

분산 동영상 부호화 시스템을 위한 부가정보 생성 기법의 성능 평가

준회원 문학수*, 종신회원 이창우*, 정회원 이성원**

Efficient Side Information Generation Techniques and Performance Evaluation for Distributed Video Coding System

Hak-soo Moon* *Associate Member*, Chang-woo Lee*° *Lifelong Member*,
Seong-won Lee** *Regular Member*

요 약

단순한 구조의 부호기를 사용할 수 있는 분산 동영상 부호화 시스템에서는 복호기에서 움직임 보상 보간 기법을 이용하여 부가정보를 생성한다. 생성된 부가정보의 정확성이 Wyner-Ziv 프레임의 복원하기 위한 패리티 정보량에 큰 영향을 미치기 때문에 부가정보의 정확한 생성이 분산 동영상 부호화 시스템의 성능에 큰 영향을 미친다. 본 논문에서는 다양한 부가정보 생성 기법의 성능을 분석하고 효율적인 부가정보 생성 기법을 제안하였고 하드웨어 구현 관점에서 각 부가정보 생성 기법을 비교하였다. 또한 분산 동영상 부호화 시스템에서 사용되는 터보 부호의 부호화를 제어 방법이 전체 성능에 미치는 영향과 부가정보 생성 기법이 전체 분산 동영상 부호화 시스템의 성능에 미치는 영향을 비교 분석하였다.

Key Words : Distributed video coding system, Side information, Motion compensated interpolation, Performance evaluation, Wyner-Ziv frame.

ABSTRACT

The side information in the distributed video coding system is generated using motion compensated interpolation methods. Since the accuracy of the generated side information affects the amount of parity bits for the reconstruction of Wyner-Ziv frame, it is important to produce an accurate side information. In this paper, we analyze the performance of various side information generation methods and propose an effective side information generation technique. Also, we compare each side information generation methods from the hardware point of view and analyze the performance of distributed video coding system using various side information generation methods.

I. 서 론

H.264 부호화 표준을 비롯한 기존의 동영상 부호화 시스템은 우수한 압축 효율을 보이지만 움직임 추

정으로 인하여 부호기의 복잡도가 매우 크다^[1]. 부호기의 복잡도를 감소시키기 위해서 분산 동영상 부호화(DVC : distributed video coding) 시스템에서는 부호기에서 움직임 추정을 하지 않고 프레임간의 상관

※ 이 논문은 2010년도 정부의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2010-0028178)

* 가톨릭대학교 정보통신전자공학과(haksoo84@catholic.ac.kr, changwoo@catholic.ac.kr), (° : 교신저자)

** 광운대학교 컴퓨터공학과 (swlee@kw.ac.kr)

논문번호 : KISC2010-07-324, 접수일자 : 2010년 7월 26일, 최종논문접수일자 : 2011년 1월 19일

성을 복호기에서 이용한다²⁾. 특히 복호기에서 Wyner-Ziv 프레임을 복원하기 위해 부가정보(side information)를 생성하게 되는데 이때 부가정보와 Wyner-Ziv 프레임간의 유사도에 따라서 분산 동영상 부호화 시스템의 성능이 크게 좌우된다. 최근에 효율적으로 부가정보를 생성하는 기법이 활발히 연구되고 있는데 순방향 움직임 추정 및 대칭 양방향 예측을 이용하여 움직임 벡터를 추정하고 가중치 벡터 중간 값 필터 등을 이용하여 움직임 벡터를 보정하는 방법이 많이 사용된다⁴⁻¹⁰⁾.

본 논문에서는 다양한 부가정보 생성 기법을 이용하여 생성한 부가정보가 분산 동영상 부호화 시스템의 성능에 미치는 영향을 비교 분석한다. 분산 동영상 부호화 시스템에서는 정확한 부가 정보 생성이 매우 중요하므로 기존의 다양한 부가정보 생성 기법의 성능 분석과 효율적인 부가 정보 생성 기법에 대한 연구는 전체 분산 동영상 부호화 시스템의 성능 향상을 위해서 필수적이다. 이를 위하여 순방향 움직임 추정(forward motion estimation) 방법 및 역방향 움직임 추정(backward motion estimation) 방법, 대칭 양방향 움직임 추정(bilateral motion estimation) 방법 등 다양한 방법을 통해 생성된 부가정보와 원 영상간의 PSNR 결과를 분석하고 효율적인 부가정보 생성 기법을 제안한다. 다양한 부가정보 생성기법을 분산 동영상 부호화 시스템에 적용시켰을 때의 성능을 해석하고 분산 동영상 부호화 시스템에서 사용되는 터보 부호의 부호화를 제어 방법에 따른 성능을 해석한다. 또한 각 부가정보 생성 기법을 하드웨어 구현 관점에서 비교 분석하였다. II장에서는 분산 동영상 부호화 시스템에 대하여 설명한다. III장에서는 부가정보 생성 방법을 위한 다양한 기법들을 설명하고 IV장에서는 부가정보 생성 기법의 하드웨어 복잡도에 대해 설명한다. V장에서는 다양한 방법을 통해 생성된 부가정보와 원 영상간의 PSNR 결과와 분산 동영상 부호화 시스템에 적용시켰을 때의 성능 및 터보 부호의 부호화를 제어에 따른 성능을 모의실험을 통하여 분석한다. 마지막 VI장에서는 결론을 맺는다.

II. 분산 동영상 부호화 시스템

분산 동영상 부호화 시스템은 그림 1에 도시한 것과 같이 크게 키 프레임(Key frame)을 부호화하는 부분과 Wyner-Ziv 프레임을 부호화하는 부분으로 나눌 수 있는데 키 프레임은 H.264와 같은 기존 부호화 시스템의 intraframe 부호화 기법으로 부호화된다³⁾. 또

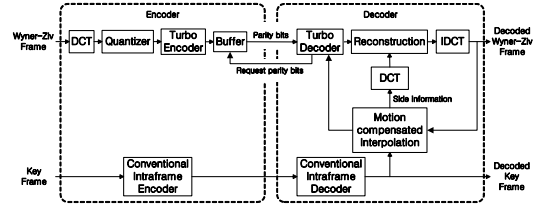


그림 1. 분산 동영상 부호화 시스템의 구조

한 Wyner-Ziv 프레임은 부호기에서 4×4 블록 크기로 나누어 DCT 영역으로 변환되고 양자화를 하게 된다. 양자화를 거친 계수들은 각 계수별로 bitplane을 만들게 되는데 이때 양자화 테이블에 따라서 bitplane 수가 결정되며 이렇게 결정된 각 bitplane을 터보 부호기의 입력으로 하여 오류 정정 부호를 위한 패리티 비트를 생성한다. 이 패리티 비트를 Wyner-Ziv 비트라고 하고 이 Wyner-Ziv 비트들을 복호기로 전송한다. 복호기에서는 키 프레임들을 이용하여 부가정보를 생성하게 되는데 이렇게 생성된 부가정보와 부호기에서 전송한 Wyner-Ziv 비트들을 이용하여 Wyner-Ziv 프레임을 복원한다.

Wyner-Ziv 비트를 생성하고 이 비트들을 이용하여 Wyner-Ziv 프레임을 복호화하기 위한 채널부호로 Shannon 한계에 근접한 성능을 보이는 터보 부호나 LDPC 부호가 사용되는데 본 논문에서는 성능평가를 위하여 터보 부호를 사용하였다. 터보 부호의 복호를 위해서는 trellis도의 각 상태(state)와 가지(branch)에 대한 확률인 $\alpha_k(S_k)$, $\beta_k(S_k)$, $\gamma_k(S_{k-1}, S_k)$ 를 계산해야 한다¹¹⁾. 먼저 다음 식 (1)에 의해서 $\gamma_k(S_{k-1}, S_k)$ 을 구하고 이를 이용하여 $\alpha_k(S_k)$, $\beta_k(S_k)$ 의 값을 계산하게 된다.

$$\gamma_k(S_{k-1}, S_k) = p(u_k)p(y_k|u_k) \quad (1)$$

이때 u_k 는 각 bitplane의 비트를 나타내고 y_k 는 복호기에서 복원된 부가정보이며 S_k 는 상태를 의미한다. 터보 부호의 복호를 위해서는 식 (1)의 $p(y_k|u_k)$ 를 계산해야 하는데 먼저 $p(u_k|y_k)$ 를 구하고 Bayesian rule을 이용하여 $p(y_k|u_k)$ 를 계산할 수 있는데 y_k 는 비트 단위의 값이 아닌 각 변환 계수의 추정치이고 u_k 는 비트 단위의 값이므로 $p(u_k|y_k)$ 를 계산하기 위해서는 원 Wyner-Ziv 프레임과 부가정보간의 차이의 확률분포가 다음과 같은 Laplacian 분포를 이룬다는 사실을 이용하여 확률을 계산해야 한다.

$$f(X) = \frac{\alpha}{2} \exp[-\alpha|X|], \quad (2)$$

$$X = WZ(i,j) - SI(i,j)$$

여기서 $WZ(i,j)$ 와 $SI(i,j)$ 는 각각 Wyner-Ziv 프레임과 부가정보의 변환 계수 값을 나타낸다. 또한 i, j 는 변환 계수 값의 위치를 나타낸다. 각 bitplane에 대한 $p(u_k|y_k)$ 를 구하는 방법을 그림 2에 도시하였다.

그림 2에서는 4개의 bitplane을 사용하는 경우 MSB부터 비트 1010에 대한 확률을 계산하는 경우를 도시하였는데 빗금 친 구간에 대한 확률 밀도를 적분함으로써 각 비트에 대한 $p(u_k|y_k)$ 값을 계산 할 수 있다.

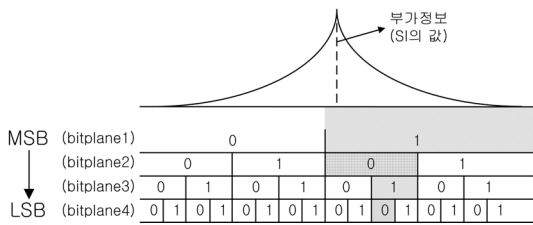


그림 2. 부가정보와 Laplacian 분포를 이용한 계산방법 (Bitplane 수가 4이고 비트 1010에 해당하는 확률을 계산 하는 경우)

III. 부가정보 생성 기법

3.1 기존의 방법

분산 동영상 부호화 시스템에서 부가정보는 키 프레임들을 이용하여 생성한다. 일반적으로 움직임 보상 보간 기법을 이용하여 부가정보를 생성하는데 그 중에서 대표적인 방법을 소개한다. 먼저 순방향 움직임 추정 방법에서는 그림 3에 도시한 것과 같이 이후 키 프레임에서 이전 키 프레임으로 움직임을 추정하면 현재 삽입 프레임에 해당하는 블록은 움직임 추정으로 생성된 벡터의 중간에 위치하게 된다고 가정한다^[2]. 이때 현재 삽입 프레임에서는 그림 3에서 도시한 것과 같이 블록 간 겹치거나(overlapped) 빈 공간(hole)이 생길 수 있다.

이런 문제를 해결하기 위해 순방향 움직임 추정 방법을 통해 생성된 벡터 중 삽입 프레임의 블록과 가장 가까이 지나가는 벡터를 블록의 가운데로 그대로 가져와 현재 삽입 프레임의 블록을 채우게 된다.

대칭 양방향 움직임 추정 방법에서는 중간 삽입 프레임의 블록을 가운데에 두고 이전 키 프레임과 이후 키 프레임에서 대칭적으로 움직임을 추정 방법을 사용한다^[8].

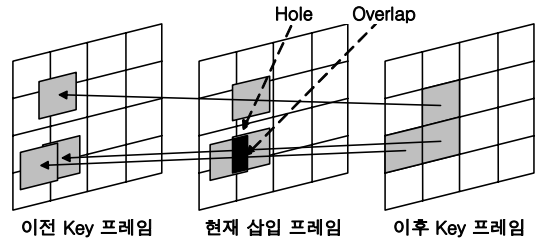


그림 3. 순방향 움직임 추정 방법

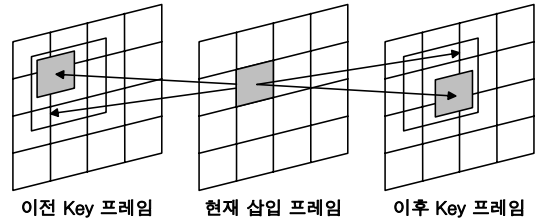


그림 4. 대칭 양방향 움직임 추정 방법

순방향 움직임 추정 방법으로 벡터를 생성하고 대칭 양방향 움직임 추정 방법을 통하여 좀 더 세밀하게 움직임을 추정을 하게 되면 전체 성능을 높일 수 있게 된다^[4-6]. 이렇게 찾은 움직임 벡터 중에서 오류가 있을 수 있는데 이를 가중치 벡터 중간 값 필터를 이용하여 보정하게 되면 전체 성능을 높일 수 있다. 가중치 벡터 중간 값 필터는 식 (3)에서 나타난 것과 같이 현재 블록과 주위 블록의 벡터들을 사용하여 현재의 블록의 벡터를 보정하는 기법인데 벡터간의 거리와 가중치를 이용하여 가중치 벡터 중간 값으로 현재 블록의 벡터를 보정한다^[7].

$$\sum_{i=1}^N w_i \|v_m - v_i\| \leq \sum_{i=1}^N w_i \|v_j - v_i\|, \quad j=1,2,\dots,N,$$

$$w_i = \frac{MSE(v_i)}{MSE(v_j)} \quad (3)$$

여기서 v_m 은 가중치 벡터 중간 값 필터를 이용하여 보정된 벡터이고 $\|v_i - v_j\|$ 는 v_i 와 v_j 간의 거리를 나타낸다. 또한 w_i 는 가중치인데 v_i 로 이전 키 프레임과 이후 키 프레임에서 가져온 블록간의 MSE 와 v_j 로 가져온 블록간의 MSE 를 이용하여 가중치를 구하게 된다.

순방향 혹은 대칭 양방향 움직임 추정 방법 이외의 대표적인 부가정보 생성 방법 중 Peixoto 등이 제안한 방법은 순방향 움직임 추정 방법과 역방향 움직임 추정 방법을 사용하여 중간 삽입 프레임을 각각 생성하

여 그 두 개의 중간 삽입 프레임의 평균을 구하는 방법이다. 그림 5에 Peixoto 등이 제안한 중간 삽입 프레임을 생성하는 방법을 도시하였다⁹⁾. 이때 역시 빈 공간이 생길 수 있는데 중간 삽입 프레임의 평균을 구함으로써 많은 부분의 빈 공간을 채울 수 있다. 아직 남은 빈 공간은 일정 크기의 마스크를 씌워서 쌍방향 움직임 추정(bidirectional motion estimation) 방법을 사용하여 채우게 된다.

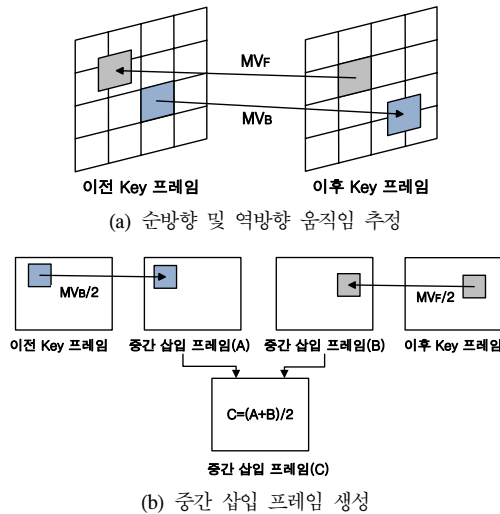


그림 5. Peixoto 등이 제안한 중간 삽입 프레임 생성 방법

3.2 제안하는 방법

본 논문에서 제안하는 부가정보 생성 방법은 두 가지이다. 첫 번째 방법은 순방향 움직임 추정 방법으로 벡터를 생성 하고 대칭 양방향 움직임 추정 방법으로 움직임 벡터를 좀 더 세밀하게 찾는다, 다시 역방향 움직임 추정 방법으로 움직임 벡터를 생성 하고 이렇게 생성한 움직임 벡터를 대칭 양방향 움직임 추정 방법으로 세밀하게 찾는다. 이렇게 하면 블록 당 2개의 움직임 벡터가 생기는데 이렇게 생성된 움직임 벡터를 보정하기 위해 가중치 중간 값 필터를 사용하게 된다. 이때 보정하고자 하는 블록을 포함한 주위 블록들의 움직임 벡터를 가지고 가중치 중간 값 필터를 적용하여 블록 당 하나의 벡터를 선택하게 되고 이 벡터를 사용해 하나의 중간 프레임을 만들어 내는 방법이다. 두 번째 방법은 첫 번째 방법과 마찬가지로 순방향으로 움직임 추정을 하고 대칭 양방향 움직임 추정 방법으로 움직임 벡터를 좀 더 세밀하게 찾는다. 그 다음에 가중치 중간 값 필터를 사용하여 하나의 중간 프레임을 만들어 낸다. 또한 역방향 움직임 추정 방법으로

움직임 벡터를 생성한 다음 대칭 양방향 움직임 추정 방법으로 다시 세밀하게 찾는다. 역시 가중치 중간 값 필터를 사용하여 움직임 벡터를 보정한 다음 또 다른 하나의 중간 프레임을 만든다. 이렇게 생성한 두 개의 중간 프레임의 평균을 내게 된다. 이러한 방법들은 기존의 방법들을 응용한 것으로서 순차적으로 기존의 방법들을 사용함으로써 성능을 향상시킬 수 있다. 특히 Peixoto 등이 제안한 방법과 기존의 방법들을 순차적으로 적용한 방법인 순방향 및 역방향으로 움직임 추정을 하고 대칭 양방향 움직임 추정 방법으로 세밀하게 찾은 다음 가중치 중간 값 필터를 사용해 보정하여 생성한 두 개의 중간 프레임의 평균을 내는 방법은 기존의 방법보다 더 좋은 결과를 얻었다. 이에 대한 모의실험 결과를 V장에 제시한다.

부가정보를 생성하기 위해 이전 키 프레임과 이후 키 프레임을 사용하는데 잡음으로 인한 움직임 추정 오류를 방지하기 위하여 저역 통과 필터(LPF)를 사용할 수 있다. 또한 움직임 추정 시에 거리에 대한 가중치를 두어 먼 거리에 있는 비슷한 블록을 잘못 선택할 확률을 낮추는 방법을 사용할 수 있다. 이렇게 저역 통과 필터를 사용하거나 거리에 대한 가중치를 사용하면 더 좋은 결과를 얻을 수 있다¹⁰⁾.

IV. 부가정보 생성 기법의 하드웨어 복잡도

부가정보를 생성하는 방법을 하드웨어적인 관점에서 고려해 보면 두 개의 프레임 메모리에서 영상의 일부를 각각 읽어 그 두 블록의 차분의 합을 구하는 방법으로 움직임 예측을 하게 된다. 이때 차분의 합을 구하는 과정에서 필요한 덧셈기에 의한 계산량은 많지 않으나 반복적인 메모리 접근에 필요한 계산량이 계산시간의 많은 부분을 차지하게 된다. 따라서 순방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법의 계산량은 움직임 벡터의 탐색영역의 크기와 밀접한 관계를 갖게 된다.

먼저 이전 키 프레임과 이후 키 프레임 사이에 하나의 중간 삽입 프레임을 생성하는 경우 순방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법이 두 키 프레임 간에 같은 크기의 움직임 벡터를 찾는다고 가정 하면 순방향 움직임 추정 방법이 대칭 양방향 움직임 추정 방법에 비하여 4배 큰 탐색영역을 가져야 한다. 그러나 대칭 양방향 움직임 추정 방법은 양쪽 키 프레임에 모두 탐색영역이 있으므로 실제 움직임 예측을 위한 메모리 접근에 의한 계산량은 순방향 움직임 추정 방법이 대칭 양방향 움직임 추정 방법에 비

하여 2배가 된다. 더욱이 순방향 움직임 추정 방법은 움직임 예측에 의해 채워지지 않는 영역이 발생할 수 있으므로 실제 계산량은 좀 더 늘어난다고 볼 수 있다. 반면 이전 키 프레임과 이후 키 프레임 사이에 하나 이상의 중간 삽입 프레임을 생성하는 경우에는 대칭 양방향 움직임 추정 방법이 순방향 움직임 추정 방법에 비해 계산량이 급격히 늘어나는 것을 확인할 수 있다. 순방향 움직임 추정 방법은 일단 최초의 두 키 프레임간의 움직임 벡터를 찾고 나면 몇 개의 중간 삽입 프레임을 생성하든 시간적 거리의 비를 적용하는 것 외에 추가적인 계산이 필요 없다. 하지만 대칭 양방향 움직임 추정 방법의 경우에는 생성하고자 하는 중간 삽입 프레임의 두 키 프레임과의 거리가 바뀌면 움직임 벡터의 상대적 크기가 바뀌므로 탐색영역이 비대칭으로 되어 한쪽이 확장되게 된다. 즉, 생성하고자 하는 중간 삽입 프레임의 시간적인 위치가 바뀌면 모든 계산을 다시 하게 되므로 생성하고자 하는 중간 삽입 프레임의 개수만큼 계산량이 증가하게 된다. 순방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법을 함께 사용하는 방법일 경우는 상황에 따라 계산량이 달라지므로 계산량을 정의하기는 어려우나 순방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법의 중간 정도의 계산량을 가지게 될 것이다. 하지만 양쪽 모두를 동시에 지원해야 하므로 하드웨어의 양에 있어서는 가장 커지는 문제가 있다.

V. 모의실험

부가정보를 생성하는 방법이 분산 동영상 부호화의 성능에 미치는 영향을 확인하기 위해 다음과 같은 분산 동영상 부호화 시스템을 사용하였다. 먼저 Wyner-Ziv 프레임은 부호기에서 4×4 블록 크기로 DCT 영역으로 변환된 후에 그림 6에 제시한 8개의 양자화 테이블 중 하나를 사용하여 양자화를 하게 된다^{11,6)}. 이를 통해 bitplane을 생성한 후에 각 bitplane에 대해서 터보 부호화를 하여 Wyner-Ziv 비트를 생성하게 된다.

복호기에서 생성된 부가정보 역시 4×4 블록 크기의 DCT 영역으로 변환되고 이렇게 변환된 부가정보와 부호기에서 전송한 Wyner-Ziv 비트를 이용해 Wyner-Ziv 프레임을 복원하게 된다. 모의실험을 위한 부가정보 생성 방법으로는 순방향 움직임 예측 방법, 대칭 양방향 움직임 예측 방법, DISCOVER 시스템에서 사용한 순방향 움직임 예측 방법으로 벡터를 생성한 다음 대칭 양방향 움직임 예측 방법을 통해 세밀하

16	8	0	0	32	8	0	0	32	8	4	0	32	16	8	4
8	0	0	0	8	0	0	0	8	4	0	0	16	8	4	0
0	0	0	0	0	0	0	0	4	0	0	0	8	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
32	16	8	4	64	16	8	8	64	32	16	8	128	64	32	16
16	8	4	4	16	8	8	4	32	16	8	4	64	32	16	8
8	4	4	0	8	8	4	4	16	8	4	4	32	16	8	4
4	4	0	0	8	4	4	0	8	4	4	0	16	8	4	0

그림 6. 양자화를 위한 8가지 테이블

게 움직임 벡터를 찾고 가중치 벡터 중간 값 필터를 사용한 방법과 Peixoto 등이 제안한 순방향과 역방향 움직임 예측을 통해 각각 만들어진 중간 프레임을 합친 다음 빈 공간을 마스크를 이용해 채우는 방법 등을 사용하였다^{4,9)}. 실험결과 순방향 및 역방향 움직임 추정 후 중간 삽입 프레임을 생성할 때 한 키 프레임의 값들만으로 중간 삽입 프레임을 생성하는 것 보다 이전 및 이후 키 프레임들을 이용하여 더 좋은 결과를 얻을 수 있다. 순방향과 역방향 움직임 예측을 위한 블록 크기는 16×16으로 하였으며 움직임 탐색 영역은 48×48로 하였다. 대칭 양방향 및 가중치 벡터 중간 값 필터를 위한 블록 크기는 8×8로 하였다. 또한 선택적으로 저역 통과 필터를 사용하였다. 실험 결과의 신뢰성을 높이기 위해서 QCIF 규격 Coastguard 300 프레임, Foreman 400 프레임, Hall monitor 300 프레임, Stefan 100 프레임의 표준 동영상들 각각 사용하였으며 Coastguard 영상과 Stefan 영상의 프레임율은 25Hz이고 Foreman 영상과 Hall monitor 영상의 프레임율은 30Hz이다. 또한 각 부가정보 생성 방법들로 만들어진 부가정보와 원 영상간의 PSNR은 식 (4)을 이용하여 구할 수 있다.

$$PSNR = 10 * \log_{10} (255^2 / MSE) \text{ dB},$$

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |u(m,n) - u'(m,n)|^2 \quad (4)$$

이때 $u(m,n)$ 는 원 영상의 화소 값, $u'(m,n)$ 는 부가정보의 화소 값을 나타낸다.

5.1 각 부가정보 생성 기법의 성능 분석

표 1은 각 영상에 대해서 여러 방법으로 생성된 부가정보와 원 영상간의 평균 PSNR 결과이다. 표 1에서 순방향, 역방향, 쌍방향 움직임 예측 방법 2는 기존에 소개한 방법 1에서 블록을 나누어 좀 더 세밀하게 찾는 방법으로 처음 블록 크기가 16×16일 때 순방향

표 1. 다양한 부가정보 생성방법에 따라 생성된 부가정보와 원 영상간의 평균 PSNR 결과

생성 방법 영상(QCIF)	Forward	Bilateral	LPF+ Forward+ Bilateral+ WVMF	Forward+ Bilateral+ WVMF	Forward+ Backward+ Bidirectional 1	Forward+ Backward+ Bidirectional 2	LPF+ Forward+ Backward+ Bilateral+ WVMF 1	LPF+ Forward+ Backward+ Bilateral+ WVMF 2
Coastguard	37.04	35.82	38.62	38.62	37.02	38.45	38.63	38.63
Foreman	34.93	29.62	36.57	36.56	35.67	36.50	36.62	36.73
Hall monitor	40.20	39.20	40.59	40.58	40.31	40.66	40.58	40.59
Stefan	28.93	24.29	30.32	30.30	29.57	30.46	30.26	30.64
전체 평균	36.41	33.45	37.67	37.66	36.76	37.63	37.68	37.76

및 역방향 움직임 추정 후 블록 크기를 8×8로 세밀하게 나눈 다음 이전에 찾은 벡터의 주위를 순방향 및 역방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법으로 좀 더 세밀하게 다시 찾는 방법이다. 저역 통과 필터, 순방향, 역방향, 대칭 양방향 움직임 예측 및 가중치 벡터 중간 값 필터 방법 1은 본 논문에서 제안한 첫 번째 방법으로 순방향 및 역방향 움직임 추정 후 각각 대칭 양방향 움직임 추정을 하고 가중치 벡터 중간 값 필터를 적용하여 움직임 벡터를 보정 할 때 순방향과 역방향으로 각각 움직임 벡터를 추정하여 그 벡터들의 주위 벡터들과 함께 가중치 벡터 중간 값 필터 방법을 사용하여 중간 프레임을 생성하는 방법이다. 본 논문에서 제안한 두 번째 방법인 저역 통과 필터, 순방향, 역방향, 대칭 양방향 움직임 예측 및 가중치 벡터 중간 값 필터 방법 2는 Peixoto 등이 제안한 방법을 변형한 방법으로 순방향 및 역방향으로 움직임 추정 후 각각 대칭 양방향 및 가중치 벡터 중간 값 필터를 수행하여 두 개의 중간 프레임을 만든 다음 합치는 방법이다.

5.2 터보 부호의 부호화율 제어 방법

또한 터보 부호기는 한 메시지 비트 당 두 비트의 패리티 비트가 생성되는데 생성된 패리티 비트는 패리티 비트율을 조절하기 위해 puncturing을 하게 되고 분산 동영상 부호화 시스템의 터보 복호기에서 패리

티 비트를 요구할 때 부호기에서 패리티 비트를 전송하게 된다. 본 논문에서는 48비트당 한 비트씩 패리티 비트를 증가시키면서 부호화율 제어를 하게 되는데 이때 좀 더 정교하게 패리티 비트 전송율을 제어하기 위해 두 RSC 부호기에 대한 패리티 비트를 동시에 전송하는 방법과 한 비트 씩 순차적으로 전송하여 복호기에서 복호화함으로써 패리티 비트율을 정밀하게 제어하는 방법의 성능을 비교하였다. 표 2에 두 RSC 부호기의 패리티 비트를 한번에 전송하는 방법과 순차적으로 전송하는 방법 간의 동일한 PSNR 결과를 얻기 위한 패리티 비트 전송율을 비교하였다.

5.3 부가정보 생성 기법이 분산 동영상 부호화 시스템의 성능에 미치는 영향 분석

복호기에서 생성된 부가정보가 분산 동영상 부호화 시스템의 성능에 미치는 영향을 분석하기 위해서 각 영상에 대해서 다양한 방법으로 만들어진 부가정보와 부호기에서 전송된 Wyner-Ziv 비트들을 가지고 복원된 Wyner-Ziv 프레임과 원 영상간의 평균 PSNR을 비교하였다. 또한 기존 영상 부호화 방법과의 성능을 비교하기 위하여 H.264 intraframe 부호화 방식의 평균 PSNR 결과도 함께 도시하였다. 그림 7은 Coastguard 영상을 분산 동영상 부호화 시스템에 적용시켰을 때의 전송된 Wyner-Ziv 비트율에 대한 Wyner-Ziv 영상과 원 영상간의 평균 PSNR 결과이다.

표 2. 동일한 PSNR 결과에서 패리티 전송 방법에 따른 패리티 전송율 비교 결과(Q=8)

영상(QCIF) 패리티 전송 방법	Coastguard	Foreman	Hall_monitor	Stefan
RSC 부호기에 대한 패리티 비트를 한번에 전송 (bits/Pixels)	0.81055	1.05991	0.47835	1.39413
RSC 부호기에 대한 패리티 비트를 순차적 전송 (bits/Pixels)	0.77577	1.02277	0.44133	1.35639
전송율 차이 (%)	3.478	3.714	3.702	3.774

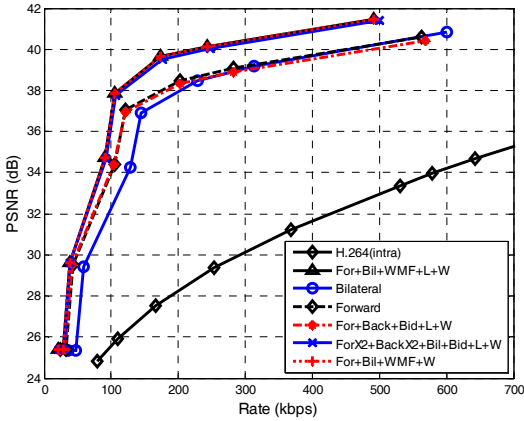


그림 7. 부가정보 생성 기법에 따른 복원된 Wyner-Ziv 영상과 원 영상간의 평균 PSNR 결과 (Coastguard)

부가정보와 원 영상간의 평균 PSNR이 높으면 만들어진 Wyner-Ziv 영상을 복원 할 때 적은 Wyner-Ziv 비트로도 높은 PSNR 결과를 보이는 것을 알 수 있는데 특히 순방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법으로 움직임 벡터를 생성하고 가중치 중간 값 필터를 사용하여 보정하는 방법과 순방향, 역방향, 쌍방향 움직임 예측 방법 2가 그 이외의 방법보다 더 좋은 결과를 보임을 알 수 있다. 또한 실험 결과를 통해 지역 통과 필터가 분산 동영상 부호화 시스템에 미치는 영향이 크지 않음을 확인 할 수 있다. 그림 8은 Foreman 영상을 분산 동영상 부호화 시스템에 적용시켰을 때의 전송된 Wyner-Ziv 비트율에 대한 Wyner-Ziv 영상과 원 영상간의 평균 PSNR 결과이다.

그림 9는 Hall monitor 영상에 대한 평균 PSNR 결과이다. Hall monitor 영상 역시 앞의 영상들과 같이

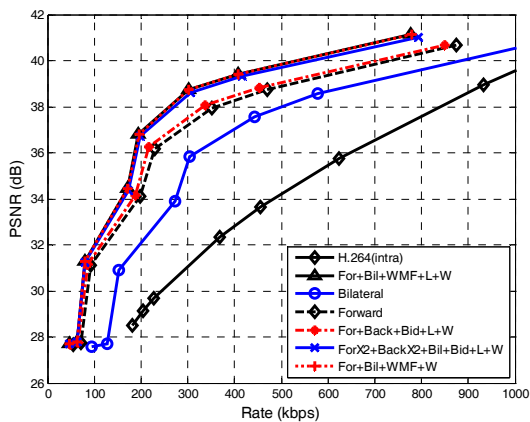


그림 8. 부가정보 생성 기법에 따른 복원된 Wyner-Ziv 영상과 원 영상간의 평균 PSNR 결과 (Foreman)

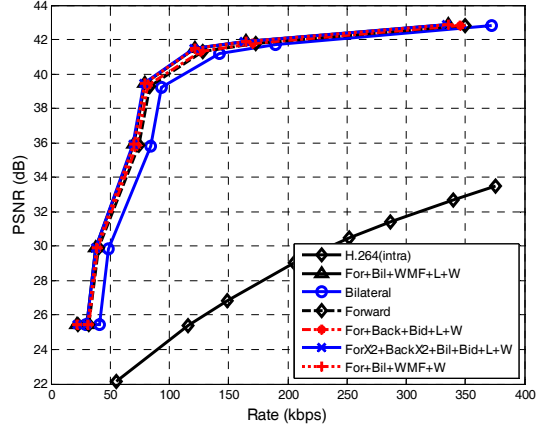


그림 9. 부가정보 생성 기법에 따른 복원된 Wyner-Ziv 영상과 원 영상간의 평균 PSNR 결과 (Hall monitor)

높은 PSNR 결과를 보이나 Hall monitor 영상이 움직임이 적은 영상이기 때문에 다른 영상보다 부가정보 생성 방법에 따라 평균 PSNR 결과가 크게 차이나지 않은 것을 볼 수 있다. 움직임이 큰 영상인 Stefan의 경우 순방향 움직임 추정 방법과 대칭 양방향 움직임 추정 방법으로 움직임 벡터를 생성하고 가중치 중간 값 필터를 사용하여 보정하는 방법과 순방향, 역방향, 쌍방향 움직임 예측 방법 2가 그 이외의 방법보다 더 좋은 결과를 보임을 알 수 있다. 하지만 대칭 양방향 움직임 추정 방법으로도 움직임 벡터를 생성하였을 경우 움직임이 큰 영상에서 부가정보를 잘 생성하지 못하여 다른 부가정보 생성 방법들과 비교하여 가장 안 좋은 결과를 보이는 것을 볼 수 있다. 이는 Wyner-Ziv 영상을 복원하기 위해 생성한 부가정보가 잘 생성되어 원 영상과 비슷하게 되면 적은

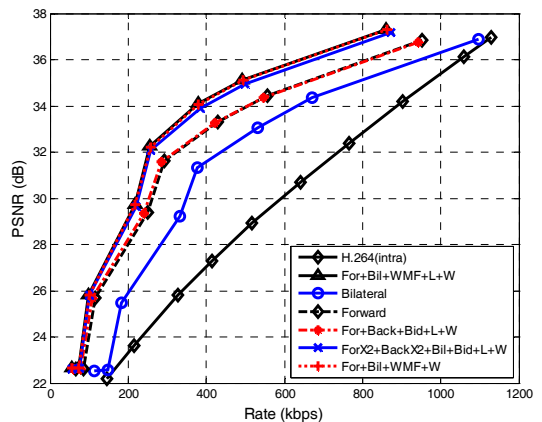


그림 10. 부가정보 생성 기법에 따른 복원된 Wyner-Ziv 영상과 원 영상간의 평균 PSNR 결과 (Stefan)

Wyner-Ziv 비트만으로도 원 영상을 복원 할 수 있음을 의미한다.

각 부가정보 생성 기법들 간의 성능을 비교해 볼 때 다양한 부가정보 생성 기법들을 이용하여 생성된 부가정보가 원 Wyner-Ziv 프레임간의 상관도가 높을 수록 같은 Wyner-Ziv 비트율에서 더 높은 PSNR 결과를 얻을 수 있음을 실험한 모든 영상에서 볼 수 있다. 특히 순방향 및 대칭 양방향 움직임 추정 방법으로 움직임 벡터를 생성하고 가중치 중간 값 필터를 사용하여 보정하는 방법과 순방향, 역방향, 쌍방향 움직임 예측 방법 2가 다른 부가정보 생성 방법들보다 더 좋은 PSNR 결과를 보임을 알 수 있다. 또한 H.264 intraframe 부호화 방법과 비교함으로써 기존의 부호화 방식과의 차이를 확인 할 수 있다.

VI. 결 론

본 논문에서는 분산 동영상 부호화 시스템에서 다양한 부가정보 생성 방법을 통해 생성된 부가정보가 분산 동영상 부호화 시스템의 성능에 미치는 영향을 비교 분석 하였다. 또한 터보 부호의 부호화율 제어를 위한 성능평가를 통해서 정교한 부호화율 제어기법을 선택하여 사용하였다. 모의실험을 통하여 비교적 복잡도가 간단한 순방향이나 대칭 양방향 움직임 추정 방식보다 상대적으로 복잡도가 높아지지만 순방향 움직임 추정으로 움직임 벡터를 찾고 대칭 양방향 움직임 추정 방식으로 좀 더 세밀하게 움직임 추정을 한 다음 가중치 벡터 중간 값 필터를 이용하여 움직임 벡터를 보정하는 것과 같이 다양한 움직임 추정 방법과 움직임 벡터 보정 방법 등을 사용해 순차적으로 생성된 움직임 벡터를 보정하였을 때 더 좋은 성능을 보이는 것을 확인하였으며 복잡도를 줄이기 위해 저역 통과 필터를 사용하지 않아도 분산 동영상 시스템의 성능에 큰 영향을 미치지 않음을 확인하였다.

참 고 문 헌

[1] T. Wiegand, G. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Syst. for Video Technol.*, Vol.13, pp.560-576, July 2003.

[2] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proc. IEEE*, Vol.93, No.1, pp.71-83, Jan. 2005.

[3] A. Aaron, S. Rane, E. Setton and B. Girod, "Transform-Domain Wyner-Ziv Codec for Video," *Proc. SPIE Visual Communications and Image Processing*, San Jose, CA, pp.520-528, Jan. 2004.

[4] J. Ascenso, C. Brites and F. Pereira, "Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding," *5th EURASIP, Slovak Republic*, pp.176-181, July 2005.

[5] X. Artigas, J. ascenso, M. Dalai, D. Kubasov, and M. quaret, "The DISCOVER codec: architecture, techniques and evaluation," *Picture Coding Symposium*, pp.1103-1120, Nov. 2007.

[6] C. Brites, J. Ascenso and F. Pereira, "Improving transform domain Wyner-Ziv video coding performance," *IEEE ICASSP*, Toulouse, France, pp.II-525-II-528, May, 2006.

[7] L. Alparone, M. Barni, F. Bartolini, V. Cappellini, "Adaptively Weighted Vector-Median Filters for Motion Fields Smoothing," *IEEE ICASSP*, Georgia, USA, Vol.4, pp.2267-2270, May 1996.

[8] B. D. Choi, J. W. Han, C. S. Kim, S. J. Ko, "Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Black Motion Compensation," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.17, pp.407-416, April 2007.

[9] E. Peixoto, R. L. Queiroz and D. Mukherjee, "On side information generation for Wyner-Ziv video coding," in *Proc. of Simposio Brasileiro de Tel.*, Rio de Janeiro, Brazil, Sept. 2008.

[10] J. Ascenso, C. Brites and F. Pereira "Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity," *IEEE ICIP*, Atlanta, USA, pp.605-608, Oct. 2006.

[11] C. Berrou, A. Glacieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. on Comm.*, Vol.44, pp.1261-1271, oct 1996.

문 학 수 (Hak-soo Moon)

준회원



2009년 2월 가톨릭대학교 정보통신전자공학부 학사 졸업
2010년 3월~현재 가톨릭대학교 정보통신전자공학과 석사과정
<관심분야> 영상통신, 영상처리

이 성 원 (Seong-won Lee)

정회원



1988년 서울대학교 제어계측공학과(공학사).
1990년 서울대학교 제어계측공학과 석사졸업
2003년 University of Southern California 전기공학과 박사
현재 광운대학교 전자정보공과대학 컴퓨터공학과 교수
<관심분야> 2D/3D 영상 신호처리, ASV, ASIP 및 SoC설계, Power-Aware Computing

이 창 우 (Chang-woo Lee)

중신회원

현재 가톨릭대학교 정보통신전자공학과 교수
<관심분야> 영상통신, 영상처리