

일반논문-04-09-4-10

## 비트 스트림 지도를 이용한 고해상도 영상의 효율적인 지역복호화

박성원<sup>a)\*</sup>, 원종우<sup>a)</sup>, 이선영<sup>a)</sup>, 김욱중<sup>b)</sup>, 김규현<sup>b)</sup>, 장의선<sup>a)</sup>

### Efficient Local Decoding Using Bit Stream Map for High Resolution Video

Sungwon Park <sup>a)\*</sup>, Jongwoo Won<sup>a)</sup>, Sunyoung Lee<sup>a)</sup>, Wookjoong Kim<sup>b)</sup>, Kyuheon Kim<sup>b)</sup>, Euee S Jang<sup>a)</sup>

#### 요 약

본 논문은 고해상도 영상 서비스를 위하여 부분 영상의 공간임의접근(Spatial Random Access)을 효과적으로 구현하기 위하여 비트 스트림 지도 (B-map)를 이용한 부호화 방법을 제안한다. 다양한 환경에서, TV 브라운관, 모니터, PDA용 LCD와 같은 표준 디스플레이 장치들은 디지털 시네마 또는 파노라마 영상과 같은 고해상도 영상을 위한 충분한 해상도와 프레임 크기를 지원하지 못하는 실정이다. 따라서 사용자는 다운샘플링된 저해상도의 영상이나 디스플레이의 크기에 맞춰진 부분 영상만을 볼 수밖에 없다. 제안한 B-map은 부호화 과정에서 생성된 비트 스트림에 대한 블록 단위의 배열 정보를 가지고 있으며, 복호 시 해당 영상의 비트 스트림보다 앞서 보내져 참조됨으로써 효과적인 지역 복호(Local decoding)를 가능하게 한다. B-map을 이용한 방식은 타일링을 비롯한 비교 대안들에 비해 효율적인 지역 복호화를 구현할 수 있음을 실험을 통해 확인하였다.

#### Abstract

In this paper, we introduce a novel coding method to efficiently enable spatial random access for high resolution video. In terms of resolution and display size, standard display devices (such as cathode-ray tubes, monitors, PDAs, and LCDs) do not sufficiently support high resolution video such as digital cinema and panoramic video. Currently, users have no choice but to view video at lower resolution as a result of down-sampling, or only a partial region of the video due to display size limitations. Our proposed method, which we call the B-map, represents the set of starting locations of the coded segments in a picture frame. This information, or B-map, is first sent to the decoder prior to the coded data stream of the frame and is then used for fast local decoding. To test our method, we compare our B-map with JPEG tiling and the JPEG Resynchronization marker. Experimental results show that the proposed coding method requires less overhead than existing methods during the same decoding time. The results show promise for future panoramic or digital cinema applications.

Keywords : B-map(Bit Stream Map), Local Decoding, Tiling, Resynchronization Marker, Spatial Random Access

## I. 서론

정보화 사회를 가능하게 한 디지털 기술의 발전은 산업

전반에 걸쳐 많은 영향을 미치고 있다. 사람과 컴퓨터, 그리고 사물을 하나로 연결하는 유비쿼터스 환경의 도래는 가전, 사무기기, 게임기 등에 다양한 미디어를 출현시키고 있다. 고해상도 영상에 대한 소비자들의 욕구 역시 나날이 증가하여 디지털 TV, HDTV, HD-DVD, 디지털 시네마 등과 같은 고해상도 디스플레이에 대한 수요가 증가하고 있으며, 디스플레이의 기술 역시 많은 발전을 이루고 있다.

a) 한양대학교 정보통신 대학원 디지털 미디어 연구실  
Digital Media Laboratory, Hanyang University.

b) ETRI 디지털방송연구단 방송미디어 연구그룹  
ETRI Digital Broadcasting Research Division Broadcasting Media  
Research Group

일례로, 현재까지 가정에서 일반적으로 사용되는 NTSC 방식의 아날로그 TV가 640×480의 해상도를 지원하는 반면, 최근의 HDTV의 경우에는 1920×1080의 해상도까지 지원한다. 뿐만 아니라, 미국 중심의 영화업계에서는 제작 및 배급 측면의 효율을 높이기 위해 DCI(Digital Cinema Initiative)를 구성하여 영화 전용 디지털 포맷을 기획하고 있는데, 이 때 사용되는 디지털 시네마의 영상 크기는 4K×2K이 기본이 되고 있다. 이와 같은 영상의 고해상도, 디스플레이의 다양화라는 현재의 흐름에서 과연 현재의 압축 기술이 고해상도 영상을 효율적으로 처리하고 있는지에 대한 고찰이 필요하다.

고해상도 영상의 압축 문제는 전통적인 영상 압축 방법과 크게 다르지 않지만, 일반 영상보다 큰 용량으로 인하여 보다 신속한 부호화/복호화 처리가 요구된다. 예를 들어, 4K×2K 이상 고해상도 영상의 복호화 시간은 352×240 크기의 프레임을 복호화하는 시간의 대략 100배로 추산할 수 있다. 현재 HDTV급이 약 2k×1k 정도의 해상도를 지원하지만 이 보다 큰 영상을 처리하기에는 부호화/복호화 시간과 물리적인 디스플레이의 크기 측면에서 완벽하게 처리한다고 말하기는 어렵다. 현재의 어플리케이션이 처리 가능한 방법으로는 다운 샘플링된 저해상도의 영상이나 원영상의 일부분만을 복원하여 보여주는 방법들이 있다. 물론 고해상도 영상의 압축 문제는 추후 디스플레이의 기술과 압축 기술이 발전함에 따라 해결되리라 생각되지만, 영상의 일부분만을 복원, 검색, 편집 가능하도록 하는 공간임의접근(spatial random access) 방법은 이를 위한 해결책 중 하나로 제시 될 수 있다. 예를 들어 사용자는 파노라마 영상과 같은 고해상도 영상에서 필요한 부분만을 복원, 검색, 편집이 가능할 것이다. 또한 일반적인 영상일지라도 PDA와 같이 작은 크기의 디스플레이에 복원되어질 경우, 이와 같은 접근 방법이 적용 가능하다.

지금까지 고해상도 영상 자체에 대한 연구는 폭넓게 이루어지지 않았다. 즉, 고해상도 영상을 일반 영상과의 차별성을 두지 않고, 기존의 압축 알고리즘의 일부 기능을 이용한 압축이 이루어졌던 것이 사실이다. 현재까지 이루어진 어플리케이션에 관한 연구 내용으로는 디스플레이를 머리에 착용한 상태에서 시선의 움직임에 따라 화면을 이동시켜 주는 시스템이나<sup>[1]</sup> 파노라마 영상을 위한 이동, 회전, 줌 등의 기능을 갖춘 Apple사의 QuickTime VR<sup>[2]</sup> 등을 예로 들 수 있다. 그러나 이들은 어플리케이션 기능 중심의

연구로써 고해상도 영상을 처리하기 위한 핵심적인 내용이라 말할 수 없다.

본 연구와 관련하여 고해상도 영상의 공간임의접근에 대하여 다룬 최근의 연구가 있다<sup>[3]</sup>. [3]에서는 압축 툴로서 JPEG-2000 타일링 기법을 사용함으로써 기본적으로 전체의 큰 영상을 작은 조각 영상으로 나누어 처리하게 된다. 이 방법은 클라이언트 컴퓨터 사용자의 화면 움직임이 input으로써 MPEG DMIF(Delivery Multimedia Integration Framework) 기반의 MPEG-4클라이언트-서버 아키텍처를<sup>[4]</sup> 통하여 실시간으로 서버 컴퓨터에 전달되게 한다. 그리하여 서버는 사용자의 움직임에 따라 복호에 필요한 타일만을 전송하게 됨으로써, 전체 영상을 보내는 것에 비해 효율적인 공간임의접근을 수행할 수 있게 된다. [3]은 실험에서, 13200×2600의 파노라마 영상을 26×6 (508×433 pixels/image)개의 타일로 나누어 처리하였다. 이를 공간임의접근이 가능한 실린더형 디스플레이에 적용 시, 약 5°/sec의 속도로 구현됨을 보여주었다. 그러나 이 속도로 360° 회전한다면 72초가 걸리는데, 이는 사용자의 입장에서 분명 만족스럽지 못한 결과일 것이다. 이러한 문제는 단순히 서버와 클라이언트 사이의 통신 속도의 문제라기보다는 복호화 알고리즘에서 그 원인을 찾을 수 있다. 즉, [3]의 시스템에서 사용한 타일의 크기는 508×433 pixel을 사용함으로써 복호화 시, 필요 이상의 많은 데이터를 전송하고 복호화하게 된다. 그렇다고 타일의 크기를 작게만 만드는 것이 해결책은 아니다. 타일의 크기가 작아질수록 타일 개수는 증가하여 각 타일의 메타 정보에 의한 오버헤드의 증가라는 새로운 문제를 만들기 때문이다.

본 논문은 고해상도 영상의 공간임의접근의 성능을 향상시키기 위하여 B-map을 이용한 개선된 접근 방법을 제안한다. 기본적으로 큰 영상을 작은 타일로 쪼개어서 처리하는 방법은 기존의 방법과 같지만 타일링의 단점인 오버헤드의 증가를 최소화하면서 복호화 시간을 줄이도록 설계되었다.

II 장에서는 이미 공간임의접근을 구현하기 위해 사용하고 있거나 이용할 수 있는 JPEG tiling과 JPEG resynchronization marker에 대하여 기술하였다. III 장에서 제안하는 B-map에 대하여 설명하였고 IV 장에서는 이들의 성능을 비교한 실험 결과를 나타내었으며, 마지막으로 V 장에서는 결론과 향후 과제에 대하여 논의해 보기로 한다.

## II. 공간임의접근을 구현하는 기존의 방법들

### 1. 공간임의접근을 구현하기 위한 접근 방법

고해상도 영상의 실시간 복호화/복원이 가능한 공간임의 접근을 구현할 시, 전체 영상을 처리하기 보다는 타일이나 매크로 블록과 같이 나누어진 각각의 작은 영상 단위로 다루는 것이 여러모로 편리할 것이다. 복호기는 사용자가 필요로 하는 부분을 포함하는 타일/매크로 블록만을 복호화하게 된다. <그림 1>은 그러한 예를 보여준다. 그림에서 Visibility 센서는 사용자의 움직임입을 감지, 예측하여 복호에 필요한 타일/매크로 블록을 미리 선정함으로써 복호화하는 시간을 벌어주는 역할을 담당하게 된다. 예를 들어, 352×240 크기의

디스플레이에 400×300의 크기로 타일링된 이미지를 보여준다고 가정해보자. 사용자의 시선이 이동함에 따라 필요한 타일의 개수는 1, 2, 또는 최대 4개까지 필요하게 될 것이다. Visibility 센서의 범위에 따라 그 이상의 개수가 필요할지도 모른다. 이러한 경우, 타일의 크기가 작으면 작을수록 불필요하게 복호화되는 부분을 최소한으로 줄일 수 있을 것이란 사실은 명백하다. 그러나 타일의 크기를 줄이는 문제는 또 다른 문제를 야기한다. 타일마다 추가되는 오버헤드의 양도 증가된 타일의 개수만큼이나 증가하기 때문이다.

우리는 간단한 실험을 통하여 고해상도 영상을 압축하기 위한 툴로써 JPEG을 사용하기로 결정하였다. JPEG을 사용한 주된 이유로는 다음과 같은 것들이 있다.

- 타일링과 Resynchronization Marker는 JPEG의 표준으로써 손쉽게 사용할 수 있다.
- 현재 4k×2k의 고해상도 이미지까지도 처리 가능하다.
- 빠른 복호화의 구현에 유리한 Intra Coding을 지원한다.

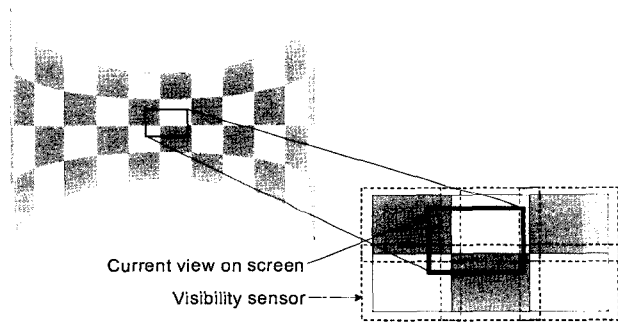


그림 1. 공간임의접근에 사용된 타일의 예([3]의 논문에서 인용)  
Fig. 1. Tiling example used for spatial random access

본 논문에서는 공간임의접근을 위한 압축 방법으로써 Inter Coding 방식은 고려하지 않았다. Inter Coding 방식은 압축 성능을 향상시키기 위하여 현재 프레임의 전후의 이미지를 이용하게 되는데, 이를 공간임의접근을 위하여 복호화하기 위해 많은 자원과 시간이 필요하기 때문이다. 따라서 실험 결과는 Intra 방식을 사용하는 경우이다. <그림 2>는 여러 가지 표준 압축 툴에 대한 PSNR(peak signal to

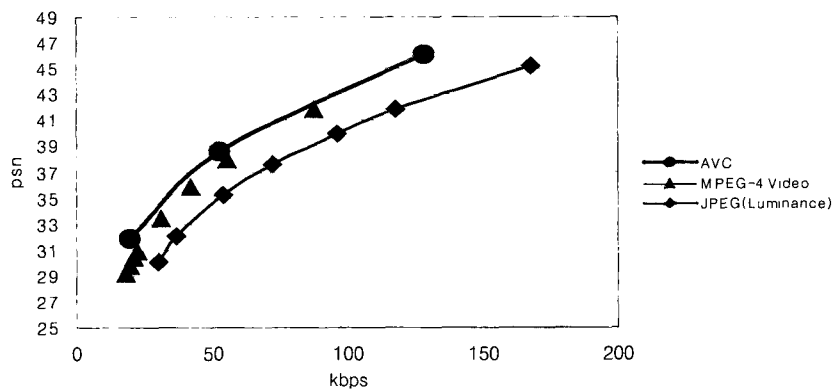


그림 2. 여러 가지 코덱에서의 PSNR 비교  
Fig. 2. PSNR comparison of codecs

표 1. 한 프레임에 걸리는 복호화 시간  
Table. 1. Decoding time per frame

압축방식	복호화 시간(sec)
JPEG(Intra 방식)	0.229
MPEG (Intra 방식)	0.348
AVC (Intra 방식)	4.275

noise ratio)을 bit-rate 별로 나타낸 그래프이다. <표 1>의 실험에 사용된 영상은 800×1200 크기의 영상을 QP값 28로 압축한 영상이며, 복호화 시간은 100 프레임을 복호화하는데 걸리는 시간의 평균값으로 계산하였다. <그림 2>에서 보는 바와 같이 압축률은 AVC가 가장 뛰어났으며, 다음으로 MPEG Video Coding 방식이 우수함을 알 수 있다. 그러나 <표 1>에서 보는 바와 같이 AVC는 높은 복잡도로 인하여 복호화 시간이 상대적으로 많이 걸리기 때문에 공간임의접근에는 알맞지 않다는 결론을 내렸다. 따라서 MPEG-4 Video Coding 방법이 가장 적절하지만 본 논문에서는 실험의 편의를 위하여 일단 JPEG을 사용하기로 하며, 추후 MPEG-4 Video Coding으로의 실험을 추가적으로 진행하기로 하였다.

영상을 나누는 방법으로는 JPEG Tiling과 JPEG Resynchronization Marker를 응용하는 방법 등을 예로 들 수 있다. 이 중 JPEG Tiling 방법은 이미 많은 응용프로그램에서 적용한 방법이며, 본 논문에서는 JPEG Resynchronization Marker를 응용하는 방법도 함께 고려해 보기로 한다.

### 2. 타일링을 이용하는 방법

타일링이란 큰 이미지를 여러 개의 작은 이미지들로 (sub-images) 나누는 것으로써 다음과 같은 목적으로 사용한다.<sup>[5]</sup>

- 하나의 타일로 주어진 스크린의 크기에 맞추어 보여준다.
- 보고 싶은 부분을 위한 공간임의접근을 제공한다.
- 영상의 해상도 단계별로 보이는 응용프로그램을 지원한다.

타일링에는 복잡도의 순서에 따라 Simple tiling, Pyramid tiling, Composite tiling의 세 가지로 구분할 수 있다. 이 중에서 공간임의접근을 위하여 가장 적합한 형태는 simple tiling으로써 simple tiling은 좌측과 하단의 경계 부분을 제외한 다른 모든 타일의 크기를 일정한 크기로 고정함으로써 다루기가 편리하다. <그림 3>은 JPEG Tiling의 Syntax 구조를 보여준다. DTI(Define Tile Image Marker) 부분에서는 이 영상이 Simple Tiling임을 선언하고 전체 이미지의 크기 등을 정의한다. 또한 각 타일의 헤더에 들어가는 메타 정보인 DTT(Define Tile Marker) 부분에서는 타일 크기와 위치에 대한 정보들을 포함한다. 이 때, DTI와 DTT는 각각 15 바이트와 18 바이트의 저장 공간을 필요하며 각 타일에는 필요에 따라 별도의 허프만 코드를 추가할 수 있다. 이러한 오버헤드는 타일의 크기가 작아질수록 즉, 타일의 개수가 많아질수록 증가하여 효과적인 공간임의접근 구현에 장애 요인이 된다.

### 3. Resynchronization Marker를 이용하는 방법

JPEG에서 Resynchronization은 전송 시 발생하는 Channel Error에 대한 대응 수단으로 개발된 것이다. JPEG 이미지 특성상 한 bit의 오류만 발생해도 전체 이미지의 복원 자체를 불가능하게 만들 수 있다. 이러한 문제를 최소화하기 위해 몇 개의 블록을 모아 하나의 기본 단위로 전송하는 환경을 Resynchronization Marker를 통해 구현한 것이다. 다음은 Resynchronization Marker의 구조를 나타낸다.

Resynchronization marker는 <그림 4>에서처럼 RST (restart) marker 부분과 ECS (Entropy Coded Segment)

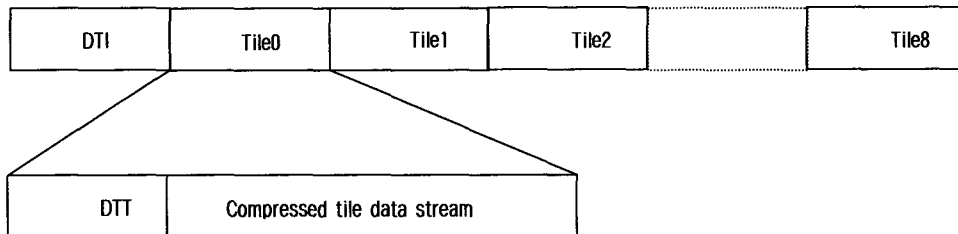


그림 3. 타일링의 구조  
Fig. 3. Tiling Syntax

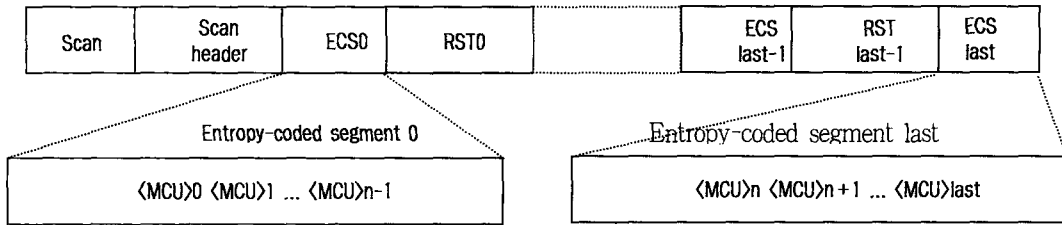


그림 4. Resynchronization 마커 구조  
Fig. 4. Resynchronization marker syntax

부분으로 나누어지고, 하나의 ECS는 한 개 또는 여러 개의 8×8 블록을 포함한다. 몇 개의 블록이 하나의 ECS에 포함되는가 하는 정보는 Image Header부분의 RTI (Restart Interval)에서 정의된다. 각 RST marker는 2바이트의 크기로 정의되어 있으며, 0에서 7사이의 값을 차례대로 반복해서 갖는다. 그리하여 데이터 전송 시, 최대 7개까지의 데이터 세그먼트 (RST+ECS 쌍)의 손실이 생겨도 순서를 잃지 않으며, 전체 영상을 못 보는 일 없이 손실되지 않은 부분만으로 그림을 재구성할 수 있게 된다.

본 논문에서는 이처럼 전송 오류의 복원 목적으로 만들어진 마커를 공간임의접근을 위한 블록 단위의 복호에 이용하는 방법을 고려해 보았다. <그림 5>는 Resynchronization marker를 이용하여 공간임의접근을 구현한 방법을 나타내었다. 사용자 측에서 보내어진 디스플레이의 좌상 좌표 정보는 복호기에서 필요한 매크로 블록만을 추출할 수 있도록 이용된다. 물론, 복호기가 타일과 디스플레이의 크기를

알고 있다는 가정이 필요하다. 앞서 말했듯이 마커는 좌상부터 스캔 방식으로 0부터 7까지 반복하여 나타나게 된다. 그림에서 각각의 매크로 블록에 있는 숫자는 마커의 번호와 마커의 누적된 개수를 나타낸다. 예를 들어, 1000×500 크기의 영상을 200×100 크기의 디스플레이에 보여줄 때, 매크로 블록의 크기가 100×100이라 가정해 보자. 만약 디스플레이의 좌상 좌표가 (150, 150)이라 하면, 필요한 매크로 블록은 12, 13, 14, 22, 23, 24번째이다. 따라서 복호기는 스캔 방식으로 차례대로 Resynchronization marker의 개수를 세어 해당되는 블록을 복호화하게 된다. 이 방법은 타일링과 B-map을 이용하는 방법에서도 유사하게 적용되었다.

전체 영상을 타일로 나누었을 때, 각 타일의 헤더 부분에 필요한 정보량(DTT 부분)은 허프만 코드는 제외하더라도 15바이트가 된다. 그러나 마커는 하나의 매크로 블록당 2바이트만 있으면 된다. 즉, 마커를 이용하는 방법은 타일링을 이용하는 방법에 비하여 최소 7.5배의 오버헤드를 줄일 수 있다. 만

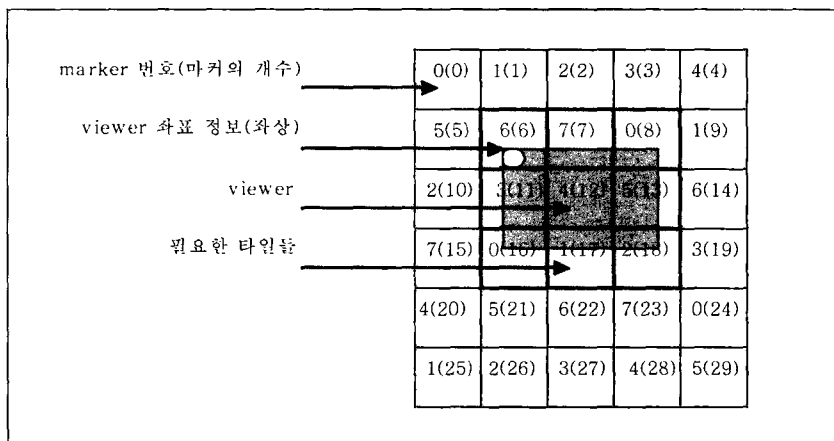


그림 5. Resynchronization marker를 이용한 공간임의접근 구현 방법  
Fig. 5. Spatial random access implementation using resynchronization markers

약 같은 양의 오버헤드를 가진다는 조건을 가정하면, 마커를 이용하는 방법은 타일링을 이용하는 방법보다 작은 서브 이미지로 자를 수 있게 된다. 따라서 마커를 이용하는 방법은 타일링을 이용하는 방법에 비해 불필요한 전송/복호를 줄여줌으로써 공간임의접근에 보다 적합한 형태라 말할 수 있다.

여기에서 고려되어야 할 사항이 있다. JPEG에서 마커의 블록 단위는 최대 16×16으로써 이보다 큰 크기의 단위 블록을 실제 사용하기 위해서는 부호기에서 비트 스트림에 대한 수정이 필요하다. 본 논문에서 복호화 시간 측정 실험은 부호기를 수정하는 대신 parsing 시간을 고려하지 않은 결과이다. 또한 마커가 지역 복호화를 위한 용도로 사용되면, 본래의 목적인 error resilience 기능은 상실되는 단점이 있다.

### III. B-map

#### 1. 제안하는 코덱 구조

본 논문에서는 타일/매크로 블록으로 나누는 과정에서 발생하는 오버헤드를 줄이면서 복호화시간을 만족할 수 있는

방법으로 Bitstream map(B-map)을 제안한다. B-map이란, ECS(Entropy-coded Segment) 길이의 정보를 나타내는 별도의 테이블을 말한다. 이는 각 프레임의 헤더부분에 위치하여 공간임의접근 구현 시 필요한 타일/매크로 블록 검색에 이용된다. 이미 2장에서 언급했듯이 B-map의 압축 틀은 JPEG을 이용한다. B-map은 부호화 과정에서 생성되는데, 영상 정보와 함께 전송된 B-map은 사용자의 입력에 따라 복호화될 타일/매크로 블록을 결정하는데 이용된다. 타일/매크로 블록을 결정하는 방법은 <그림 5>에서 설명하는 방식과 유사하다. 타일이나 마커를 이용할 시, 복호기는 타일/매크로 블록의 헤더 정보에 의하여 각각의 타일/매크로 블록을 검색했다면, B-map 구조에서 복호기는 프레임의 시작부분에서 단위 블록의 위치와 길이를 B-map을 통해 미리 파악함으로써 필요한 블록을 검색할 수 있다. 선택된 블록들의 이미지 데이터는 JPEG decoder 부분에서 복호화되어 디스플레이 크기에 맞는 이미지로 재구성된다. 제안된 구조는 <그림 6>과 같다.

#### 2. B-map의 Syntax

B-map의 구조는 <그림 7>과 같다. 원본 영상을 부호화

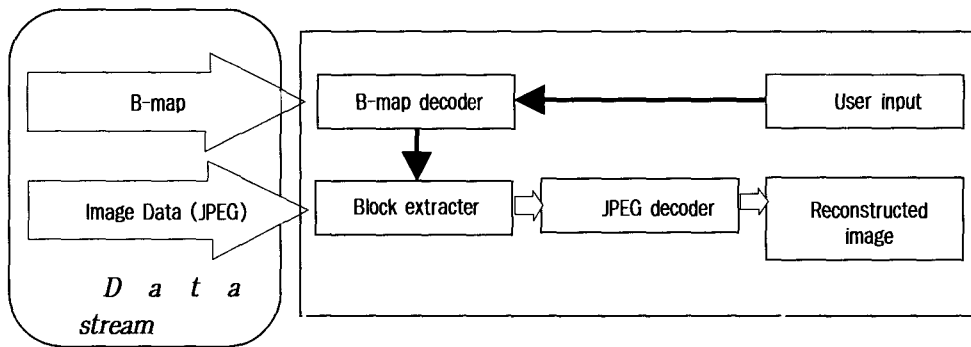


그림 6. 제안하는 B-map Codec  
Fig. 6. Proposed B-map Codec

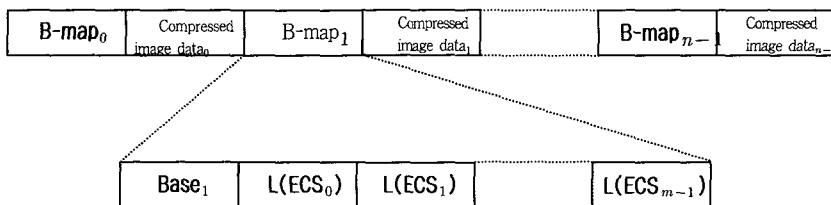


그림 7. B-map의 구조  
Fig. 7. B-map Syntax

할 때, 각 프레임 별로 ECS의 길이를 DPCM(Differential Pulse Code Modulation)하여 이진 표현으로 B-map을 생성한다. B-map은  $i$  번째 ECS 길이( $X_{i-1}$ )를 참조하여  $X_i$ 와  $X_{i-1}$ 의 차의 절대값( $B_i$ )을 취한다.

$$B_i = |X_i - X_{i-1}| \quad (1)$$

고해상도 영상의 특성 상 인접한 픽셀들의 유사성이 높으므로, 인접한 ECS들 간의 길이도 유사하다. 따라서 B-map은 DPCM(Differential Pulse-Code Modulation)을 이용하여 표현함으로써 ECS의 길이를 효과적으로 표현한다. 또한  $B_i$  값을 효율적으로 전송하기 위하여, 이진 표현(unary numeral system)을 사용한다. 예를 들어  $B_i$  값이 4로 계산되었다면 '0001'로 표현됨으로써 <표 2>와 같이 표현된다.  $B$ 값 1,4,2,4의 값을 바이트 단위로 보낸다면 효율이 떨어지므로, 이진표현으로 표현한다면 부호 비트를 포함하여 15비트로 표현 가능하다. 단, B-map의 시작부에는 DPCM에서 기준이 되는 Base 값이 포함되어야 하며, 부호 비트는 B-map 표현과는 독립적으로 처리한다. 이렇게 이진 표현된 정보는 산술 부호화(Arithmetic Coding) 과정을 한 번 더 거치게 됨으로써 오버헤드의 양을 최소로 하게 된다. 하지만 산술 부호와 과정의 높은 복잡도는 시스템에 많은 영향을 주기 때문에 본 논문에서는

표 2. B-map 이진 표현의 예  
Table. 2. An example of binary B-map expression)

	ECS(i-1)	ECS(i)	ECS(i+1)	ECS(i+2)	ECS(i+3)
ECS 길이	162	163	167	165	161
DPCM( $B_i$ )	X	1	4	2	4
부호 비트	X	+	+	-	-
B-map 표현	X	1	0001	01	0001

두 가지 방법을 모두 고려하여 보았다.

B-map은 복호화 시 각 프레임의 헤더 정보에 포함되어 전송된다. 복호기는 B-map으로부터 획득한 ECS 길이 정보와 디스플레이의 위치, 프레임의 크기 등을 고려하여 비트 스트림 상에서 복호화에 필요한 블록의 정보만을 추출, 복호화하게 된다. 즉, B-map 정보는 지역 복호화를 구현하기 위해 비트 스트림 상에서 필요한 정보만을 빠르게 추출하게 함으로써 효율적인 복호화를 가능하게 한다.

#### IV. 실험 결과

##### 1. 실험 조건

본 논문에서는 고해상도 영상의 공간임의접근을 구현하는 각각의 방법들을 비교하고자 다음과 같은 실험을 하였다. 첫 번째는 공간임의접근을 구현하는 가상의 사용자 패턴을 가정하여 각각의 방법에 따라 걸리는 복호화 시간을 측정하였고, 두 번째로 전체 이미지를 다양한 크기의 서브 이미지로 나누었을 때, 추가되는 오버헤드 비트의 양을 계산해 보았다. 기본적인 시뮬레이션 조건은 다음과 같다. <표 3>는 실험에 사용된 기본 조건을 나타내었고, <그림 8>

표 3. 실험 기본 조건  
Table. 3. Basic experimental conditions

테스트 영상	- "lobby" (출처: Immersive Media Company) - 해상도: 2400x1200 (RGB 24bits) - 프레임: 100 frame
Computer spec	- CPU: P4 2.4GHz - RAM: 480MB
타일 크기	- 16x16 through 600x600
디스플레이 해상도(Fig. 2)	- 720x480 - 352x240

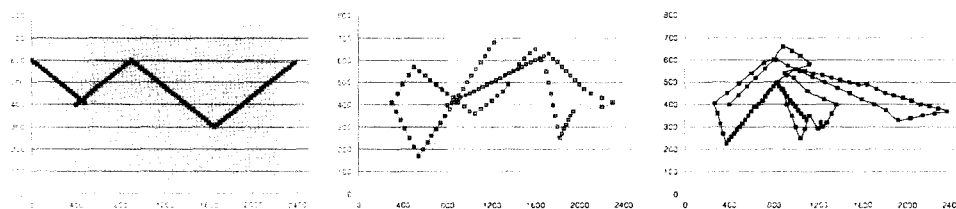


그림 8. 디스플레이 윈도우 움직임 경로  
Fig. 8. Three paths of the display movement

은 디스플레이 윈도우의 이동 경로를 나타낸다.

### 2. 복호화 시간

이 실험에서는 16×16에서부터 600×600에 이르는 다양한 크기의 타일/매크로 블록에 대하여 타일링, 마커, B-map을 이용한 공간임의접근 구현 시 걸리는 복호화 시간을 측정하였다. JPEG의 특성상, 모든 타일/매크로 블록의 크기는 16의 배수 단위로 구분하였다. 사용되는 타일/매크로 블록의 위치에 따라 복호화 시간이 달라질 가능성을 배제하기 위하여 3가지 경로를 고려하여 보았다. 또한 다양한 디스플레이의 크기에 대한 고려를 하기 위하여 2가지 크기(352×240, 720×480)에 적용시켜 보았다. 결과는 다음과 같다.

본 실험에서 타일링, 마커, 그리고 압축하지 않은 B-map을 이용하는 방법에 대한 복호화 시간은 모두 같다. 왜냐하면 각각의 방법에 대하여 헤더 부분의 스트림을 Parsing하

는데 걸리는 시간은 매우 작아서 시스템의 성능에는 거의 영향을 주지 않는다는 가정에 parsing 시간은 복호화 시간에 고려하지 않았기 때문이다. 또한 복호에 필요한 타일/매크로 블록의 선택 방법은 모두 <그림 5>에서 설명한 방식을 기반으로 이루어졌기 때문에 즉, 같은 크기의 타일이나 매크로 블록에서는 필요한 타일/매크로 블록도 일치하기 때문에 복호화 시간도 똑같이 걸리게 된다. B-map의 경우 ECS의 길이 정보의 양을 줄이기 위하여 산술 부호화를 통한 압축된 형태의 결과도 함께 비교해 보았다. 그러나 부호화된 B-map 정보를 복호화 하는 과정은 산술 부호의 높은 복잡도로 인하여 전체 복호화 시간에 많은 영향을 미치는 것을 알 수 있었다.

### 3. 추가 정보량 (오버헤드)

오버헤드는 공간임의접근을 구현하기 위하여 전체 이미

표 4. 720×480 디스플레이에서의 복호화 시간  
Table. 4. Decoding time on 720×480 display

적용 방식	경로	블록 크기별 복호화 시간(msec)				
		16×16	48×48	80×80	120×120	600×600
JPEG (without partitioning)	경로 1-3	229.3	229.3	229.3	229.3	229.3
Tile, Marker, B-Map (Unary coding)	경로 1	36.4	41.8	55.4	74.5	119.52
	경로 2	35.8	40.3	55.4	74.5	122.68
	경로 3	37.45	42.6	55.4	74.5	123.82
B-Map (Unary with Arithmetic Coding)	경로 1	51.4	54.8	65.3	83	123.52
	경로 2	50.8	53.3	65.3	83	126.68
	경로 3	52.45	55.6	65.3	83	127.82

표 5. 352×240 크기의 디스플레이에서의 복호화 시간  
Table. 5. Decoding time on 352×240 display

적용 방식	패턴	블록 크기별 복호화 시간(msec)				
		16×16	48×48	80×80	120×120	600×600
JPEG (without partitioning)	경로 1-3	229.3	229.3	229.3	229.3	229.3
Tile, Marker, B-Map (Unary coding)	경로 1	15.05	24.8	33.04	45.48	61.66
	경로 2	15.23	25.1	33.11	45.48	84.46
	경로 3	15.95	25.31	33.84	45.48	48.06
B-Map (Unary with Arithmetic Coding)	경로 1	30.05	39.8	44.04	54.48	65.66
	경로 2	30.23	40.1	44.11	54.48	88.46
	경로 3	30.95	40.31	44.84	54.48	52.06



표 6. 오버헤드 비교표  
Table 6. Overhead comparison table

적용방식	블록 크기별 프레임의 크기(오버헤드 비율) 단위: bytes(%)				
	16×16	48×48	80×80	120×120	600×600
Tile	356171 (90.04)	206171 (10.00)	194171 (3.60)	190421 (1.60)	187541 (0.06)
Marker	209921 (12.01)	189921 (1.33)	188321 (0.48)	187821 (0.21)	187437 (0.01)
B-Map (Unary coding)	196170 (4.67)	193216 (3.09)	191950 (2.42)	191789 (2.33)	191196 (5.22)
B-Map(Unary with Arithmetic Coding)	193807 (3.41)	188650 (0.66)	187959 (0.29)	187714 (0.16)	187502 (0.04)

\* 원 영상의 크기 : 187421 bytes

지를 여러 개의 서브 이미지로 나누는 과정에서 발생한다. 이중 타일링에서는 하나의 타일당 15바이트의 오버헤드가 발생하며, 마커에서는 하나의 매크로 블록당 2바이트의 오버헤드가 일정하게 발생한다. 반면 B-map에서는 DPCM 방식을 사용하기 때문에 고정된 크기의 오버헤드가 발생하는 것은 아니지만 ECS 길이 정보를 위한 오버헤드가 발생한다. 그러나 B-map의 경우, 같은 이미지에서 ECS 길이는 비슷할 것이기 때문에 DPCM 방식에 의한 오버헤드는 많지 않을 것임을 예상할 수 있다. <표 6>은 타일링, 마커, B-map에 대하여 발생하는 오버헤드의 양을 측정한 결과이다.

실험 결과에서 타일/매크로 블록의 크기가 작아질 때, 오버헤드의 양은 제곱으로 증가한다는 것을 알 수 있었다. 타일링의 경우 서브 이미지의 크기를 16×16으로 한다면, 오버헤드가 원본 영상의 90%까지 추가되는 결과를 볼

었다. 이러한 결과는 타일링을 이용하는 방법으로는 서브 이미지의 크기를 줄이는데 한계를 가지고 있음을 보여준다. <그림 14>에서 B-map의 그래프가 다른 방식들과 차이를 보이는 보인다. 이유는 서브 이미지의 크기가 커지면 ECS의 단위가 커지면서 DPCM의 성능이 떨어지면서 오히려 비트 수가 증가하기 때문인데 이러한 경우에는 이진 표현보다는 십진 표현이 더 좋다. 앞서 언급했듯이 블록의 크기가 작을수록 필요 없는 연산을 줄여주어 복호화 시간을 단축시킬 수 있다. 서브 이미지의 크기가 600×600에서 48×48 사이에서는 마커를 이용한 방법이 가장 작은 오버헤드를 가지는 것을 알 수 있었다. 그러나 서브 이미지를 16×16으로 했을 경우 마커의 경우, 12.01% 정도의 오버헤드를 가지는 것에 반해 B-map은 4.7%(산술 부호화 시 3.4%)의 오버헤드를 갖는다. 따라서 B-map의 경우, 블록의 크기가 작으면 작을수록 유효한 방법이 될 수 있음

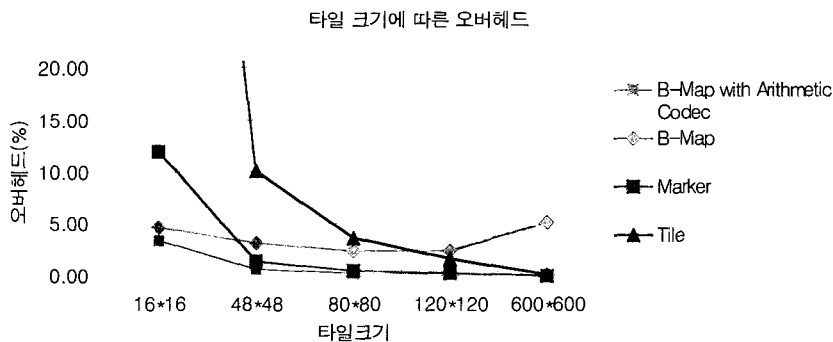


그림 14. 타일 크기에 따른 오버헤드 비교 그래프  
Fig. 14. Overhead comparison graph according to tile sizes

을 알 수 있다. 산술 부호화를 적용한 B-map은 이진 표현만으로 만들어진 B-map 보다 작은 오버헤드를 갖지만, 복잡도의 증가는 역시 치명적인 약점이다.

## V. 결 론

본 논문에서는 고해상도 영상의 임의지역접근을 위한 방법으로 타일링, 마커를 이용하는 방법과 새롭게 제안하는 B-map에 대한 비교 실험을 하였다. B-map은 서브 이미지의 크기가 작으면 작을수록 유리한 방식이라 할 수 있다. 상대적으로 작은 B-map의 서브 이미지는 불필요한 전송과 복호화를 줄일 수 있기 때문에 고해상도 영상의 지역 복호화에 가장 적합한 방식이라 말할 수 있다.

본 논문에서 B-map 방식은 실험의 용이함을 이유로 압축 톨로써 JPEG을 사용하였다. 하지만 앞서 언급했듯이 JPEG보다는 MPEG-4 Video Coding 이 더 좋은 성능을 발휘할 것으로 생각된다. 따라서 향후 MPEG-4 Video Coding을 이용한 실험이 이루어져야 할 것이다. 또한, 본 논문에서는 복잡도와 복호화 시간의 이유로 intra coding 방식만을 고려하였지만, 보다 높은 압축률을 위해서는 inter coding 방식을 사용하는 방법에 대한 고찰이 필요하리라 생각된다. 마지막으로 B-map의 사용은 헤더 부분에 추가적인 부호화로 인한 복잡도의 증가로 비트 에러에 더욱 민감하게 될 수 있는데 이 또한 해결해야 할 과제이다.

## 참 고 문 헌

- [1] Thomas Pintaric, Ulrich Neumann, Albert Rizzo, "Immersive Panoramic Video", Proceedings of the 8th ACM International Conference on Multimedia, October 2000.
- [2] S.E. Chen, "QuickTime VR An Image-Based Approach to Virtual Environment Navigation", Proc. ACM SIGGRAPH, August 1999.
- [3] Carsten Grnheit, Aljoscha Smoli, and Thomas Wiegand, "Efficient representation and interactive streaming of high-resolution panoramic views", Proc. ICIP 2002, IEEE International Conference on Image Processing, Rochester, NY, USA, September 22-25, 2002.
- [4] A. Smolic, Y. Huo, J. Guether and T. Selinger, "Demonstration of Streaming of MPEG-4 3-D Scenes with Live Video", ISO/IEC JTC1/SC29/WG11, MPEG01/M7811, Pattaya, Thailand, December 2001.
- [5] ISO/IEC 10918-[1~4] JPEG specification, 1994.
- [6] Sun Young Lee, Hyun Jong Oh, Sung Won Park, Euee S. Jang, Wook-Joong Kim, Myungsuk Ki, Kyuheon Kim "Study and experiments on the random access (partial decoding and rendering) of panoramic image" ISO/IEC JTC1/SC29/WG11 M10098, Brisbane, Australia October 2003
- [7] Euee S. Jang, Sun Young Lee, Hyun Jong Oh, Jong Woo Won, Wook-Joong Kim, Myungsuk Ki, Kyuheon K "Some results on local decoding of panoramic video (EE1)" ISO/IEC JTC1/SC29/WG11 M10333, Hawaii, USA December 2003
- [8] Euee S. Jang, Sun Young Lee, Hyun Jong Oh, Jong Woo Won, Wook-Joong Kim, Myungsuk Ki, Sangwoo Ahn, Injae Lee, Kyuheon Kim "B-map method for the partial decoding and rendering of panoramic video" ISO/IEC JTC1/SC29/WG11 MPEG2004/M10632, Munchen, Germany, March 2004.

## 저 자 소 개



박 성 원

- 2003년 2월 : 한양대학교 기계공학과 공학사
- 2003년 2월 ~ 현재 : 한양대학교 소프트웨어공학과 석사과정
- 주관심분야 : MPEG-4, 3DAV

저 자 소 개



원 종 우

- 2003년 2월 : 한양대학교 전자전기공학과 공학사
- 2003년 2월~현재 : 한양대학교 소프트웨어공학과 석사과정
- 주관심분야 : 영상압축, 3DAV, 디지털 신호처리



이 선 영

- 1996년 : 숭실대학교 전산학 학사
- 2003년~현재 : 한양대학교 소프트웨어공학과 석사과정
- 주관심분야 : MPEG-4, 비디오 부호화, 컴퓨터 그래픽



김 욱 중

- 1993년 2월 : KAIST 전기 및 전자공학과 학사
- 1995년 2월 : KAIST 전기 및 전자공학과 석사
- 1999년 8월 : KAIST 전기 및 전자공학과 박사
- 1999년 12월~현재 : 한국전자통신연구원 방송미디어연구그룹 선임연구원
- 주관심분야 : 3D 비디오 처리, 대화형 콘텐츠



김 규 현

- 1989년 2월 : 한양대학교 전자공학과 졸업(공학사)
- 1992년 9월 : 영국 University of Newcastle upon Tyne 전기전자공학과 졸업(공학석사)
- 1996년 7월 : 영국 University of Newcastle upon Tyne 전기전자공학과 졸업(공학박사)
- 1996년~1997년 : 영국 University of Sheffield reserach fellow
- 1997년~현재 : 한국전자통신연구원 대화형미디어연구팀장
- 2001년 12월~현재 : ISO/IEC JTC1 (MPEG) 한국대표단장
- 2004년 9월~현재 : AWF (Asian-Pacific Wireless Forum) TG3 (Task Group) 의장
- 주관심분야 : 영상처리, 멀티미디어통신, 대화형 및 이동 방송 시스템



장 의 선

- 1991년 : 전북대학교 전자계산기공학 공학사
- 1994년 : 뉴욕주립대 버펄로 캠퍼스 (SUNY at Buffalo) ECE M.S.E.E.
- 1996년 : 뉴욕주립대 버펄로 캠퍼스 (SUNY at Buffalo) ECE Ph.D.
- 2000년~현재 : IEEE International Conference on Multimedia and Expo Technical Committee Member & Session Chair
- 2000년~현재 : 방송공학회 편집위원
- 2000년~현재 : MPEG Forum 전문위원
- 2003년 2월~현재 : 한양대학교 정보통신대학 정보통신학부 조교수
- 주관심분야 : 영상압축, MPEG-4, 컴퓨터 그래픽스