

**요 약**

본 고에서는 최근 표준화가 진행 중인 H.26L 부호기의 화면 내 예측기법에 대해서 살펴보고자 한다. 기존의 H.261, MPEG-1, MPEG-2등에서는 블록별로 독립적으로 DCT 기반의 변환부호화를 수행하였으나, H.26L의 화면내 부호화 방식은 공간 예측 및 정수형 변환을 사용한다. 또한, H.261에서 시작해서, MPEG-4에 이르기까지 지속적으로 사용되어 온 8x8 화소 블록과는 달리 4x4 또는 16x16단위의 처리, 휘도 성분과 색도 성분의 독립적인 처리 등 많은 새로운 부호화 방식이 도입되었다. 효과적인 예측 모드 부호화를 위해 주변 블록의 모드들을 이용한 예측 부호화 방식이 도입되었다. 기존의 가장 대표적인 화면내 부호화기인 H.263에 비해서 H.26L방식이 같은 화질에서 약 40%의 추가적인 압축 효과가 있음을 확인하였다.

**1. 서 론**

본 고에서는 차세대 고성능 동영상 압축 기법인

H.26L의 화면내 부호화 기법(intra coding method)에 대해서 다룬다. 동영상 압축기법의 경우, 부호화하고자 하는 첫 번째 화면 또는 임의의 접근(random access)을 위한 화면의 경우 현재 화면의 정보만을 이용해서 부호화 하고, 이런 부호화 방식을 화면내 부호화 기법(intraframe coding)이라고 부른다. H.261, H.263, MPEG-1, -2, -4등의 표준 기법들에서 화면내 부호화 기법은 공간방향의 중복성(spatial redundancy)을 줄이기 위해 DCT(Discrete Cosine Transform)기반의 변환 부호화 기법(transform coding)을 이용해 왔다. 이들 표준 부호화 기법들에서 사용되는 화면내 부호화 기법은 화면내의 일정크기의 블록(공히 8x8 화소 블록)들을 DCT를 이용해서 변환을 수행하고, 변환 계수(transform coefficient)들에 대해서 양자화(quantization)와 가변장 부호화(Variable Length Coding)을 수행함으로써 원하는 압축률을 얻었다.

이때 압축률은 양자화시에 사용되는 양자화 간격(quantization step size)에 의해서 조절이 된다. 변화 및 양자화과정만을 가지는 기존의 H.261, MPEG-1, MPEG-2등의 방식과 달리 MPEG-4 및 H.263 Annex I 기법의 경우, 양자화 과정 직전에 변환 영역에서 변환 계수 예측 부호화(transform coefficient prediction coding) 기법을 도입하여 압축률을 향상 시켰다.

최근 표준화가 진행되고 있는 H.26L기법의 경우, 공간 영역에서 다양한 예측 기법을 도입해서 압축률을 향상 시켰다. 기존의 방식에 비해서 아래와 같은 차별화된 화면내 부호화 기법을 사용하고 있다.

- 변환 부호화를 위한 블록의 크기는 4x4을 사용함.<sup>1)</sup>
- 변환 방식으로 실수형 kernel을 가지는 DCT 변환이 아닌 정수형 kernel의 정수형 변환(Integer transform)을 사용함.
- 화소영역에서 방향성 예측에 의한 예측 오류 화소에 대해서 변환 부호화를 수행함.
- 주변 블록의 예측 모드를 이용한 확률기반의 예측 모드 부호화 기법

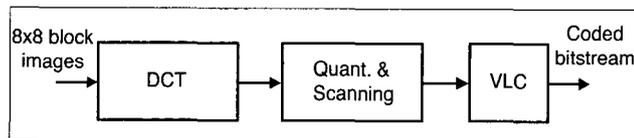
본 고에서는 H.26L에서 도입된 상기 특성들에 대해서 살펴보고, H.26L 화면내 부호화 기법을 기존의 H.263 기법과 비교해서 성능을 평가하고자 한다. 본 고의 구성은 다음과 같다.

2절에서는 기존의 대표적인 화면내 부호화 기법인 H.263 화면내 부호화 기법을 살펴본다. 3절에서는 H.26L에서 사용된 화면내 부호화 기법을 설명하고, H.263기법과 성

능 비교를 4절에서 기술한다. 5절에서 본 고를 요약해서 나타내고자한다.

## 2. H.263 화면내 부호화 기법

〈그림 1〉은 H.263 화면내 부호화 기법의 전체 블록도를 보여준다. 입력 영상은 매크로 블록(Macroblock) 단위로 분할된다. 매크로 블록은 휘도 성분(luminance component)의 경우에 16x16 화소로 구성되어 있고, 색도 성분(chrominance component)들의 경우 휘도성분에 해당하는 위치의 8x8의 블록으로 구성된다. 그리고, 휘도 성분은 다시 4개의 8x8블록으로 나뉘어 진다. 그래서 각 매크로 블록은 6개의 블록으로 구성된다. 각 블록들은 〈그림 1〉에서 보인 바와 같이 DCT를 통해서 변환 계수로 변환된다. 각 변환 계수들은 독립적으로 양자화 과정을 통해서 정수화 된다. 이 과정에서 에너지가 작은 변환계수들은 0으로 변환되고, 특히 DCT의 변환의 에너지 집중 특성에 의해 양자화된 고주파 계수들은 0인 값을 가질 확률이 높아진다. 양자화된 변환계수들은 zigzag 스캔을 통해서 일차원 신호로 변환된다. 이때 zigzag 스캔은 저주파에서 고주파 성분으로 정렬시키는 효과를 가지고 있다. 또한, 양자화에 의해서 많이 발생된 0을 통계적



〈그림 1〉 H.263 화면내 부호화 방법

1) ABT(Adaptive Block Transform)을 사용하는 경우 다양한 크기의 블록 변환을 사용하지만 본 고에서는 ABT 기법은 고려하지 않는다.

인 중복성을 제거하기 위해서 (RUN, LEVEL, EOB)을 이용한 RLC(Run Length Coding)을 사용한다. 여기에서 RUN은 일차원으로 스캔된 양자화 계수들에서 연속된 0의 계수를 의미하고, LEVEL은 이들 연속된 0인 계수들 다음에 오는 0이 아닌 계수를 의미한다. 또한, EOB는 블록의 마지막을 의미한다. 이들 (run, level, EOB)를 새로운 심볼(symbol)로 간주해서 가변장 부호화를 수행한다.

H.263의 선택 부호화 기법중에 Annex I (Advanced Intra Coding Mode)에서는 DCT 변환 영역에서 공간 예측을 통해서 화면내 부호화 성능을 개선시켰다. 자세한 H.263 Annex I 부호화 방식은 [3]에서 잘 분석되어 있다.

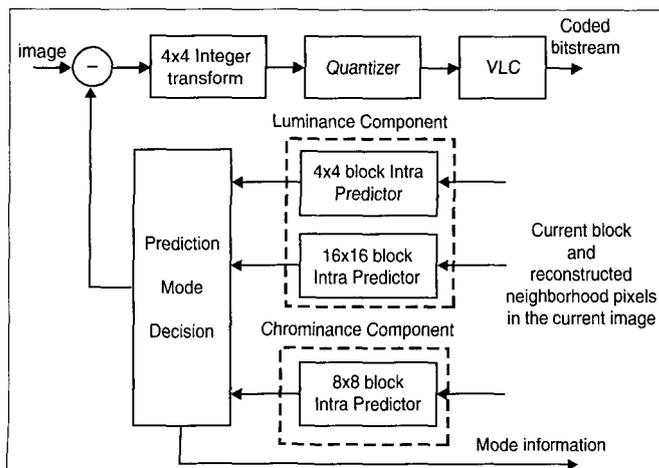
### 3. H.26L 화면내 예측 기법(1)

〈그림 2〉은 H.26L 화면내 부호화 기법의 전체 블록도를 보여준다. 입력 영상은 H.263 방식과 같이 매크로 블록 (Macroblock) 단위로 분할되고, 각

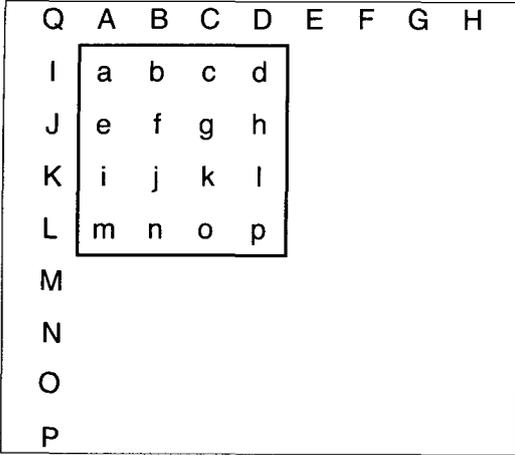
매크로 블록은 6개의 블록으로 구성된다. 또한, 각 블록들은 4x4 크기의 부분블록(subblock)으로 분할된다. 매크로 블록내에는 총 24개의 4x4 블록들로 구성된다. 각 매크로 블록별로 공간방향의 예측을 통해서 예측차 블록을 만들고, 이들 차블록들을 변환 부호기를 통해서 부호화 된다. 예측 방식은 예측 시에 사용되는 블록의 크기에 따라서 두 가지의 종류로 나눌 수 있다.

첫째, 4x4 크기의 블록 예측(Intra-4x4 mode)의 경우, 매크로 블록내의 매 4x4 휘도 블록별로 방향성을 고려한 공간예측을 수행한다. 일반적으로 예측 시에는 휘도 성분의 예측 시에 사용한 파라미터들을 이용해서 색도 성분의 예측시에는 사용하나, H.26L에서는 휘도 성분의 예측과 독립적으로 색도 성분에 대해서 예측을 수행한다. 색도 성분의 대역폭이 휘도 성분에 비해서 좁고, 4:2:0 형식의 입력 영상의 경우에 색도 성분의 해상도도 낮기 때문에 예측 시에는 8x8 단위로 예측을 수행하고, 두개의 색도 블록을 구분하지 않고 같은 예측방식을 사용한다. 그래서 전송해야 할 예측 모드 정보들을 휘도 성분의 경우, 매 4x4 단위의 예측 모드들 16개와 색도 성분의 예측 시 사용한 모드 정보 1개를 전송하게 된다.

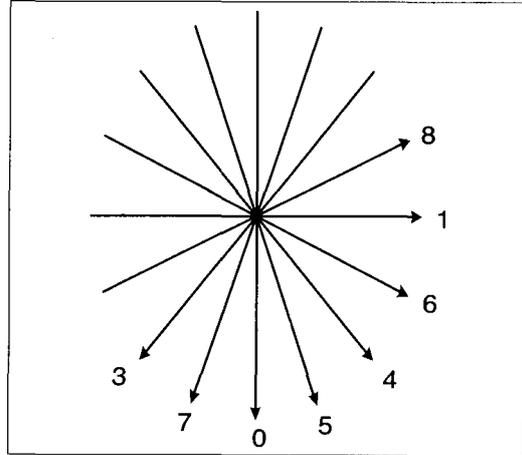
일정한 기울기로 천천히 변화하는 영상의 부분의 경우에는 4x4단위의 예측 방식은 불필요한 예측 모드의 전송량이 높을 수 있고, 블록화 현상을 유발할 수 있다[6]. 이런 부분의 영상의 경우에는 16x16 크기의 블록 예측(Intra-16x16 mode)을 수행하게 된다.



〈그림 2〉 H.26L 화면내 부호화 방법



〈그림 3〉 공간방향의 예측을 위한 화소들의 위치도



〈그림 4〉 화면내 예측의 방향 및 예측 모드명

### 3.1 4x4 휘도블록을 위한 화면내 예측 방법 (Intra-4x4 mode)

〈그림 3〉은 4x4 블록의 공간방향의 예측에 사용되는 화소들을 보여준다. 그림에서 소문자 'a'에서 'p'는 현재 부호화하고자 하는 블록내의 화소를 의미하고, 대문자 'A'부터 'Q'는 이미 부호화된 주변 화소들을 의미하고, 이들은 공간방향의 예측에 사용된다. 주변 화소들은 아래와 같은 상황에서는 사용할 수가 없다.

- 화면밖 또는 현재 슬라이스 밖에 존재하는 경우
- 그들이 raster scan 순서에서 현재 매크로 블록 다음에 오는 매크로 블록에 속하는 경우
- 현재의 4x4블록보다 다음에 보내지는 경우
- constrained\_intra\_pred가 1인 경우에 주변화소들이 화면밖 부호화 매크로 블록 (non-intra macroblock) 방식으로 부호화 된 경우

휘도 신호의 경우에 9가지의 예측모드가 존재하고, 예측 방향 및 예측 모드명을 〈그림 4〉에서 나타내었다. 그림에서 빠져있는 모드 2는 DC예측에 해당한다.

각 예측모드에 대한 자세한 예측 방법은 아래와 같다.

#### 3.1.1 Mode 0: vertical prediction

$$\begin{aligned} \text{Pred}(a, e, i, m) &= A, \\ \text{Pred}(b, f, j, n) &= B, \\ \text{Pred}(c, g, k, o) &= C, \\ \text{Pred}(d, h, l, p) &= D \end{aligned}$$

여기에서 Pred(.)는 현재 화소의 예측치를 나타낸다.

#### 3.1.2 Mode 1: horizontal prediction

이 모드는 I, J, K, L만이 존재하는 경우만 사용된다.

$$\begin{aligned} \text{Pred}(a, b, c, d) &= I, \\ \text{Pred}(e, f, g, h) &= J, \\ \text{Pred}(i, j, k, l) &= K, \\ \text{Pred}(m, n, o, p) &= L. \end{aligned}$$

#### 3.1.3 Mode 2: DC prediction

주변화소 A, B, C, D, I, J, K, L이 모두 존재하는 경우에는 평균치인  $(A+B+C+D+I+J+K+L+4) \gg 3$ 을 이용해서 현재 블록의 모든 화소들을 예측하고, A, B, C, and D가 존재하지 않고 I,

J, K, 및 L만 존재하는 경우 현재 블록의 모든 화소들은  $(I+J+K+L+2) \gg 2$  값으로 예측한다. 여기서  $x \gg n$ 은  $x$ 를  $2^n$ 으로 나누는 연산을 의미한다. 주변화소 I, J, K, and L이 존재하지 않고, A, B, C, and D만 존재하는 경우 현재 블록의 모든 화소들은  $(A+B+C+D+2) \gg 2$  값으로 예측한다. 만일 모든 주변 화소들이 존재하지 않는 경우에 현재 블록내의 모든 화소들은 고정된 예측치인 128을 이용해서 예측한다.

#### 3.1.4 Mode 3: diagonal down/left prediction

$$\begin{aligned} \text{Pred}(a) &= (A + 2B + C + I + 2J + K + 4) \gg 3 \\ \text{Pred}(b, e) &= (B + 2C + D + J + 2K + L + 4) \gg 3 \\ \text{Pred}(c, f, i) &= (C + 2D + E + K + 2L + M + 4) \gg 3 \\ \text{Pred}(d, g, j, m) &= (D + 2E + F + L + 2M + N + 4) \gg 3 \\ \text{Pred}(h, k, n) &= (E + 2F + G + M + 2N + O + 4) \gg 3 \\ \text{Pred}(\ell, o) &= (F + 2G + H + N + 2O + P + 4) \gg 3 \\ \text{Pred}(p) &= (G + H + O + P + 2) \gg 2 \end{aligned}$$

#### 3.1.5 Mode 4: diagonal down/right prediction

$$\begin{aligned} \text{Pred}(m) &= (J + 2K + L + 2) \gg 2 \\ \text{Pred}(i, n) &= (I + 2J + K + 2) \gg 2 \\ \text{Pred}(e, j, o) &= (Q + 2I + J + 2) \gg 2 \\ \text{Pred}(a, f, k, p) &= (A + 2Q + I + 2) \gg 2 \\ \text{Pred}(b, g, \ell) &= (Q + 2A + B + 2) \gg 2 \\ \text{Pred}(c, h) &= (A + 2B + C + 2) \gg 2 \\ \text{Pred}(d) &= (B + 2C + D + 2) \gg 2 \end{aligned}$$

#### 3.1.6 Mode 5: vertical-left prediction

$$\begin{aligned} \text{Pred}(a, j) &= (Q + A + 1) \gg 1 \\ \text{Pred}(b, k) &= (A + B + 1) \gg 1 \\ \text{Pred}(c, \ell) &= (B + C + 1) \gg 1 \\ \text{Pred}(d) &= (C + D + 1) \gg 1 \\ \text{Pred}(e, n) &= (I + 2Q + A + 2) \gg 2 \\ \text{Pred}(f, o) &= (Q + 2A + B + 2) \gg 2 \end{aligned}$$

$$\text{Pred}(g, p) = (A + 2B + C + 2) \gg 2$$

$$\text{Pred}(h) = (B + 2C + D + 2) \gg 2$$

$$\text{Pred}(i) = (Q + 2I + J + 2) \gg 2$$

$$\text{Pred}(m) = (I + 2J + K + 2) \gg 2$$

#### 3.1.7 Mode 6: horizontal-down prediction

$$\begin{aligned} \text{Pred}(a, g) &= (Q + I + 1) \gg 1 \\ \text{Pred}(b, h) &= (I + 2Q + A + 2) \gg 2 \\ \text{Pred}(c) &= (Q + 2A + B + 2) \gg 2 \\ \text{Pred}(d) &= (A + 2B + C + 2) \gg 2 \\ \text{Pred}(e, k) &= (I + J + 1) \gg 1 \\ \text{Pred}(f, \ell) &= (Q + 2I + J + 2) \gg 2 \\ \text{Pred}(i, o) &= (J + K + 1) \gg 1 \\ \text{Pred}(j, p) &= (I + 2J + K + 2) \gg 2 \\ \text{Pred}(m) &= (K + L + 1) \gg 1 \\ \text{Pred}(n) &= (J + 2K + L + 2) \gg 2 \end{aligned}$$

#### 3.1.8 Mode 7: vertical-right prediction

$$\begin{aligned} \text{Pred}(a) &= (2A + 2B + J + 2K + L + 4) \gg 3 \\ \text{Pred}(b, i) &= (B + C + 1) \gg 1 \\ \text{Pred}(c, j) &= (C + D + 1) \gg 1 \\ \text{Pred}(d, k) &= (D + E + 1) \gg 1 \\ \text{Pred}(\ell) &= (E + F + 1) \gg 1 \\ \text{Pred}(e) &= (A + 2B + C + K + 2L + M + 4) \gg 3 \\ \text{Pred}(f, m) &= (B + 2C + D + 2) \gg 2 \\ \text{Pred}(g, n) &= (C + 2D + E + 2) \gg 2 \\ \text{Pred}(h, o) &= (D + 2E + F + 2) \gg 2 \\ \text{Pred}(p) &= (E + 2F + G + 2) \gg 2 \end{aligned}$$

#### 3.1.9 Mode 8: horizontal-up prediction

$$\begin{aligned} \text{Pred}(a) &= (B + 2C + D + 2I + 2J + 4) \gg 3 \\ \text{Pred}(b) &= (C + 2D + E + I + 2J + K + 4) \gg 3 \\ \text{Pred}(c, e) &= (J + K + 1) \gg 1 \\ \text{Pred}(d, f) &= (J + 2K + L + 2) \gg 2 \\ \text{Pred}(g, i) &= (K + L + 1) \gg 1 \\ \text{Pred}(h, j) &= (K + 2L + M + 2) \gg 2 \end{aligned}$$

$$Pred(\ell, n) = (L + 2M + N + 2) \gg 2$$

$$Pred(k, m) = (L + M + 1) \gg 1$$

$$Pred(o) = (M + N + 1) \gg 1$$

$$Pred(p) = (M + 2N + O + 2) \gg 2$$

### 3.2 16x16 휘도 블록을 위한 화면내 예측 방법 (Intra-16x16 mode)

영상 신호가 천천히 변화하는 경우에 4x4 블록단위의 화면내 변환 부호화는 블록킹 현상이 심하게 나타날 수 있다[6]. 이런 영역의 화면내 변환부호화를 위해서는 16x16단위의 예측을 사용하는 것이 화질측면에서 장점을 가진다. 16x16 예측방법은 다음과 같은 네 종류의 예측방법이 있다. 설명을 위해서 다음과 같은 기호를 사용하고자 한다. P(x, y), x=0,...,15, y=0,...,15를 16x16의 휘도블록에서 가로축으로 x번째, 세로축으로 y번째에 위치한 화소로 두자. 또한 P(-1, y)를 현재의 휘도 블록 왼쪽의 화소로 두고, P(x, -1)을 현재 블록 위의 화소로 나타내자. 이런 주변 화소들은 다음과 같은 경우에는 이용할 수 없는 것으로 간주한다.

- 주변 화소가 화면 또는 현재 슬라이스의 밖에 존재하는 경우
- constrained\_intra\_pred가 1인 경우에 주변화소들이 화면간 부호화 매크로 블록 (non-intra macroblock) 방식으로 부호화 된 경우

#### 3.2.1 Mode 0: vertical prediction

주변 화소들인 P(x, -1)이 존재할 경우에만 사용이 될 수 있다.

$$Pred(x, y) = P(x, -1), x, y=0..15$$

#### 3.2.2 Mode 1: horizontal prediction

주변 화소들인 P(-1, y)이 존재할 경우에만 사용

이 될 수 있다.

$$Pred(x, y) = P(-1, y), x, y=0..15$$

#### 3.2.3 Mode 2: DC prediction

$$Pred(x, y) = \left( \sum_{x'=0}^{15} P(x', -1) + \sum_{y'=0}^{15} P(-1, y') + 16 \right) \gg 5, x, y=0..15 \quad (1)$$

만일 주변 화소들인 P(x, -1) 또는 P(-1, y)만 존재하는 경우에 매크로 블록내의 모든 화소들의 다음 식들에 바탕을 두고 예측한다.

$$Pred(x, y) = \left( \sum_{y'=0}^{15} P(-1, y') + 8 \right) \gg 4, x, y=0..15 \quad (2)$$

$$Pred(x, y) = \left( \sum_{x'=0}^{15} P(x', -1) + 8 \right) \gg 4, x, y=0..15 \quad (3)$$

만일 주변 화소들인 P(x, -1)과 P(-1, y) 모두 존재하지 않는 경우에 매크로 블록내의 모든 화소들은 고정된 예측치인 128을 사용한다.

#### 3.2.4 Mode 3: Plane prediction

주변 화소 P(x, -1)과 P(-1, y)이 모두 존재하는 경우에 사용된다.

$$Pred(x, y) = Clip\left( (a + b(x-7) + c(y-7) + 16) \gg 5 \right)$$

여기에서

$$a = 16(P(-1, 15) + P(15, -1))$$

$$b = (5 * H + 32) \gg 6$$

$$c = (5 * V + 32) \gg 6$$

$$H = \sum_{x=1}^8 x (P(7+x, -1) - P(7-x, -1)) \quad (4)$$

$$V = \sum_{y=1}^8 y (P(-1, 7+y) - P(-1, 7-y)) \quad (5)$$

### 3.3 색도 블록을 위한 화면내 예측 방법

매크로 블록내의 색도 화소들은 16x16블록의 휘도 예측방법과 유사하게 예측한다. 그러나 휘도블록에서 사용된 예측 방법과는 독립적인 예측 방법이 사용된다. 휘도블록 중에 일부가 화면내 부호화 방식으로 부호화 되면, 전체 색도 블록은 화면내 부호화 방식으로 부호화 된다. 설명의 편이를 위해서 다음과 같은 기호를 사용한다.  $P(x,-1)$ ,  $x=0..7$ 과  $P(-1,y)$ ,  $y=0..7$ 은 각각 현재 색도 블록의 상위 및 좌측에 위치한 화소들로 정의하자.  $Pred(x, y)$ ,  $x, y = 0..7$ 은 현재 색도 블록의 예측치를 의미하고, 아래와 같은 네 가지 방법에 의해서 예측된다. 다음과 같은 경우에 주변화소  $P(x,-1)$  또는  $P(-1,y)$ 은 존재하지 않는 것으로 간주 한다:

- 주변 화소들이 화면 또는 현재 슬라이스 밖에 존재하는 경우
- constrained\_intra\_pred가 1인 경우에 주변화소들이 화면간 부호화 매크로 블록(non-intra macroblock) 방식으로 부호화 된 경우

$P(x,y)$ 가 존재하지 않는 경우에는 128인 값으로 간주한다. 아래에 기술된 Mode 0 및 Mode 1의 예측 시에  $P(x,y)$ 는  $\{1,2,1\}/4$ 인 필터를 이용해서 필터링을 해서 사용한다. 슬라이스나 화면 경계에서는 화소를 반복해서 필터링을 한다.

#### 3.3.1 Mode 0: vertical prediction

$$F(0, -1) = (P(0, -1) + P(1, -1) + 1) \gg 1$$

$$F(x, -1) = (P(x-1, -1) + 2P(x, -1) + p(x+1, -1) + 2) \gg 2, x=1..6$$

$$F(7, -1) = (P(6, -1) + P(7, -1) + 1) \gg 1$$

$$Pred(x, y) = F(x, -1), x, y=0..7$$

#### 3.3.2 Mode 1: horizontal prediction

$$F(-1, 0) = (P(-1, 0) + P(1, -1) + 1) \gg 1$$

$$F(-1, y) = (P(-1, y-1) + 2P(-1, y) + p(-1, y+1) + 2) \gg 2, x=1..6$$

$$F(-1, 7) = (P(-1, 6) + P(-1, 7) + 1) \gg 1$$

$$Pred(x, y) = F(-1, y), x, y=0..7$$

#### 3.3.3 Mode 2: DC prediction

주변화소  $P(-1,n)$  및  $P(n,-1)$ 이 모두 존재하는 경우에는 다음 식에 의해서 예측된다.

$$Pred(x, y) = \left( \left( \sum_{n=0}^7 P(-1, n) + P(n, -1) \right) + 8 \right) \gg 4, x, y=0..7 \quad (6)$$

주변화소  $P(-1,n)$  또는  $P(n,-1)$ 이 존재하지 않는 경우에는 다음 식에서 의해서 예측된다.

$$Pred(x, y) = \left( \left( \sum_{n=0}^7 P(-1, n) \right) + 4 \right) \gg 3, x, y=0..7 \quad (7)$$

$$Pred(x, y) = \left( \left( \sum_{n=0}^7 P(n, -1) \right) + 4 \right) \gg 3, x, y=0..7 \quad (8)$$

모든 주변 화소들이 존재하지 않는 경우에 고정된 예측치 128을 사용한다.

#### 3.3.4 Mode 3: plane prediction

$$Pred(i, j) = Clip1((a + b(i-3) + c(j-3) + 16) \gg 5), i, j=0..7$$

여기에서

$$a = 16(P(-1, 7) + P(7, -1))$$

$$b = (17 * H + 16) \gg 5$$

$$c = (17 * V + 16) \gg 5$$

$$H = \sum_{i=1}^4 i (P(3+i, -1) - P(3-i, -1)) \quad (9)$$

$$V = \sum_{j=1}^4 j (P(-1, 3+j) - P(-1, 3-j)) \quad (10)$$

Clip1(.)은 [0, 255]사이의 값으로 끊기(truncation) 연산을 의미한다.

### 3.4 예측 모드 선택 방법(3)

다양한 예측 방법들이 존재하는 경우에 예측 모드의 선정은 전체적인 성능에 매우 큰 영향을 미치게 된다. 특히 4x4단위 공간 예측의 경우, 전송해야 할 예측 모드 개수가 매크로 블록별로 17개(휘도 성분 16개 및 색도 성분 1개)로 매우 많기 때문에 단순히 예측 성능만으로 모드를 선정하는 것은 매우 비효율적일 수 있다. 높은 압축율의 성능을 얻기 위해서 모드 선정 시에 다음과 같은 Lagrange multiplier 방법에 바탕을 두고 각 블록별로 모드를 선택한다.

$$cost(p) = SAD(p) + \lambda mbits(p) \quad (11)$$

여기에서,  $\lambda$ 는 다음과 같이 정의된다.

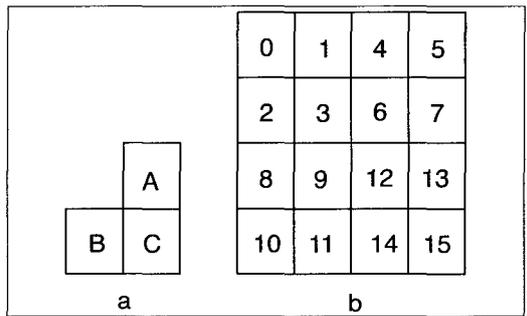
$$\lambda = 0.85 \left( \frac{qp}{3} \right)^2 \quad (12)$$

$p$ 는 예측 방식을 의미하고,  $SAD(p)$ 는  $p$  예측방식에 의해서 발생한 예측차 블록의 SAD(Sum of Absolute Difference)를 의미한다. 특히, 주의해야 할 점은 예측차 블록의 SAD를 구할 때 예측차 블록을 정규화된 Hadarmard 변화를 수행한 후에 나온 예측차 신호를 이용해서 SAD를 구한다는 점이다.  $mbits(p)$ 는  $p$  예측방식을 사용할 때 발생할 모드 비트를 의미한다. 그리고  $\lambda$ 는 Lagrange multiplier를 의미한다. 일반적으로 Lagrange multiplier 값을 결정하는 것은 매우 많은 계산량을 요구한다. 그러나 H.26L의 JM에서 제시하고 있는 방법은 다음과 같이 양자화 간격의 함수로 정의되는 고정된 식

을 이용하므로, 기존의 operational optimization 방식에 비해서 계산량은 매우 작다. 이런 접근 방식에 대한 자세한 이론적인 배경은 [4]에서 잘 설명되어 있다.

### 3.5 예측 모드 부호화 방법

앞서 설명한 바와 같이 H.26L의 화면내 예측의 경우, 특히 4x4 블록단위의 공간 예측의 경우에는 매 블록별로 17개의 예측 모드들을 전송해야 한다. 이런 개별적인 블록 단위의 모드를 표현하는데 매우 많은 부가 정보를 요구하기 때문에, 좀 더 효율적인 모드 정보의 부호화가 필요하다. 블록의 크기가 4x4로 매우 작기 때문에 현재 블록의 예측 모드 값은 인접한 블록들에서 사용된 예측 모드들과 높은 상관성을 가진다. <그림 5(a)>에서 C는 현재 블록의 위치를 의미하고, A와 B블록은 인접 블록들을 의미하고, <그림 5(b)>는 화면내 예측 정보의 전송 순서를 나타낸다.



<그림 5> 화면내 예측모드의 부호화 방법: (a)현재블록 C 및 예측모드의 예측에 사용될 인접 블록 A, B의 위치; (b)화면내 예측 정보의 전송 순서

현재 블록 C의 예측 모드는 인접 블록 A와 B를 부호화하는데 사용한 예측 모드값에 의해서 예측이 된다. 즉, 인접 블록 A와 B가 모두 예측 모드 0으로 예

측된 경우에 현재 블록의 예측 모드가 0일 확률이 높을 것이라는 것을 쉽게 예상할 수 있다. 예측 모드의 부호화 시에는 1 bit의 use\_most\_probable\_mode라는 정보를 이용한다. use\_most\_probable\_mode가 1인 경우에 C의 예측 모드는 주변 블록 A, B모드에 따른 가장 확률적으로 발생하기 쉬운 모드로 예측되었음을 나타낸다. use\_most\_probable\_mode가 0인 경우에는 3bits로 현재 블록 C의 예측 모드를 나타낸다.

Intra-16x16 mode인 경우에 모드 정보는 macroblock type정보와 함께 부호화된다. I slice인 경우에는 macroblock type은 Intra\_4x4와 Intra\_16x16\_x\_y\_z의 두 가지 형태들이 존재한다. 여기에서 Intra\_16x16\_x\_y\_z은 Intra-16x16 mode를 이용해서 화면내 예측을 하였음을 의미하고, 또한 x는 3.1절에서 사용한 화면내 예측 모드값을 의미한다. 참고로 y는 nc를 의미하며, H.263의 CBPC(Coded Block Pattern-Chrominance)에 해당하는 의미를 가지고, z는 AC를 의미한다.

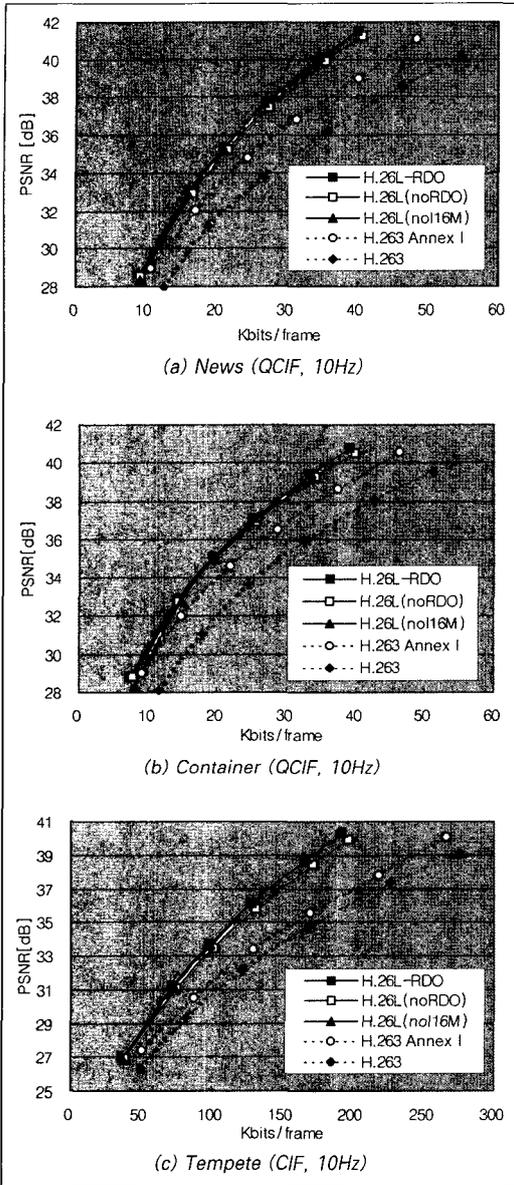
#### 4. 성능 평가

H.26L 화면내 부호기의 성능을 평가하기 위해서 H.263 및 H.263 Annex I 방식과 성능을 비교하였다. 실험에 사용한 소프트웨어 구현들은 H.26L은 JVT(Joint Video Team)의 JM(Joint Model) 4.1을 사용하였고[2], H.263의 경우[5]의 구현을 이용하였다. 성능 비교에 사용한 H.26L 부호기는 ABT(Adaptive Block Transform)은 사용하지 않았다. 또한, 엔트로피 부호기에는 CABAC(Context-based Adaptive Binary Adaptive Coding)을 사용하지 않고, UVLC(Universal VLC)를 사용하였다.

H.26L과 같은 종류의 엔트로피 부호기를 사용하기 위해서 H.263 및 H.263 Annex I에서도 H.263 Annex E SAC(Syntax-based Arithmetic Coder)를 사용하지 않고 VLC를 사용하였다.

본 고에서는 다루는 내용은 화면내 부호화 기법들의 성능 평가이므로, 모든 영상들은 화면내 부호화를 수행한 후 평균치를 비교하였다. 실험에 사용한 영상들은 QCIF(Quarter Common Intermediate Format)의 공간해상도와 시간해상도는 10Hz로 사용했고, 'News', 'Container' 영상 100 frame을 사용했고, 10Hz의 CIF 영상 'Tempete' 250 frame을 사용하였다. 양자화시에는 H.263 및 H.263 Annex I 방식의 경우에 양자화 간격을 4, 5, 7, 10, 15, 25로 고정된 후 화면당 발생 비트율과 PSNR(Peak Signal to Noise Ratio)를 구했고, H.26L기법들은 H.263 Annex I 방식과 비슷한 화질을 발생하는 양자화 간격들인 23, 25, 28, 31, 34을 사용하였다.

〈그림 6〉의 율왜곡(Rate-Distortion) 그래프를 살펴보면 전반적으로 H.26L방식이 H.263방식에 비해서 높은 성능 개선 효과를 가짐을 알 수 있다. 또한, DCT 영역에서 공간방향의 예측 부호화를 H.263 Annex I 방식도 H.263 기본 사양에 비해서 높은 성능 개선 효과가 있음을 알 수 있다. 'News' 영상의 경우, 30Kbits/frame 정도의 발생 비트율에서 H.26L방식은 H.263 Annex I 및 H.263방식들에 비해서 약 3.8dB 및 2.0dB정도의 화질 개선효과가 있음을 알 수 있다. PSNR이 34dB를 기준으로 해서 H.26L방식은 H.263 Annex I 및 H.263방식들에 비해서 약 25% 및 40% 정도의 비트율 감소효과를 보여준다. 또한, H.26L방식에서 모드 선정 시에 율왜곡 최적 선정 방식을 사용한 경우에 이를 사용하지 않은 경우에 비해서 약 0.4dB정도의 화질 개선효과가 있음을 확인 할 수 있었다. 그러



〈 그림 6 〉 H.263 및 H.26L 화면내 부호화기들의 성능 비교; H.26L-RDO(RD 최적 모드 선택 기법 사용), H.26L(RD 최적모드 선택기법 사용하지 않음); H.26L(nol16M) (RD 최적모드선택 사용, Intra-16x16 mode 사용하지 않음); H.263 Annex I (Advanced Intra Prediction 모드사용); H.263(기본 사양만 사용, 독립적인 블록변환기법)

나, Intra\_16x16 모드의 경우에 약간의 개선효과가 있는 것으로 확인되었으나, 그 개선량은 무시할 만

〈 표 1 〉 양자화 간격에 따른 Intra\_4x4 및 Intra\_16x16모드 중 Intra\_16x16 모드가 사용된 비율 (%);

23	5.10	18.75	9.37
25	10.15	23.09	10.24
28	11.75	31.84	11.72
31	14.93	37.87	13.32
34	20.88	43.85	15.30
40	34.53	49.71	24.42

하였다. Intra\_16x16 모드가 최초로 도입된 것은 1999년 10월 RedBank 회의 때에 Tempete 영상의 천천히 변화하는 영역인 하늘 배경 부분에서 Intra\_4x4 모드만 사용할 경우, 심한 블록화 현상으로 주관적 화질의 저하가 있다는 주장에 의해서 도입되었다. 자세한 것은 Q15-I-46 기고안 [6]을 참고하기 바란다. 본 고에서 수행한 실험에서도 객관적 화질에서 개선은 매우 무시할 만한 수준으로 확인되었다.

### 5. 요약

본 고에서는 최근 표준화가 진행 중인 H.26L 부호기의 화면내 예측기법에 대해서 살펴보고자 한다. 기존의 H.261, MPEG-1, MPEG-2등에서는 블록별로 독립적으로 DCT 기반의 변환부호화를 수행하였으나, H.26L의 화면내 부호화 방식은 공간 예측 및 정수형 변환을 사용한다. 또한, H.261에서 시작해서, MPEG-4에 이르기까지 지속적으로 사용되어 온 8x8 화소 블록과는 달리 4x4 또는 16x16단위의 처리, 휘도 성분과 색도 성분의 독립적인 처리 등 많은 새로운 부호화 방식이 도입되었다. 효과적인 예측 모드 부호화를 위해 주변 블록의 모드들을 이용한 예측 부호화 방식이 도입되었다.

기존의 가장 대표적인 화면내 부호화기인 H.263에 비해서 H.26L방식이 같은 화질에서 약 40%의 추가적인 압축률을 주는 것을 확인하였다. 같은 비트율에서 약 3dB의 화질 개선이 있음을 보였다. 그러나 H.26L방식은 기존의 방식에 비해서 많은 복잡

도를 증가를 통해서 이루진 만큼, 이를 수용할 수 있는 플랫폼에서 활용해야 한다는 제한 요소를 가지고 있다. 이런 단점을 극복하기 위한 복잡도 개선에 관한 연구가 추가적으로 이루어져야 할 것으로 판단된다.

● 참고 문헌 ●

(1) T. Wiegand, Joint Final Committee Draft (JFCD) of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Doc. # JVT-D157d5, Aug. 2002.  
 (2) JM1.4.1 (H.26L encoder reference software), available on the anonymous ftp site: ftp://lmvc.org/jvt-exports  
 (3) 유국열, "DCT 계수 이축을 이용한 H.263 기반 화면내 부호화 성능 개선", 한국통신학회 논문집, 제 26권 제 8 호, pp. 1168-1175, 2001년 8월.  
 (4) G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74-90, Nov. 1998.  
 (5) ITU-T Recommendation H.263, Video coding for low bitrate communication, Draft H.263, January 1998.  
 (6) University of British Columbia, Image Processing Lab., H.263+ (H.263 ver. 2.0) encoder software, <http://www.ee.ubc.ca/image>  
 (7) Ram R. Rao, et. al, Comments on the H.26L Test Model, Doc. Q15-I-46, RedBank, Oct. 1999.

필자 소개



유국열

- 1991년 : 경북대학교 공과대학 전자공학과 졸업 (공학사)
- 1993년 : 한국과학기술원 전기및전자공학과 졸업 (공학석사)
- 1998년 : 한국과학기술원 전기및전자공학과 졸업 (공학박사)
- 1998년~2001년 : 삼성전자 중앙연구소 책임 연구원
- 2001년~현재 : 영남대학교 전자정보공학부 전임강사
- 주관심분야 : 표준동영상압축기법, 멀티미디어통신, 스트리밍