

일반논문-06-11-1-11

단일 엔코더 및 디코더를 이용하는 다시점 비디오 시스템

김학수^{a)}, 김윤^{a)}, 김만배^{a)†}

Multi-View Video System using Single Encoder and Decoder

Haksoo Kim^{a)}, Yoon Kim^{a)} and Manbae Kim^{a)†}

요약

인터넷을 통한 데이터 전송 기술의 발전은 다양한 실감 콘텐츠를 확산시키는 역할을 하고 있고, 이러한 실감 콘텐츠의 하나로 다시점 비디오가 연구되고 있다. 다시점 비디오는 여러 대의 카메라로부터 획득한 영상을 사용자에게 전달해 원하는 시점에서 영상을 시청할 수 있도록 하는 차세대 영상이다. 다시점 영상을 공급하기 위해서는 일반적으로 카메라 개수와 같은 개수의 엔코더와 디코더가 필요하며, 또한 상대적으로 큰 전송 대역폭을 필요로 한다. 본 논문에서는 상기 문제점을 해결하기 위해서 단일 엔코더와 디코더만을 이용하는 다시점 비디오 시스템을 제안한다. 부호화에서는 다시점 YUV 비디오를 GOP 단위로 믹싱하여 1개의 YUV 스트림을 만든 후에, 단일 H.264/AVC 엔코더로 부호화하여 압축 비트열을 생성한다. 또한 복호화도 단일 디코더 및 이를 제어하는 스케줄러로 구성된다. 스케줄러는 디코더 이용률과 프레임 스킵을 활용하여 각 시점 영상들이 동등한 복호화 프레임 개수를 가지게 한다. 또한 프레임 스킵에서는 비용 함수를 이용하여 H.264/AVC 프로파일들에 따른 효율적인 프레임 선택 방법을 제안한다. 다양한 메인 및 베이스라인 프로파일 다시점 영상들을 활용하여 제안 방법의 성능을 검증하였다. 실험 결과는 각 시점 영상이 동등한 디코더 이용률을 얻는 것을 보여준다. 또한 rate-distortion 커브를 이용하여 제안한 방식과 동시 방식의 성능을 검증하였다.

Abstract

The progress of data transmission technology through the Internet has spread a variety of realistic contents. One of such contents is multi-view video that is acquired from multiple camera sensors. In general, the multi-view video processing requires encoders and decoders as many as the number of cameras, and thus the processing complexity results in difficulties of practical implementation. To solve for this problem, this paper considers a simple multi-view system utilizing a single encoder and a single decoder. In the encoder side, input multi-view YUV sequences are combined on GOP units by a video mixer. Then, the mixed sequence is compressed by a single H.264/AVC encoder. The decoding is composed of a single decoder and a scheduler controlling the decoding process. The goal of the scheduler is to assign approximately identical number of decoded frames to each view sequence by estimating the decoder utilization of a GOP and subsequently applying frame skip algorithms. Furthermore, in the frame skip, efficient frame selection algorithms are studied for H.264/AVC baseline and main profiles based upon a cost function that is related to perceived video quality. Our proposed method has been performed on various multi-view test sequences adopted by MPEG 3DAV. Experimental results show that approximately identical decoder utilization is achieved for each view sequence so that each view sequence is fairly displayed. As well, the performance of the proposed method is examined in terms of bit-rate and PSNR using a rate-distortion curve.

Keyword: multi-view video, single decoder/encoder, scheduling, frame skip, H.264/AVC

a) 강원대학교 컴퓨터정보통신공학과

Dept. of Computer, Info. & Telecom, Kangwon National Univ.

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (광주과학기술원 ITRC 실감방송연구센터 IITA-2005-C1090-0502-0022)

† 교신저자 : 김만배(manbae@kangwon.ac.kr)

I. 서론

정보 통신의 기술 발달과 사용자의 요구에 따라 현재 우리 주위에는 오락, 게임, 입체 영화 등 다양한 콘텐츠들이 산재되어 있다. 실감 콘텐츠는 HDTV에 이어 3DTV, 3D mobile phone 등을 통해서 제공되고 있고^[1,2], 인터넷을 통한 데이터 전송 기술은 다양한 콘텐츠를 확산시키는 역할을 하고 있다. 소비자들은 인터넷이나 다른 전송 방법을 통해 다양한 콘텐츠를 보고 듣고 즐기면서, 단순 소비 형태의 콘텐츠 이용에서 대화형 콘텐츠, 실감 콘텐츠의 이용을 요구하는 수준에 이르고 있다. 한 예로 여러 대의 카메라를 통해 획득한 다시점 비디오(multi-view video)의 제공은 사용자들의 요구를 충족시킬 수 있는 좋은 대안이 될 수 있다. 따라서, 하나의 정적 시점보다는 다수의 시점 영상을 제공하고, 사용자가 임의로 시점(view)을 변경하면 해당 시점 영상 콘텐츠를 공급하는 것이 가능하다.^[3,4,5]

다시점 비디오 처리 기술은 다시점 비디오를 입력으로 일련의 처리기술을 제공하는 것을 목적으로 한다. 이 과정에서 다시점 비디오 획득, 압축, 다중화, 전송 및 복원, 합성, 디스플레이가 주요 연구 기술이다. 대용량의 다시점 비디오의 저장 및 전송을 위한 압축 기법 연구가 MPEG 3DAV의 MVC 그룹을 중심으로 활발히 진행 중이다^[6]. MPEG MVC에서는 현재 Fraunhofer-HHI에서 제안한 다시점 비디오의 엔코딩 방법을 참조 모델로 확정하였다^[7]. Hierarchical B picture와 adapted prediction structure를 활용하는 방식으로, H.264/AVC의 Decoded Picture Buffer (DPB)의 개수가 $2 * GOP_length + number_of_views$ 만큼 증가한다. 엔코딩 단계에서는 효율적인 메모리 관리를 위해서 프레임 재배열(reordering)을 하게 되는데, 모든 시점의 첫 번째 영상들에서 시작하여 시간 축으로 GOP의 남은 프레임들을 지그재그(zigzag) 스캔으로 한개의 YUV 파일을 만든 후에 [8]에서 제안한 방식으로 부호화한다. 디코더는 엔코더의 역순으로 복호화하게 된다.

기존 다시점 비디오 시스템의 방식은 다시점 입력 비디오 신호가 들어오면 카메라 개수와 같은 개수의 엔코더를 사용해 개별적인 압축 비트열(compressed bitstream)으로 만드는 동시(simulcast) 방식이다. 각 비트열은 MUX를 통

해서 전송되는데, MUX는 일반적으로 해당 전송 규격에 맞추어 데이터 구조를 만든 후에 채널로 전송한다. Vetro 등이 제안한 다시점 비디오 전송 시스템에서는 다수의 카메라로부터 획득된 영상을 독립적으로 각각 부호화 하며, 개별 채널을 통해 사용자에게 전달한다^[5,9,10]. 따라서 카메라 수 만큼의 전송 채널이 필요하다. 이렇게 전달된 부호화 영상은 수신 측에서 각각의 디코더를 통해 복호화되며 최종적으로 사용자에게 전달된다.

Fecker 등이 제안한 방식은 위의 방식과 유사하나, MUX에서 압축 비트열을 한개의 데이터 스트림을 만든 후 전송한다^[11,12]. 단말에서는 DEMUX를 통해 각각의 압축 스트림으로 분리한 후, 다시점 디코더를 이용하여 영상을 복원한다. 그리고 두 방식 모두 카메라 개수와 같은 수의 엔코더 및 디코더가 필요하다. 요약하면, K 개의 다시점 영상을 동시에 전송하기 위해서는 K 개의 채널을 필요로 하며, 또한 K 개의 엔코더와 K 개의 디코더를 필요로 한다. 이러한 시스템의 복잡성으로 인해 다시점 비디오의 전송을 위한 실용적인 시스템 구현에는 어려움이 있다.

상기 문제를 해결하기 위해 본 논문에서는 단일 엔코더와 단일 디코더를 이용한 다시점 비디오 처리 방법을 제안한다. 제안 방법에서는 K개의 YUV 비디오 시퀀스를 입력 받아 비디오 믹서(video mixer)를 통해 1개의 단일 YUV 스트림을 생성한다. 단일 스트림을 생성할 때에는 카메라의 개수, MPEG GOP 크기, 엔코더 프로파일 등의 정보가 필요하다. 단일 스트림은 단일 엔코더를 사용하여 압축 부호화하고, 이렇게 만들어진 부호화 영상은 단일 채널로 전송이 가능한 장점이 있다. 믹싱된 비디오 데이터는 기존 H.264/AVC를 포함하는 MPEG 비디오 압축을 이용하여 부호화가 가능한 장점이 있다. 또한 믹싱 기법에서는 각 시점 비디오를 GOP (Group of Picture) 단위로 나눈 후에, 각 GOP 단위 순서로 연결하여 단일 스트림을 만든다. 단일 스트림의 압축은 MPEG MVC에서 채택한 H.264/AVC를 기본으로 사용한다^[6,13,14]. 기존 방법들은 N 개의 시점에 대해서 N 개의 엔코더를 사용하고 있는 것에 반해서, 상대적으로 H.264/AVC 엔코더는 1 개만 사용되기 때문에 시스템 복잡도가 감소되지만, 채널 대역폭 및 엔코딩하는 매크로블록의 개수 등은 영향이 없다. 제안 방식은 [8]에서도

(Group of Picture) 단위로 나누어 부호화한다. 그림1(b)의 부호화 단계에서는 각 시점 시퀀스는 동일한 GOP 구조(예: N=4이면 IPPPPI...)을 가지게 된다. $GOP_i(V_k)$ 는 k번째 시점 시퀀스의 i번째 GOP를 가리킨다. 이러한 시퀀스 구조는 그림1(b)처럼 1개의 단일 스트림 형태로 만들어져 부호화된다. 즉, $GOP_1(V_1)$ $GOP_1(V_2)$ $GOP_1(V_3)$ $GOP_1(V_4)$ $GOP_2(V_1)$ $GOP_2(V_2)$ 순으로 정렬한 후에 H.264/AVC 로 부호화한다. $GoGOP_i = \{GOP_i(V_1), GOP_i(V_2), GOP_i(V_3), GOP_i(V_4)\}$ 이다.

이렇게 만들어진 부호화 데이터는 단일 채널을 통해 전송 가능하며, 수신 측에서는 역으로 부호화 데이터를 해당 시점 영상들로 분리하여 디코더를 통해 복호화한다. 동시 방식과 달리 1개의 엔코더만 필요하며 복호화 과정에서도 1개의 디코더를 사용하는 장점을 가지고 있다. I/P 프레임 이외에 B 프레임을 사용하는 메인 프로파일도 유사하게 부호화된다.

Ⅲ. 다시점 비디오의 복호화

Ⅱ장에서는 다시점 영상들의 비디오 믹싱 및 부호화 방법을 설명하였다. 이어서 본 장에서는 다시점 비디오 복호화기(multi-view video decoder)를 제안한다. 다시점 비디오 복호화는 부호화 데이터를 복호화하는 H.264/AVC 디코더와 이를 제어하는 스케줄러(scheduler)로 구성된다.

스케줄링은 H.264/AVC의 베이스라인 프로파일과 메인

프로파일에 따라 다르며, 각 시점마다 균등한 디코딩 기회를 갖기 위해 각 시점의 디코딩 시간을 이용하는 프레임 스킵/드롭(frame skip/frame drop) 방법을 제안한다. 스케줄러는 각 시점 시퀀스의 디코더 이용률(decoder utilization)의 계산, 이에 따라 시점의 복호화 순서를 결정하는 시점 우선 순위 결정 (view priority decision) 및 프레임 스킵/드롭(frame skip, frame drop)으로 구성된다. 하나의 복호화기에서 일정 시간 내에 (예: 30 fps) 처리해야 하는 프레임 개수는 한정되어 있기 때문에, 스케줄러를 이용하여 각 시점의 복호화되는 프레임 개수를 효율적으로 조절한다. 프레임 드롭에서 발생하는 문제는 시각 영상 화질(visual image quality)의 저하 (degradation)인데, 효율적으로 드롭되는 프레임의 선택 방법을 조사한다^[16,17]. 마지막으로 베이스라인 프로파일과 메인 프로파일에서의 프레임 스킵 방법을 설명한다.

1. 제안하는 다시점 복호화기

다시점 복호화기의 구조는 그림 2와 같다. 입력으로 들어온 압축 비트열은 스트림 스플리터(Stream Splitter)에서 GOP 단위로 분리되어 해당 시점 버퍼 Buffer (V_k)에 저장된다. $k = 1, \dots, K$, K 는 카메라의 개수이다. 버퍼에 저장된 시점 영상들의 복호화는 스케줄러가 제어한다.

스케줄러는 각 시점 영상을 GOP 단위로 처리한다. GOP 단위로 각 시점의 복호화 시간(decoding time) D_k 를 구한 후에, D_k 로부터 디코더 이용률(Decoder Utilization) U_k 를 계산한다. U_k 값으로부터 다음 GOP에서 복호화해야 하는

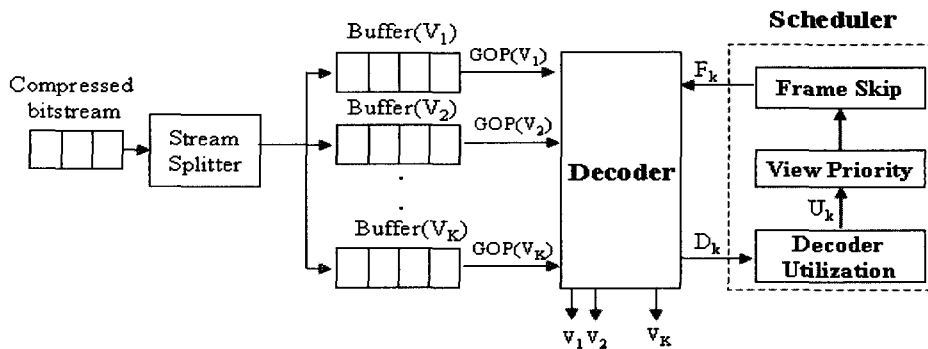


그림 2. 제안하는 다시점 복호화기의 블록도
Fig. 2. Block diagram of the proposed multi-view decoder

프레임 개수를 결정한다. 다음에는 프레임 개수가 적은 시점부터 다음 GOP에서의 시점 우선순위(View Priority)를 정한 후에 복호화를 시작한다. 즉, 각 시점이 동등한 복호화 시간을 가지도록 하여, 각 시점이 균등한 재생 기회를 가지도록 구현한다.

프레임 스킵(Frame Skip)에서는 각 시점의 이전 GOP의 디코더 이용률을 활용하여 프레임 스킵 방법을 결정한다. 다음 GOP의 복호화가 시작되기 전에 디코더 사용률이 높은 시점의 프레임을 상대적으로 많이 드롭함으로써 모든 시점들이 공평하게 복호화되도록 한다. 해당 버퍼의 모든 시점 프레임들의 디코딩이 끝나면 디코더 이용률을 다시 계산되어 다음 GOP의 프레임 스킵을 결정하게 된다.

제안 방법의 장점은 I 장에서 언급한 것처럼, 한 개의 디코더만을 사용한다는 장점이 있다. 단일 디코더를 사용함으로써 불필요한 시스템 리소스의 사용을 방지하며 제안된 프레임 스킵 방법의 효율적인 적용이 가능하다. 또한 K개의 연속적인 GOP에서 실제 복호화되는 프레임 개수는 시스템 성능에 따라 다르기 때문에, 최대 성능을 유지하기 위해서 적응적으로 조정이 가능하다.

2. 디코더 이용률

K 개의 시점 영상에서 각 시점은 N 개의 프레임으로 구성되어 있다면, GoGOP의 프레임 개수는 (K · N)이다. 따라서 (K · N) 프레임을 정해진 프레임 율(frame rate) (예: 24 fps, 30 fps)을 만족하며 모두 복호화하는 것은 시스템 성능 등의 문제로 어려움이 있다. 따라서 (K · N) 개의 프레임에서 일부는 드롭된다. 그림 3은 GoGOP 내에서의 복호화 시간의 예를 보여준다. D_k 는 GOP(V_k)의 프레임 디

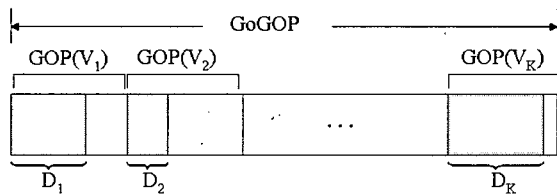


그림 3. GoGOP의 GOP들의 복호화 시간
Fig. 3. Decoding time of GOP in GoGOP

코딩 시간이면 시점 V_k 의 디코더 이용률 U_k 는 다음과 같이 계산된다.

$$U_k = \frac{D_k}{\sum_{k=1}^K D_k} \tag{1}$$

여기서 $\sum_{k=1}^K U_k = 1$ 이다.

3. 시점 우선 순위

현재 GOP에서의 U_k 값에 따라 다음 GOP에서 복호화되는 시점의 우선 순위는 매 GOP마다 동적으로 변화한다. U_k 가 작다는 것은 상대적으로 큰 시점에 비해 적은 개수의 프레임을 복호화되었음을 의미한다. 따라서 다음 GOP에서 우선적으로 복호화하여 재생의 프레임율을 동등하게 유지시켜 주는 역할을 한다. 표 1에서 View 4는 가장 작은 U 값을 가지므로 다음 GOP에서는 가장 큰 시점 우선 순위를 할당받는다.

표 1. U와 시점 우선 순위
Table 1. U and view priority

View	Current GOP's U	Next GOP's View Priority
1	0.35	4
2	0.24	3
3	0.20	2
4	0.21	1

4. 프레임 스킵

프레임 스킵은 디코더 이용률을 이용하여 각 시점마다 스킵되는 프레임의 개수를 결정하는 Skipped Frame Decision 과 복호화되는 프레임들을 선택하는 Decoded Frame Selection의 두 과정으로 구성된다. 그림 4의 블록도는 프레임 스킵 방법을 보여준다. Skipped Frame Decision에서는 각 시점의 입력값 U_k 로부터 각 시점에서 드롭되는 개수를 결정하는 값 l_k 를 계산한다. 계산된 값으로부터

Decoded Frame Selection은 연속 스킵(sequential skip)과 비연속 스킵(non-sequential skip)의 두 방법을 수행한다. 베이스라인 프로파일 시퀀스이면 연속 방법으로 복호화되는 프레임을 결정하고, 메인 프로파일이면 두 가지 스킵 방법을 모두 사용할 수 있다. 두 방법의 시각 영상 화질(visual image quality)은 비용함수(cost function)에 의해 판단된다 [16, 17]. 최종적으로 각 GOP의 프레임들중에서 실제 복호화되는 프레임 번호가 얻어진다.

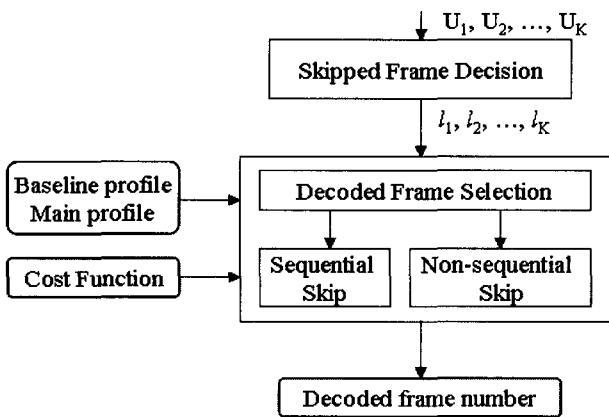


그림 4. 제안하는 프레임 스킵의 블록도
Fig. 4. Block diagram of the proposed frame skip

4.1 Skipped frame decision

N 개의 프레임으로 구성된 1 개의 GOP(V_k)가 있다고 하자. l 번째 프레임 F_l 을 드롭하면 다음 $F_{l+1}, F_{l+2}, \dots, F_N$ 도 드롭된다고 가정한다. 이 경우에 실제 디코딩되는 프레임 개수는 $(l-1)$ 이다. 따라서 GoGOP에서 총 디코딩되는 프레임 개수는 다음 식과 같다.

$$(l_1 - 1) + (l_2 - 1) + \dots + (l_K - 1) = \sum_{k=1}^K l_k - K, \quad (2)$$

여기서 l_k 는 GOP(V_k)에서 스킵되는 첫 프레임 번호이다. GoGOP에서 최적의 성능은 $(K \cdot N)$ 개의 프레임, 즉 모든 프레임을 디코딩 했을 때가 된다. 하지만, 실제 다시점 영상에서 모든 프레임의 디코딩이 가능한 시스템 성능은 불가능하기 때문에, GoGOP에서 디코딩되는 프레임 개수를 S

로 가정한다, $S = [0, K \cdot N]$. 이 값은 시스템 성능에 따라 증감이 가능하다. 따라서 식 (2)는

$$\sum_{k=1}^K l_k - K = S \quad (3)$$

이므로, 다음과 같은 결과를 얻는다.

$$\sum_{k=1}^K l_k = S + K \quad (4)$$

이 조건을 만족하는 l_k 값을 구하기 위하여 식 (2)을 활용한다. 계산된 U_k 에 따라 l_k 는 역선형 매핑을 이용하여 구한다. 즉 $U = [0, 1]$ 은 $k = [S, 1]$ 로 역 매핑된다. 수식으로 표현하면 다음과 같다.

$$\hat{l} = -(S-1)U + S \quad (5)$$

모든 K 개의 시점에 대해 합하면,

$$\sum_{k=1}^K \hat{l}_k = \sum_{k=1}^K [-(S-1)U + S] = S \cdot (K-1) + 1 \quad (6)$$

식 (4) 및 식 (6)으로부터 GOP(V_k)에서 스킵하는 첫 프레임 번호 l_k^* 는 다음과 같이 계산된다.

$$l_k^* = \lfloor (S+K) \frac{\hat{l}_k}{\sum_{k=1}^K \hat{l}_k} \rfloor, \quad (7)$$

여기서 $\lfloor \cdot \rfloor$ 는 floor 함수이다.

이렇게 구해진 l_k^* 값을 이용해 실제 프레임이 스킵하는 방법중에는 스킵번호 전까지 재생하고, 그 다음 프레임부터는 모두 드롭하는 간단한 방식이 있다. 또한 시각 영상 화질을 이용하여 드롭되는 영상들의 간격을 조절한 방법이 있다. 물론, 이 방식은 매 프레임마다 드롭 유무를 결정해야 하는 복잡성이 있다.

4.2 Decoded frame selection

복호화되는 프레임 개수가 결정이 되면, 전체 GOP 프레임 중에서 어떤 프레임을 선택하는 지를 결정한다. 그림 5에서 보면 (a)는 연속 스킵이고, (b)는 비연속 스킵이다.

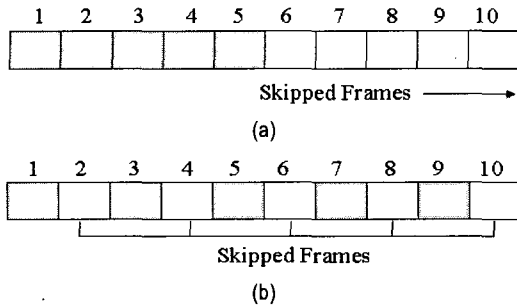


그림 5. 스킵 방법: (a) 연속 스킵 (b) 비연속 스킵
Fig. 5. Skip methods: (a) sequential skip, (b) non-sequential skip

그림 5(a)의 단점으로 bursty discard의 발생이 매 GOP마다 있기 때문에, 시각 영상 화질(visual image quality)이 낮아지는 문제가 발생한다. 예를 들어 매 2 프레임마다 1프레임을 드롭하는 그림 5(b)보다, GOP 프레임의 전반 1/2을 디코딩하고, 나머지 후반 1/2을 드롭하는 것이 재생 불연속으로 화질의 저하를 유발한다. 이러한 시각 영상 화질을 객관적으로 정량화한 것이 비용 함수(cost function)이다 [14, 15]. 비용 함수는 다음과 같이 계산된다; (1) frame i 가 연속적으로 발생하는 드롭 프레임에 속하고, 동시에 k_i 번째의 연속적인 프레임이라면, 해당 프레임의 cost c_i 는 k_i 이다, (2) 연속적이 아닌 2개의 드롭 프레임 간의 거리가 d_i 이면, cost c_i 는 $c_i = 1 + 1/\sqrt{d_i}$, 그리고, (3) 전체 비용 함수 $\phi(G)$ 는 $\phi(G) = \sum_{j \in F/G} c_j$ 이다. 여기서 F는 시퀀스에서 드롭되는 프레임들의 집합이고, G는 고려하는 GOP내의 전체 프레임의 집합이다. 그림 5의 (a) 와 (b)의 비용함수 값은 다음과 같이 계산된다.

$$\phi_A(G) = c_6 + c_7 + c_8 + c_9 + c_{10} = 1 + 2 + 3 + 4 + 5 = 14$$

$$\begin{aligned} \phi_B(G) &= c_2 + c_4 + c_6 + c_8 + c_{10} \\ &= 1 + (1 + \frac{1}{\sqrt{2}}) + 1 + (\frac{1}{\sqrt{2}}) + (1 + \frac{1}{\sqrt{2}}) = 6.82 \end{aligned}$$

따라서, $\phi_B(G)$ 가 $\phi_A(G)$ 보다 크다. 반대로 그림5(a)보다 (b)가 우수한 시각 영상 화질을 가진다. 같은 개수의 프레임을 드롭하지만 실제로 계산되는 비용은 다른 결과를 가져온다.

일반적으로 H.264/AVC의 베이스라인 프로파일은 I, P 프레임만 사용하므로 디코딩은 상대적으로 간단하다. P 프레임은 전방위 예측 (forward prediction)으로 부호화되기 때문에, 이전 참조 프레임이 존재해야 복원이 가능하다. 다음은 베이스라인 프로파일에서 사용하는 GOP의 구조를 보여준다. N=6이고, M=1이다.

$$I_1 P_1 P_2 P_3 P_4 P_5 I_2 P_6 P_7 P_8 \dots$$

프레임 드롭은 다음과 같이 수행된다.

- 1) 프레임이 드롭되면 다음 P_1, \dots, P_5 프레임 모두가 드롭된다.
- 2) P_1 이 드롭되면 P_2, \dots, P_5 프레임들이 모두 드롭된다. 따라서 P_i 프레임이 드롭되면, P_{i+1} 부터의 모든 프레임들이 드롭된다.
- 3) 따라서 현재 GOP에서 3 개의 프레임 및 다음 GOP에서 2 개의 프레임을 복호화하게 되면, $I_1 P_1 P_2$ 및 $I_2 P_6$ 가 재생된다.

베이스라인 프로파일은 앞서 설명한 연속 스킵만 사용이 가능하다. 그러므로, 복호화되는 프레임들이 연속적으로 발생하고, 마찬가지로 드롭되는 프레임들도 연속적으로 발생한다.

베이스라인 프로파일과 다르게, 메인 프로파일에서의 선택 순서는 프레임 타입 (frame type)에 따라 달라진다. 따라서 스킵되는 첫 프레임의 번호 l 이 정해지면, 즉 복호화되는 프레임 개수가 결정되면, 다음 단계는 프레임의 선택이다. 표 2는 GOP 프레임간의 예측구조에 따르는 프레임의 선택 순서의 예를 들어 보여준다.

Ex. 1은 GOP 구조는 IPPP...를 사용했을때의 결과로 베이스라인 프로파일에 해당된다. P 프레임은 이전 I와 P를 참조하여 복호화되므로 순차적으로 선택 번호가 결정된다.

스킵 번호 $l = 4$ 이면 I P1 P2 P3 가 선택된다.

Ex. 2는 main profile에서 $N=9, M=3$ 이다. 선택순서는 I P1 P2 B1 B3 B5 B2 B4 B6 이다. 스킵 번호 $l = 6$ 이면 I P1 P2 B1 B3 B5 가 복호화 프레임으로 선택된다.

Ex. 3은 main profile에서 $N=8, M=4$ 이다. 선택순서는 I P1 P4 B2 B5 B3 B6 이다. 스킵 번호 $l = 5$ 이면 I P1 P4 B2 B5 가 복호화 프레임으로 선택된다.

표 2. GOP 프레임의 선택 순서
Table 2. The selection order of GOP frames

Ex 1	Baseline profile (N=6,M=1)	I	P1	P2	P3	P5	P6				
	Selection order	1	2	3	4	5	6				
Ex 2	Main profile (N=9,M=3)	I	B1	B2	P1	B3	B4	P2	B5	B6	
	Selection order	1	4	7	2	5	8	3	6	9	
Ex 3	Main profile (N=8,M=4)	I	B1	B2	B3	P1	B4	B5	B6		
	Selection order	1	3	5	7	2	4	6	8		
Ex 4	Main profile (N=10,M=5)	I	B1	B2	B3	B4	P1	B5	B6	B7	
	Selection order	1	3	5	6	9	2	4	6	8	10

IV. 실험 결과

사용된 실험 다시점 테스트 동영상은 일본 KDDI에서 제공한 aquarium, flamenco, objects, 및 race 등이다^[5]. 그림 6은 사용된 영상들을 보여주고, 영상의 해상도는 모두 320x240이다. H.264/AVC JM(Joint Model) 7.6 참조 소프트웨어를 사용하여 구현하였으며, bit-rate는 512Kbps, QP 값은 28로 고정했다. Intra Decoded Refresh(IDR)은 0으로 하였다. IDR로 코딩된 픽처는 수신측에서의 압축 스트림에 대한 임의의 접근을 위해 사용하며 IDR로 코딩된 프레임이 오면 디코더의 모든 버퍼를 초기화한다. 본 논문에서는 각 프레임이 속한 뷰를 알아내기 위하여 슬라이스 헤더의 frame_num를 이용하였는데 IDR은 frame_num 파라미터를 0으로 셋팅하기 때문에 본 논문에서는 사용하지 않았다.

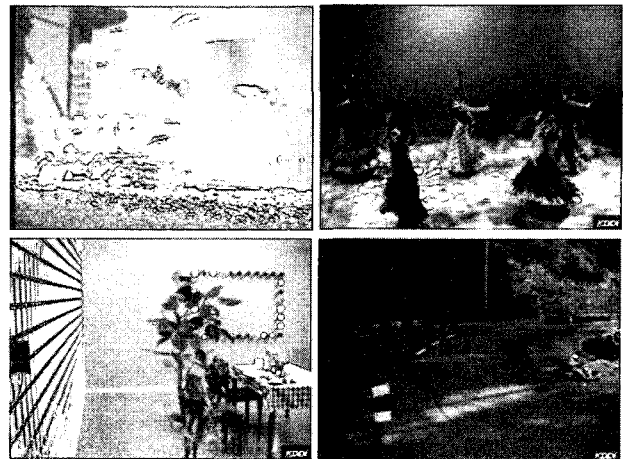


그림 6. KDDI 실험 영상 시퀀스
Fig. 6. Test image sequences of KDDI

믹싱 테스트 동영상은 베이스라인 및 메인 프로파일에서 다양한 (K, N, M) 값을 혼합하여 제작하였다. II 장에서 소개한 방법으로 믹싱 영상을 제작하고, H.264/AVC로 부호화하였다. 표 3은 베이스라인 프로파일 및 메인 프로파일의

표 3. 실험 베이스라인 프로파일 및 메인 프로파일 동영상. N_T = 전체 영상 개수
Table 3. Test baseline and main profile sequences. N_T = total frames

Test sequence		K	N	M	N_T	Encoded file
Baseline Profile	b11.yuv	1	4		100	b11.264
	b12.yuv	1	12		300	b12.264
	b21.yuv	2	12		300	b21.264
	b22.yuv	2	24		300	b22.264
	b41.yuv	4	12		400	b41.264
	b42.yuv	4	24		400	b42.264
	b81.yuv	8	12		500	b81.264
b82.yuv	8	24		500	b82.264	
Main Profile	m11.yuv	1	4	1	80	m11.264
	m12.yuv	1	12	2	120	m12.264
	m21.yuv	2	12	3	240	m21.264
	m22.yuv	2	12	2	240	m22.264
	m23.yuv	2	12	6	240	m23.264
	m41.yuv	4	12	6	480	m41.264
	m42.yuv	4	12	4	480	m42.264
	m43.yuv	4	16	8	512	m43.264
	m81.yuv	8	12	3	576	m81.264
m82.yuv	8	24	6	576	m82.264	

실험 동영상을 보여준다. b21.yuv는 flamenco1.yuv, aquarium1.yuv를 이용한 믹싱 시퀀스이다. 시점 개수 K는 2이며, N은 12이다. 만들어진 영상 개수는 300 장이다. 제작된 믹싱 동영상을 H.264/AVC로 부호화하여 b21.264 압축 화일을 생성하였다. 믹싱 입력 영상을 서로 다른 시퀀스의 조합으로 제작한 이유는 H.264/AVC의 특수한 참조 프레임의 특성으로 인해 영상 화질의 저하를 검증하기 위해서이고, 실제 같은 유형의 시퀀스의 처리도 가능하다. m21.yuv는 flamenco1.yuv, aquarium1.yuv를 이용한 믹싱 영상으로, K=2, N=12, M=3이다. 만들어진 영상 개수는 240 장이다. 얻어진 믹싱 동영상을 H.264/AVC로 부호화하여

m21.264 압축 파일을 생성하였다. 다른 압축 화일들도 유사하게 제작하였다.

표 4는 4개의 베이스라인 프로파일의 복호화 결과를 보여준다. 실험영상의 각 View의 첫 번째 GOP는 시스템의 초기화를 위해 사용하며 다음 GOP의 프레임스킵 예측을 위해 사용하지 않는다. b21.264는 2개의 시점으로 구성되며, N은 12이다. 10개의 GOP의 디코딩 결과 I 프레임의 경우 9개가 디코딩 되었으며, V₁과 V₂의 디코딩 타임 및 디코더 이용률은 비슷하게 사용된 것을 알 수 있다. 이에 따른 이용률 역시 50% 정도를 나누어서 사용한 것을 관찰할 수 있다. 다른 3개의 실험 결과도 유사한 결과를 보여준다.

표 4. 실험 베이스라인 프로파일 시퀀스의 성능
Table 4. The performance of test baseline profile sequences

b21.264	View		V ₁	V ₂
	Decoded Frames	I	9	9
		P	53	55
		Total	62	64
	Decoding Time (msec)		5753	5597
Decoder Utilization		0.506	0.493	

b41.264	View		V ₁	V ₂	V ₃	V ₄
	Decoded Frames	I	9	9	9	9
		P	27	27	27	27
		Total	36	36	36	36
	Decoding Time (msec)		3362	3295	3417	3203
Decoder Utilization		0.253	0.248	0.257	0.241	

b81.264	View		V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈
	Decoded Frames	I	4	4	4	4	4	4	4	4
		P	6	6	6	6	6	6	6	6
		Total	10	10	10	10	10	10	10	10
	Decoding Time (msec)		954	938	907	874	860	968	892	905
Decoder Utilization		0.130	0.128	0.124	0.119	0.117	0.132	0.122	0.124	

b82.264	View		V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈
	Decoded Frames	I	4	4	4	4	4	4	4	4
		P	8	8	8	8	8	8	8	8
		Total	12	12	12	12	12	12	12	12
	Decoding Time (msec)		1547	1531	1580	1423	1327	1576	1406	1515
Decoder Utilization		0.129	0.128	0.132	0.119	0.111	0.132	0.118	0.127	

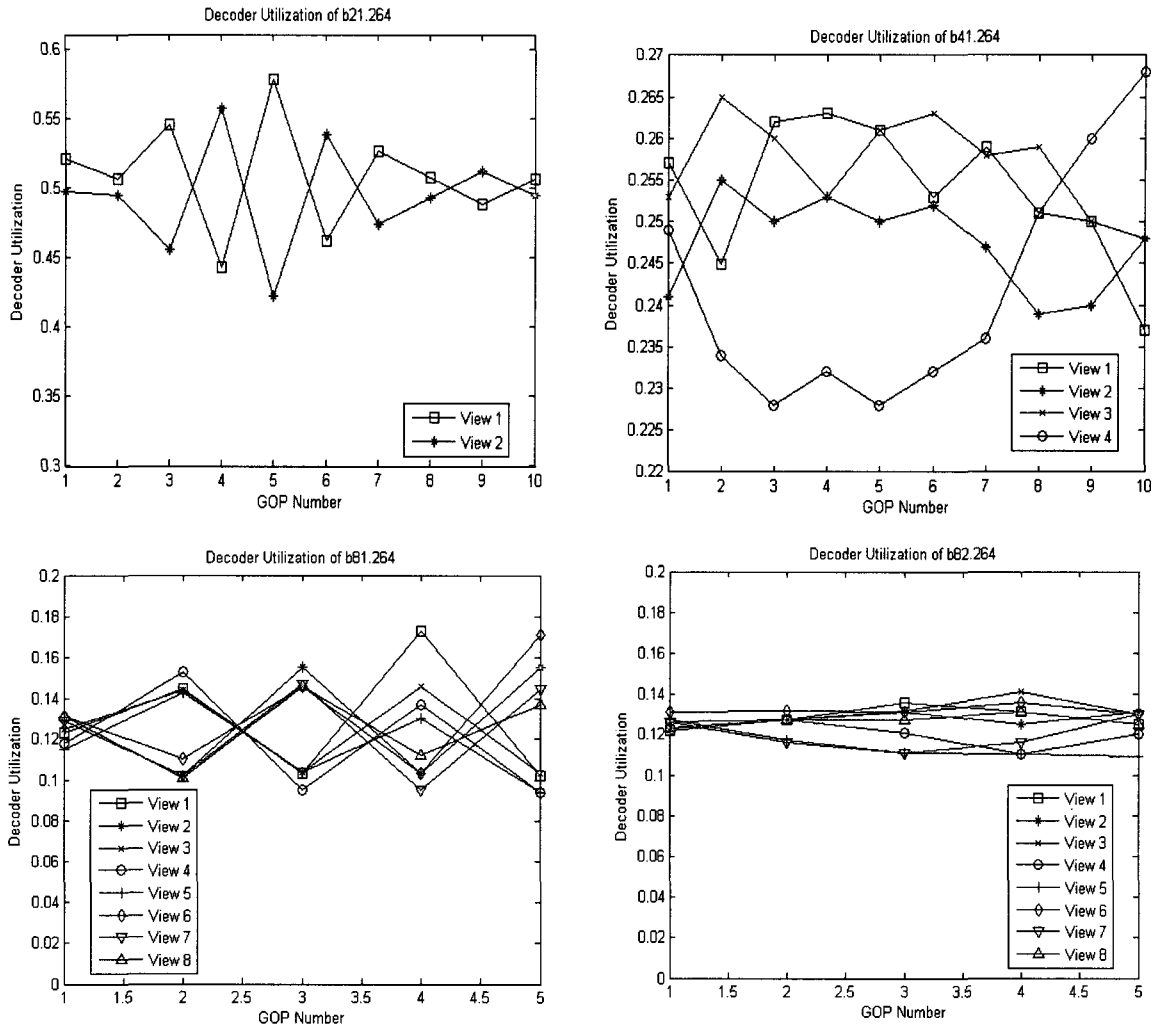


그림 7 베이스라인 프로파일 실험 시퀀스 GOP들의 디코더 이용률의 변화
 Fig. 7. The variation of decoder utilization for baseline profile sequences

다. 그림 7은 베이스라인 프로파일 시퀀스들의 GOP들의 디코더 이용률의 변화를 보여준다. b21.264는 복호화 과정에서 유사한 값으로 수렴하는 것이 관찰된다. 다른 시퀀스들은 어느 정도 변화가 있다. b41.264의 경우는 4개의 view 이므로, 평균적으로 0.25의 디코더 이용률을 가지게 되는데, 그래프는 0.25에서 크게 다르지 않는 범위에서 디코더 이용률을 지속적으로 유지한다. 다른 두 시퀀스도 유사하게 관찰된다.

표 5는 메인 프로파일 실험 영상을 사용해 부호화한 영

상의 복호화 성능을 보여준다. 실험 영상 m21.yuv는 2개의 시점으로 구성되며, $N=12$, $M=3$ 이다. GoGOP에서 복호화되는 프레임 개수 S 는 N 값을 사용하였다. V_1 은 I, P, B 프레임이 각각 9, 25, 17개가 복호화되었다. V_2 도 유사한 결과를 보여준다. 디코더 이용률은 0.53 및 0.46으로 약간의 차이를 보이고 있는데, 이는 마지막 복호화되는 B 프레임의 영향 때문이다. 비용함수 값은 연속스킵과 비연속스킵이 각각 254, 117.9 (V_1), 202 및 90.6 (V_2)으로 비연속스킵이 상대적으로 작은 값, 즉 좋은 시각 화질을 얻게 된다.

표 5. 실험 메인 프로파일 시퀀스의 성능

Table 5. The performance of test main profile sequences

m21.264	View		V ₁	V ₂
	Decoded Frames	I	9	9
		P	25	26
		B	17	22
		Total	51	57
	Decoding Time (msec)		5044	4574
	Decoder Utilization		0.538	0.461
	Cost	Sequential	254	202
Non-sequential		117.9	90.6	

m22.264	View		V ₁	V ₂
	Decoded Frames	I	9	9
		P	30	33
		B	12	15
		Total	51	57
	Decoding Time (msec)		5045	4562
	Decoder Utilization		0.526	0.473
	Cost	Sequential	270	231
Non-sequential		174.7	147.7	

m23.264	View		V ₁	V ₂
	Decoded Frames	I	9	9
		P	9	9
		B	37	35
		Total	55	53
	Decoding Time (msec)		4609	4702
	Decoder Utilization		0.487	0.512
	Cost	Sequential	187	200
Non-sequential		113.8	120.8	

m41.264	View		V ₁	V ₂	V ₃	V ₄
	Decoded Frames	I	9	9	9	9
		P	9	9	9	9
		B	9	9	9	9
		Total	27	27	27	27
	Decoding Time (msec)		2842	2561	2778	2484
	Decoder Utilization		0.264	0.241	0.262	0.231
	Cost	Sequential	405	405	405	405
Non-sequential		231.3	231.3	231.3	231.3	

m81.264	View		V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈
	Decoded Frames	I	5	5	5	5	5	5	5	5
		P	3	3	3	3	2	2	2	3
		B	0	0	0	0	0	0	0	0
		Total	8	8	8	8	7	7	7	8
	Decoding Time (msec)		702	719	765	704	767	890	860	717
	Decoder Utilization		0.111	0.115	0.122	0.113	0.128	0.149	0.144	0.114
	Cost	Sequential	297	297	297	297	308	308	308	297
		Non-sequential	251.1	251.1	251.1	251.1	277.4	277.4	277.4	251.1

그림 8은 메인 프로파일 시퀀스들의 GOP들의 디코더 이용률의 변화를 보여준다. 그래프는 베이스라인 프로파일과

유사한 결과를 보여준다.

그림 9는 S에 따른 두개의 메인 프로파일 시퀀스

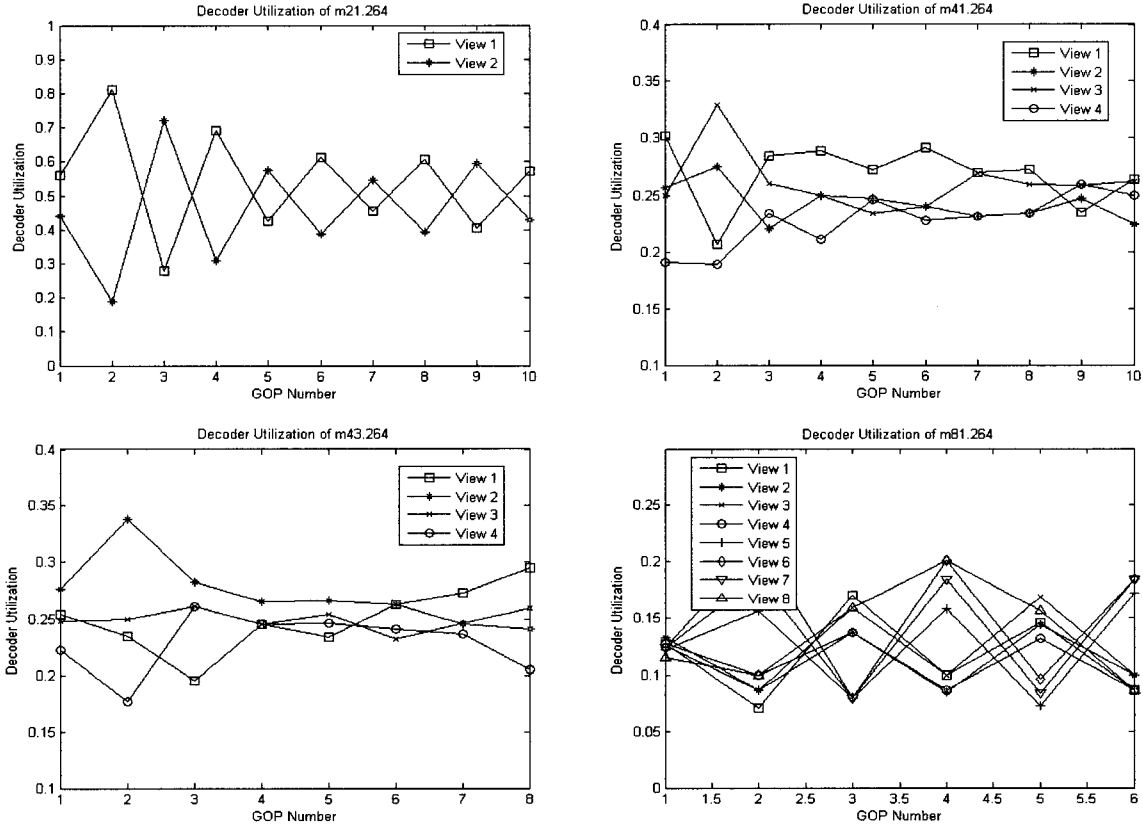


그림. 8 메인 프로파일 실험 시퀀스 GOP들의 디코더 이용률의 변화
 Fig. 8. The variation of decoder utilization for main profile sequences

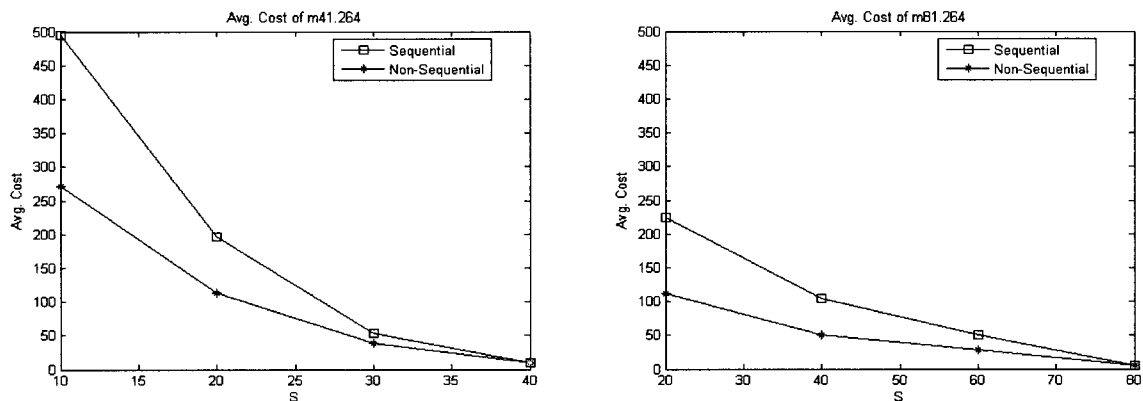


그림 9. 디코딩 가능한 최대 프레임 개수 S에 따른 메인 프로파일 시퀀스의 비용 함수의 변화
 Fig. 9. The variation of cost functions for main profile sequences with respect to S

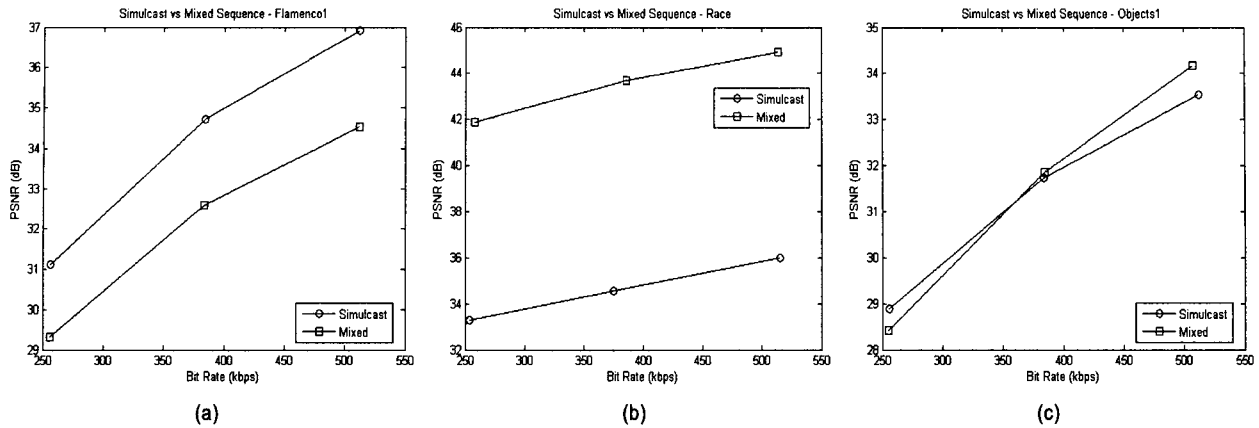


그림 10. 동시방식 시퀀스와 비디오 믹싱 시퀀스의 성능 비교
 Fig. 10. Performance comparison of simulcast sequence and video-mixed sequence

m41.264와 m81.264의 비용 함수의 변화를 그래프로 보여 준다. 예측되는 것처럼, S가 증가하면 비용은 선형 비례적으로 감소하는 것이 관찰된다. GoGOP의 전체 프레임 개수는 $K \cdot N$ 이므로, m41.264와 m81.264는 각각 48장 및 96장이다. S=40 및 80에서 비용 함수는 근사적으로 0값을 가지게 된다.

마지막으로 그림 10은 동시 시퀀스와 비디오 믹싱 시퀀스의 Rate-Distortion 커브를 보여준다. GOP 구조는 모두 IBBP... 로 N은 12이다. 그림 10 (a)은 flamenco1의 실험영상을 사용하였는데 두개의 시퀀스들을 믹싱하여 믹싱 시퀀스를 만들었다. 동일 비트율에서 PSNR(Peak-to-Signal Noise Ratio)이 평균 2 dB정도 낮은 결과를 보여준다. 그림 10 (b)은 race1에서 4 개의 시퀀스로 얻은 믹싱 시퀀스의 실험 결과를 보여주는데, 반대로 믹싱 시퀀스에서 평균 8 dB 우수한 결과가 얻어진다. 그림 10 (c)의 objects1은 8 개의 sequence를 사용했는데, 두 방법이 유사한 PSNR 값을 보여주고 있다. 실험 결과는 부호화 성능은 다시점 비디오의 영상 운동과 관련이 있는 것을 보여준다.

V. 결론 및 향후 연구

본 논문에서는 단일 H.264/AVC 엔코더 및 디코더를 이용하는 다시점 비디오의 효율적인 부호화 및 복호화 시스

템을 제안하였다. 주요 목적으로 다시점 비디오가 요구하는 기존 고성능 시스템과 비교하여 단일 엔코더 및 디코더를 사용하는 것이다. 이를 위해 GOP 단위의 비디오 믹싱 방법을 제안하였다. 비디오 믹싱을 통해 단일 인코더와 단일 디코더만으로 구현 가능한 다시점 비디오의 효율적인 시스템을 제안하였으며, 따라서 구현한 시스템은 기존의 H.264/AVC 코덱의 수정 없이 제안 방법의 적용이 가능하다. 또한 H.264/AVC에서 사용하는 두가지 프로파일에 적합한 스케줄링 기법을 제안하고, 비용함수를 이용하는 선택적 프레임 스킵 방법을 제안하였다. 실험에서는 다양한 다시점 비디오를 이용하여 각 시점 시퀀스가 균등하게 재생되는 것을 보여주었고, Rate-Distortion 커브를 활용하여 동시 방식과의 성능을 비교하였다.

향후 연구로는 1) 제안 방법은 1개의 디코더를 사용하지만, $N (\geq 2)$ 디코더를 사용하면 더 우수한 성능이 얻어진다. 즉, 디코더의 개수를 증가하면 높은 프레임률을 얻을 수 있고, 또한 제안 방법의 적용도 용이하다; 2) 향후 다시점 비디오 시스템에서는 인터넷 등에서 클라이언트가 원하는 시점영상(들)을 요청하면, 서버는 해당 시점영상(들)만을 전송하는 응용도 중요시되고 있다. 이를 위해서 제안 방법을 N 개의 시점 영상 중에서 해당 시점들만 부호화가 가능하도록 설계하는 것이 필요하고, 또한 MPEG-21 Multimedia Framework의 DIA(Digital Item Adaptation)의 개념을 활용하는 응용 개발도 연구 중이다.

참 고 문 헌

[1] 허남호, 이광순, 유응식, 이진환, 안충현, "An HDTV-compatible 3DTV broadcasting system," ETRI Journal, Vol. 26, No. 2, Apr. 2004.

[2] 강훈중, 김대회, 윤국진, 조숙희, 안충현, "3차원 비디오 처리 기술 동향," 전자통신동향분석, 제19권 제4호, 2004년 8월.

[3] 이영렬, 이용기, "3DAV 기술 동향" 전자공학회지, 제32권 9호, 2005년 9월.

[4] S. Lee, K. Lee, C. Han, J. Yoo, M. Kim and J. Kim, "View-switchable stereoscopic HD video over IP networks for next generation broadcasting," SPIE Photonic East, Vol. 6016, Oct. 24-26 2005.

[5] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding approaches for end-to-end 3D TV systems," Picture Coding Symposium, Dec. 2004.

[6] MPEG/ISO/IEC JTC1/SC29/WG11 "Call for Proposals on Multi-view Video Coding," N7327, Poznan, Poland, July 2005.

[7] MPEG/ISO/IEC JTC1/SC29/WG11 "Description of Core Experiments in MVC," N7798, Bangkok, Thailand, Jan. 2006.

[8] MPEG/ISO/IEC JTC1/SC29/WG11 "Multiview Coding using AVC," M12945, Bangkok, Thailand, Jan. 2006.

[9] 김태완, 김제우, 송혁, 김용환, 정혁구, 최병호 "다시점 영상 전송을 위한 효율적인 (De)Steamer," 16회 신호처리학술대회논문집, 16권 1호, 2003.

[10] Q. Zhang, W. Zhu and Y-Q Zhang, "Resource allocation for multimedia streaming over the Internet," IEEE Trans. on Multimedia, Vol. 3, No. 3, Sep. 2001.

[11] 이용주 "다시점영상처리를 위한 역다중화기의 설계, 연세대학교 전기전자공학과 석사논문.

[12] Ulrich Fecker and André Kaup, "Transposed Picture Ordering for Dynamic Light Field Coding," ISO/IEC JTC1/SC29/WG11 MPEG2004/M10929, Redmond USA, July 2004.

[13] I. Richardson, H.264 and MPEG-4 video compression, Wiley, 2003.

[14] 정제창 (역), H.264/AVC 비디오 압축 표준, 홍릉과학출판사.

[15] MPEG/ISO/IEC JTC1/SC29/WG11, "Requirements on Multi-view Video Coding," N6501, Redmond, USA, July 2004.

[16] K. Chebrolu and R. Rao, "Selective frame discard for interactive video," IEEE International Conference on Communications, Vol. 7, pp. 4097-4102, June 2004.

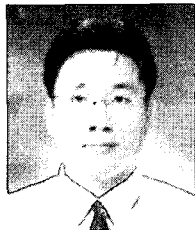
[17] Z.-L. Zhang, S. Nelakuditi, R. Agarwal, and R. Tsang, "Efficient selective frame discard algorithms for stored video delivery across resource constrained networks," IEEE INFOCOM, 1999.

저 자 소 개



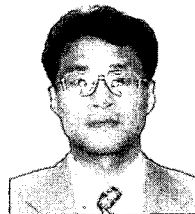
김 학 수

- 2004년 2월 : 강원대학교 정보통신공학과 학사
- 2006년 2월 : 강원대학교 컴퓨터정보통신공학과 석사
- 주관심분야 : 비디오 신호 처리, 미디어 스트리밍, 컴퓨터 그래픽스



김 윤

- 1993년 : 고려대학교 전자공학과 학사
- 1995년 : 고려대학교 전자공학과 석사
- 1995년 ~ 1999년 : LG전자 안양연구소 주임연구원
- 2003년 : 고려대학교 전자공학과 박사
- 2004년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 조교수
- 주관심분야 : 디지털 영상 신호 처리, 영상 압축, 영상 통신



김 만 배

- 1983년 : 한양대학교 전자공학과 학사
- 1986년 : University of Washington (Seattle) 전기공학과 석사
- 1992년 : University of Washington (Seattle) 전기공학과 박사
- 1993년 : Georgetown University, Visiting Scientist
- 1996년 : University of Rochester, Visiting Scientist
- 1992년 ~ 1998년 : 삼성종합기술원 신호처리/소프트웨어 Lab 수석연구원
- 1998년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 부교수
- 주관심분야 : 입체영상처리, 실감방송시스템, Image/Video 신호 처리