

논문 2008-45SP-1-2

센서네트워크상의 응용을 위한 터보 복호화 오류정정 기법을 이용한 경량화 비디오 부호화 방법

(Low Complexity Video Encoding Using Turbo Decoding Error Concealments for Sensor Network Application)

고 봉 혁*, 심 혁 재*, 전 병 우**

(Bonghyuck Ko, Hiuk Jae Shim, and Byeungwoo Jeon)

요 약

종래의 움직임보상 변환 부호화 기술은 부호화기가 복호화기에 비해 매우 복잡한 구조를 갖는다. 하지만 센서네트워크와 같은 에너지 제한 환경을 위한 경량화 부호화기의 필요성이 대두됨에 따라 부호화기 복잡도와 에너지소비의 대부분을 차지했던 움직임 예측/보상과정을 없애기 위한 새로운 부호화 구조에 대한 연구가 이루어져 왔다. Wyner-Ziv 코딩 기술은 이를 가능하게 하는 대표적인 기술로서 터보 코드와 같은 채널 코드를 이용하여 프레임과 보조정보 사이의 잡음을 제거하여 영상을 복원한다. 이때 부호화기는 단순히 현재 프레임에 대한 패리티 정보만을 생성할 뿐 프레임간의 유사성을 이용하는 어떠한 과정도 수행하지 않기 때문에 매우 간단한 구조를 갖게 된다. 하지만 Wyner-Ziv 코딩 구조에서는 잡음이 많은 보조영상을 이용하여 복호화 할 경우 터보 코드의 복호화 오류가 발생한다. 이러한 복호화 오류는 특히 영상 간 움직임이 많거나 occlusion이 존재하는 경우 더 많이 발생하여 마치 복원된 영상에 Salt & Pepper 같은 잡음이 나타난다. 이러한 잡음은 비록 그 발생빈도는 적지만 복원된 영상의 주관적인 화질을 상당히 떨어뜨린다. 본 논문은 심볼단위의 Wyner-Ziv 코딩구조하의 초경량 부호화 기술과, 잘못된 필터 적용으로 영상의 texture를 손상시키는 경우를 최소화하기 위하여 복호화 시 각 화소에 터보 코드 복호화 오류가 있는지 여부를 판단하여 선택적으로 미디언 필터를 적용시키는 기술을 제안한다. 제안된 방법은 종래의 움직임보상 변환 부호화 기술과 비교하여 현저하게 연산량이 감소된 경량화 부호화 기술로서 터보 코드 복호화 오류로 발생하는 잡음과 영상의 texture를 구분하여 필터를 적용함으로써 복원된 영상의 주관적인 화질과 PSNR을 동시에 개선한다. 실험결과 PSNR의 경우 평균 최고 약 0.8dB에 달하는 성능이득 효과를 보였다.

Abstract

In conventional video coding, the complexity of encoder is much higher than that of decoder. However, as more needs arises for extremely simple encoder in environments having constrained energy such as sensor network, much investigation has been carried out for eliminating motion prediction/compensation claiming most complexity and energy in encoder. The Wyner-Ziv coding, one of the representative schemes for the problem, reconstructs video at decoder by correcting noise on side information using channel coding technique such as turbo code. Since the encoder generates only parity bits without performing any type of processes extracting correlation information between frames, it has an extremely simple structure. However, turbo decoding errors occur in noisy side information. When there are high-motion or occlusion between frames, more turbo decoding errors appear in reconstructed frame and look like Salt & Pepper noise. This severely deteriorates subjective video quality even though such noise rarely occurs. In this paper, we propose a computationally extremely light encoder based on symbol-level Wyner-Ziv coding technique and a new corresponding decoder which, based on a decision whether a pixel has error or not, applies median filter selectively in order to minimize loss of texture detail from filtering. The proposed method claims extremely low encoder complexity and shows improvements both in subjective quality and PSNR. Our experiments have verified average PSNR gain of up to 0.8dB.

Keywords : DSC, DVC, Wyner-Ziv coding, Turbo decoding error, Selective median filter, Refinement

* 학생회원, **정회원, 성균관대학교 정보통신공학부

(School of Information and Communication Engineering, Sungkyunkwan University)

* 이 논문은 2005년도 교육인적자원부의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임

(KRF-2005-041-D00557)

접수일자: 2007년10월4일, 수정완료일: 2007년12월31일

I. 서 론

MPEG-1/2/4와 H.26X 같은 기존의 비디오 압축 기술은 부호화기가 복호화기에 비해 복잡도가 상당히 높은 구조로 이루어져 있다. 이는 부호화기가, 이웃한 프레임들 간에 존재하는 상관도를 이용하는 움직임 예측(ME : Motion Estimation)과 움직임 보상(MC : Motion Compensation)을 통하여 중복성(Redundancy)을 제거하며 이에 소요되는 연산량이 상당하기 때문이다. 따라서 이와 같은 기존의 비디오 압축 기술은 부호화기에 허용되는 연산능력과 전력소모량이 극히 제한적인 센서 네트워크와 같은 응용환경에서는 사용할 수 없다.

Distributed Source Coding (DSC)과 Distributed Video Coding (DVC)은 바로 이러한 부호화기의 복잡도 문제를 해결하기 위한 방법의 하나로 최근 각광을 받고 있다. 이는 상관도(Correlation)가 있는 소스들을 각각 독립적으로 부호화하여도 복호화를 서로 연계해서 한다면, 각 소스를 같이 (예측) 부호화하여 얻을 수 있는 부호화 이득을 동일하게 얻을 수 있다는 것을 수학적으로 증명한 Slepian-Wolf 이론에 근거한다^[1~2]. 비디오 압축 기술의 구현관점에서 봤을 때 이는 프레임 간 중복성을 줄이기 위해 부호화기에서 행하였던 모든 처리절차를, 부호화 이득에 별다른 손실 없이 복호화기 쪽으로 옮길 수 있음을 의미하므로, 센서네트워크와 같은 새로운 환경에 대응하는 초경량화 부호화가 가능해진다.

현재 경량 부호화를 위한 한 방법으로서의 DVC 기술에 관한 연구는 크게 두 가지의 흐름으로 요약 할 수 있다. 하나는 미국의 스탠포드 대학과 유럽의 공동연구 그룹인 DISCOVER 그룹 등에서 활발히 연구 중인 Wyner-Ziv 코딩 기술이며 나머지 하나는 미국 버클리

대학의 PRISM 기술이다^[3~4]. 전자는 복호화기에서 현재 프레임과 이웃 프레임간의 유사성을 이용하여 현재 프레임에 대한 보조정보(Side Information)를 만들고, 이 보조정보와 원래 프레임간의 차이를 가상채널(Virtual Channel)의 잡음으로 간주한 후, 강력한 채널 코딩 복호화 기법을 적용하여 이 잡음을 제거하는 방식이다^[3]. 이 과정에서 부호화기는 단순히 현재 프레임에 대한 채널코딩 패리티 정보만을 생성하게 되므로 기존의 압축 기술과 같은 움직임 예측이나 보상 과정이 필요 없어 매우 간단한 부호화 구조를 갖게 된다. Wyner-Ziv 코딩은 크게 화소 영역에서 동작하는 PDWZ(Pixel Domain Wyner-Ziv Coding)와 변환 영역에서 동작하는 TDWZ(Transform Domain Wyner-Ziv Coding)로 나눌 수 있다.

후자인 PRISM 기술은 Wyner-Ziv 코딩과 구조적으로 매우 상이하지만, Wyner-Ziv 코딩은 I-B-I-B에 상응하는 반면, PRISM은 I-P-P-P에 상응하는 구조를 가진다. 따라서 현재 프레임에 대한 보조정보를 항상 전 프레임에서 움직임 탐색을 통하여 확보하게 되며, 보조정보의 정확한 생성을 위해 블록마다 HASH나 CRC등과 같은 블록 시그니처를 전송한다. 또한 영상의 부호화는 Multi-level Coset Code를 이용하는 것이 Wyner-Ziv 코딩과 구별된다^[4].

Wyner-Ziv 코딩 기술은 부호화기로부터 전송받은 패리티를 이용하여 보조정보에 있는 잡음을 제거하는 것을 복호화 과정으로 한다. 따라서 보조정보가 정교할수록 그리고 터보 코드의 오류정정 능력이 좋고 부호화기에서 전송해주는 패리티의 양이 충분할수록 더 나은 복호화 성능을 보인다. 하지만 반대로 보조정보를 잘못 만들거나 터보 코드의 오류정정 능력에 비해 패리티의

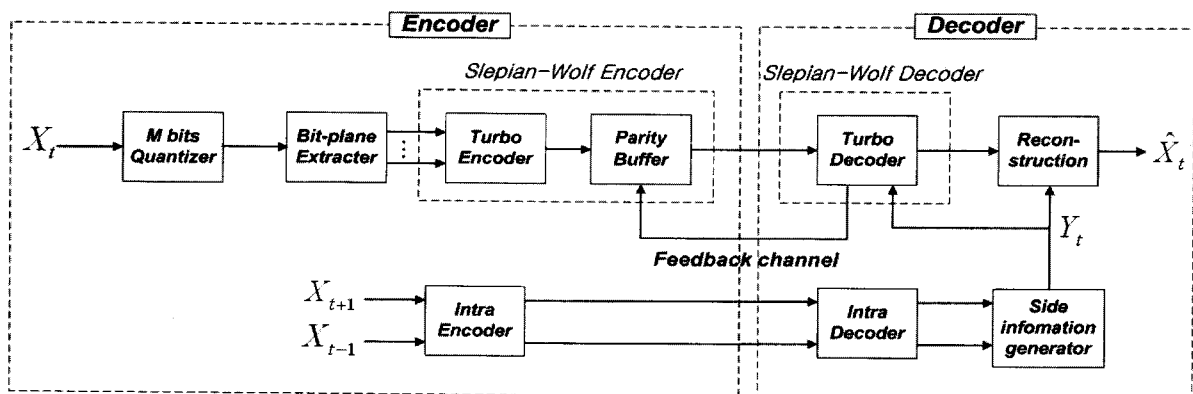


그림 1. 일반적인 Wyner-Ziv 코딩 구조
Fig. 1. General structure of the Wyner-Ziv coding.

양이 충분하지 않을 경우에는 터보 코드 복호화 오류가 발생한다. 이러한 터보 코드 복호화 오류는 트렐리스 (Trellis) 상에서 잘못된 천이과정을 최적의 천이과정으로 선택하게 하며 때로는 원래 화소 값과는 상이하게 다른 값으로 복호화하기도 한다. 이는 복원된 영상에 마치 Salt & Pepper와 같은 잡음을 발생시켜 주관적인 화질을 상당히 열화 시키는 주요 원인이 된다.

따라서 본 논문은 센서네트워크와 같은 새로운 환경에서 사용하기 위한 경량화 부호화기 구현을 위해, 화소영역 Wyner-Ziv 코딩 기술을 사용하되, 기존 방법의 문제점인 터보 코딩의 복호화 오류를 시공간 유사성 적용을 통하여 판별하고 보정하는 새로운 경량화 비디오 부호화 방법과 이에 상응하는 개선된 부호화 방법을 제안한다. 본 논문의 구성은 다음과 같다. 우선 II장에서는 화소영역 Wyner-Ziv (PDWZ) 코딩 기술의 전반적인 구조 및 복호화 과정에 관하여 설명한다. III장에서는 본 논문에서 제안하는 터보 코드 복호화 오류 판별 방법 및 보정방법에 관하여 언급하며, IV장에서는 실험 및 그 결과에 대해서 논의한다. 마지막으로 V장에서는 결론과 향후 연구과제에 대해서 논의한다.

II. 화소영역 Wyner-Ziv 비디오 부호화

본 장에서는 PDWZ의 구조와 부호화 및 복호화 과정에 대해서 설명한 후, 본 논문에서 제안한 PDWZ에 대해서 설명한다.

그림 1은 일반적인 PDWZ의 부호화 구조이다. 그림에서 알 수 있듯이 부호화기는 M 비트 양자화기와 비트 플레인 추출기, 터보 부호화기, 그리고 패리티 버퍼로 구성되어 있으며 복호화기는 터보 복호화기와 보조 정보생성기, 그리고 영상복원기로 이루어져 있다.

1. 부호화 및 복호화 과정

Wyner-Ziv 코딩에서는 프레임을 크게 두 가지로 분류한다. 하나는, 실제 Wyner-Ziv 코딩이 적용되는 프레임 (Wyner-Ziv 프레임으로 부름)으로서, 본 논문이 사용하는 PDWZ에서는 매 짝수 번째 프레임에 해당한다. 또 하나는, 키 프레임으로, PDWZ에서는 Wyner-Ziv 프레임 전후의 홀수 번째 프레임에 해당한다. 이 키 프레임은 보조정보를 생성하기 위해 부호화기에서 복호화기로 보내주는 정보로서, 통상 인트라 부호화 되어 전송된다. 복호화기에서는 먼저 전송되어온 키 프레임을 복원한 후 이를 이용하여 Wyner-Ziv 프레임에 해당하는

보조정보를 만든다. 부호화기는 Wyner-Ziv 프레임을 부호화하며 생성되는 패리티를 버퍼에 저장하였다가 복호화기의 요청에 따라 생성된 패리티 중 일부를 복호화기에 전송한다. 복호화기의 터보 복호화기는 이 패리티를 받아 양자화된 화소값의 복호화를 실행하며, 사전에 정의한 임계값 이하로 에러율이 내려가지 않는 경우 부호화기에 추가적인 패리티를 요청한다. 몇 차례의 이러한 패리티 요청과 전송을 통하여 에러율이 임계값 이하로 내려가면 터보 복호화기는 패리티 요청을 종료하고 복호화된 신호를 영상복원기로 보낸다. 영상복원기는 터보복호화기에서 받은 복호화된 신호와 보조정보를 이용하여 Wyner-Ziv 프레임의 화소 값을 복원한다.

2. M 비트 양자화기와 비트 플레인 추출기

Wyner-Ziv 코딩에 사용되는 M 비트 양자화기는 화소의 상위 M 비트를 제외한 하위 비트들을 버리는 uniform scalar 양자화기이다. 따라서 M 비트 양자화기를 사용하였을 경우 프레임마다 총 M개의 비트 플레인이 생성되며 다음의 일련의 터보 부호화, 복호화 과정은 각각의 비트 플레인 마다 독립적으로 적용된다.

3. 터보 부호화기

터보 부호화기는 두 개의 RSC(Recursive Systematic Convolutional Code)와 하나의 interleaver 그리고 각각의 RSC로부터 생성된 패리티를 저장하는 패리티 버퍼로 구성된다. 비트 플레인 추출기를 통하여 얻어진 각각의 비트 플레인은, 첫 번째 RSC에는 그대로, 그리고 두 번째 RSC에는 interleaver를 거쳐서 입력된다.

Wyner-Ziv 코딩은 보조정보와 원본 프레임간의 차이를 잡음으로 가정하며 이를 실제 전송채널 상의 잡음과 구별하여 '가상의 채널 잡음'(Virtual Channel Noise)이라 부른다. 보조정보는 이런 가상의 채널 잡음이 섞여있는 원본 프레임으로 간주되기 때문에 각각의 RSC에서 생성되는 systematic 정보는 복호화기로 전송되지 않는다. 또한 Wyner-Ziv 코딩에 쓰이는 터보 코드는

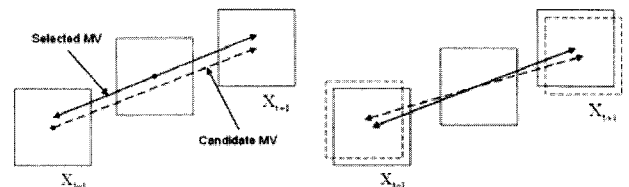
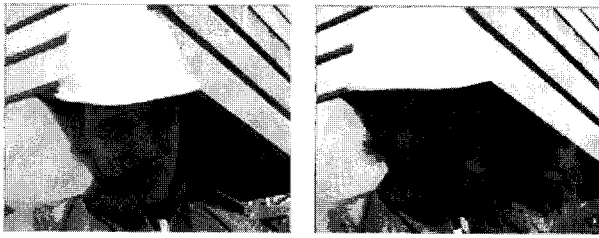


그림 2. 선형 움직임을 가정한 움직임 보간법
Fig. 2. Linear motion interpolation.



(a) 프레임 248

(b) 프레임 254

그림 3. 움직임 보간법으로 만들어진 보조정보 (포만 시퀀스, 지멘스 로고 없는 것)

Fig. 3. Side information from interpolation. (Foreman sequence without Siemens logo)

RCPT(Rate Compatible Punctured Turbo Code)로 생성된 패리티를 버퍼에 저장하며 복호기의 요청에 따라 그 중 일부를 puncturing하여 전송한다. 본 논문에서는 puncturing시 패리티가 트렐리스 상에서 동일한 간격으로 위치할 수 있게 puncturing 테이블을 만들어 사용하였다.

4. 보조정보 생성기

일반적으로 보조정보는 보조정보 생성기에서 전후 프레임 간 선형 움직임을 가정한 보간법에 의해 만들어진다. 물론 보외법을 사용하여 보조정보를 생성할 수도 있으나 성능 면에서 보간법이 보외법에 비해 앞서기 때문에 대부분의 경우 이 방법이 사용되고 있다.

복호화기에서는 부호화기로부터 패리티 정보를 받기 전에 Wyner-Ziv 프레임 X_t 에 대한 보조정보 Y_t 를 만들어야 한다. 따라서 실제적으로는 Wyner-Ziv 프레임 복호화 전에, 앞뒤 키 프레임인 X_{t-1} 과 X_{t+1} 을 먼저 복원하며 이후 X_{t+1} 에서 X_{t-1} 방향으로 움직임 예측을 수행한다. 이렇게 얻어진 움직임 벡터 중 프레임 X_t 의 임의의 블록 위치에서 해당 블록을 가장 가까이 통과하는 벡터를 그 블록의 움직임 벡터로 사용한다. 하지만 후보 움직임 벡터를 보정없이 사용하면 블록 간에 빈자리가 생기거나 겹치는 일이 생길 수 있기 때문에 실제로는 그림 2에서와 같이 후보 움직임 벡터를 평행이동하여 자신의 위치에 맞게 조정한 후 사용한다. 이렇게 Wyner-Ziv 프레임의 움직임 벡터를 키 프레임을 이용하여 추정된 이후에는 전방향 움직임 벡터와 역방향 움직임 벡터를 대칭으로 이동시키며 키 프레임에서 선택되어지는 후보 블록들 간 정합오류가 최소가 되는 쌍을 탐색영역으로부터 찾아낸다. 그리고 그렇게 찾은 블록 쌍을 보간하여 보조정보를 만든다^[5]. 근본적으로 Wyner-Ziv 코딩은 보조정보를 잘 만들수록, 즉 제거되

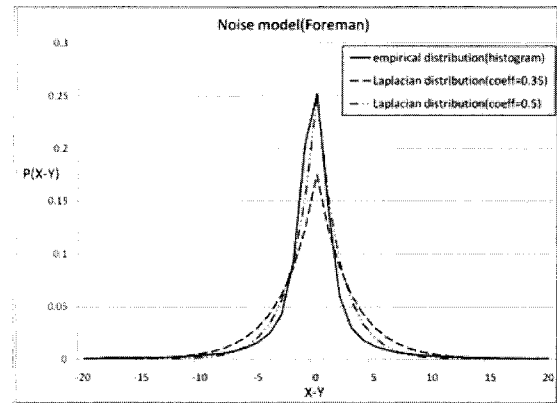


그림 4. 잡음모델(Foreman 영상)

Fig. 4. Noise model(Foreman sequence).

어져야 할 잡음이 적게 생성될수록 전체적인 시스템의 부호화 효율이 좋아진다. 따라서 앞서 언급한 방법 이외에도 움직임 벡터필드에서 선택적으로 비정상적인 움직임 벡터를 찾아내는 방법 등 더욱 개선된 보조정보를 만드는 방법에 관한 많은 연구가 이루어졌다^[6]. 본 논문에서는 앞서 설명한 기본적인 방법^[5]을 이용하여 보조정보를 생성하였으며, Wyner-Ziv 코딩의 성능 확인을 위하여 다른 PDWZ 방법과 같이 키 프레임은 무 손실 압축을 가정하였다. 그림 3은 예로써 생성된 Foreman 영상의 보조정보이다. 보조정보를 만들 때 프레임 간에 움직임이 많지 않은 경우에는 그림 3(a)와 같이 정상적으로 보조정보가 생성된다. 하지만 이와는 달리 프레임 간 움직임이 심한 경우 또는 occlusion이 존재할 경우에는 그림 3(b)와 같이 보조정보 내에 많은 오류가 발생하게 된다. 그림 4는 Foreman 영상의 Wyner-Ziv 프레임과 보조정보간의 잡음 히스토그램과 이를 라플라시안 분포로 모델링한 것이다. 하지만 실제 Wyner-Ziv 코딩을 함에 있어서는 복호화기에서 원본영상에 대한 정보를 알 수 없기 때문에 잡음모델을 라플라시안으로 근사화 시킨다해도 영상에 따른 정확한 계수값을 알 수 없다. 따라서 실제로는 기존에 제안되었던 방법^[3]과 마찬가지로 다양한 영상을 통해 얻어진 통계적인 계수값을 사용한다.

5. 터보 복호화기

터보 복호화기는 MAP(Maximum A Posteriori) 알고리즘^[7], Log-MAP 알고리즘^[8], Sub-MAP 알고리즘^[9], Viterbi 알고리즘^[10] 등 여러 가지 알고리즘을 사용하여 구현할 수 있다. 이중 MAP알고리즘이 연산량이 많기는 하지만 성능면에서 다른 알고리즘보다 좋기 때문에 현재 일반적인 Wyner-Ziv 코딩에서는 이를 주로 사용

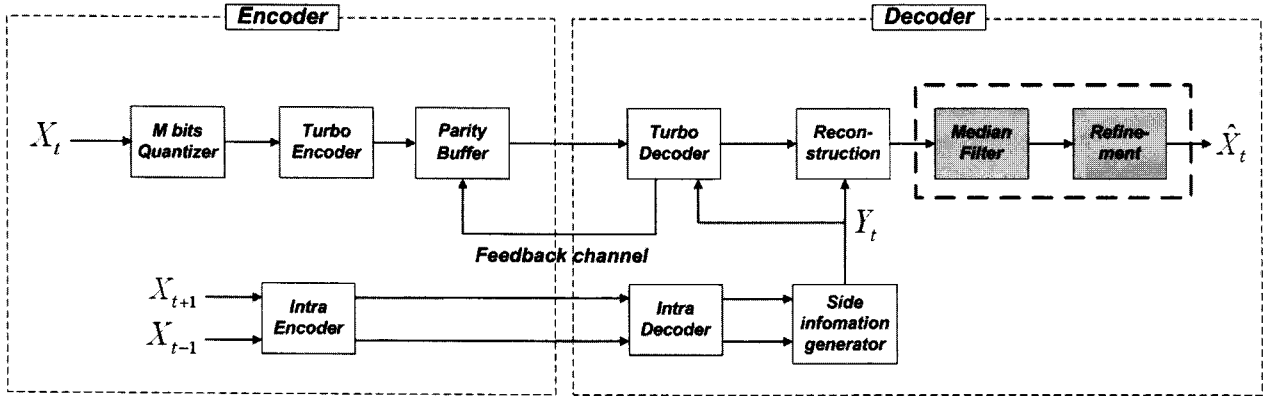


그림 5. 제안하는 Wyner-Ziv 코딩 구조

Fig. 5. Structure of the proposed Wyner-Ziv coding.

한다. 터보 복호화기와 MAP 알고리즘의 설명을 위해 사용하는 기호는 다음과 같이 요약할 수 있다.

- $X = [x_1, \dots, x_N]$: 현재 프레임의 특정 비트 플레인으로 총 N 개의 비트로 구성된 비트 스트림
- $Y = [y_1, \dots, y_N]$: 보조정보 중 특정 비트 플레인에 해당하는 비트스트림
- $Y_i^j = [y_i, \dots, y_j]$, $1 \leq i \leq j \leq N$: 보조정보 중 일부구간, 즉 i 번째에서 j 번째까지의 비트스트림
- s_k : 트렐리스에서 k 번째 상태

MAP(Maximum A Posteriori) 알고리즘은 먼저, 입력 비트 x_k 에 대해 log-likelihood ratio인 $L(x_k)$ 를 구한다.

$$L(x_k) = \log \frac{P(x_k = 0 | Y)}{P(x_k = 1 | Y)}$$

$L(x_k)$ 값이 양수일 경우에는 x_k 를 '0'으로, 음수일 경우에는 '1'로 판별한다. 이 확률 값은 식(1)과 같이 두 개의 상태에 관련된 확률 값과 하나의 천이에 관한 확률 값으로 나눌 수 있다^[7~8]. q 가 $\{0, 1\}$ 중의 하나의 값을 갖는다고 하고 χ_q 가 q 에 대한 트렐리스상의 모든 천이값을 표시한다 하면,

$$\begin{aligned} P(x_k = q | Y) &= P(x_k = q, Y) / P(Y) \\ P(x_k = q, Y) &= \sum_{\chi_q} P(s_{k-1} = s', s_k = s, Y) \\ P(s_{k-1} = s', s_k = s, Y) &= \\ P(s_{k-1} = s', Y_1^{k-1}) &\times P(s_k = s, y_k | s_{k-1} = s') \\ &\times P(Y_{k+1}^N | s_k = s) \end{aligned} \quad (1)$$

이 값들은 다시 식(2)와 같은 메트릭 표현으로 치환할 수 있다.

$$\begin{aligned} \alpha_k(s) &= P(s_k = s, Y_1^k) \\ &= \sum_{s_{k-1}=s'} \alpha_{k-1}(s') \gamma_k(s', s) \\ \beta_k(s) &= P(Y_{k+1}^N | s_k = s) \\ &= \sum_{s_{k+1}=s'} \beta_{k+1}(s') \gamma_k(s, s') \\ \gamma_k(s', s) &= P(s_k = s, y_k | s_{k-1} = s') \\ &= P(x_k = q) P(y_k^x | x_k = q) P(y_k^p | p_k) \end{aligned} \quad (2)$$

이 경우 $P(Y)$ 는 상수와 마찬가지로 이 때문에 결과적으로 $L(x_k)$ 값은 식(3)으로 정리할 수 있다.

$$L(x_k) = \log \left(\frac{\sum_{\chi_0} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}{\sum_{\chi_1} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)} \right) \quad (3)$$

식(2)로부터 터보 복호화기는 트렐리스 상의 처음 상태에서 마지막 상태로 진행해가며 천이 메트릭으로부터 전방향 상태 메트릭을 구하고 마지막 상태에 도달하면 다시 반대 방향으로 진행해가며 역방향 상태 메트릭을 구한다. 이렇게 해서 구한 상태 메트릭 값과 천이 메트릭 값을 이용하여 $L(x_k)$ 값을 계산한다. 그리고 그 결과로부터 에러율을 산출하며 이 값이 임계값 이하로 내려갈 경우 복호화를 끝마치고 그렇지 못할 경우 피드백 채널을 통하여 부호화기에 추가적인 패리티 요청을 한다.

6. 영상복원

터보 부호화가 끝나면 복호화기에서는 원본 프레임의 양자화된 화소 값을 복원하게 된다. 영상복원 과정은 양자화 과정에서 버려진 나머지 하위 비트들을 식(4)와 같은 영상복원 규칙을 적용하여 보상한다.

$$\begin{aligned} \hat{X}_t(i, j) &= 2^{8-M} \times q(i, j), q(i, j) > Q(Y_t(i, j)) \\ \hat{X}_t(i, j) &= Y_t(i, j), q(i, j) = Q(Y_t(i, j)) \\ \hat{X}_t(i, j) &= 2^{8-M} \times [q(i, j) + 1], \\ & q(i, j) < Q(Y_t(i, j)) \end{aligned} \quad (4)$$

$\hat{X}_t(i, j)$: t번째 프레임 (i,j)위치의 복원된 화소값
 $q(i, j)$: 터보 코드로부터 복원된 양자화된 화소값
 $Y_t(i, j)$: t번째 프레임 (i,j)위치의 보조정보 화소값
 $Q(Y_t(i, j))$: $Y_t(i, j)$ 의 양자화된 화소값

7. 제안하는 Wyner-Ziv 기반 경량화 부호화구조

그림 5는 본 논문에서 제안하는 PDWZ 기반의 경량화 부호화기로 앞서 설명한 종래의 PDWZ와 달리, 심볼 단위에서 작동하는 터보 코드를 사용하고 그림에 표기한 것과 같이 제안하는 미디언 필터와 refinement 모듈을 추가하여 에러정정에 따른 화질을 개선한다.

먼저 비트플레인 추출기를 없애고 종래의 비트 단위에서 동작하는 터보 코드 대신 심볼 단위에서 동작하는 Hyper-trellis 터보 코드를^[11] 사용함으로써 채널 코드의 정정능력을 향상시킨다. 이는, 일반적인 채널상의 잡음은 물리적으로 비트별로 전송되므로 비트단위로 가해지는 것으로 모델링하지만, DVC에서의 가상채널 잡음은 보조정보와 Wyner-Ziv 프레임과의 차이이므로, 가상의 채널잡음은 심볼 단위로 주어지는 것이 실제와 더 가까울 것이기 때문이다. 기존의 비트 플레인 기반의 Wyner-Ziv 코딩은 이런 심볼 단위에서의 잡음모델을 비트 단위로 천이하는 트렐리스에 적용하기 위해 심볼

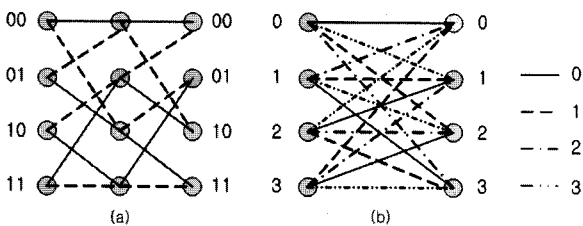


그림 6. 비트단위 트렐리스(a) 와 하이퍼 트렐리스(b)
 Fig. 6. Bit-level trellis(a) and hyper-trellis(b).

을 구성하는 비트들 간의 연관성이 없음을 가정하였다. 하지만 실제 영상신호는 화소를 구성하는 비트들 간에 연관성이 존재하며 그렇기 때문에 비트들 간 연관성이 없음을 가정한 잡음 모델의 부정확성은 결과적으로 동작하는 채널코드의 성능저하로 이어진다^[11].

Hyper-trellis는 그림 6에서 보는 바와 같이 몇 개의 비트 단위 천이를 묶어서 하나의 심볼 단위 천이로 보고 트렐리스를 구성한다. 예를 들자면 입력이 '00', '01', '10', '11'로 들어왔을 때 2번의 비트 단위 천이가 일어나는 그림 6(a)의 비트 단위 트렐리스에서 비하여, 그림 6(b)의 Hyper-trellis에서는 '0', '1', '2', '3'이 들어왔을 때 한 번의 천이가 일어나는 것으로 모델링 하며, 모든 메트릭을 비트 단위가 아닌 심볼 단위에서 계산하게 된다.

다음으로 본 논문의 refinement 모듈은 기존의 방식을^[12] 확장한 것으로 refine을 해야 하는 블록을 선정하는 방법과 움직임 예측과 보상을 수행하는 방법에 있어서 선택적 미디언 필터의 결과를 함께 이용하도록 함으로써 오류정정 후에 잔류하는 약간의 오류가 있더라도 결과 영상의 화질을 개선하도록 하는 것이다.

III. 제안한 복호화 오류 판별 및 정정방법

1. 터보 복호화 오류와 문제점

그림 7은 생성된 보조정보를 이용하여 기존의 Wyner-Ziv 코딩 방법으로 복원한 영상을 나타낸다. 그림을 통해 알 수 있듯이 움직임이 많아 보조정보에 잡음이 많은 경우가 그렇지 않은 경우에 비해 복원된 영상의 화질이 상당히 떨어지는 것을 알 수 있다. 터보 부호화기는 주어진 패리티와 보조정보를 이용하여 트렐리스 상의 처음 상태에서 마지막 상태에 이르는 가장 적합한 천이과정을 추정한다. 하지만 위와 같이 패리티 정보가 영상 내 잡음의 양에 비해 충분하지 못할 경우

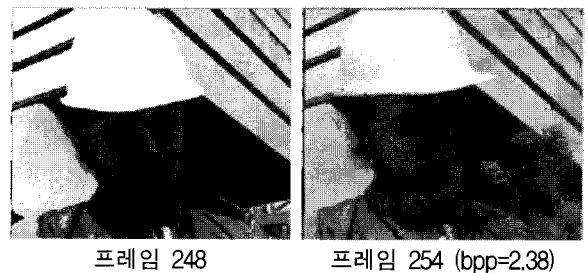


그림 7. 복원된 영상
 Fig. 7. Reconstructed frame.

에는 잘못된 천이 과정을 가장 적합한 것으로 판단한다. 문제는 이렇게 잘못 판단된 천이는 특히 잡음이 많이 섞여 있는 화소와 그에 이웃해 있는 화소 주위에서 종종 해당위치의 화소 값과는 판이하게 다른 값으로 복호화 된다는 것이다. 영상을 보면 이는 마치 Salt & Pepper같은 잡음으로 보이며 특히 보조정보 내 잘못된 보간으로 만들어진 블록 영역에 이런 오류가 밀집해서 나타나게 된다. 그리고 영상에 이와 같은 잡음이 있게 되면 비록 그 발생 빈도수가 작더라도 주관적인 화질을 심각하게 저하 시킨다. 영상처리 이론에 따르면 이와 같은 Salt & Pepper형태의 잡음 보정에 미디언 필터와 같은 비선형필터가 효과적인 것으로 알려져 있다. 하지만 위와 같은 터보 코드 복호화 오류로 인해 생기는 잡음은 발생 빈도수도 작을뿐더러 주로 특정 블록영역에 밀집해서 나타나기 때문에 미디언 필터를 그대로 적용시키는 것은 부적당하다. 우선 미디언 필터를 사용하기 위해서는 이웃화소의 신빙성이 높아야 하기 때문에 이러한 종류의 잡음이 밀집해서 나타나는 것을 방지해야 한다. 본 논문에서는 이를 위해 프레임을 12개의 하위 블록으로 나누고, 이웃한 화소들을 모두 다른 하위블록에 배정함으로써 서로간의 영향을 최소화 하였다. 그리고 발생하는 잡음이 특정 영역에 밀집해서 발생하는 것을 방지하기 위해 터보 부호화전에 interleaving 과정을 수행한다. 하지만 이렇게 발생하는 잡음을 프레임 전체에 분산시켜 주변 화소의 신빙성을 높인다 해도 필터를 적용시켰을 때 잡음을 제거해서 얻는 이득보다 필터링 때문에 영상의 texture에 생기는 손실이 더 크면 바람직하지 않기 때문에, 터보 코드 복호화 오류를 보정하기 위해서는 모든 화소의 위치에서 일괄적으로 필터를 적용하는 대신, 해당 화소에 이와 같은 오류가 발생하였는지 여부를 판별하고 그에 따라 선택적으로 미디언 필터를 적용시킬 수 있는 방안이 필요하다.

2. 터보 코드 복호화 오류 판별 방법

표 1은 터보 코드 복호화 오류의 통계적 발생 특성을 나타내고 있다.

- E_b : 복원된 영상 화소와 그에 상응하는 보조 영상 화소의 상위 M 비트가 서로 다른 사건
- E_c : 복원된 영상의 화소와 그에 상응하는 원본 영상 화소의 상위 M 비트가 다른 사건 (즉, 터보 코드 복호화 오류가 발생한 경우로 실제 실험에서는 원본영상과 복원된 영상의 상위 비트를 직접 비교하

표 1. 복호화 오류의 통계적 특성(M=4, 임계값=0)
Table 1. Statistical property of channel decoding error.

Sequence	$P(E_c)$	$P(E_b)$	$P(E_b E_c)$	$P(E_M E_{b,e})$
Akiyo	5.26×10^{-4}	0.53×10^{-2}	0.70	0.64
Mother & Daughter	0.21×10^{-4}	1.32×10^{-2}	0.79	0.81
Coastguard	1.64×10^{-4}	5.08×10^{-2}	0.69	0.62
Mobile and Calendar	1.53×10^{-4}	4.07×10^{-2}	0.64	0.59
Foreman	2.66×10^{-4}	4.59×10^{-2}	0.80	0.75
Stefan	6.70×10^{-4}	6.16×10^{-2}	0.74	0.64
Average	3.00×10^{-4}	3.63×10^{-2}	0.73	0.68

* 실험에 사용한 영상 시퀀스의 크기는 QCIF이다.

여 구하였다.)

- E_M : 선택적 미디언 필터를 적용한 사건^[13]
- $E_{b,e}$: $E_{b,e} = E_b \cap E_e$ (즉, 복원된 영상의 화소가 그에 상응하는 보조 영상 화소의 상위 M 비트와도 다르고 원본 영상의 상위 M비트와도 다른 사건)

미디언 필터의 주 적용대상은 터보 코드 복호화 오류가 있는 화소 중 Salt & Pepper같은 잡음이 있는 화소로 자신의 이웃들과는 판이하게 다른 값을 보이는 화소들이다. 이러한 화소들은 주변과의 차이가 크기 때문에 해당위치의 보조정보와도 크게 값 차이가 날 것으로 예상할 수 있다. 표 1에서 볼 수 있는 것처럼 영상을 복원한 후 터보 코드 복호화 오류가 발생한 화소 중 평균적으로 약 73%가 보조 정보와 상위 4 비트가 다르다는 것을 확인할 수 있다. 즉, 값이 보조정보와 최소 16 이상(이 경우 M값이 4이므로) 차이가 나는 모든 오류가 있는 화소는 앞의 약 73%에 해당한다. 따라서 필터의 적용대상이 되는 잡음이 있는 화소는 대부분 보조정보와 상위 4 비트가 다른 화소 중에 있을 것이라 가정할 수 있으며 이 위치에 미디언 필터를 적용할 경우 대부분의 오류를 정정할 수 있을 것이라 예상할 수 있다. 그리고 마지막 열의 $P(E_M|E_{b,e})$ 값을 보면 평균적으로 실제 보조정보와 상위 4 비트가 다른 복호화 오류가 발생하는 화소 중 약 68%에 미디언 필터가 적용될 수 있음을 확인할 수 있다. 그렇기 때문에 복원된 화소들 중

상위 4비트가 보조정보와 다른 화소에 대해서만 선택적인 미디언 필터를 적용시키면 이웃화소와 값 차이가 크게 나는 오류를 상당량 정정할 수 있을 것으로 예상할 수 있다. 하지만 이와 같은 방법으로 오류가 없는 픽셀에 필터가 적용되는 경우를 줄인다 하더라도 여전히 오류가 아닌 화소에 필터가 적용되는 경우가 많이 발생하는 것을 실험을 통하여 확인할 수 있었다. 그렇기 때문에 본 논문에서는 선택적 미디언 필터에 임계값을 추가적으로 두어 사용한다^[13]. 이는 복원된 화소가 이웃 화소 중 최고값보다 임계값 이상으로 크거나 또는 최소값보다 임계값 이하로 작을 때에만 미디언 필터를 적용함으로써 영상 고유의 texture에 필터가 잘못 적용되는 것을 좀 더 억제하기 위함이다.

3. 미디언 필터 적용 오류 보상 방법

선택적 미디언 필터를 적용하면 영상 고유의 texture에 필터가 적용되는 것을 피하며 오류를 다수 정정할 수 있으나 문제는 여전히 texture에 필터가 적용되는 경우가 발생한다는 것이다. 따라서 본 논문에서는 이렇게 필터가 적용되어버린 texture를 보상하는 방법과 같이 적용한다.

그림 5의 제안하는 refinement 모듈의 처리절차는 다음과 같다. 우선 refinement 모듈은 모든 블록에 대해서 블록 내 상위 M 비트 값이 보조정보의 값과 다른 화소가 존재하는지의 여부를 파악한다. 만약 상위 M비트가 다른 화소가 있을 경우 양쪽에 있는 키 프레임으로부터 반복적인 움직임 예측을 한다. 이때에 정합도 계산 알고리즘은 WMAD(Weighted Mean Absolute Difference)를 사용하며 상위 M비트가 같은 화소에는 계산 시 $1/2^M$ 의 가중치를 두게 한다^[12]. 그리고 본 논문에서는 여기에 해당 블록에 미디언 필터가 적용된 화소가 있을 경우, 이는 계산의 정확도를 위해 정합도 계산에서 제외한다. 움직임 보상 시 기존의 방법은 해당 블록에 터보 코드 복호화 오류가 거의 없다는 것을 가정하여 상위 M 비트가 같은 화소의 경우에만 움직임 보상을 하였다. 상위 M 비트가 다를 경우에는 키 프레임에서 찾아낸 후보 블록의 동일 위치에 잡음이 있다고 판단하기 때문이다. 하지만 터보 코드 복호화 오류가 있을 때, 특히 주변 화소들과 판이하게 다른 값을 갖는 화소의 경우, 키 프레임에서 현재 블록에 거의 흡사한 블록을 찾는다 해도 오류가 있는 화소의 경우 상위 M 비트 값이 달라 움직임 보상의 대상에서 제외되는 경우가 발생할 수 있다. 따라서 본 논문에서는 상위 M 비트

값이 다르더라도 미디언 필터가 적용된 화소의 경우에만 대해서는 후보 블록에서 움직임 보상을 수행한다.

IV. 실험 방법 및 결과

1. 실험 방법

본 논문에서 제안하는 경량화부호화기법과 터보 복호화 오류보정방법을 평가하기 위하여 Foreman 영상(Siemens 로고 없는 것)과 Stefan 영상 각각 100 프레임에 대하여 실험을 하였다. 이 두 영상 시퀀스를 선택한 것은, 움직임의 많고 적음과 영상 texture의 복잡도에 따라 종래 방법에 비해 본 논문의 제안 방법이 얼마나 성능을 개선할 수 있을 것인가를 보기 위함이다.

실험은 다음 세 가지 경량화 부호화 방법의 비교로 이루어졌다.

- PDWZ : Hyper-trellis 기반 터보 코드를 사용한 종래의 PDWZ 구조 경량화 부호화기^[11]
 - PDWZ+Refine : Hyper-trellis 기반 PDWZ에 종래의 refinement 방법을 적용한 경량화 부호화기^[11~12]
 - Proposed : Hyper-trellis 기반 PDWZ에 확장된 선택적 미디언 필터와 refinement를 사용한 제안 경량화 부호화기
- 구체적 실험 조건은 다음과 같다.
- M비트 양자화기 : 기존에 제안되었던 방법들과 동일하게 M={1, 2, 3, 4}에 대해서 실험.
 - 터보 부호화기 : $G = \left[\frac{1 + D + D^3 + D^4}{1 + D^3 + D^4} \right]$ 를 생성 함수로 사용하였으며 프레임 당 서브블록은 12개, puncturing 주기는 32 사용.
 - 보조정보 생성기 : 블록 크기는 8x8이며 움직임 예측을 위한 탐색영역은 ± 16 , 움직임 벡터 미세조정을 위한 탐색영역은 ± 3 이며 프레임간 거리는 1:1로 가정하였다. 키 프레임은 무손실 압축을 가정함.
 - 미디언 필터 : 임계값으로 20을 사용.
 - Refinement : 전방향과 역방향 탐색영역은 보조 정보를 생성할 때의 크기와 동일하며 양방향 반복 탐색영역의 크기는 ± 8 임.

전송률은 기존의 방법과 마찬가지로 Wyner-Ziv 프레임에 대한 값을 계산하였으며, PSNR은 원본 영상과 복원된 영상 간에 측정된 값이다.

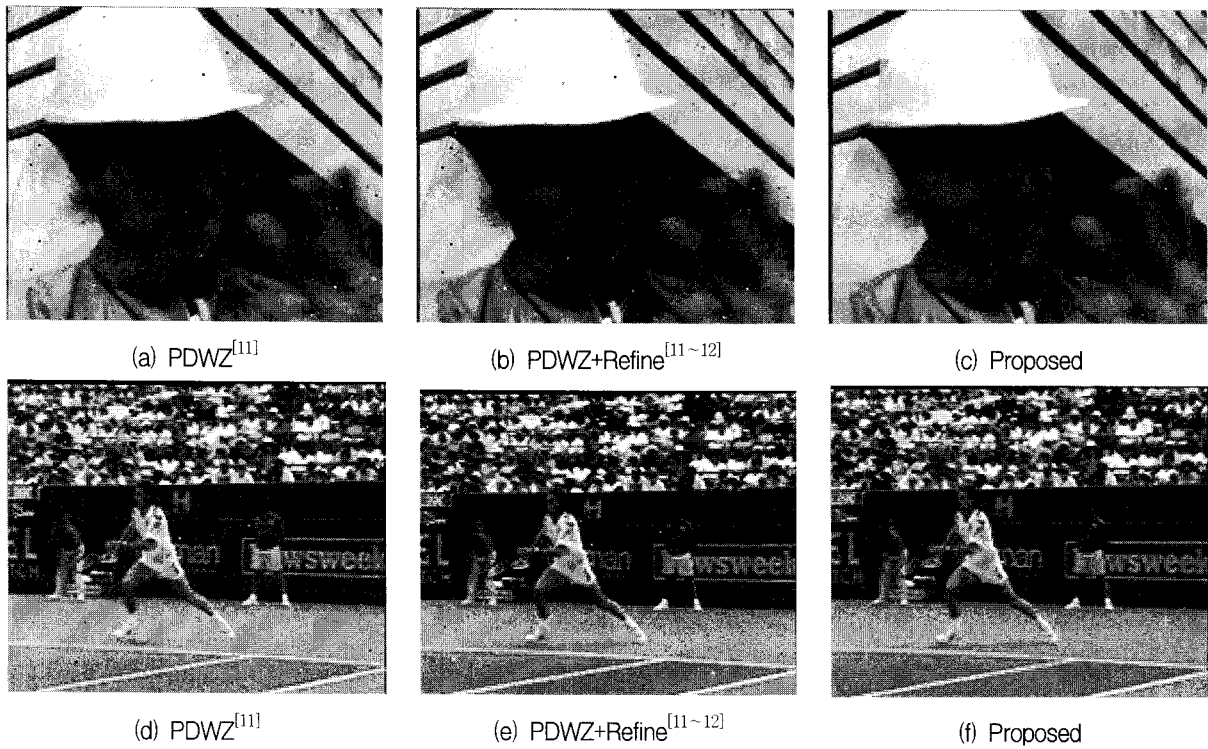


그림 8. 복원된 영상의 주관적 화질 비교

((a)~(c) Foreman 영상의 254번째 프레임, (d)~(f) Stefan 영상의 252번째 프레임)

Fig. 8. Comparison of subjective quality ((a)~(c) Foreman sequence frame 254, (d)~(f) Stefan sequence frame 252).

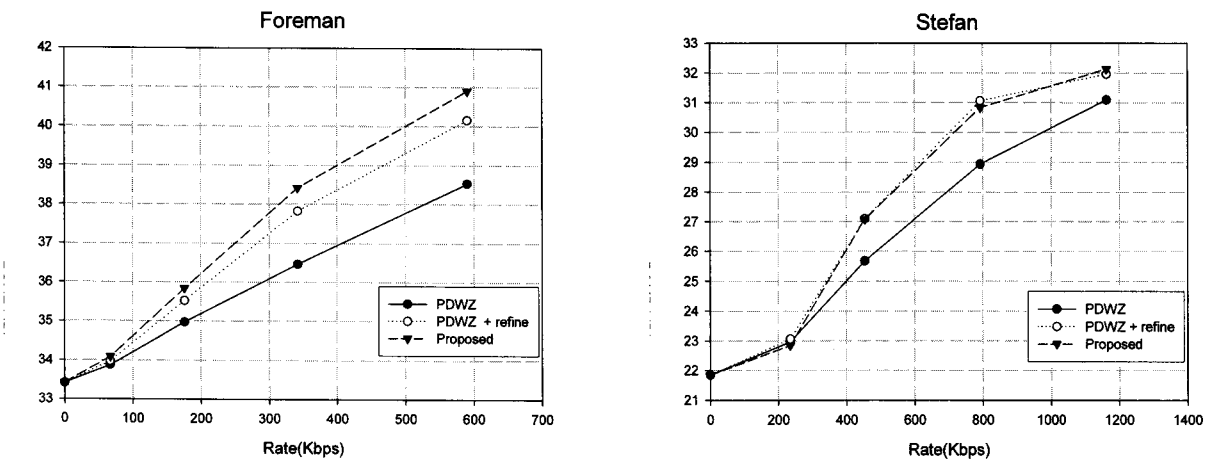


그림 9. R-D 성능 비교

Fig. 9. R-D performance.

2. 실험 결과

그림 8의 (a)~(c)는 Foreman의 254번째 프레임에 대한 각각의 복원 영상으로 제안된 방법에서처럼 미디언 필터를 적용함으로써 터보 코드 복호화 오류로 인한 주관적 화질열화가 많이 개선되었음을 보여주는 것이다.

그림 8의 (d)~(f)의 Stefan에서도 역시 본 논문에서 제안하는 방법을 이용함으로써 주관적 화질이 좋아졌음을 확인할 수 있다. 기존의 미디언 필터도 이러한 종류의

잡음을 제거할 수는 있지만 문제는 영상의 고유한 texture에 손상이 간다는 것이었다. 그렇기 때문에 영상 내 이러한 잡음이 빈도수가 많지 않은 경우에 일괄적으로 미디언 필터를 적용하면 PSNR측면에 상당히 많은 손실이 있게 된다. 하지만 본 논문의 선택적 미디언 필터는 각 화소위치에서 해당화소에 채널 복호화 오류가 있는지 여부를 판단하고 필터를 적용시키기 때문에 texture의 손상을 최대한 억제 할 수 있다. 결과적으로

texture의 손상을 줄임으로써 잡음을 제거해서 얻는 이득이 PSNR 이득으로 나타나게 된다. 실제 그림 8에 나와 있는 영상에서 원본 Wyner-Ziv 프레임과의 PSNR 측정값은 (a)가 31.95dB (b)가 33.3dB (c)가 36.17dB로 제안한 방법이 (a)에 비해서는 4.22dB 좋아 졌고 (b)에 비해서는 2.87dB 좋아 졌다.

그림 9는 화질평가를 위한 Foreman 영상과 Stefan 영상의 R-D 곡선이다. 그래프를 통해 알 수 있듯이 Foreman 영상의 경우 주관적인 화질이 향상은 물론 PSNR 측면에서도 최대 0.8dB에 달하는 많은 화질 향상이 있었음을 알 수 있다. 하지만 이와는 달리 Stefan 영상의 경우 비록 주관적인 화질의 향상은 있었지만 제안하는 방법으로 PSNR 이득은 얻지 못하였음을 알 수 있다. 이는 Stefan의 경우 texture의 복잡도가 높기 때문에 잡음을 제거해서 얻는 이득이 texture의 손상으로 감쇄되었기 때문이다.

V. 결 론

본 논문에서는 복호화 과정에서 발생하는 터보 코드 복호화 오류를 정정하기 위한 기술로 선택적 미디언 필터를 사용한 센서네트워크상의 응용을 위한 경량화 비디오 부호화기를 제안하였다. 제안한 방법은 여러 가지 정보를 이용하여 각 화소 위치에서 해당 화소에 터보 코드 복호화 오류가 있는지 여부를 판단하고 필터를 적용시켜 영상 고유 texture의 손상을 최소화 시키면서 발생한 복호화 오류를 정정할 수 있었다. Foreman 실험 결과를 보면 알 수 있듯이 제안하는 방법은 주관적 화질에서도 많은 개선을 보였지만 동시에 R-D성능에 있어서도 최대 0.8dB에 이르는 성능 이득을 보인다. 하지만 제안한 방법을 이용하여 터보 코드 복호화 오류가 아닌 영상의 texture를 필터의 적용대상에서 가능한 제외시키더라도 texture의 복잡도가 매우 높은 경우에는 필터에 의한 손상을 억제하기가 어렵다. 그렇기 때문에 향후에는 이러한 문제를 해결하기 위해 영상의 texture가 복잡한 경우 좀 더 정교하게 터보 코드 복호화 오류 여부를 결정할 수 있는 방안과 나아가 블록 내 오류가 아닌 texture에 필터가 적용되는 경우가 많이 발생하더라도 이를 보상 할 수 있는 더 정교한 refinement 기술에 대한 연구가 필요하다.

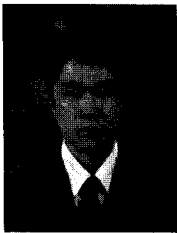
참 고 문 헌

- [1] J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, pp. 471 - 480, July 1973.
- [2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Information Theory*, vol. 22, pp. 1-10, July 1976.
- [3] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv coding for video: Applications to compression and error resilience," *Proc. of IEEE Data Compression Conference, DCC-2003, Snowbird, UT, March 2003.*
- [4] R. Puri and k. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conference on communication, Control, and Computing, Allerton, IL, October, 2002.*
- [5] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Slovak Republic, July 2005.*
- [6] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini, "Adaptively weighted vector-median filters for motion fields smoothing," *Proc. of IEEE ICASSP, Georgia, USA, May 1996.*
- [7] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, pp 284-287, March 1974.
- [8] S. S. Pietrobon, "Implementation and performance of a turbo/map decoder," *International Journal of Satellite Communications*, vol. 16, pp. 23-46, 1998.
- [9] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data distributed by time varying inter-symbol interference," *Globecom '90, San Diego, USA, pp. 1679-1684, December 1990.*
- [10] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260 - 269, April 1967.
- [11] A. Avudainayagam, J. M. Shea, and D. Wu, "Hyper-tellis decoding of pixel-domain Wyner-Ziv video coding," *Globecom '2005, St.Louis, USA, November 2005.*

[12] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding," Proc. of IEEE International Conference on Advanced Video and Signal Based Surveillance, Como, Italy, September 2005.

[13] 박운기, 이상희, 전병우, "효과적인 잡음 제거를 위한 간략한 전처리 방법," 방송공학회논문지 제6권 제2호, pp. 139-147, 2001.

저 자 소 개



고 봉 혁(학생회원)
 2006년 아주대학교 전자공학부 졸업(학사).
 2007년~현재 성균관대학교 정보통신공학부 석사과정
 <주관심분야 : 멀티미디어 영상압축, 신호처리>



전 병 우(정회원)
 1985년 서울대학교 전자공학과 졸업(학사).
 1987년 서울대학교 전자공학과 졸업(석사).
 1992년 Purdue Univ, School of Elec. 졸업(공학박사).
 1993년~1997년 삼성전자 신호처리연구소 수석연구원
 1997년~현재 성균관대학교 정보통신공학부 부교수
 <주관심분야 : 멀티미디어 영상압축, 영상인식, 신호처리>



심 혁 재(학생회원)
 2000년 성균관대학교 전자공학과 졸업(학사).
 2002년 성균관대학교 정보통신공학부 졸업(석사).
 2002년~현재 성균관대학교 정보통신공학부 박사과정

<주관심분야 : 멀티미디어 영상압축, 신호처리>