

일반논문-09-14-1-06

**B 슬라이스의 압축 효율 향상을 위한 개선된 양방향 대칭 예측 부호화 방법**정 봉 수<sup>a)</sup>, 원 광 현<sup>a)</sup>, 전 병 우<sup>a)‡</sup>**Improved Bi-directional Symmetric Prediction Encoding Method for Enhanced Coding Efficiency of B Slices**Bongsoo Jung<sup>a)</sup>, Kwanhyun Won<sup>a)</sup>, and Byeungwoo Jeon<sup>a)‡</sup>

## 요 약

H.264/AVC의 B 슬라이스의 부호화 효율 향상과 두 개의 움직임벡터를 탐색하는데 소요되는 계산량 감소를 위하여 양방향 대칭(Bi-directional Symmetric) 기법이 개발된 바 있다. 이 기법은, 전방향과 역방향 참조영상 각각에 대하여 움직임 벡터를 구하고 이 두 개를 각각 다 전송하는 종래의 양방향 예측기법과는 달리, 전방향 참조영상에 대해 움직임 벡터를 찾는 동시에 역방향 참조영상에 대한 역방향 움직임 벡터를 전방향 참조영상, 역방향 참조영상, 그리고 현재 영상간의 상대적 거리를 고려한 대칭(Symmetric) 구조로 동시에 계산하여 추정하는 방법이다. 이 기법에 따르면, 전방향 움직임 벡터가 정해지면, 역방향 움직임벡터는 이와 대칭적으로 계산하여 얻어 지므로 움직임벡터 추정 복잡도를 반으로 줄이고, 전방향 움직임벡터만을 전송하도록 하여 부호화할 움직임벡터의 양도 줄일 수 있다. 그러나 이 방법은 항상 전방향 움직임 벡터를 기준으로 역방향 움직임 벡터를 계산하여 얻다 보니, 장면전환등의 경우 오히려 역방향 움직임벡터를 기준으로 전방향 움직임벡터를 추산하는 것이 더욱 효율적인 경우도 있다. 본 논문에서는 전방향 참조영상에 대한 움직임 벡터를 중심으로 역방향 움직임 예측벡터를 추정하는 방법을 일반화시켜, 역방향 움직임 벡터를 중심으로 전방향 움직임 벡터를 추산하여 사용하는 방법을 제안하고 아울러 기존 방법과 제안 방법을 왜곡 관점에서 최적으로 선택하여 사용하는 방법을 제안한다.

## Abstract

A bi-directional symmetric prediction technique has been developed to improve coding efficiency of B-slice and to reduce the computational complexity required to estimate two motion vectors. On the contrary to the conventional bi-directional mode which encodes both forward and backward motion vectors, it only encodes a single forward motion vector, and the missing backward motion vector is derived in a symmetric way from the forward motion vector using temporal distance between forward/backward reference frames to and from the current B picture. Since the backward motion vector is derived from the forward motion vector, it can halve the computational complexity for motion estimation, and also reduces motion vector data to encode. This technique always derives the backward motion vector from the forward motion vector, however, there are cases when the forward motion vector is better to be derived from the backward motion vector especially in scene changes. In this paper, we generalize the idea of the symmetric coding with forward motion vector coding, and propose a new symmetric coding with backward motion vector coding and adaptive selection between the conventional symmetric mode and the proposed symmetric mode based on rate-distortion optimization.

Keywords : B pictures, B slices, Bi-directional, Bi-prediction, Symmetric coding, H.264/AVC

a) 성균관대학교 정보통신공학부

School of Information & Communication Eng., Sungkyunkwan Univ.

‡ 교신저자 : 전병우(bjeon@skku.edu)

※ 이 논문은 과학기술부의 재원으로 한국과학재단 국가지정 연구실사업으로 수행된 연구임(R0A-2006-10286-0(2008))

## 1. 서 론

H.264/AVC는 ITU-T VCEG과 ISO/IEC MPEG이 연합

한 JVT(Joint Video Team)에 의해 표준화된 압축 기술으로써<sup>[1]</sup>, 종래의 다른 비디오 압축표준과 같은 블록기반 부호화 방식을 유지하지만 더욱 정교화된 기술들을 사용함으로써 우수한 비트율 왜곡(Rate-Distortion) 성능을 보여준다. H.264/AVC에는 I 슬라이스, P 슬라이스, B 슬라이스<sup>[2]</sup> 이렇게 크게 3가지의 영상부호화 종류가 있다. I 슬라이스는 각 블록에 대하여 현재 영상 내의 공간적 상관도(spatial correlation)가 가장 높은 공간예측 모드를 선택한 후, 현재 블록과 현재 블록의 주변에 위치한 복원된 블록의 해당화 소와의 오차를 부호화 하고, P 슬라이스는 현재 영상의 현재블록과 시간적 상관도(temporal correlation)가 가장 높은 블록을 전방향 참조영상(forward reference)에서 찾은 후, 그 블록과 현재 블록과의 오차를 부호화 한다. B 슬라이스는 현재 영상과 시간적 상관도가 가장 높은 블록을 전방향과 역방향 또는 어느 한쪽의 두 참조영상의 조합들 중에서 찾은 후, 그 블록과 현재영상 블록과의 오차를 부호화 한다. 이때 B 슬라이스의 예측신호는 두 참조영상내의 참조블록들을 선형 조합하여 얻는다. P 슬라이스는 전방향 참조영상만을 이용하지만 B 슬라이스는 양방향의 참조영상들까지 이용하기 때문에 상대적으로 시간적 상관도를 더욱 더 효과적으로 이용할 수 있게 되고, 결국 예측 성능이 향상되어 뛰어난 압축효율을 얻을 수 있다. 이러한 B 슬라이스는 예측방향에 따라 다시, 전방향 예측(forward prediction), 역방향 예측(backward prediction), 직접 예측(direct prediction)<sup>[3]</sup>, 그리고 양예측(bi-predictive)기법으로 분류할 수 있다. 전방향 예측은 현재 영상 블록과 상관도가 가장 높은 블록을 움직임 탐색을 통해 이전 참조영상(즉 전방향 참조영상)에서 찾는 방법으로 P 슬라이스와 동일한 예측 방법

이다. 역방향 예측은 이와는 달리 이후 참조영상(즉 역방향 참조영상)에서 움직임 탐색하여 찾는 방법이다. 직접 예측은 현재 블록의 움직임벡터의 전송을 생략하는 대신, 시간적(시간적 직접예측-Temporal Direct Mode) 또는 공간적(공간적 직접예측-Spatial Direct Mode)으로 이웃한 블록의 움직임벡터로부터 현재 블록의 전방향과 역방향 움직임벡터를 추산(derivation)하여 얻어내는 방법이다. 양예측(Bi-predictive) 기법은 종래 비디오 부호화 방법에서의 양방향 예측(Bi-directional prediction) 기법을 좀 더 일반화 한 것이다. 즉, 양방향 예측은 전방향 참조영상에 대한 전방향 움직임벡터와 역방향 참조영상에 대한 역방향 움직임벡터를 사용하도록 하기 때문에 반드시 예측방향이 전방향과 역방향으로 동시에 존재 하지만, 양예측(Bi-predictive) 기법은 이 제약조건을 없애, 전방향(또는 역방향) 움직임벡터 둘만이 사용되는 경우도 가능하게 한 것이다. 그림 1은 양예측과 양방향 예측의 예를 나타내었다.

양예측 또는 양방향 예측 기법은 예측효율을 높여 인여 데이터의 압축효율을 증대시키지만 2개의 움직임 벡터를 추정하여야 하고 또한 이들 정보를 전송해야 하기 때문에 움직임 벡터 탐색의 연산 복잡도와 움직임 벡터의 데이터량이 증가하는 문제가 생길 수 있다.

이러한 단점을 극복하고 예측 효율을 높이면서 움직임 벡터 부호화의 효율을 높이기 위해 Tourapis<sup>[3]</sup> 등은 시간축 직접모드와 공간축 직접 모드를 매크로블록 수준과 슬라이스 수준에서 비트율 왜곡 최적화 관점에서 선택하는 기법을 제안하였다. 또한, Ji<sup>[4]</sup> 등은 양방향 예측의 움직임 벡터 부호화 효율을 높이면서 조인트 양방향 추정(Joint motion estimation)의 복잡도를 낮출 수 있는 양방향 예측 기법을 제안하였다. 이 방법은, 예측한 전방향 참조블록과 역방향 참조블록에 대한 움직임 벡터 두 개를 전부 다 보내는 것이 아니라, 전방향 움직임벡터만을 보내고 역방향 움직임벡터는 전송되어진 전방향 움직임 벡터를 현재영상을 중심으로 각각의 참조영상에 대한 거리를 고려하여 대칭시켜서 얻어내는 방법으로, 통상 대칭 모드(Symmetric mode) 방법으로 불리운다. 또한 종래의 양방향 예측기법처럼 전방향 참조영상에서 움직임 벡터를 찾고 역방향 참조영상에서 또 다른 움직임 벡터를 찾는 것과는 달리, 전방향 참조영상에서

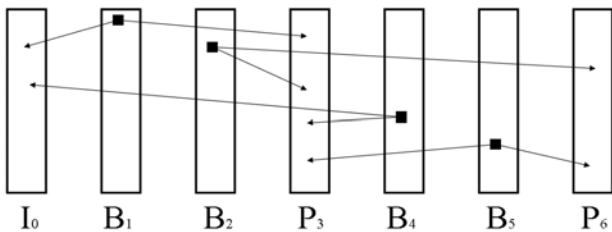


그림 1. 양방향 예측(B<sub>1</sub>, B<sub>5</sub>)과 양예측(B<sub>2</sub>, B<sub>4</sub>)  
 Fig. 1. Bi-directional prediction(B<sub>1</sub>, B<sub>5</sub>) and bi-predictive prediction(B<sub>2</sub>, B<sub>4</sub>)

움직임 벡터를 찾으면서 동시에 현재 영상을 중심으로 참조될 역방향 참조영상에 대한 움직임 벡터를, 두 참조영상과 현재 영상간의 거리를 고려한 대칭구조를 이용하여 역방향 움직임벡터를 계산에 의하여 구하면서 움직임 벡터 탐색을 하도록 하여 양방향 예측에서 얻는 화질 증대와 움직임 벡터 두 개를 다 보내지 않도록 하여 비트전송률 감소의 효과 모두를 얻도록 하는 방법이다. 그러나, 이 방법은 전방향 움직임 벡터만을 이용하여 시간적 거리를 고려하면서 대칭되게 역방향 움직임 벡터를 얻기 때문에, 만약 현재 영상을 기준으로 과거의 전방향 참조영상까지의 거리가 상대적으로 미래의 역방향 참조영상까지의 거리보다 멀 경우, 시간적 상관도에 따라서 과거의 전방향 움직임 벡터가 미래의 역방향 움직임 벡터에 비해서 정확도가 떨어질 수 있는 단점이 있다.

본 논문에서는 과거영상에 대한 전방향 움직임 벡터를 중심으로 역방향 움직임 예측 벡터를 추정하는 기존 대칭 모드 방법<sup>[4]</sup>을 일반화 시켜, 역방향 참조 영상에 대한 역방향 움직임 벡터를 중심으로 전방향 움직임 벡터를 추정하여 사용할 수 있도록 하는 방법을 제안한다. 또한 기존의 대칭 예측 방법과 제안한 대칭 예측 방법을 매크로블록 수준에서 비트율 왜곡(Rate-distortion) 관점에서 최적으로 선택하여 사용하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 우선 II장에서는 종래의 양방향 대칭방법의 전반적인 구조 및 동작설명을 하고 그 한계에 대해서 이야기를 한다. III장에서는 그 한계를 극복하기 위한 제안 방법에 대해서 설명하고, 기존 양방향 대칭 방법과 제안하는 방법을 적용적으로 선택하는 방법에 대해서 설명한다. IV장에서는 실험 및 그 결과에 대해서 논의할 것이다. 마지막으로 V장에서는 결론과 향후 연구에 대해서 논의 한다.

## II. 종래의 양방향 대칭 예측기법

### 1. 양방향 예측 기법

그림 2는 H.264/AVC 표준에서의 양방향 예측(Bi-direc

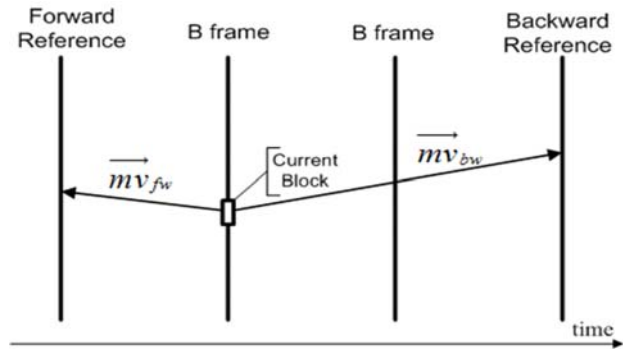


그림 2. 양방향 예측 모드  
Fig. 2. Bi-directional prediction mode

tional prediction)을 나타낸 것이다. 그림 2에 나타난 것과 같이 양방향 예측기법은 과거 참조 프레임에 대한 움직임 벡터(이하, 전방향 움직임 벡터)와 미래 참조프레임에 대한 또 다른 움직임 벡터(이하, 역방향 움직임 벡터)를 탐색한 후, 전방향 참조 프레임에서의 예측 블록과 역방향 참조 프레임에서의 예측 블록의 가중합을 현재 블록의 예측 블록으로 사용하여 현재블록을 예측 부호화한다. 식 (1)은 전방향 움직임 벡터와 역방향 움직임 벡터 위치의 블록들의 가중합으로 예측 블록을 생성하는 수식을 나타낸 것이다.

$$\hat{B}(\overrightarrow{mv_{fw}}, \overrightarrow{mv_{bw}}) = w_{fw} B_{fw}(\overrightarrow{mv_{fw}}) + w_{bw} B_{bw}(\overrightarrow{mv_{bw}}) \quad (1)$$

여기서,  $B_{fw}(\cdot)$ 는 전방향 움직임 벡터( $\overrightarrow{mv_{fw}}$ ) 위치의 블록을,  $B_{bw}(\cdot)$ 는 역방향 움직임 벡터( $\overrightarrow{mv_{bw}}$ ) 위치의 블록을 말하며,  $w_{fw}$ 와  $w_{bw}$ 는 각각 전방향 예측치와 역방향 예측치에 대한 가중치 값, 그리고  $\hat{B}(\cdot)$ 는 예측 블록을 의미한다. 식 (1)에서 최적의  $\hat{B}(\cdot)$  예측 블록을 찾기 위해 식 (2)의 예측 비용값을 계산한 후, 이중 최소의 예측 비용을 가지는 전방향 움직임 벡터(Forward motion vector)와 역방향 움직임 벡터(Backward motion vector)를 양방향 예측 벡터로 선택하며, 이 들을 이용하여 최적의 양방향 예측 참조블록을 결정한다.

$$\hat{B}_{best}(\overrightarrow{mv_{fw}}, \overrightarrow{mv_{bw}}) = \min_{(\overrightarrow{mv_{fw}}, \overrightarrow{mv_{bw}}) \in (SR_{fw}, SR_{bw})} COST(B, \hat{B}(\overrightarrow{mv_{fw}}, \overrightarrow{mv_{bw}})) \quad (2)$$

여기서,  $COST(\cdot)$  는 비용함수 (Cost function)로 일반적으로 현재 블록과 예측 블록의 차이의 절대치 합(SAD)이나 비트율 왜곡 최적화(Rate-distortion optimization) 방법<sup>[7]</sup> 등을 사용한다.  $B$ 는 현재 부호화 할 블록,  $\hat{B}(\cdot)$ 는 식 (1)과 같이 전방향과 역방향 참조프레임 블록의 가중합으로 만든 예측 블록이다. 또한  $\overrightarrow{mv}_{fw}$ 와  $\overrightarrow{mv}_{bw}$  는 각각 전방향(Forward)과 역방향(Backward) 움직임 벡터값을,  $SR_{fw}$ 와  $SR_{bw}$  는 전방향 움직임 탐색범위의 크기와 역방향 움직임 탐색범위의 크기를,  $\hat{B}_{best}(\cdot)$  는 양방향 예측부호화를 위한 최적 예측 블록이 된다. 여기서 양방향 예측부호화를 위한 최적의 전방향 움직임 벡터( $\overrightarrow{mv}_{fw}$ )와 역방향 움직임 벡터( $\overrightarrow{mv}_{bw}$ )를 찾기 위해서는 조인트 움직임 추정(Joint motion estimation)<sup>[5]</sup>을 수행하는데, 조인트 움직임 추정의 경우 움직임 탐색이  $SR_{fw} \times SR_{bw}$  번 필요하여 매우 많은 연산량과 참조프레임 메모리의 접근이 요구되므로 실제 영상 압축에서 사용하기에는 현실적 문제가 많다. 이러한 문제점으로 인하여, 일반적으로 양방향 예측은 식 (3)에 나타난 것처럼 전방향 움직임 벡터와 역방향 움직임 벡터를 독립적으로 탐색한다.

$$\begin{aligned} \hat{B}_{fw}(\overrightarrow{mv}_{fw}) &= \min_{\overrightarrow{mv}_{fw} \in SR_{fw}} COST(B, \hat{B}(\overrightarrow{mv}_{fw})) \\ \hat{B}_{bw}(\overrightarrow{mv}_{bw}) &= \min_{\overrightarrow{mv}_{bw} \in SR_{bw}} COST(B, \hat{B}(\overrightarrow{mv}_{bw})) \end{aligned} \quad (3)$$

여기서,  $\hat{B}_{fw}(\cdot)$ 는 전방향 참조 프레임에서 찾은 최적의 예측 블록을,  $\hat{B}_{bw}(\cdot)$ 는 역방향 참조 프레임에서 찾은 최적의 예측 블록이다.

그 다음, 식 (4)에서와 같이 식 (3)에서 독립적으로 찾은 전방향 움직임 벡터와 역방향 움직임 벡터 위치의 예측 블록들을 가중합 하여 양방향 예측을 수행한다.

$$\hat{B}_{best}(\overrightarrow{mv}_{fw}, \overrightarrow{mv}_{bw}) = w_{fw} \hat{B}_{fw}(\overrightarrow{mv}_{fw}) + w_{bw} \hat{B}_{bw}(\overrightarrow{mv}_{bw}) \quad (4)$$

여기서  $\hat{B}_{best}(\cdot)$  는 식 (3)에서 독립적으로 찾은 최적의 전방향과 역방향 움직임 벡터를 이용한 최적의 양방향 움직임 보상 블록을 나타낸다. 식 (3)과 (4)와 같이 독립적으로 양방향 움직임 추정을 수행한 경우 필요한 탐색 횟수는  $SR_{fw} + SR_{bw}$  로 낮은 복잡도가 요구되지만 대신 양방향 예

측 성능이 떨어지며, 이는 예측 후 얻어지는 잉여데이터의 부호화 비트 발생량의 증가를 수반한다. 또한 전방향 움직임 벡터와 역방향 움직임 벡터를 각각 독립적으로 전송해야 하기 때문에 움직임 벡터의 부호화 데이터량이 많이 발생하는 문제점이 있다.

## 2. 종래의 양방향 대칭 예측 기법

종래 양방향 예측 방법의 단점을 개선하기 위해 Ji<sup>[4]</sup>등은 양방향 대칭 예측 기법(이하, 전방향 움직임 벡터 기준 양방향 대칭 예측 모드(forward symmetric bi-directional mode)라 한다)을 제안하였다. 이 방법은 전방향 움직임 벡터만을 전송하고, 역방향 움직임 벡터는 복호화기에서 미리 정해진 선형 계산식(아래의 식(5) 참조)에 따라 산출하여<sup>[5]</sup> 양방향 예측에 필요한 역방향 움직임 벡터를 구한다. 복호화기는 전방향 움직임 벡터를 기준으로 역방향 움직임 벡터를 계산식에 의해 구한 후 이 둘을 이용하여, 양방향 예측방법에 따른 양방향 예측 참조블록을 구한다. 전방향 움직임 벡터 기준 대칭 모드에서 역방향 움직임 벡터는 전방향 움직임 벡터에 대칭되게 계산하여 사용하기 때문에 결국 전방향 움직임 벡터만을 전송하면 된다. 따라서 움직임 벡터 부호화에 들어가는 비트량을 줄일 수 있다.

그림 3은 전방향 움직임 벡터 기준 대칭 모드를 나타낸 것이다. 전방향 기준 양방향 대칭 움직임 벡터를 추정할 경우, 이에 상응하는 역방향 움직임 벡터는 다음 식 (5)에 의

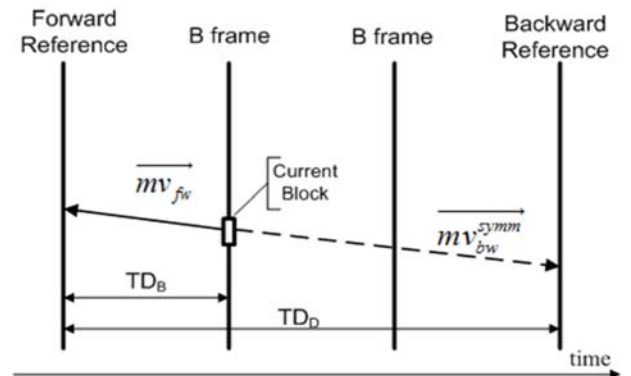


그림 3. 종래의 전방향 움직임 벡터 기준 양방향 대칭 모드  
Fig. 3. Conventional forward symmetric bi-directional mode

해 구한다.

$$\overrightarrow{mv_{bw}^{symm}} = - \frac{(TD_D - TD_B)}{TD_B} \times \overrightarrow{mv_{fw}} \quad (5)$$

여기서  $TD_D$ 는 전방향 참조영상과 역방향 참조영상 사이의 시간상 거리이고,  $TD_B$ 는 현재 영상과 전방향 참조영상 사이의 거리이다.  $\overrightarrow{mv_{fw}}$ 는 부호화할 현재 블록의 전방향 움직임벡터를,  $\overrightarrow{mv_{bw}^{symm}}$ 는 전방향 움직임 벡터에 대칭인 역방향 움직임 벡터를 나타낸다.

이러한 종래의 양방향 대칭 기법을 이용한 부호화 방법은 전방향을 중심으로 역방향까지 동시에 조인트 움직임 추정(Joint motion estimation)이 가능함에 따라 독립적으로 전방향과 역방향 움직임을 찾아 양방향 예측을 수행하는 방법과 비교하여 별도의 복잡도 증가없이 움직임보상의 정확도를 높일 수 있다. 또한 전방향 움직임 벡터만을 전송하여 움직임 벡터의 비트량을 효과적으로 감소시킬 수 있다. 하지만, 일방적으로 전방향 움직임 벡터만을 전송하고, 전방향 탐색의 지정된 탐색 범위 내에서 탐색하기 때문에 현재 부호화하는 영상의 위치가 전방향 참조영상과 멀어지는 경우 움직임 예측 성능이 떨어지는 문제점이 발생한다. 또한 영상 압축에서 움직임 벡터의 전송은 이에 대한 예측 움직임 벡터와 추정된 움직임 벡터와의 차이 값을 전송하는 것으로 동작이 되는데, 현재 블록에서 전방향 움직임 벡터 부호화를 위한 전방향 예측 벡터와의 상관도가 떨어질 경우 오차값이 크게 발생하여 움직임 벡터의 부호화 효율이 크게 떨어지게 된다. 또한 전방향과 역방향의 참조영상 사이의 영상 움직임 특성이 전방향 움직임 벡터에 의해 정해지지 않고 다른 요인에 의해 결정이 되는 경우, 기존 양방향 대칭 기법은 성능이 떨어질 우려가 있다.

### III. 제안하는 선택적 양방향 대칭 예측기법

본 장에서는 종래 양방향 대칭 예측방법의 한계점에 대한 해결방법을 제안하고 그 구조와 동작을 설명한다. 그리고, 기존의 양방향 대칭 방법과 본 논문의 제안 양방향 대칭

방법을 유효곡(Rate-distortion) 관점에서 선택적으로 대칭 예측 모드를 정하여 사용하는 방법을 제안한다.

#### 1. 역방향 움직임 벡터 기준 양방향 대칭 예측 기법

제안 방법은 기존 전방향 움직임 벡터 기준 대칭 예측 방법을 일반화하여, 역방향 움직임 벡터만을 전송하고, 전방향 움직임 벡터는 복호화 과정에서 기존 양방향 대칭 방법과 유사하게 거리를 고려하여 선형적 비율에 따라서 구할 수 있도록 한 역방향 움직임 벡터 기준 양방향 대칭 예측 방법을 추가한 방법이다. 제안 역방향 움직임 벡터 기준 대칭 예측에서 전방향 움직임 벡터( $\overrightarrow{mv_{fw}^{symm}}$ )는 역방향 움직임 벡터에 대칭되게 계산하기 때문에 기존의 방법과는 정반대로 역방향 움직임 벡터만을 전송하면 된다. 따라서 현재 영상이 전방향 참조영상에서 멀어져 오히려 역방향 참조영상과 가까워질 경우, 오히려 역방향 움직임 벡터로부터 전방향 움직임 벡터를 추정하는 것이 더욱 정확도를 높일 수 있고, 역방향 움직임 벡터 부호화를 위한 역방향 예측 벡터와의 상관도도 올라갈 것이므로 결국 움직임 벡터의 부호화 효율 또한 높아져 전송할 비트량이 감소할 것이다.

그림 4는 제안하는 양방향 대칭 예측기법을 나타낸 것이다. 전방향 움직임 벡터는 다음 식 (6)과 같이 역방향 움직임 벡터의 선형 비율로 계산된다.

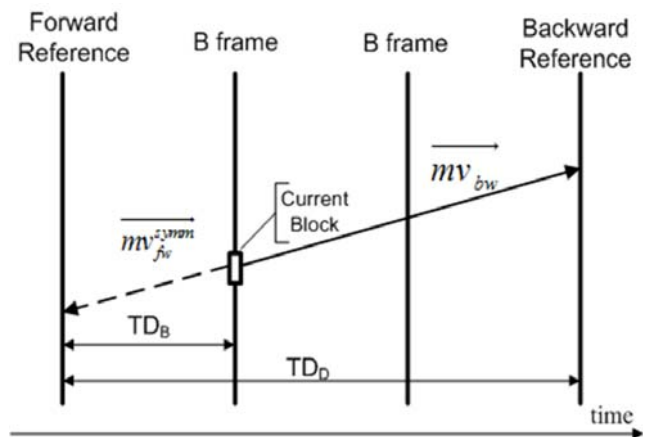


그림 4. 제안하는 역방향 움직임 벡터 기준 양방향 대칭 모드  
Fig. 4. Proposed backward symmetric bi-directional mode

$$\overrightarrow{mv_{fw}^{symm}} = -\frac{TD_B}{(TD_D - TD_B)} \times \overrightarrow{mv_{bw}} \quad (6)$$

여기서  $TD_D$ 와  $TD_B$ 는 이전 식 (1)에서의 전방향 참조영상과 역방향 참조영상 사이의 시간상의 거리와 현재 영상과 전방향 참조영상 사이의 시간상의 거리이다. 그리고  $\overrightarrow{mv_{bw}}$ 는 부호화할 현재 블록에서의 역방향 움직임 벡터이다.

## 2. 윌왜곡 최적화 매크로블록 수준 선택적 대칭 모드

종래 양방향 대칭 모드는 H.264/AVC의 B 슬라이스에서 양방향 예측 모드를 전방향 움직임 벡터 기준 대칭 모드로 대체하여 사용하였다. 제안 방법은 종래 양방향 대칭 모드를 좀 더 일반화 시킨 것으로, 종래 전방향 움직임 벡터 기준 대칭모드와 제안한 역방향 움직임 벡터 기준 대칭모드 방법 중 윌왜곡 최적화 관점에서 상대적으로 더 부호화 효율과 정확도가 높은 하나를 선택하여 전송하는 것으로서, 전방향 대칭 모드가 선택되었는지, 아니면 역방향 대칭 모드가 선택되었는지 복호화기가 알지 못하기 때문에, 이에 대한 정보 또한 복호화기에 보내준다. 전송할 움직임 벡터를 선택하는 방법은 윌왜곡 관점에서 아래 식(7)에 의해서 각각의 비용<sup>[7]</sup>을 계산하여, 그 비용들 중 가장 최소인 것을 선택하는 것이다.

$$J(s, c, MODE | QP, \lambda_{MODE}) = SSD(s, c, MODE | QP) + \lambda_{MODE} R(s, c, MODE | QP) \quad (7)$$

여기서  $\lambda_{MODE}$ 는 양자화 매개변수  $QP$ 에 따라 바뀌는 라그랑지안 멀티플라이어(Lagrangian multiplier)<sup>[8]</sup>이다. 그리고  $s$ 는 현재블록을,  $c$ 는 해당  $MODE$ 로 부호화 후 복원된 영상을 나타낸다.  $SSD(\cdot)$ 는 현재 영상과 복원된 영상의 왜곡값으로 양방향 예측의 경우, 현재 영상 블록을 기준으로 전방향 움직임 벡터만큼 이동한 전방향 참조영상의 블록과 역방향 움직임 벡터만큼 이동한 역방향 참조영상 블록 값을 이용하여 구한 예측값과 현재 원본 블록과의 오차를 계산한 왜곡 수치이다. 마지막으로  $R(\cdot)$ 은 최종적으로 전송되어지는 데이터가 차지하는 비트율을 나타내는 것으

로 식 (8)과 같은 부호화 정보가 포함된다.

$$R = R_{mv} + R_{ref} + R_{MODE} + R_{residue} \quad (8)$$

여기서,  $R_{mv}$ 는 움직임 벡터의 비트량으로 양방향 예측의 경우 전방향 움직임 벡터와 역방향 움직임 벡터에 대한 비트가 포함되고 양방향 대칭 모드는 전방향이나 역방향 중 하나의 움직임 벡터에 대한 비트만 포함된다.  $R_{MODE}$ 는 B 슬라이스에서 예측 모드의 정보로, 양방향 대칭 기법이 선택 되었을 때 전송되는 움직임 벡터가 전방향의 것인지 아니면 역방향의 것인지를 알리는 플래그 한 비트가 포함된다.  $R_{residue}$ 는 부호화 모드와 양자화 계수에 따라 잉여데이터 부호화에 필요한 비트량을 나타낸다.

그림 5는 양방향 예측모드 두 가지를 보여주는 그림이다. 그림 4에 있는 종래 전방향 움직임 벡터 기준 대칭모드와 제안한 역방향 움직임 벡터 기준 대칭모드 윌왜곡 최적화 관점에서 상대적으로 더 부호화 효율과 정확도가 높은 하나를 선택하여 전송하여야 하는데, 움직임예측에는 이 두 가지 대칭예측 모드 외에도 다른 예측방법들이 있기 때문에 이들과의 효율성 비교 또한 중요하다. 이를 위해, 본 논문에서는 선택적 양방향 예측 모드를 수용하는 다음 2가지 방법 - 제안방법 I과 제안 방법 II -의 예측 모드를 제시한다.

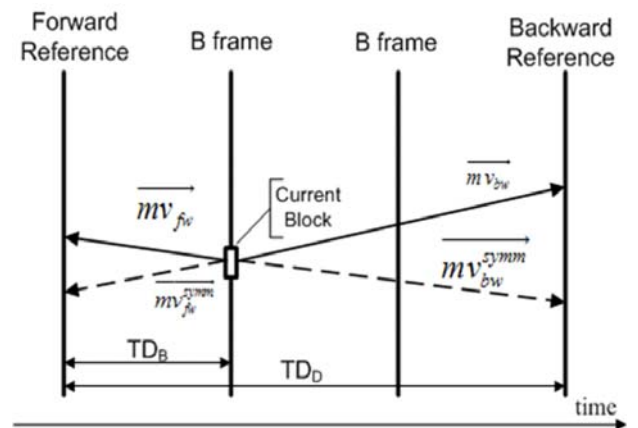


그림 5. 선택적 양방향 대칭 예측

Fig. 5. Selective bi-directional symmetric prediction



· 제안방법 I의 모드 정보

B 슬라이스 :  $MODE \in \{INTRA\text{모드, 전방향 움직임 벡터 기준 대칭 모드, 역방향 움직임 벡터 기준 대칭 모드, 전방향 예측 모드, 역방향 예측 모드, 직접모드}\}$

· 제안방법 II의 모드 정보

B 슬라이스 :  $MODE \in \{INTRA\text{모드, 양방향 예측 모드, 전방향 움직임 벡터 기준 대칭 모드, 역방향 움직임 벡터 기준 대칭 모드, 전방향 예측 모드, 역방향 예측 모드, 직접모드}\}$

여기서, 제안방법 I은 H.264/AVC의 기존 양방향 예측 모드를 사용하지 않고 제안하는 대칭 모드만 사용하는 방법이고, 제안방법 II는 양방향 예측 모드에 대칭 모드를 추가한 방법이다. 그림 6은 제안하는 양방향 예측모드를 적용적으로 선택하는 기법을 그림으로 나타낸 것이다. 그림 6 (a)는 제안방법 I로서, 이것은 식 (9)와 (10)에 따라 각각의 대칭 모드의 윌왜곡 비용을 계산하고 식 (11)을 사용하여 최소 값을 가지는 모드를 최적의 양방향 대칭 모드로 선택한다.

$$J_{Symm_{fw}} = D_{Symm_{fw}} + \lambda R_{Symm_{fw}} \quad (9)$$

$$J_{Symm_{bw}} = D_{Symm_{bw}} + \lambda R_{Symm_{bw}} \quad (10)$$

$$\text{Best Symmetric Mode} = \min \text{COST}(J_{Symm_{fw}}, J_{Symm_{bw}}) \quad (11)$$

식 (9)에서  $J_{Symm_{fw}}$  는 전방향 기준 양방향 대칭모드에서 산출한 윌왜곡 비용이고, 식 (10)에서  $J_{Symm_{bw}}$  는 역방향 기준 양

방향 대칭모드로 선택될 경우 필요한 윌왜곡 비용이다. 그리고 "Best Symmetric Mode"는 가장 최소 비용 방법을 선택하는 함수이다. 이러한 제안방법은 부호화 효율이 좋은 방향의 움직임 벡터를 선택하여 부호화 할 수 있으므로, 전송하는 움직임 벡터의 부호화 효율을 더욱 증대시킬 수 있고, 움직임 예측 에러블록의 부호화 효율을 높일 수 있다. 하지만 전방향 참조영상과 역방향 참조영상 사이의 움직임이 규칙적이지 못한 경우나 움직임이 선형적이지 않은 경우도 많기 때문에, 대칭 모드 사용으로 양방향 예측 모드 그 자체를 대체하는 데에는 문제점이 있다. 따라서 제안방법 II는 이러한 문제를 해결하고자 그림 6(b)에 나타난 것과 같이, 종래 양방향 예측 모드를 대체하지 않고 대칭모드를 새로이 추가하는 방법이다. 그림 6(b)의 제안방법 II에서 각각의 양방향 예측의 윌왜곡 비용은 식 (12), (13), 그리고 (14)와 같이 계산한다.

$$J_{Symm_{fw}} = D_{Symm_{fw}} + \lambda R_{Symm_{fw}} \quad (12)$$

$$J_{Symm_{bw}} = D_{Symm_{bw}} + \lambda R_{Symm_{bw}} \quad (13)$$

$$J_{Bi} = D_{Bi} + \lambda R_{Bi} \quad (14)$$

여기서  $R_{Symm_{fw}}$  는 전방향 기준 양방향 대칭모드로 부호화 하는 경우 발생하는 비트량으로 움직임 벡터는 전방향 움직임 벡터에 대해서만 비트가 발생한다.  $R_{Symm_{bw}}$  는 역방향 기준 양방향 대칭모드로 부호화하는 경우 발생하는 비트량으로 움직임 벡터는 역방향 움직임 벡터에 대해서만 비트가 발생한다. 또한 식 (12)와 (13)은 어떤 양방향 대칭모드를 사용할 것인가에 대한 추가 플래그에 대한 비트를 포함하여 계산한다.  $J_{Bi}$  는

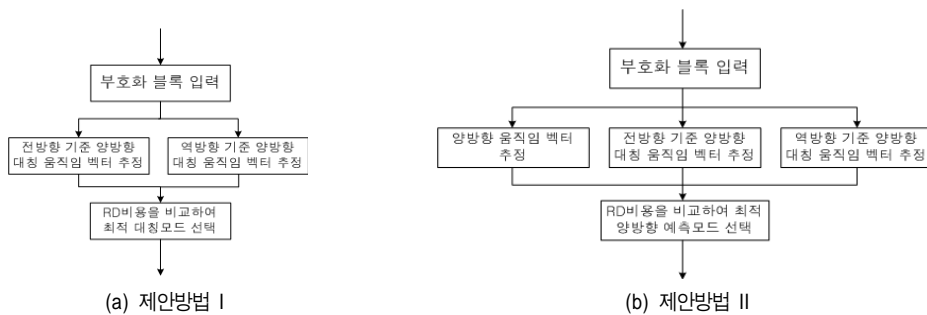


그림 6. 제안하는 선택적 양방향 대칭 예측 방법  
Fig. 6. Proposed selective bi-directional symmetric prediction

양방향 예측 모드에 대한 윌왜곡 비용으로 종래 H.264/AVC의 양방향 예측 모드와 동일하다. 식 (15)는 식 (12), (13), (14)의 비용을 비교하여 최적의 양방향 예측 모드를 선택하는 식이다. 여기서 "Best Bi-directional Mode"는 가장 최소 비용을 가지는 양방향 예측 모드를 선택하는 함수이다.

$$\text{Best Bi-directional Mode} = \min \text{COST}(J_{\text{Symm}_{fw}}, J_{\text{Symm}_{bw}}, J_{\text{Bi}}) \quad (15)$$

#### IV. 실험 및 결과

제안하는 선택적 양방향 대칭 모드에 대한 부호화 성능을 검증하기 위해서 JM 12.2 참조소프트웨어<sup>[8]</sup>를 변형하여 실험 소프트웨어를 구현하였다. 객관적 성능평가를 위해 H.264/AVC의 B슬라이스(Anchor)와 종래의 대칭 모드(Symmetric Mode), 제안방법 I과 제안방법 II의 선택적 양방향 대칭 기법들의 성능비교를 하였다. 또한 본 논문의 양방향 대칭 모드는 16X16 매크로블록이 더 작게 분할되지 않은 16X16 블록크기에 대해서만 적용시켰다. 실험 영상은 초당 30 프레임의 CIF 포맷을 가진 Mobile&Calendar, Paris, Foreman, Hall monitor, Stefan영상을 사용하였다. 실험

표 1. 실험 조건  
Table 1. Test conditions

움직임 탐색범위	± 16
움직임 탐색 방법	전역 탐색 방법
움직임 벡터	해상도 1/4 축소
참조 영상	P 슬라이스: 전방향 1장 B 슬라이스: 전방향 1장 역방향 1장
하다마드(Hadamard) 변환	사용
RD 최적화	사용
엔트로피 부호화	CAVLC
양자화 매개변수	28, 32, 36, 40
B 슬라이스 개수	1장, 2장, 3장
루프 필터	사용

험 조건은 표 1과 같으며, 제안방법이 주는 성능 개선만을 측정하기 위해 제안방법이 적용되는 B 슬라이스에 대해서만 성능을 평가하였다. 또한 B 슬라이스를 1장만 가지는 IBPBP구조, 2장을 가지는 IBBPBBP구조, 그리고 3장을 가지는 IBBBPBBP구조에 대해서 각각 실험하였다.

H.264/AVC의 B 슬라이스(Anchor)의 부호화 성능을 비교기준으로 하여, 기존의 양방향 대칭 기법(Symmetric

표 2. 제안 방법에 대한 성능 비교  
Table 1. Performance comparison of proposed methods

Sequence	# B slice	Symmetric Mode		Proposed I		Proposed II	
		BDPSNR [dB]	BDBR [%]	BDPSNR [dB]	BDBR [%]	BDPSNR [dB]	BDBR [%]
Mobile	IBPBP	0.53	-12.32	0.57	-13.06	0.59	-13.54
	IBBPBBP	0.16	-3.89	0.26	-6.16	0.28	-6.57
	IBBBP	0.04	-1.08	0.10	-2.36	0.15	-3.58
Paris	IBPBP	0.21	-4.65	0.23	-4.97	0.26	-5.78
	IBBPBBP	0.04	-0.90	0.07	-1.55	0.09	-2.07
	IBBBP	0.01	-0.14	0.02	-0.52	0.05	-1.12
Foreman	IBPBP	0.00	-0.06	0.02	-0.62	0.10	-2.95
	IBBPBBP	-0.01	0.34	0.02	-0.49	0.09	-2.52
	IBBBP	-0.05	1.35	-0.04	0.98	0.04	-0.96
Hall	IBPBP	0.02	-1.03	0.03	-1.37	0.05	-2.48
	IBBPBBP	0.03	-1.05	0.06	-2.36	0.07	-2.75
	IBBBP	0.01	-0.51	0.03	-1.28	0.05	-1.84
Stefan	IBPBP	-0.06	1.23	-0.03	0.58	0.09	-1.94
	IBBPBBP	0.01	-0.22	0.10	-2.02	0.20	-3.92
	IBBBP	-0.04	0.76	0.02	-0.33	0.11	-2.03
Average	IBPBP	<b>0.14</b>	<b>-3.37</b>	<b>0.16</b>	<b>-3.89</b>	<b>0.22</b>	<b>-5.34</b>
	IBBPBBP	<b>0.05</b>	<b>-1.14</b>	<b>0.10</b>	<b>-2.52</b>	<b>0.15</b>	<b>-3.58</b>
	IBBBP	<b>-0.01</b>	<b>0.08</b>	<b>0.03</b>	<b>-0.70</b>	<b>0.08</b>	<b>-1.91</b>

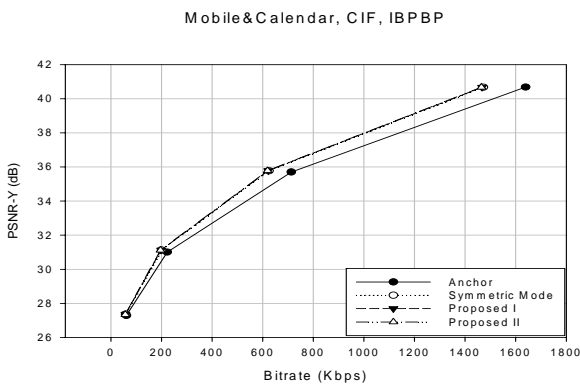


Mode), 제안방법 I과 제안방법 II의 선택적 양방향 대칭 기법에 대해서 BDPSNR과 BDBR 값을 구하였다<sup>[9]</sup>. 여기서, BDPSNR의 (+) 부호와 BDBR의 (-) 부호는 모두 제안 부호화 기술의 성능향상을 나타낸다.

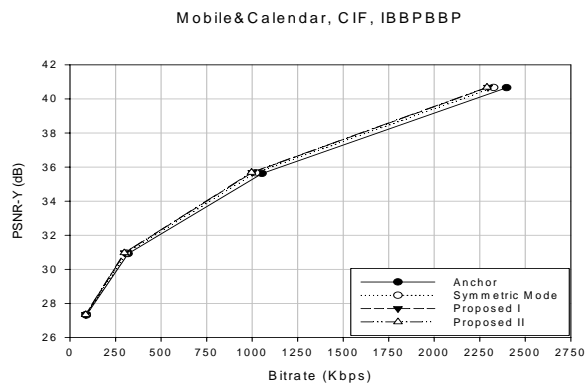
표 2의 결과를 살펴보면 우선 B 슬라이스의 개수가 적을수록 제안방법의 양방향 대칭 모드의 부호화 효율이 우수함을 확인할 수 있다. 특히 IBPBP 부호화 구조에서 Mobile&Calendar 영상을 부호화 하는 경우, 제안방법은 -13% 이상의 매우 높은 부호화 효율 향상을 보여 주었다. 이것은 Mobile&Calendar 영상은 선형 움직임이 많이 발생하고 B 슬라이스가 참조하는 I나 P슬라이스와의 시간축상의 거리가 양방향 모두 가장 가깝기 때문에 시간축 상관도 또한 높아, 제안하는 양방향 대칭 모드의 성능이 우수하게 나타남을

의미한다. 또한, 표 2에서 B 슬라이스의 개수가 많아질수록 종래의 Symmetric Mode나 제안방법 모두 부호화 효율이 감소함을 알 수 있다. 이것은 B 슬라이스의 증가는 B 슬라이스와 참조 프레임 사이의 거리가 멀어지기 때문에 움직임 벡터의 선형특성이 많이 줄어들어 발생하는 것이다. 특히, Foreman과 Stefan 영상은 움직임 벡터의 선형 특성이 매우 적고 움직임이 복잡한 영상으로, 양방향 예측 모드가 아닌 양방향 대칭 모드를 사용하는 종래의 Symmetric Mode와 제안방법 I의 경우, 오히려 부호화 효율이 감소하는 경우가 발생함을 볼 수 있다. 하지만 제안방법 II는 종래의 양방향 예측 모드 또한 수용하기 때문에 움직임이 선형적이지 않은 영상에 대해서도 부호화 효율이 증가함을 알 수 있다.

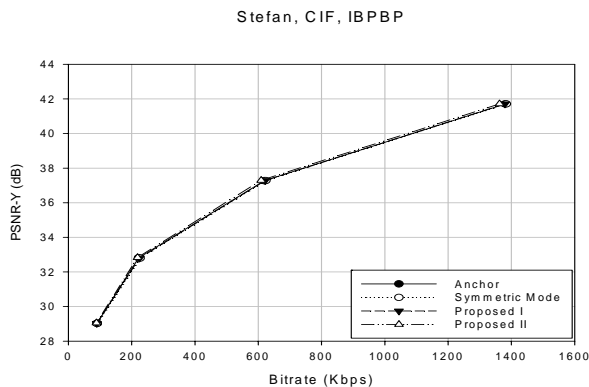
한편, Hall monitor나 Paris영상과 같이 정적인 영상이 대



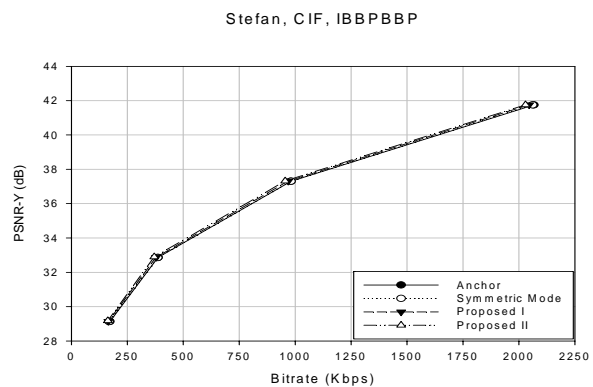
(a) Mobile&Calendar, IBPBP



(b) Mobile&Calendar, IBPBPP



(c) Stefan, IBPBP



(d) Stefan, IBPBPP

그림 7. 비트율 왜곡 성능의 비교

Fig. 7. Performance comparison of rate-distortion

부분을 차지하는 경우 양방향 움직임 모드보다는 B 스킵(SKIP)모드의 선택이 대부분을 차지하기 때문에 다른 영상에 비해서 상대적으로 화질의 개선이 적다. 표 2의 실험 결과에서 IBPBP 구조로 부호화 한 경우, 종래의 대칭 모드(Symmetric Mode)는 평균 -3.37%, 제안방법 I은 평균 -3.98%, 제안방법 II는 평균 -5.34%의 부호화 효율을 향상되었다. IBBPBP구조에서는, 종래의 대칭 모드(Symmetric Mode)는 평균 -1.14%, 제안방법 I은 평균 -2.52%, 제안방법 II는 평균 -3.58%로 부호화 효율이 향상되었으며, IBBPBP구조에서는, 종래의 대칭 모드(Symmetric Mode)는 평균 0.08%로 부호화 손실을, 제안 방법 I은 -0.7%, 제안 방법 II는 -1.91% 부호화 효율 향상을 가져왔다. 특히, B 슬라이스의 개수가 증가할수록 종래의 대칭 모드(Symmetric Mode) 보다 제안방법 I과 제안방법 II의 성능향상이 더욱 두드러지게 나타난다. 이것은 B 슬라이스의 개수가 증가 하여 선형 움직임 벡터가 감소하더라도 제안방법은 윌 왜곡 최적화 관점에서 더 좋은 양방향 예측 모드를 선택하기 때문이다.

그림 7은 IBPBP와 IBBPBP 부호화 구조에서 Mobile&Calendar와 Stefan영상에 대해 비트율 왜곡 성능 비교 그래프를 나타낸 것이다. 이를 보면 역시 제안방법의 경우 Mobile&Calendar 영상에서 가장 많이 성능이 상승한 것을 알 수 있다. 위의 모든 실험 결과를 통하여, 제안한 역방향 대칭 모드와 기존의 대칭 모드를 선택적으로 사용하게 되면 전체적으로 성능이 향상되기 때문에 화질개선과 비트 발생률 측면에서 매우 효과적인 기법이라고 말할 수 있다.

#### IV. 결론 및 향후 연구과제

본 논문은 종래의 양방향 예측 및 양방향 대칭 기법의 한계점을 극복하기 위한 개선된 양방향 예측 기법을 제안하였다. 개선된 양방향 대칭 기법은 전방향 움직임 벡터를 기준으로 대칭되게 역방향 움직임 벡터를 계산하여 부호화하는 기존 방법과 반대로 역방향 움직임 벡터를 기준으로 대칭되게 전방향 움직임 벡터를 계산할 수 있도록 하고, 부호화 할때 매크로블록마다 최적 윌 왜곡 관점에서 이 둘중

의 하나를 선택하여 부호화 하는 방법이다. 전방향 참조영상과 현재 영상과의 시간적 상관도가 낮아질수록 전방향 움직임 벡터의 움직임 추정의 정확도가 떨어지고, 전방향의 예측 움직임 벡터의 상관도가 떨어질 경우, 최종 부호화할 때는 전방향 움직임 벡터와 예측 벡터의 오차를 부호화하기 때문에 결국 부호화 효율이 떨어질 수밖에 없다. 그래서 이를 극복하기 위해 역방향에서도 동일한 원리를 이용해서 부호화를 해보고 윌 왜곡 비용을 측정하여 최소의 비용이 사용되는 기법을 적응적으로 선택을 하여 부호화함으로써 화질 개선과 비트 발생률의 감소라는 이득을 취할 수가 있었다. 제안 방법과 기존 방법을 적응적으로 선택하는 기법을 사용했을 때, H.264/AVC의 양방향 예측과 비교하여 IBPBP 구조에서 B슬라이스의 화질은 약 0.16(dB)~0.22(dB) 개선을 가져왔고, 비트율은 약 3.89%~5.34% 부호화 효율 향상을 가져왔다.

#### 참 고 문 헌

- [1] Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)," Doc. JVT-G050r1, Mar. 2003.
- [2] M. Flierl and B. Girod. "Generalized B pictures and the Draft JVT/H.264 Video Compression standard", IEEE Trans. on Circuits and Systems for Video Tech., Vol. 13, NO. 7, July 2003.
- [3] A. M. Tourapis, F. Wu, and S. Li, "Direct Mode Coding for Bipredictive Slices in the H.264 Standard", IEEE Trans. on Circuits and Systems for Video Tech., Vol. 15, NO. 1, Jan 2005.
- [4] X. Ji, D. Zhao, W. Gao, Q. Huang, S. Ma, and Y. Lu, "New bi-prediction techniques for B pictures coding," IEEE Intern. Conf. on Multimedia and Expo, June 2004.
- [5] S.-W. Wu and A. Gersho, "'Joint Estimation of Forward and Backward Motion Vectors for Interpolative Prediction of Video'", IEEE Trans. on Image Processing, vol.3, no.5, Sep. 1994.
- [6] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," IEEE Signal Process. Mag., vol. 15, no. 6, pp. 74-90, 1998.
- [7] T. Wiegand and B. Girod, "Lagrange Multiplier Selection In Hybrid Video Coder Control," IEEE Int. Conf. on Image Processing, pp. 542 - 545, vol.3., 2001.
- [8] Joint Video Team Reference Software, Version 12.2 (JM12.2). [Online]. Available: <http://bs.hhi.de/~suehring/tml/download/>.
- [9] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," in ITU-T VCEG SG16 Q.6 13th Meeting, VCEG-M33, Austin, Texas, USA, Apr. 2001.

저 자 소 개



정 봉 수

- 2002년 : 성균관대학교 전자전기컴퓨터공학부 졸업(학사)
- 2004년 : 성균관대학교 정보통신공학부 졸업(석사)
- 2004년~현재 : 성균관대학교 정보통신공학부 박사과정
- 주관심분야 : Resource-Aware 영상압축, 멀티미디어 VLSI, 에러강인 디지털 비디오



원 광 현

- 2007년 : 광운대학교 컴퓨터공학과 졸업(학사)
- 2007년~현재 : 성균관대학교 정보통신공학부 석사과정
- 주관심분야 : 멀티미디어 영상압축, 신호처리



전 병 우

- 1985년 : 서울대학교 전자공학과 졸업(학사)
- 1987년 : 서울대학교 전자공학과 졸업(석사)
- 1992년 : Purdue Univ., School of Elec. 졸업(공학박사)
- 1993년~1997년 : 삼성전자 신호처리연구소 선임/수석연구원
- 1997년~현재 : 성균관대학교 정보통신공학부 교수
- 주관심분야 : 멀티미디어 영상압축, 영상인식, 신호처리