EstBL_{\alpha} &= f_{mode}(A, B, C, D, E, F, G, H, I, J, K, L, M) \\ &= f_{mode}(P(3,0), P(3,1), P(3,2), P(3,3), \\ &\quad Q(3,0), Q(3,1), Q(3,2), Q(3,3), \\ &\quad S(0,3), S(1,3), S(2,3), S(3,3), R(3,3)) \quad (2) \end{aligned}$$

또한  $\alpha$ 블록의 다음 블록인  $\beta$ 블록을 예측하기 위해서는 블록 Q가 P와 같이 되고, 블록 P가 R과 같이 되고, T가 Q와 같이 되며, 블록  $\alpha$ 가 블록 S와 같이 된다. 그러므로 참조 화소들은 이전에 복원된 블록들의 위쪽과 아래에 경계에 있는 7개의 화소들로 구성할 수 있다. 그림 2.(b)는 다음 블록들의 예측에 사용되는 참조 화소들을 구성하기 위하여 복원되어야 할 7개의 참조 화소들을 나타내며, 하나의 블록 내에서 복원되어야 할 7개의 참조 화소들은 색으로 표시하였다. 따라서 Intra-prediction을 수행하기 위해서는 블록 전체를 복원할 필요 없이 블록 경계에 있는 7개의 화소만 복원하면 된다. 본 논문에서는 이러한 사실에 근거하여 참조 블록을 복원하지 않고 경계화소 7개만을 복원하게 함으로써 4x4 역 정수 DCT의 복잡도를 80%이상 감소시킨 고속 역 정수 DCT를 개발한다.

복원 블록은 차이 블록과 예측 블록이 더하여 저서 구해진다. 따라서 참조 화소로 사용되는 7개의 경계 화소들은 다음과 같이 구하여 진다.

$$\widetilde{Rec}(i, j) = \widetilde{Res}(i, j) + EstBL(i, j)$$

여기서  $\{(i, j)\} = \{(0,3), (1,3), (2,3), (3,0), (3,1), (3,2), (3,3)\}$ . 예측 블록의 화소  $EstBL(i, j)$ 는 식(2)와 같이

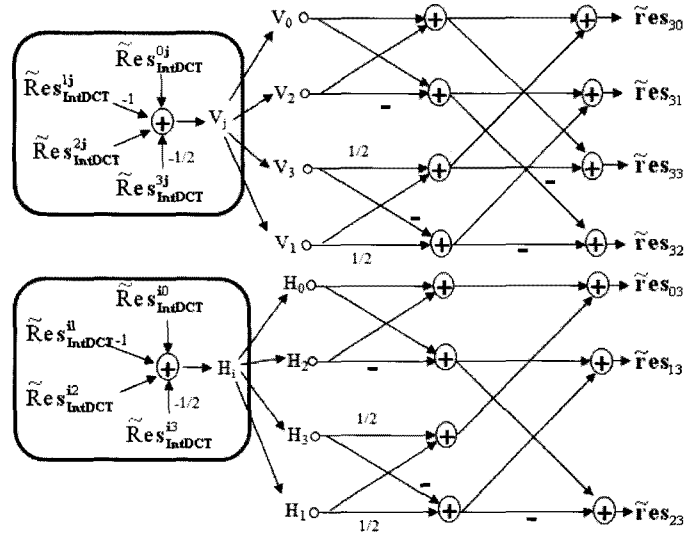


그림 3. 7-IntIDCT 구조  
Fig. 3. 7-IntIDCT structure.

이전의 참조 화소들로부터 예측 모드에 따라 각 경계 화소의 위치에 맞게 구하여 진다. 차이 영상 블록의 7개의 화소들  $\widetilde{Res}(i,j)$ 은 양자화 된 정수 DCT 차이 계수 블록  $\widetilde{Res}_{\text{IntDCT}}$ 에서 경계선에 있는 7개의 화소들만 역 정수 DCT하여 구하면 된다. 이는 통상 4x4 크기의  $\widetilde{Res}_{\text{IntDCT}}$ 으로부터 4x4 크기의 차이 영상 블록  $\widetilde{Res}$ 를 전체를 구하지 않고 7개의 경계 화소만을 구하여도 I-slice 부호화를 위한 Intra prediction이 가능하다는 것을 의미한다. 본 논문에서는 개발된 고속 역 정수 DCT를 7-Integer Inverse DCT (7-IntIDCT)라고 한다. 제안한 7-IntIDCT를 적용하면 4x4 역 정수 DCT를 적용한 방법과 동일하게 예측 블록을 생성할 수 있기 때문에 계산량을 현저히 감소시킬 수 있다.

역 정수 DCT kernel을  $C_i$ 라 하고, 복원된 차이 블록을  $\widetilde{res}$ 라 할 때, 정수 DCT 차이 계수 블록  $\widetilde{Res}_{\text{IntDCT}}$ 의 역 정수 DCT 관계식은 다음과 같다<sup>[5]</sup>.

$$\widetilde{res} = C_i^t \cdot \widetilde{Res}_{\text{IntDCT}} \cdot C_i \quad (3)$$

$$\text{여기서 } C_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \text{이다.}$$

역 정수 DCT kernel의 symmetry 특성을 이용하여 식(3)을 direct 2-D 형식으로 다음과 같이 표현할 수 있다<sup>[6]</sup>.

$$\begin{bmatrix} \widetilde{res}_{00} \\ \widetilde{res}_{01} \\ \widetilde{res}_{02} \\ \widetilde{res}_{03} \\ \widetilde{res}_{10} \\ \widetilde{res}_{11} \\ \widetilde{res}_{12} \\ \widetilde{res}_{13} \\ \widetilde{res}_{20} \\ \widetilde{res}_{21} \\ \widetilde{res}_{22} \\ \widetilde{res}_{23} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1/2 & 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 & 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 & 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 & 1 & -1 & 1 & -1/2 \\ 1 & 1 & 1 & 1/2 & -1 & -1 & -1 & -1/2 \\ 1 & 1/2 & -1 & -1 & -1 & -1/2 & 1 & 1 \\ 1 & -1/2 & -1 & 1 & -1 & 1/2 & 1 & -1 \\ 1 & -1 & 1 & -1/2 & -1 & 1 & -1 & 1/2 \end{bmatrix} \begin{bmatrix} \widetilde{Res}_{\text{IntDCT}}^{00} + \widetilde{Res}_{\text{IntDCT}}^{20} \\ \widetilde{Res}_{\text{IntDCT}}^{01} + \widetilde{Res}_{\text{IntDCT}}^{21} \\ \widetilde{Res}_{\text{IntDCT}}^{02} + \widetilde{Res}_{\text{IntDCT}}^{22} \\ \widetilde{Res}_{\text{IntDCT}}^{03} + \widetilde{Res}_{\text{IntDCT}}^{23} \\ \widetilde{Res}_{\text{IntDCT}}^{10} + \widetilde{Res}_{\text{IntDCT}}^{30}/2 \\ \widetilde{Res}_{\text{IntDCT}}^{11} + \widetilde{Res}_{\text{IntDCT}}^{31}/2 \\ \widetilde{Res}_{\text{IntDCT}}^{12} + \widetilde{Res}_{\text{IntDCT}}^{32}/2 \\ \widetilde{Res}_{\text{IntDCT}}^{13} + \widetilde{Res}_{\text{IntDCT}}^{33}/2 \end{bmatrix} \quad (4)$$

여기서  $\widetilde{res}_{ij}(i,j=0,1,2,3)$ 는 복원된 차이 블록의 (i,j)에 위치한 화소들이고,  $\widetilde{Res}_{\text{IntDCT}}^{ij}$ 는 정수 DCT 차이 계수 블록의 (i,j)에 위치한 계수들이다.

<그림 2.(b)>에서 보듯이 차이 블록의 경계 화소들은  $\widetilde{res}_{03}$ ,  $\widetilde{res}_{13}$ ,  $\widetilde{res}_{23}$ ,  $\widetilde{res}_{33}$ ,  $\widetilde{res}_{30}$ ,  $\widetilde{res}_{31}$ ,  $\widetilde{res}_{32}$ 이며,  $\widetilde{Res}(i,j)$ 을 구성한다.

따라서 7개의 차이 블록의 경계 화소들은 다음과 같이 계산된다.

$$\begin{aligned} H_0 &= \widetilde{Res}_{\text{IntDCT}}^{00} - \widetilde{Res}_{\text{IntDCT}}^{10} + \widetilde{Res}_{\text{IntDCT}}^{20} - \widetilde{Res}_{\text{IntDCT}}^{30}/2 \\ H_1 &= \widetilde{Res}_{\text{IntDCT}}^{01} - \widetilde{Res}_{\text{IntDCT}}^{11} + \widetilde{Res}_{\text{IntDCT}}^{21} - \widetilde{Res}_{\text{IntDCT}}^{31}/2 \\ H_2 &= \widetilde{Res}_{\text{IntDCT}}^{02} - \widetilde{Res}_{\text{IntDCT}}^{12} + \widetilde{Res}_{\text{IntDCT}}^{22} - \widetilde{Res}_{\text{IntDCT}}^{32}/2 \\ H_3 &= \widetilde{Res}_{\text{IntDCT}}^{03} - \widetilde{Res}_{\text{IntDCT}}^{13} + \widetilde{Res}_{\text{IntDCT}}^{23} - \widetilde{Res}_{\text{IntDCT}}^{33}/2 \end{aligned}$$

표 1. 4x4 역 정수 DCT와 제안한 7-IntIDCT의 복잡도

Table 1. Complexity comparison of 4x4 integer IDCT and proposed 7-IntIDCT.

	Addition	Shift	Memory Access	Structure
Fast IDCT	96	48	192	Recursive
Direct IDCT	240	112	64	Non-recursive
7-IntIDCT	59	26	8	Non-recursive

표 2. 각 예측 모드에 따른 참조 화소들을 생성하기 위한 4x4 블록 당 총 연산량 비교

Table 2. Total computational complexity of 4x4 block for producing reference pixels of each intra prediction mode.

Mode		공간영역 기법				제안 기법			
블록 Mode	예측 Mode	덧셈	정수 곱셈	Shift	Memory Access	덧셈	정수 곱셈	Shift	Memory Access
4x4	0	112	0	48	196	66	0	26	12
	1	112	0	48	196	66	0	26	12
	2	120	0	49	200	74	0	27	16
	3	133	1	61	208	86	1	39	15
	4	133	0	62	208	87	0	40	16
	5	138	0	64	208	85	0	38	15
	6	138	0	63	208	85	0	38	15
	7	137	0	63	208	85	0	38	15
16x16	8	137	0	58	201	72	0	30	11
	0	112	0	48	196	66	0	26	12
	1	112	0	48	196	66	0	26	12
	2	120	0	49	200	74	0	27	16
	3	241	42	74	241	101	14	35	41

$$\begin{aligned}
 V_0 &= \widetilde{Res}_{IntDCT}^{00} - \widetilde{Res}_{IntDCT}^{01} + \widetilde{Res}_{IntDCT}^{02} - \widetilde{Res}_{IntDCT}^{03} / 2 \\
 V_1 &= \widetilde{Res}_{IntDCT}^{10} - \widetilde{Res}_{IntDCT}^{11} + \widetilde{Res}_{IntDCT}^{12} - \widetilde{Res}_{IntDCT}^{13} / 2 \\
 V_2 &= \widetilde{Res}_{IntDCT}^{20} - \widetilde{Res}_{IntDCT}^{21} + \widetilde{Res}_{IntDCT}^{22} - \widetilde{Res}_{IntDCT}^{23} / 2 \\
 V_3 &= \widetilde{Res}_{IntDCT}^{30} - \widetilde{Res}_{IntDCT}^{31} + \widetilde{Res}_{IntDCT}^{32} - \widetilde{Res}_{IntDCT}^{33} / 2 \\
 \widetilde{res}_{30} &= V_0 + V_1 + V_2 + V_3 / 2 \\
 \widetilde{res}_{31} &= V_0 + V_1 / 2 - V_2 - V_3 \\
 \widetilde{res}_{32} &= V_0 - V_1 / 2 - V_2 + V_3 \\
 \widetilde{res}_{33} &= V_0 - V_1 + V_2 - V_3 / 2 \\
 \widetilde{res}_{03} &= H_0 + H_1 + H_2 + H_3 / 2 \\
 \widetilde{res}_{13} &= H_0 - H_1 / 2 - H_2 - H_3 \\
 \widetilde{res}_{23} &= H_0 - H_1 / 2 - H_2 + H_3
 \end{aligned} \tag{5}$$

여기서  $H_0, H_1, H_2, H_3, V_0, V_1, V_2, V_3$ 은 데이터간의 dependency가 없는 중간 연산과정이다.

그림 3은 7-IntIDCT의 구조를 보여준다. 그림 3와 식(5)에서 알 수 있듯이 7-IntIDCT의 계산 과정에서는 수직 성분과 수평 성분 간에 연관성이 없고, 중간 연산 과정에서의 데이터간의 dependency없는 non-recursive 구조이다. 따라서 2차원의 계산 시 수반되는 대각선 방향의 계산이 필요 없는 1차원의 계산으로 할 수 있으며, 메모리 access의 횟수를 급격히 감소시킨다.

표 1은 제안한 7-IntIDCT와 4x4 역 정수 DCT의 복잡도를 분석한 것이다. 표 1에서 보듯이 기존의 4x4 고속 역 정수 DCT 보다 덧셈과 shift가 50%이상, memory access 횟수는 90%이상 감소한다. Pentium processor에서 4x4 고속 역 정수 DCT와 7-IntIDCT의

계산 시간을 측정하면 7-IntIDCT의 계산 시간이 4x4 고속 역 정수 DCT의 계산 시간에 10%이하로 된다. 따라서 그림 3에서 보듯이 예측 블록의 생성을 위한 참조 화소들은 7-IntIDCT를 이용하여 복원된 경계 화소들과 블록의 예측 모드에 따라 경계화소의 위치에 맞게 각각 계산된 예측 화소들을 더하여 구한다. 그러므로 참조 화소들을 생성하는 총 복잡도는 예측 블록을 생성하기 위해 요구되는 각 예측 모드에 따른 연산량과 표 1의 연산량을 합한 것이다.

표 2는 참조 화소들을 생성하는 총 복잡도를 분석한 것이다. 표 2에서 보듯이 기존의 모든 블록을 복원하여 참조 블록을 생성하는 공간 영역의 방법보다 덧셈과 shift 연산이 평균 40%, memory access 횟수는 90%이상 감소한다.

#### IV. 실험 및 고찰

본 절에서는 제안하는 기법의 성능을 화질과 복잡도를 평가하여 제안 기법의 성능을 검증한다. 이를 위하여 공간 영역에서 모든 블록을 복원하여 I slice를 만드는 공간 영역의 방법과 제안하는 7-IntIDCT를 이용하

여 참조되는 화소들만 복원하여 I slice를 만드는 방법의 화질과 복잡도를 비교하였다. 영상의 intra prediction mode의 빈도수를 명확하게 비교하기 위하여 시험 동영상들은 MPEG/JVT에서 제시하는 HD (1920x1080)의 해상도로 “Table Setting”, “Playing Cards”, “Rolling Tomato”를 사용하였다. H.264/AVC로 부호화된 bit stream은 MPEG에서 제시한 JM9.8로 생성하였다. 또한 제안한 방법은 JM9.8기반으로 디코더에 구현하였다. Profile은 main으로 하였고 양자화 계수  $Q_p$ 는 20으로 하였다<sup>[7]</sup>.

그림 4는 “Table Setting”영상으로 공간영역 방법과 제안 방법의 화질을 비교한다. 그림 4(a),(b)에서 보듯이 공간영역 방법과 제안 방법으로 생성된 영상들은 완벽하게 동일하다. 또한 그림 4(c)에서 보듯이 이 영상들의 차이 영상에서도 차이가 전혀 발생되지 않음을 알 수 있다. 그림 5는 “Playing Cards”영상으로 공간영역 방법과 제안 방법의 화질을 비교한다. 그림 4와 같이 그림 5에서도 차이가 전혀 발생되지 않음을 알 수 있다.

제안한 방법의 복잡도를 분석하기 위하여 시험 동영상들을 부호화 할 때 발생하는 intra prediction mode의 빈도수를 측정하였다. 각 시험 동영상의 intra

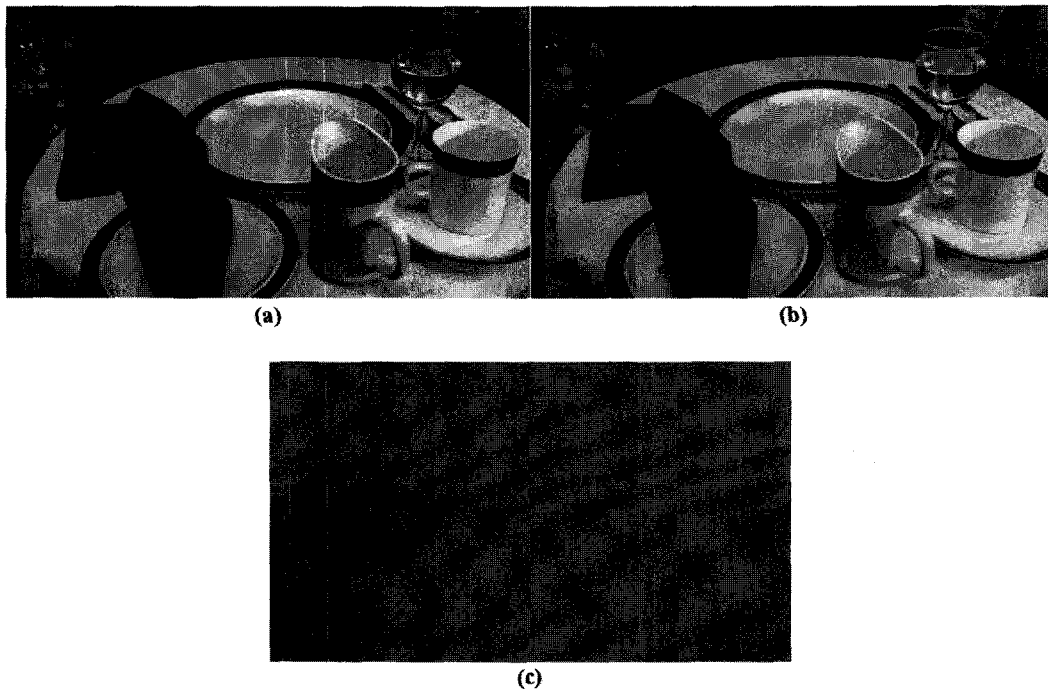


그림 4. “Table Setting” 영상에서의 화질비교. (a)공간영역기법으로 생성한 영상. (b) 제안기법으로 생성한 영상. (c) (a)영상과 (b)영상의 차이 영상.

Fig. 4. Image quality comparison in “Table Setting” (a) Image generated by spatial domain method (b) Image generated by proposed method (c) Difference image from (a) and (b).

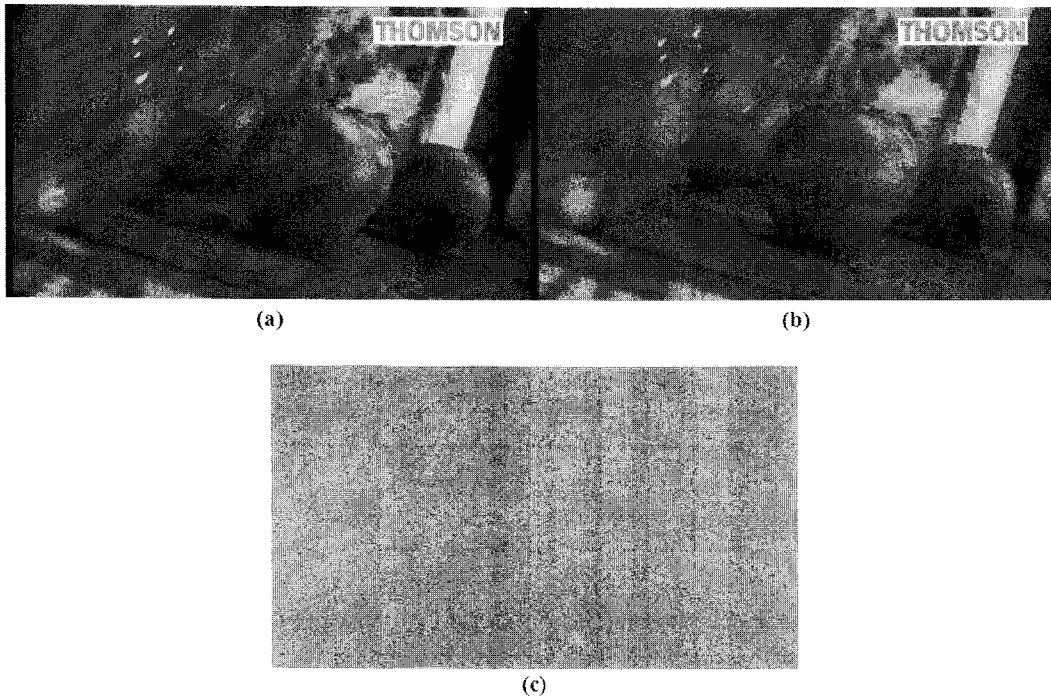


그림 5. “Rolling Tomato” 영상에서의 화질비교. (a)공간영역기법으로 생성한 영상. (b) 제안기법으로 생성한 영상. (c) (a)영상과 (b)영상의 차이 영상.

Fig. 5. Image quality comparison in “Rolling Tomato” (a) Image generated by spatial domain method (b) Image generated by proposed method. (c) Difference image from (a) and (b).

표 3. “Table Setting” 영상의 intra prediction mode의 빈도수

Table 3. Frequency of each intra prediction mode in “Table Setting”.

예측 Mode	0	1	2	3	4	5	6	7	8
4x4 Mode	8%	10%	9%	1%	9%	3%	7%	3%	3%
16x16 Mode	4%	3%	5%	5%					
8x8 UV Mode	8%	10%	6%	5%					

표 4. “Table Setting” 영상의 총 연산량

Table 4. Total computational complexity in “Table Setting”.

	공간 영역 기법				제안 기법			
	덧셈	곱셈	Shift	Memory Access	덧셈	곱셈	Shift	Memory Access
Update ref.	36.12	2.20	5.76	17.44	22.96	1.98	5.43	8.41
IDCT/7-IntIDCT	96.96	0	48.48	193.92	59.59	0	26.26	8.08
Total	133.08	2.20	54.24	211.36	82.55	1.98	31.69	16.49

표 5. “Playing Cards” 영상의 intra prediction mode의 빈도수

Table 5. Frequency of each intra prediction mode in “Playing Setting”.

예측 Mode	0	1	2	3	4	5	6	7	8
4x4 Mode	2%	10%	7%	8%	8%	6%	8%	2%	1%
16x16 Mode	2%	9%	10%	9%					
8x8 UV Mode	2%	1%	9%	7%					

prediction mode의 빈도수는 표 3, 표 5, 표 7과 같다. 공간 영역 방법에서의 연산량은 표 1의 fast 정수

IDCT 연산량과 참조 블록을 생성하는 연산량을 더한 표 2의 연산량이다. 반면 제안하는 방법에서의 연산량

표 6. "Playing Cards" 영상의 총 연산량

Table 6. Total computational complexity in "Playing Cards".

	공간 영역 기법				제안 기법			
	덧셈	곱셈	Shift	Memory Access	덧셈	곱셈	Shift	Memory Access
Update ref.	45.72	4.06	8.12	21.08	14.11	2.94	6.43	10.89
IDCT/7-IntIDCT	96.96	0	48.48	193.92	59.59	0	26.26	8.08
Total	142.68	4.06	56.6	215	73.7	2.94	32.69	18.97

표 7. "Rolling Tomato" 영상의 intra prediction mode의 빈도수

Table 7. Frequency of each intra prediction mode in "Rolling Tomato"

예측 Mode	0	1	2	3	4	5	6	7	8
4x4 Mode	10%	9%	6%	7%	4%	7%	6%	3%	5%
16x16 Mode	4%	1%	11%	6%					
8x8 UV Mode	8%	6%	1%	5%					

표 8. "Rolling Tomato" 영상의 총 연산량

Table 8. Total computational complexity in "Rolling Tomato".

	공간 영역 기법				제안 기법			
	덧셈	곱셈	Shift	Memory Access	덧셈	곱셈	Shift	Memory Access
Update ref.	41.46	2.74	7.08	19.69	12.25	2.94	5.65	9.71
IDCT/7-IntIDCT	96.96	0	48.48	193.92	59.59	0	26.26	8.08
Total	138.42	2.74	55.56	213.61	71.84	2.94	31.91	17.79

은 표 1에서 7-IntIDCT의 연산량과 참조화소를 update 하는 연산량을 더한 표 2의 연산량이다. 참조 블록 또는 참조 화소들을 update하는 연산량들은 intra prediction mode의 빈도수와 표 2에서 제시된 각 mode에서의 요구되는 연산량을 곱한 것이다. 표 4, 표 6, 표 8는 "Table Setting", "Playing Cards", "Rolling Tomato"의 I-slice를 공간 영역 방법과 제안하는 방법으로 생성 시 필요한 연산량을 비교한 것이다. 표들에서 보듯이 총 연산량의 덧셈 연산은 공간영역의 방법에 비하여 평균 44.7%, shift 연산은 평균 41.7%, memory access는 평균 91% 감소하는 것을 볼 수 있고, 곱셈 연산은 거의 같다는 것을 볼 수 있다. 결과적으로 제안하는 7-IntIDCT를 이용한 방법은 공간영역의 방법과 똑같은 화질을 제공하며, 7-IntIDCT를 사용하여 복호화 과정을 단축시키고 예측 블록을 생성하지 않으며 7개의 예측 화소만을 생성하기 때문에 공간영역의 방법보다 연산량을 60%이상 감소시킬 수 있다.

## V. 결 론

본 논문에서는 고속 역 정수 DCT인 7-IntIDCT 기법을 사용하여 블록의 경계면에 있는 7개의 경계 화소를 복원하여 intra prediction에 필요한 참조 픽셀들로 구성하는 방법을 제시하였다.

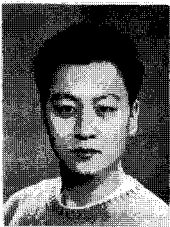
개발된 기법을 H.264/AVC의 참조 소프트웨어로 구현하여 H.264/AVC에 적용하였고, 기존의 공간 영역 기법과 동일한 I-slice 영상을 얻을 수 있음을 검증하였다. 또한 제안하는 7-IntIDCT 기법을 이용한 고속 intra prediction 수행 기법의 복잡도는 공간 영역의 기법보다 평균 40%이상 낮출 수 있음을 실험을 통하여 확인하였으며, 공간 영역의 기법보다 평균 50%이상 빠르게 I-slice 영상을 생성할 수 있다. 그러므로 제안된 기법은 H.264/AVC를 이용하는 다양한 응용부분을 위하여 H.264/AVC의 I-slice를 고속으로 처리할 수 있는 효율적인 방법이라고 결론을 내릴 수 있다.



참 고 문 헌

- [1] B. Girod, A. Aaron, S. Rane and D. Rebello-Monedero, "Distributed Video Coding", Proc. IEEE, Vol. 93, pp. 71-83, 2005.
- [2] ISO/IEC, JTC1/SC29/WG11, *Information Technology-coding of audio-visual objects, Part 2: Video, FDIS 14496-2 Oct. 1998.*
- [3] Iain E. G. Richardson, *H.264 and MPEG-4 Video Compression, Wiley, Hoboken, 2004.*
- [4] "Draft ITU-T Rec. FDIS of Joint Video Spec. (H.264 | ISO/IEC 14496-10 AVC)", JVT of MPEG and VCEG, Doc. JVT-G050r1, May 2003.
- [5] H. S. Malvar, A. Hallapuro, M. Karczewicz and L. Kerofsky, "Low-Complexity Transform and Quantization in H.264/AVC" IEEE Trans. Circuits. Syst., Video Technol., vol. 13, No. 7, pp. 598-603, July 2003.
- [6] Z. Cheng, C. Chen, B. Liu, and J. Yang, "High Throughput 2-D Transform Architectures for H.264 Advanced Video Coders", IEEE Asia-Pacific Conference on Circuits and Systems, pp. 1141-1144, Dec. 2004.
- [7] [http://iphome/hhi.de/shehring/tml/download/old\\_jm/jm9.8.zip](http://iphome/hhi.de/shehring/tml/download/old_jm/jm9.8.zip), Joint Video Team, Reference Software Version 9.8

저 자 소 개



오 형 석(학생회원)  
 2005년 경희대학교 전자공학과  
 학사졸업  
 2007년 경희대학교 일반대학원  
 전자전파공학과 석사졸업  
 2007년~현재 경희대학교  
 일반대학원  
 전자전파공학과 박사 과정  
 <주관심분야 : 영상/동영상부호화, 영상처리>



신 동 인(학생회원)  
 2009년 경희대학교 전자정보학부  
 학사졸업  
 2009년~현재 경희대학교  
 일반대학원  
 전자전파공학과  
 석사 과정  
 <주관심분야 : 영상/동영상부호화, 영상처리>



김 원 하(정회원)-교신저자  
 1985년 연세대학교  
 전자공학과 학사졸업  
 1988년 Univ. of Wisconsin-  
 Madison 전기공학과  
 석사졸업  
 1997년 Univ. of Wisconsin  
 -Madison 전기공학과  
 박사졸업  
 1996년 1월~7월 (미) Motorola, 연구원  
 1997년 8월~2000년 2월 Los Alamos National  
 Lab. 연구원  
 2003년 8월 명지대학교 정보통신공학과 조교수  
 2003년 8월~현재 경희대학교 전자정보대학  
 전자공학과 부교수  
 <주관심분야 : 영상/동영상부호화, 영상처리>