



# 분산비디오코딩을 바라보는 새로운 눈

서덕영·김철근·박은주·박정아 (경희대학교 전자정보학부)

## I. 서론

지난 2008년 촛불집회 기간 동안 노트북에 카메라를 장착해서 WiBRO로 송출하는 개인 방송국이 크게 활약하였다. 가장 피크인 6월 1일에는 개인방송국의 채널수가 2500개였고 접속자수는 1백20만명이었다. 이제 시설이 좋은 방송국에서 개개인에게 전송하는 형태에서, 빈약한 장비를 가진 개인이 방송을 송출하는 형태의 서비스의 중요도가 커지고 있다. 분산비디오코딩(DVC, distributed video coding)은 이러한 서비스에서 중요한 역할을 할 것이다.

본고에서는 DVC에서 몇 가지 연구할 분야를 제시한다. 압축률은 엔트로피(entropy)에 반비례한다. 엔트로피는 무질서도 또는 불확실성의 정도(degree of uncertainty)로 번역된다. 어떤 정보 X의 정보량은 X의 엔트로피 H(X)로 표시된다. 보통의 압축방식은 이 H(X)를 줄이는데 초점을 맞추고 있다. 그런데 수신측에서 X에 대해서 어느 정도 알고 있다면 불확실성이 적어지면서 보내야 할 정보량을 줄여도 된다. 여기서 '알고 있다면'에 해당하는 것이 보조정보(SI, side information) Y이다. Y를 알 때, X의 불확

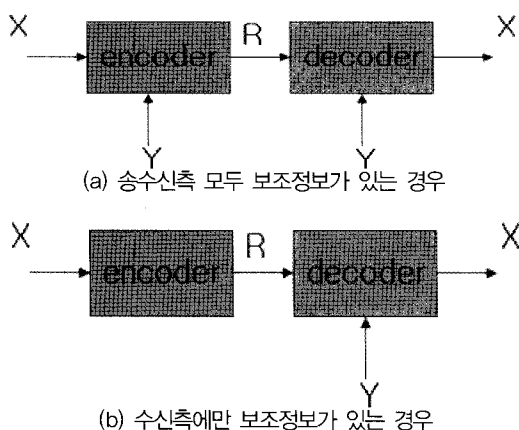
실성은 수식으로는 조건엔트로피(conditional entropy)  $H(X|Y)$ 로 표시된다.  $H(X) > H(X|Y)$  라면, DVC의 효과가 생기는 것이다.

본고에서는 DVC에 대하여 3가지 새로운 개념을 제시하고자 한다. 첫째, 계산량과 압축률의 상관관계가 있을 것이라는 생각이다. 즉, 계산량을 늘리면 압축률을 향상시킬 수 있을 것이라는 생각이다. 둘째, 보조정보 Y가 한 가지 종류가 아니라, 여러 종류라면 더 좋을 것이라는 생각이다. 셋째는 패킷 단위로 전송되는 경우에 DVC에 적합한 패킷화(packetization)이 있을 것이라는 생각이다.

## II. 약간의 정보이론적인 상식

DVC는 새로운 코딩방식이므로 이를 잘 이해하기 위해서는 약간의 정보이론적인 상식이 필요하다. 주머니에 4개의 서로 다른 색의 공이 있다고 하자. 내가 그것을 꺼내고 넣고 하는 일을 하면서 꺼낸 공의 색깔을 순서대로 상대방에게 전송한다고 하자. 각각의 색깔은 몇 비트로 인코딩하면 될까?

“빨강 하양 빨강 노랑 파랑 노랑 ……” 이



<그림 1> 분산코딩의 개념 (전송비트율 R)  $H(X|Y)$  [10]

렇게 전송하면 한 색깔을 알려주는데 “빠 ㅏ 큰 ㅓ ㅓ ㅓ” 처럼 6바이트가 필요하다. 그런데 4개의 서로 다른 심볼(symbol)을 하양 00 빨강 01 노랑 10 파랑 11으로 송수신측이 약속한다면 “010001101110...”으로 전송하여 정보를 전달할 수 있다. 이때 한 색깔당 평균 정보량 2비트를 사용하면 된다. 이것이 엔트로피이다. 정보량  $I(X)$ 는  $\log_2(1/pX)$ 비트이며 각각의 확률  $pX$ 가 1/4씩 이니까, 각 심볼의 정보량은 2비트가 되고 정보량의 평균인 엔트로피는  $H(X)=2$ 비트가 된다.

확률이 같은 8개의 심볼에 대해서는  $H(X)=3$ 비트가 될 것이다. 그런데 여기서 <그림 1> (a)과 같이 보조정보 Y를 송신측과 수신측에서 가지고 있다고 생각하자. 예를 들면, “Y는 X와 1비트 이하로 틀린 정보이다.”가 보조정보가 된다. 송신측에서 수신측이 가진 Y를 알고 있으면 어떨까? 예를 들어,  $X=101$ 이며  $Y=100$ 이라고 하자. 이 경우 2비트만 전송하여도 X를 알아낼 수 있다. Y가 X와 같으면 00, 첫 번째 비트가 틀리면 01, 두 번째 비트가 틀리면 10, 세 번째 비트가 틀리면 11을 보내기로 약속했다면, 이 경우에는 11을 전송할 것이고, 수신측에서는 Y에서

세번째 비트를 고쳐서 X가 101임을 알아낼 수 있다. Y가 X와 같은 경우, 첫 번째 비트가 틀린 경우, 두 번째 비트가 틀린 경우, 세 번째 비트가 틀린 경우 이렇게 4가지 경우가 있고 각각의 확률이 같다면, 조건 엔트로피  $H(X|Y)=2$ 비트가 된다.  $H(X|Y) < H(X)$ 이므로 압축이 가능한 것이다.

만일 <그림 1> (b)과 같이 송신측에서 수신측이 가진 Y를 모르면 어떻게 될까? 이때에도 “Y는 X와 1비트 이하로 틀린 정보이다.”라는 보조정보를 이용해서 압축이 가능하다. 아까와 같이  $X=101$ ,  $Y=100$ 이라고 하자. 그런데 우리는 X의 첫번째 비트와 두번째 비트를 XOR한 비트와 두번째 비트와 세번째 비트를 XOR한 비트를 전송한다. 즉, 예에서는 11을 전송한다. 그런데 8개의 심볼중에 11이 되는 것은 010과 101뿐이다. 그런데 이 둘 중에서  $Y=100$ 과 1비트가 틀린 것은 101이므로 X가 101임을 알게 된다.

위와 같은 개념을 Slepian과 Wolf가 생각해내었고[10], Wyner와 Ziv는 이 개념을 아날로그 값에 적용하는 방법을 생각해내었다. 따라서 DVC에서는 압축하려는 X의 불확실성을 줄이기 위해 Y가 필요한데, Y가 X에 가까우면 가까울수록 압축효율이 좋아진다. 극단적으로 Y가 X와 같다면 정보를 전송하지 않아도 Y를 가지고 X를 알 수 있는 것이다.

### III. 계산량이 증가하면, 압축률이 좋아진다?

Y를 X와 가깝게 만들수록 압축효율이 좋아진다. 그런데 DVC가 ‘분산’인 것은 Y를 만드는 것을 수신측에서 담당하기 때문이다. 계산의 부담을 수신측으로 넘긴 것이다. 그런데 계산을 많이 하면 할수록 더 좋은 Y를 만들 수 있다. 자연



스럽게 계산량과 압축율간에 거래관계(trade-off)가 있음을 생각해볼 수 있다. 이것은 기존의 비디오 코딩에서 움직임 예측을 위한 참조프레임(reference frame)의 수가 많을수록 계산량이 증가하지만 압축률이 좋아지는 것과 같은 이치이다.

### 1. 기존 비디오 코딩의 디코더와 인코더의 복잡도

현재 MPEG-4[1] 와 H.264/AVC[2]와 같은 비디오 코딩 표준에서 압축효율을 높이기 위한 방법들은 디코더보다 인코더측의 복잡도가 높다. 대표적으로 디코더에서의 움직임 보상(motion compensation)을 위하여 인코더에서 움직임 예측/보상(motion estimation/compensation)을 수행하고 모션벡터(motion vector)를 부호화한다. 이러한 구조에서는 인코더의 복잡도가 디코더의 복잡도보다 높을 수 밖에 없다. [3]에서는 디코더의 복잡도 증가를 통해 압축 효율을 증가시키는 방법을 제안하였다. 디코더에서의 움직임 벡터 유도(decoder side Motion vector derivation)로 움직임 벡터를 예측하여 움직임 벡터에 대한 불확실성을 줄이는 것이다. 이에 따라, 움직임 벡터를 표현하는 데 사용되는 비트량을 감소시켜 압축효율을 증가시키는 것이다. 이러한 방법은 인코더와 디코더 모두의 복잡도를 증가시킨다.

### 2. 분산비디오코딩의 디코더와 인코더의 복잡도

분산 비디오 코딩은 인코더의 복잡도를 디코더로 옮긴 구조를 가지고 있다. <표 1>과 <표 2>

<표 1> Encoding Time Complexity for Qi=4,8 and GOP 2,4,8 – QCIF at 15 Hz<sup>[5]</sup>

QP	H.264 / AVC			DISCOVER						Ratios			
	Intra	No Motion		Using LDPCs			Using Turbo Codes			H.264/AVC Intra vs DISCOVER			
		GOP 2	GOP 4	GOP 8	GOP 2	GOP 4	GOP 8	GOP 2	GOP 4	GOP 8	GOP 2	GOP 4	GOP 8
40	32.67	33.14	33.64	32.97	19.44	11.58	8.02	19.14	11.26	7.58	1.68	2.82	4.08
39	33.30	33.81	34.41	33.72	19.75	11.80	8.09	19.47	11.42	7.69	1.69	2.82	4.11
38	33.78	34.34	34.88	34.15	20.05	11.95	8.22	19.73	11.57	7.79	1.69	2.83	4.14
34	37.11	37.60	38.21	37.34	21.67	12.94	8.79	21.39	12.48	8.33	1.71	2.87	4.22
34	37.13	37.61	38.29	37.34	21.68	13.00	8.80	21.39	12.53	8.34	1.71	2.86	4.22
32	38.94	39.55	40.18	39.23	22.78	13.66	9.37	22.38	13.14	8.79	1.71	2.85	4.15
29	42.33	43.11	43.73	42.72	24.49	14.58	9.94	24.13	14.07	9.34	1.73	2.90	4.26
25	48.56	48.89	49.54	48.17	27.72	16.81	11.23	27.36	15.98	10.55	1.75	2.92	4.32

<표 2> Decoding Time Complexity for Qi=4,8 and GOP 2,4,8 – QCIF at 15 Hz<sup>[5]</sup>

QP	H.264/AVC			DISCOVER						
	INTRA	No Motion		Using LDPC			Using Turbo Codes			
		GOP 2	GOP 4	GOP 8	GOP 2	GOP 4	GOP 8	GOP 2	GOP 4	GOP 8
38	1.55	1.56	1.56	1.53	430.27	710.16	986.87	441.70	691.77	891.15
37	1.58	1.61	1.56	1.56	485.89	777.28	1048.44	513.44	797.13	995.11
37	1.61	1.64	1.58	1.57	531.17	895.20	1226.48	580.31	909.39	1150.86
34	1.64	1.66	1.70	1.66	796.69	1525.55	2168.61	875.44	1464.41	1886.16
33	1.70	1.75	1.77	1.69	824.23	1628.69	2330.06	932.67	1565.42	2024.59
31	1.72	1.81	1.78	1.80	1144.45	2254.94	3193.92	1230.25	2093.83	2708.85
30	1.75	1.88	1.81	1.81	1497.25	2802.98	3856.55	1566.94	2668.91	3375.72
26	1.92	2.05	2.03	1.94	2461.73	4462.90	5872.40	2501.95	4309.85	5428.45

는 DISCOVER 분산 비디오 코딩[4]의 인코딩 복잡도와 디코딩 복잡도를 H.264/Intra와 비교하여 나타내었다. 인코딩은 H.264/AVC Intra가 DISCOVER 분산 비디오 코딩보다 2~4배정도 복잡하고 디코딩은 DISCOVER 분산 비디오 코딩이 H.264/AVC Intra보다 300~3000배정도 복잡한 것을 볼 수 있다[5]. DISCOVER 분산 비디오 코딩의 디코딩이 복잡한 이유는 SI(side information)을 생성과 채널 디코딩, 피드백을 이용한 반복적인 처리 때문이다.

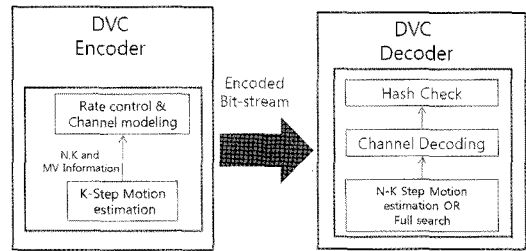
### 3. 분산 비디오 코딩의 피드백 채널의 제거와 복잡도 분배

분산 비디오 코딩은 움직임예측을 인코더에서 수행하지 않기 때문에 인코더는 디코딩에 필요한 비트량을 정확하게 알 수 없다. DISCOVER

분산 비디오 코딩에서는 인코더는 디코딩에 필요한 비트량을 예측하여 디코더에게 전송하고 디코더는 디코딩에 실패할 경우에 피드백채널을 통하여 추가 전송을 요청한다<sup>[6]</sup>. 하지만 인코더와 디코더간의 피드백 채널의 사용은 상호활동적인 디코딩이 필수적이기 때문에 오프라인 어플리케이션에서 사용이 제한된다. 이러한 문제점을 줄이기 위해서는 인코더에서 빠르게 디코더 Residual과 필요한 페리티 비트량을 예측할 수 있는 방법이 필요하다<sup>[7]</sup>. 인코더에서의 비트량 예측의 정확도는 기본적으로 인코더의 복잡도에 비례하게 된다. 분산 비디오 코딩은 디코더와 인코더간의 유연한 복잡도 분배가 가능하므로<sup>[8]</sup> 이러한 특징을 활용하여 필요한 예측 정확도에 따라서 복잡도를 분배할 수 있을 것이다. <sup>[9]</sup>에서는 인코더측에서의 bitplane motion estimation을 제안하였다. 분산 비디오 코딩에서는 움직임이 복잡한 경우 SI의 품질이 떨어질 수 밖에 없고 이에 따라서 디코딩이 실패한 확률은 높아지게 된다. <sup>[9]</sup>에서는 인코더에서의 간단한 bitplane motion estimation을 통하여 SI의 품질을 향상시켰고 이는 인코더의 복잡도의 증가가 비트 에러를 감소시켰음을 보여준다.

#### 4. 분산 비디오 코딩의 단계적 움직임 예측을 통한 인코더와 디코더의 복잡도 분배

압축 효율의 증가를 위해 무분별하게 인코더의 복잡도를 올리는 것은 분산 비디오 코딩의 장점인 '가벼운 인코더'의 특징을 죽이는 것이다. 분산 비디오 코딩의 장점을 살리기 위해서는 단순하고 복잡도의 분배가 가능한 움직임 예측 방법을 사용하여야 한다.



$N$  : Total step size,  $K$  : Encoder step size

〈그림2〉 단계적 움직임 예측 및 보상을 이용한 분산 비디오 코딩

기존의 단순한 움직임 예측 방법으로 3 step search, 2D log search, conjugate directions search등이 있다. 이러한 단계적인 움직임 예측 방법을 분산 비디오 코딩에 적용할 경우 복잡도 분배를 유연하게 할 수 있다. 예를 들어 <그림 2>는 3 step search를 분산 비디오 코딩에 적용하였을 경우를 나타낸다.  $N$ 은 총 스텝 크기를 나타내고  $K$ 는 인코더측에서 수행할 스텝 크기를 나타낸다. 인코더에서 움직임 예측 범위에 따라서  $N$ 의 크기가 정해지면 복잡도의 증가량을 고려하여 인코더에서 수행한  $K$ 의 크기를 결정한다.  $K$ -step의 움직임 예측을 통해 얻은 움직임 정보를 디코더에게 전달하면 디코더에서는 ( $N$ ,  $K$ ) 정보와 MV정보를 이용하여 좁아진 범위 안에서 움직임 예측을 수행할 수 있다. 이를 통하여 SI품질향상과 비트량 예측의 정확도를 향상시킬 수 있을 것이다.

#### IV. 여러 개의 보조정보를 사용하면, 압축률이 좋아진다?

Slepian/Wolf의 정보 이론 경계(bound) 개념 [10]에 의해서 참조하는 보조 정보가 많을수록  $X$ 에 대한 불확실성이 떨어지고, 따라서, 압축률

을 높일 수 있을 것이다. 다음 수식에서와 같이 보조정보를  $Y_1, Y_2, Y_3, Y_4$ 로 늘려가면서 조건 엔트로피도 줄어들 것이다.

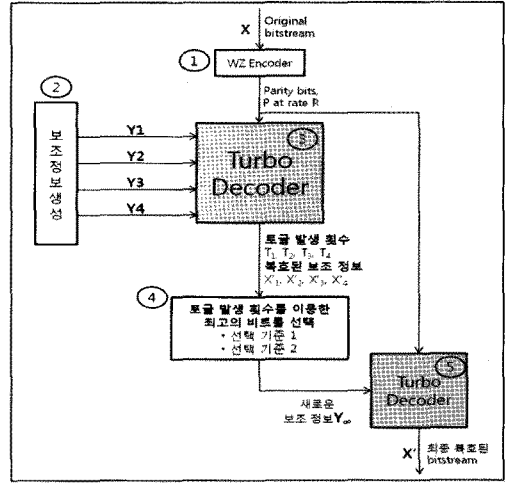
$$H(X|Y_1, Y_2, Y_3, Y_4) \leq H(X|Y_1, Y_2, Y_3) \leq H(X|Y_1, Y_2) \leq H(X|Y_1)$$

이러한 접근은 3-D motion refinement<sup>[11]</sup><sup>[12]</sup>과 multiple side information<sup>[11][13]</sup>기술을 사용하는 방법으로 소개되었다. 이들은 두 개의 보조 정보 스트림의 직접적인 조합을 통하여 압축률을 높이고 있다. 예상 여러 확률에 기초를 둔 터보 복호기는 어떤 보조 정보가 주어진 블록을 복호해내는 데 사용될 것인지를 결정한다.

여러 가지 보조정보는 어떻게 만들 수 있을까? 보조정보를 만들기 위해 영상을 보간(interpolation)하는 방법 A, B, C가 있다고 할 때, 각각의 방법으로 만들어진 보조정보 영상의 어떤 부분은 A 방법으로 만든 부분이 가장 X에 가깝고, 어떤 부분은 B방식으로 만든 부분이 가장 X에 가까울 수 있다. 여러 개의 보조정보 영상을 가지고 가장 X에 가까운 부분부분으로 조각조각 맞춘다면 각각의 방식 하나만 사용하는 경우보다 더 X에 가까운 보조영상을 만들 수 있을 것이다.

그런데 문제는 수신측에서는 X를 모르므로 어떤 것이 더 가까운 것인지를 판단할 수 없다는데 있다. 본고에서는 터보 복호기를 사용하여 복수의 보조 정보의 후보들로부터 생성된 데이터에서 가장 에러가 적은 정보를 선택하는 방법을 제시하고, 같은 양의 전송 정보를 가지고 복호화가 성공할 확률을 높일 수 있음을 보인다. 이를 위하여 복호화 과정에서 포함된 채널 복호화 중에서 얻어지는 정보를 이용한다.

가장 유사한 부분을 결정하는 터보 복호기 터보 코드는 원래의 비트 열에 대한 패리티 비트열,



〈그림 3〉 후보프레임 결정 방법 Flowchart

인터리빙 된 비트 열에 대한 패리티 대해서 번갈아 가면서 복호화를 반복한다. 이 반복(iteration) 과정이 수행되는 동안에 데이터가 더 나은 정보 즉, 원래의 정보로 복호되지만 이 과정에서 고쳐진 데이터가 여러 번 바뀔 수 있다. 예를 들면, 처음 복호 과정에서 100개 중 10번째 데이터(1)와 20번째 데이터(0)의 에러를 검출하여 복호를 해서 원래의 데이터(1→0, 0→1)로 고쳤으나 두 번째 복호 과정에서 그 두 데이터가 다시 에러(0→1, 1→0)로 바뀌어 버리는 경우이다. 반복 과정에서 여러 번 변하게 되는 부분을 '토글(toggle)'이라고 정의한다. 토글이 생긴다는 것은 데이터가 계속하여 바뀌는 부분이기 때문에 바뀌지 않는 부분보다 신뢰성이 떨어진다고 할 수 있다. 토글이 발생하는 부분을 원래의 데이터에서 채널의 영향으로 에러가 발생할 확률이 높은 부분이라고 가정한다. 이러한 가정을 이용하여 DVC에 적용시킨다. DVC의 복호기 측에서 각 블록 당 여러 가지 보조 정보(후보 프레임)를 생성하고 그 중 좋은 것을 선택하면 이를 원래의 정보라고 판단할 수 있는데, 본 논문에서는 보조



정보를 선택하는 데 있어서의 판단 기준이 반복 시 토글 발생 횟수를 계산하여 최소의 토글 발생 횟수 블록을 신뢰성 있는 블록으로 결정하는 것이다. 토글을 터보 복호화기에서 사용되는 과정이 <그림 3>에 나타내었고, 각 번호별 구체적인 내용은 다음과 같다.

- ① WZ 부호화기를 통해서 전송 정보 X는 실제 패리티 전송률 R에 맞는 패리티 스트림을 생성하게 되고, 이는 터보 복호기로 전달된다.
- ② 복수 개의 서로 다른 방법으로 만들어진 후보 프레임의 비트열  $Y_n$ 를 생성하여 터보 복호기로 전달한다.
- ③ WZ 부호기에서 패리티 비트열을 비트율 R로 평칭링하여 전송한다. 각 후보 별로 패리티 비트 열과 보조 정보 비트 열 들을 복호화 하여 복호화 된 비트열  $X' n$ 를 각각 생성한다. 이 때, 복호화 과정에서 발생하는 비트별 토글수  $T_n$ 를 저장한다.
- ④ 다수의 후보를 비교해 가면서 최적의 보조 정보  $Y_{\infty}$ 를 만든다. 각 비트 별로 최소의 토글 수에 해당하는 후보의 복호화 된 비트를 저장하여 보조 정보  $Y_{\infty}$ 를 만든다. 이 때, 4개 중 최소의 토글 수를 가진 후보가 1개라면 이 후보를 선택하고, 2개 이상이라면 선택 기준이 필요하다.

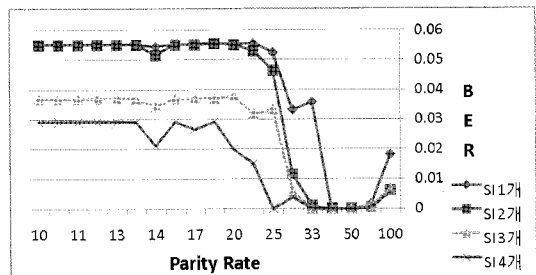
전송된 패리티 열 P를 이용하여  $Y_{\infty}$ 를 한번 더 복호화하여 최종 복호화된  $X'$ 를 만들어낸다. 그리고 토글이 발생하는 횟수로 좋은 블록을 선택하는 데에 있어서 토글 발생 횟수가 동일한 경우 후보를 택하는 기준이 필요하다.

- 기준 1 : 좌에서 우로, 위에서 아래로 진행 하면서 블록 별 토글 발생 횟수를 모두 누적한 후보 프레임 별 토글 발생 횟수가 가장 적은 후보 프레임의 비트를 선택한다.

- 기준 2 : 같은 토글 발생 횟수가 나타나는 주변의 정보를 이용하여 주변의 최소 토글 발생 횟수에 해당하는 블록을 선택한다. 여기서, 주변이라는 범위는 터보 코더에서 패리티 비트열의 평칭링 간격에 따라 변경된다.

위에서 제시한 두 가지 선택 기준에 대한 가정은 전체적(프레임 단위)으로 잘 복호되는 정보가 부분적(블록별)으로도 잘 복호 될 가능성이 더 크다는 것이다.

<그림 4>는 본고에서 제안한 복수개의 보조 정보와 터보 코드의 특성인 토글 방법을 적용시키면 터보 복호화 후 BER이 얼마나 떨어지는지를 보여주는 그래프이다. 랜덤하게 보조 정보를 4개 생성하고 이를 각각 복호화 한 후 토글 발생 횟수로 좋은 보조 정보의 블록을 선택하여 <그림 3>에서 나타낸  $Y_{\infty}$ 을 생성한 결과를 비교한 것이다. 같은 BER을 얻고자 할 경우 제안한 방식에 이용되는 보조 정보의 개수가 증가할수록 적은 패리티만으로도 복호가 가능하다는 것을 보여준다.



<그림 4> 사용하는 SI개수에 따른 BER 비교

## V. DVC에 적합한 패킷화 방식

앞으로의 통신망은 all-IP망으로 진화한다. 따라서, DVC서비스는 대부분 인터넷 패킷 통신망을 통해서 이루어 질 것이다. DVC의 압축은 일정 부분 채널 코딩으로 인해 이루어지는데 [14], 채널코딩의 특성상 손실이 한 곳으로 집중되면 채널코딩의 기능을 할 수 없게 되지만, 손실이 분산되면 복원할 수 있는 가능성이 높아진다. DVC에서 사용하는 채널코드는 대표적으로 터보 코드[15]와 LDPC코드 또는 LDPCA 코드[16]가 있다. DVC는 무선 단말기에서 사용될 가능성이 많으며<sup>[17]</sup>, 무선 통신망에서는 패킷 손실이 자주 일어나게 된다. 이러한 사실에 기초하여, 패킷 손실에 대비한 대책이 연구되고 있다.

한 예로, <sup>[18]</sup>에서는 터보 인코더의 출력인 패리티의 에러 발견에 대한 가능성을 높이기 위해 CRC 생성기를 고용한다. <sup>[18]</sup>는 WZ 프레임을 수개의 부분 영상으로 나누고, 각각의 부분 영상들에 대한 비트플레인을 추출한다. 추출한 비트플레인들을 터보 인코더와 CRC 생성기에 각각 넣고 그 출력을 디코더에 보내는 방법을 사용한다. 즉, 터보 코드를 사용하여 만든 패리티 비트와 CRC 비트들을 디코더에 보낸다. 이렇게 DVC의 패리티 패킷 손실에 대비한 대책은 여러 가지 방법으로 연구되고 있다.

### 1. 터보코드와 패킷 손실의 영향

일반적으로 채널 코드는 에러 분포가 넓을 때 좋은 효율을 나타내고 에러 분포가 좁을 때 효율이 나쁘다. 터보코드는 다른 채널 코드들에 비해 에러 분포가 좁을 때 효율이 좋지만 어느 한계보다 넓은 영역에서 집중적으로 에러가 생기면 에



〈그림 5〉 터보 코드의 성능

러를 복원할 수 없다.

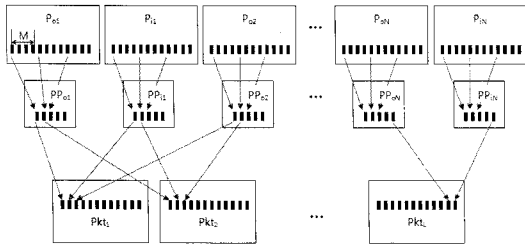
〈그림 5〉는 터보 코드의 성능을 눈으로 보기 쉽게 그린 것이다. 그림에서 검은색으로 선을 그은 것이 에러를 표시한 것인데, 위에 있는 것은 에러를 고치기 전의 데이터를 나타낸 것이고 아래에 있는 것은 에러를 고친 후의 데이터를 나타낸 것이다. 에러가 넓게 퍼져 있는 부분은 터보 코드가 에러를 잘 고치지만, 에러 분포가 좁은 부분은 터보 코드의 에러 수정 능력이 떨어짐을 한 눈에 알 수 있다. 이렇게, 터보 코드도 에러 분포가 좁은 데이터에 대해서는 에러정정 능력이 떨어진다.

### 2. 교차 인터리빙 패리티 패킷 구성 방식

DVC 복호화기에 도착한 패리티 패킷의 에러가 고르게 분산되어 있다면 위에서 보았듯이, 터보 코드는 그 에러를 잘 고칠 것이다. 그러나, 실제로는 에러가 어느 곳에 생길지 예측할 수 없다. 그래서 터보코드의 출력을 다시 한 번 인터리빙하여 패킷화하는 방법을 고안하였다. 터보코드는 자체적으로 burst error에 대처하기 위해 인터리빙 패리티를 만들지만, 원래의 패리티와 인터리빙 패리티를 패킷화 할 때, 이것을 다시 한 번 인터리빙하는 방법이다.

이하에서는 이 방법을 교차 인터리빙이라 하며, 아래에 더 자세한 방법을 소개하였다.

교차 인터리빙 방식: 〈그림 6〉는 교차 인터리빙하는 방식을 보여준다. 평처링 패리티 비트열 PPon와 PPin을 L개의 패킷에 되도록 고르게 분

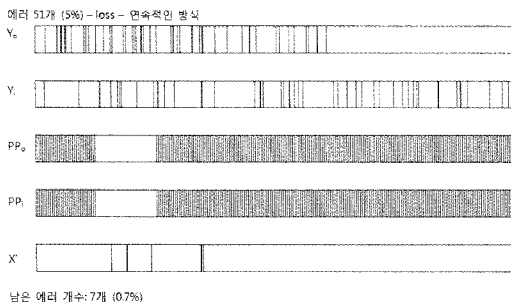


<그림 6> 교차 인터리빙 패리티패킷 구조도

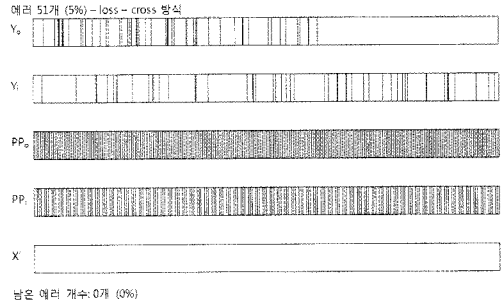
포되도록 한다. 이를 테면, PPon와 PPin에서 j 번째 비트는 (j modulo L)+1번째 패킷에 속하게 한다.

### 3. 실험 결과

<그림 7>와 <그림 8>은 각각 같은 크기의 패리티 패킷을 손실한 상황을 가정하여, 연속적인 패리티 구성 방식을 사용했을 때의 터보 복호결과와 교차 인터리빙 구성 방식을 사용했을 때의 터보 복호결과를 나타낸 것이다. 일단 원래 보조 정보  $Y_0$ , 원래 보조정보를 인터리빙한  $Y_i$ , 복호화한 결과인  $X'$ 에 대해서는  $X$ 와 비교했을 때의 에러를 표시한 것이다. 다음으로  $PP_0$ 와  $PP_i$ 는 복호화기가 패리티 패킷을 받았을 때 패리티 비트가 존재하는 위치를 검은 색으로 나타내었다. 즉, 흰색 부분은 이레이저(erasure)이다. 총 8개의



<그림 7> 손실 패킷 존재시 기존의 연속적인 방식을 사용했을 때의 복호 결과



<그림 8> 손실 패킷 존재시 제안하는 교차 인터리빙 방식을 사용했을 때의 복호 결과

패리티 패킷 중에서 2번째의 패킷을 손실한 상황을 가정하였다. <그림 7>에서 보여진 것처럼 연속적인 패리티 패킷 구성 방식을 사용하였을 때는 에러 개수가 7개가 남았고, 교차 인터리빙 패리티 패킷 구성 방식을 사용하였을 때는 <그림 8>에서 보여진 것처럼 에러가 하나도 남지 않았다.

위의 실험 결과는 교차 인터리빙 방식이 연속적인 방식보다 더 좋은 결과를 보이는 최상의 조건에서 실험한 것이다. 그러나, 패킷당 데이터의 수를 많게 하면, 교차 인터리빙 방식에 비해 연속적인 방식을 사용했을 때 남아있는 에러 개수가 안정적이다. 교차 인터리빙 방식을 사용하면 에러를 고칠 때는 모두 고치지만, 에러를 못 고칠 때는 성능이 훨씬 나쁘다.

## VI. 마치며

본고에서는 앞으로 DVC에서 연구될 수 있는 분야 중에 3가지를 제시하였다. 다수의 보조정보를 이용하여 불확실성을 줄인다는 개념, 계산량을 늘임으로서 압축율을 향상시킨다는 개념은 기존의 비디오 코딩에도 적용할 수 있는 개념이다. 또한 향후 네트워크가 패킷교환방식으로 통일되므로 DVC 데이터의 패킷화 방식에 대한 연





구도 필요하다.

DVC가 앞으로 광범위하게 쓰일지는 의문이다. 그러나, DVC에서 적용되는 개념들은 보통의 비디오 코덱에서 사용할 수 있는 개념이 있을 수 있다. 반대로 기존 코덱에서 사용되었던 방법중에서도 DVC에 적용할 수 있는 방법이 있을 것이다.

참고문헌

- [1] "Information technology - Generic coding of audio-visual objects: Visual," ISO/IEC 14496-2, 1998.
- [2] "Advanced video coding for Generic audio visual services," ITU-T Recommendation H.264, March, 2005.
- [3] S. Kamp, M. Evertz, M. Wien, "Decoder side motion vector derivation for inter frame video coding," ICIP 2008, pp1120-1123, October, 2008.
- [4] <http://www.discoverdvc.org/>
- [5] <http://www.img.lx.it.pt/~discover/compl exity.html>
- [6] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The Discover codec: Architecture, techniques and evaluation," Picture Coding Symposium (PCS), November, 2007.
- [7] J. L. Martinez, G. Fernandez- Escribano, H. Kalva, W. A. R. J. Weerak kody, W. A. C. Fernando, A. Garrido, "Feedback free DVC architecture machine learning," ICIP 2008, pp1140-1143, October, 2008.
- [8] R. Puri, A. Majumdar, K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," IEEE Transactions on Image Processing, Vol.16, No.10, October 2007.
- [9] T. Clerckx, A. Munteanu, J. Cornells, P. Schelkens, "Distributed video coding with shared encoder/decoder complexity," ICIP 2007, pp.414-421, October 2007.
- [10] J. D. Slepian, J. K. Wolf, "Noiseless coding of correlated information sources," IEEE Transactions on Information Theory, Vol. IT-19, pp.471-480, July 1973.
- [11] A. B. B. Adikari, W. A. C. Fernando, W. A. R. Weerakkody, "Side information improvement in DVC with two side information streams and 3D motion refinement," Canadian Conference on Electrical and Computer Engineering, pp.32-35, April 2007.
- [12] A. B. B. Adikari, W. A. C. Fernando, W. A. R. Weerakkody, H. K. Arachchi, "Sequential motion estimation using luminance and chrominance information for distributed video coding of Wyner-Ziv frames," IEE Electronics Letters, Vol.42, Issue 7, pp.398-399, March, 2006.
- [13] A. B. B. Adikari, W. A. C. Fernando, W. A. R. Weerakkody, "Multiple side information streams for distributed video doding," IET Electronics Letter, Vol.42, Issue 25, pp1447-1449, March 2006.
- [14] Y. Zhao, J. Garcia-Frias, "Turbo compression /joint source-channel coding of correlated binary sources with hidden Markov correlation," EURASIP Signal Processing

Journal Elsevier, Vol.86, No.11, pp.3115-3122, November, 2006.

- [15] J. Garcia-Frias, "Compression of correlated binary sources using Turbo codes," Communications Letters, IEEE Vol.5, Issue 10, October, 2001.
- [16] D. Varodayan, A. Aaron, B. Girod, "Rate-adaptive distributed source coding using low-density parity-check codes," Signals Systems and Computers Conference Record of the Thirty-Ninth Asilomar Conference, November, 2005.
- [17] F. Pereira, P. Correia, E. Acosta, L. Torres, C. Guillemot, M. Ouaret, F. Dufaux, T. Ebrahimi, R. Leonardi, M. Dalai, S. Klomp, "Distributed coding for video services, d-4," Project of the IST FET program of the European Union within the FP6/2002/IST/C Call, March, 2006.
- [18] W. Chien, L. J. Karam, G. P. Abousleman, "Rate-distortion based selective decoding for pixel-domain distributed video coding," ICIP 2008, pp.1132-1135, October, 2008.

저자소개



서 덕 영

1980년 2월 서울대학교 핵공학과 학사  
 1986년 6월 미국 조지아텍 핵공학과 석사  
 1990년 6월 미국 조지아텍 전기·컴퓨터공학과 박사  
 1990년 9월 ~ 1992년 2월 생산기술연구원 HDTV 연구  
 단 선임연구원  
 1992년 3월 ~ 현재 경희대학교 전자정보학부 교수  
 주관심 분야 : networked video



김 철 근

2005년 2월 경희대학교 전자공학과 학사  
 2007년 2월 경희대학교 전자공학과 석사  
 2008년 3월 ~ 현재 경희대학교 전자·전파공학과 박사  
 과정  
 주관심 분야 : 무선 데이터 통신, 동영상 신호처리,  
 IPTV

### 저자소개



**박 은 주**

2007년 8월 경희대학교 컴퓨터공학과 학사  
2008년 3월~현재 경희대학교 전자·전파공학과 석사과정  
주관심 분야 : 멀티미디어 통신, 무선 데이터 통신



**박 정 아**

2009년 2월 경희대학교 전자·전파공학과 학사  
2009년 3월~현재 경희대학교 전자·전파공학과 석사과정  
주관심 분야 : 멀티미디어 통신, 무선 데이터 통신