

파노라마 동영상 스트리밍을 위한 적응적 스트립 압축 기법

(Adaptive Strip Compression for Panorama Video Streaming)

김보연[†] 장경호^{**} 구상옥^{***} 정순기^{****}
 (Bo Youn Kim) (Kyung Ho Jang) (Sang Ok Koo) (Soon Ki Jung)

요약 일반적인 동영상 스트리밍 시스템은 좁은 시각영역을 갖는 동영상을 원격의 사용자에게 제공한다. 서버 측에 팬/틸트(pan/tilt) 카메라를 부착한 시스템은 클라이언트가 카메라의 방향을 조작할 수 있도록 함으로써 보다 넓은 시각영역을 제공할 수 있다. 하지만, 하드웨어 팬/틸트 카메라를 이용한 스트리밍 시스템은 다수의 사용자가 동시에 접속할 경우, 모든 사용자에게 각자가 원하는 시점에서의 영상을 제공할 수 없으며, 카메라 이동에 의한 지연이 발생한다는 단점이 있다. 본 논문에서는 소프트웨어 팬/틸트 카메라를 구현하였으며, 이를 이용한 새로운 파노라마 동영상 스트리밍 시스템을 제안한다. 제안된 시스템은 넓은 시각영역을 가지는 파노라마 영상을 획득한 후, 각 사용자에게 원하는 부분영상을 제공함으로써 다수의 사용자에게 자유로운 시점 이동을 제공해 준다. 이때, 방대한 양의 파노라마 동영상을 원격의 사용자에게 전송하기 위해서는 효율적인 압축 방법이 요구된다. 본 논문에서는 전체 영상을 좁은 스트립 영상들로 분할하고, 클라이언트 측에서 원하는 시각영역에 해당하는 스트립 영상들만을 독립적으로 압축하는 적응성 스트립 압축 기법을 제안한다. 제안된 시스템은 적응성 스트립 압축 기법을 사용하여 파노라마 동영상 압축, 전송함으로써 사용자의 시각영역 변화 및 새로운 사용자의 접속에 빠르게 대응할 수 있으며 클라이언트 전송 시 데이터양을 효과적으로 감소시킬 수 있다.

키워드 : 파노라마 영상, 파노라마 동영상, 적응적 스트립 압축

Abstract Traditional live video streaming systems support the limited field of view (FOV) of image to the remote users. A server system based on the pan/tilt camera provides a user with wide view by changing the view direction of the camera mechanically. But, when many clients try to access to the server, this system can not offer their own view to every user simultaneously, and moreover it has the delay by camera motion. In order to offer wide views to several users, we propose new streaming system using the panorama image that has wide view. Our system is a kind of implementation of software pan/tilt camera. The server acquires panorama video and sends a part of the video to clients. Then, each client can control their own view. We need the effective way to reduce the average transmission data size and server burden to the compression because generally the full size of panorama video is too big to be served by the real-time streaming. To solve this problem, we propose an strip-based video compression and adaptive transmission of the compressed multiple strip videos. Experimental results show that our system can be adapted quickly to the change of view and the number of clients. Furthermore, proposed method effectively reduce the transmission data.

Key words : Panoramic Image, Panorama Video, Adaptive Strip Compression

1. 서론

데이터 전송기술의 발달과 인터넷 멀티미디어 기술의 발달로 단순한 문자와 그림만이 아닌 소리와 동영상에 이용한 정보의 공유가 가능하게 되었다. 과거 인터넷을 통해 음악 파일이나, 동영상 파일을 재생하기 위해서는 재생에 필요한 파일을 서버(server)에서 모두 다운로드한 후 파일을 재생하는 것이 일반적이었다. 이 경우 사용자의 컴퓨터에 내용량의 데이터를 저장할 공간이 필

† 정 회 원 : 한국전자통신연구소 CG 기반기술 연구팀
 bykim@etri.re.kr
 ** 정 회 원 : 경북대학교 가상현실연구실 연구원
 khjang@vr.knu.ac.kr
 *** 학 생 회 원 : 경북대학교 컴퓨터공학과
 sokoo@vr.knu.ac.kr
 **** 종 신 회 원 : 경북대학교 컴퓨터공학과 교수
 skjung@knu.ac.kr
 논문접수 : 2004년 10월 4일
 심사완료 : 2005년 11월 7일

요하며, 데이터를 모두 다운받을 때까지 기다려야 하는 문제점이 있다. 더구나, 파일을 서버로부터 전송 받는 도중 서버와의 접속이 끊어졌을 때, 서버에 재접속하여 재생하고자 하는 파일을 처음부터 다시 다운 받아야 하는 번거로움이 있다.

스트리밍(Streaming)이란 클라이언트(client)가 서버로부터 데이터를 전송 받을 때, 모든 데이터를 다운 받는 것이 아니라 필요한 데이터를 실시간으로 전송받을 수 있는 멀티미디어 기술로서, 음악, 동영상, 영화 등 대용량의 멀티미디어 데이터를 인터넷을 통해서 먼 거리에 있는 다수의 사용자에게 서비스하는데 최근 널리 이용되고 있다. 스트리밍은 서버 측에서 예전에 이미 만들어진 데이터를 가지고 서비스 하는 경우에도 유용하지만, 서버에서 실시간으로 데이터를 생성하여 이를 즉시 클라이언트에게 전송하는 경우에 더욱 이용가치가 커진다. 생방송이나 생중계와 같은 기능을 인터넷을 통해 손쉽게 할 수 있기 때문이다.

서버 측에서 고정된 카메라로부터 획득된 영상을 실시간으로 다수의 사용자들에게 제공하고자 할 때 문제점 중의 하나는, 모든 클라이언트가 서버에서 제공해주는 고정된 영역만을 일방적으로 보게 된다는 것이다. 이때 카메라 렌즈(lens)의 시각영역(field of view)에 따라 클라이언트가 제공받는 영상의 영역도 제한된다. 클라이언트가 단지 고정된 영역만을 서비스 받는 것이 아니라, 서버와 클라이언트 간의 상호작용을 통해 클라이언트가 원하는 시점을 자유롭게 선택할 수 있도록 하기 위해서 하드웨어로서 이용할 수 있는 것이 팬/틸트(pan/tilt) 카메라이다[1]. 현재 상용화 되고 있는 Canon의 Web-View Livescope[2]나 iCanTeK의 iCanView[3] 등과 같이 원격으로 팬/틸트 카메라를 제어하는 시스템들은 주로 넓은 영역을 감시하기 위한 감시카메라 시스템이나 원격지간 회의 등에 사용되고 있다. 그러나 이러한 시스템들은 팬/틸트 카메라를 제어하기 위한 고가의 카메라 서버 또는 스트리밍 서버를 요구하며, 다음과 같은 한계점들이 있다. 첫째는 클라이언트 측에서 원하는 시점으로 이동하기 위해서는 하드웨어 제어가 필요하며, 이에 따른 반응 시간이 오래 걸린다. 보다 중요한 문제로서 두 번째는, 서버에 다수의 클라이언트가 접속해 있을 경우, 다수의 클라이언트가 동시에 카메라의 방향을 조작할 수 없으므로 모든 클라이언트에게 원하는 영상을 제공해 주기가 어렵다는 것이다. 따라서 본 논문에서는 넓은 시각영역을 가지는 파노라마(panorama)[4] 영상을 획득한 후 다수의 사용자에게 원하는 부분영상을 제공함으로써 소프트웨어적으로 팬/틸트 카메라를 구현한 시스템을 제안한다.

파노라마 영상을 인터넷을 통하여 전송하고 제한적이

나마 서버와 사용자 간의 상호작용을 지원해주는 기존의 여러 시스템들이 존재한다. 먼저 Chen[5]은 파노라마 영상을 기반으로 하여 회전(rotation)과 확대(zoom)가 가능한 영상 기반의 가상현실 시스템인 QTVR(Quick-Time VR)을 제안했는데, 제한적이지만 핫 스팟(Hot Spot)을 이용하여 다른 영역으로의 이동성을 제공하기도 한다. 그러나 QTVR에서는 영상 전체를 서버로부터 모두 다운 받아야 재생이 가능하다. C. Zhang[7]은 동심원 모자의 영상[6]을 RBC(Reference Block Codec)[8]로 압축하여 서버와 클라이언트 사이에 상호작용이 가능한 브라우저를 제안하였다. 서버 측에서 RBC로 압축된 영상을 캐쉬(cache)로 전송하고, 클라이언트 측에서는 캐쉬에 저장된 영상을 렌더링(rendering)하는 전략을 사용함으로써 렌더링 속도를 향상시켰다. C. Grunheit[9]가 제안하는 시스템은 대용량의 높은 해상도를 가지는 파노라마 영상에 대한 축소/확대와 시점 이동을 제공한다. Grunheit의 시스템은 대용량의 파노라마 영상(13200×2600)을 일정한 크기(512×512)의 독립된 영상으로 분할하여 클라이언트가 파노라마 영상의 특정 부분을 요구할 경우, 특정부분에 해당되는 영상만을 압축하여 전송하는, 서버와 클라이언트 사이에 상호작용이 가능한 시스템이다. 이 시스템에서는 확대 시에도 높은 해상도를 가지는 영상을 서비스하기 위해 해상도에 따라 계층적으로 압축 형식(format)을 달리하는 스케일러블 코딩(scalable coding) 기법을 이용하여 압축하는데, 스케일러블 코딩은 일반적인 압축 기술인 MPEG-2[5,10,11]압축에 비해 압축률이 떨어진다. 기존의 세 시스템 모두 시간의 흐름에 따른 연속적인 영상이 아닌 한 장의 정지 영상을 사용한다. 본 논문에서는 한 장의 정지 영상이 아니라 연속적인 파노라마 영상, 즉 파노라마 동영상을 실시간으로 전송 및 재생하며 사용자의 시점 이동이 가능한 시스템을 제안한다.

그림 1과 같이 파노라마 동영상은 여러 시점에서의 동영상을 연결하여 만들어지기 때문에 일반 동영상보다 데이터의 크기가 매우 크다. 그러므로 파노라마 동영상을 전송할 때에는 전송 데이터양을 줄이기 위해 효율적인 동영상 압축 기술이 요구된다. 본 논문에서는 파노라마 영상 전체를 전송하는 것이 아니라 클라이언트들이 요구하는 영역만을 압축하여 전송함으로써 압축 및 전송 데이터양을 줄이는 방법을 제안한다. 이를 통해 파노라마 동영상을 효율적으로 다수의 사용자에게 스트리밍할 수 있으며, 서버 측의 압축 부담도 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 파노라마 동영상 스트리밍 시스템의 전체 구성 및 파노라마 영상 전송을 위한 새로운 압축 방법에 대해 제안한다. 3장에서는 제안하는 방법으로 파노라마 영상을 압

축하고 전송할 경우, 사용자가 시점을 이동할 때와 새로운 사용자가 접속할 때와 같은 상황에서의 효율적 전송 방법에 대해 살펴본다. 4장에서는 실험 및 결과를 보이며 5장에서 결론을 맺는다.

2. 파노라마 동영상 스트리밍 시스템

본 장에서는 파노라마 동영상 스트리밍 시스템의 전체 구성을 살펴보고, 파노라마 영상의 효율적인 전송을 위한 새로운 압축방법을 제안한다.

2.1 시스템 개요

본 절에서는 파노라마 영상을 생성하는 방법과 생성된 파노라마 영상을 다수의 클라이언트에게 전송하는 방법에 대해 설명한다. 파노라마 영상은 동일한 위치에서 각각 다른 방향을 가지는 3 대의 카메라를 설치하고 각 카메라를 통해 얻어진 영상들을 정합하여 생성한다[4]. 그림 1과 같이 생성된 파노라마 영상을 입력으로 하여 서버는 시간의 흐름에 따라 입력되는 영상을 순차

적으로 압축하여 클라이언트에 전송한다.

제안하는 파노라마 동영상 스트리밍 시스템의 전체 구성은 그림 2와 같다. 파노라마 동영상 스트리밍 시스템은 크게 파노라마 영상을 생성하고, 생성된 파노라마 영상을 압축하는 서버와 이를 전송받아 재생하는 클라이언트로 나누어진다. 서버는 파노라마 영상을 실시간으로 얻기 위해 보정된 여러 대의 카메라(multi-camera)를 이용한다. 여러 대의 카메라를 통해 얻은 영상을 영상 모자이크(image mosaics) 기법을 이용하여 파노라마 동영상을 생성한다[4]. 제안된 시스템에서 서버는 파노라마 영상 중 클라이언트가 요청하는 영역, 즉 전체 파노라마 영상의 일부분만을 압축하여 전송한다. 예를 들어 그림 2와 같이 각 클라이언트가 서버에게 A, B, C 영역의 영상을 요청한다면, 서버는 전체 파노라마 영상 중 클라이언트가 요청하는 A, B, C 영역만을 각각 압축하여 각 클라이언트에게 전송한다.

본 시스템에서는 각 클라이언트가 자신이 원하는 영

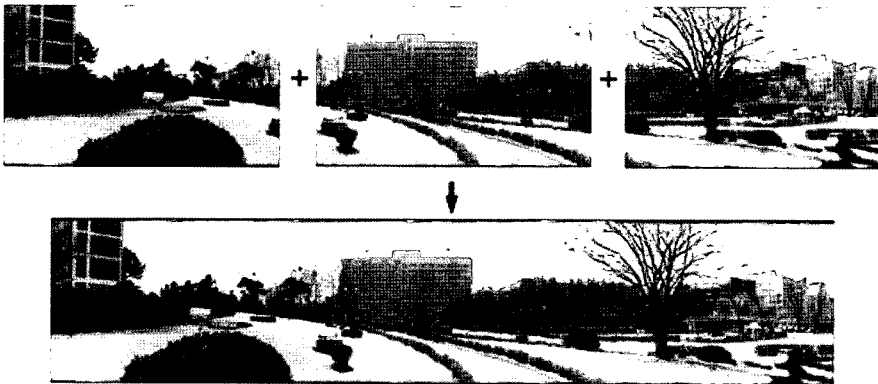


그림 1 파노라마 영상 생성

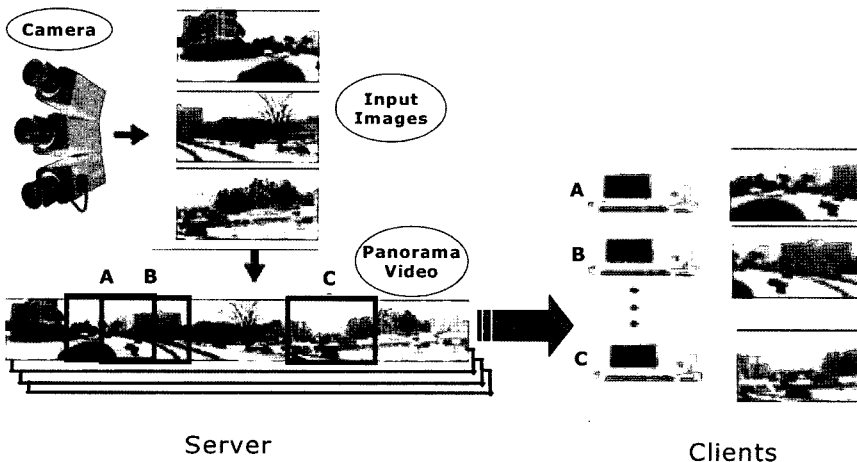


그림 2 파노라마 동영상 스트리밍 시스템

역의 영상을 서버로부터 전송받다가 새로운 영역을 서버에 요청하면, 서버는 클라이언트의 요청에 따라 새로운 영역의 영상을 제공해 주므로 다수의 클라이언트에 대한 시점 이동을 지원해 줄 수 있다. 보다 자세한 파노라마 동영상 압축 방법은 2.2절에서 살펴본다.

일반적인 스트리밍 시스템이 하나의 카메라를 통해 입력된 전체 영상을 전송하므로 클라이언트 측의 요구를 고려할 필요가 없는 단 방향 시스템인데 반해, 본 논문에서 제안하는 파노라마 스트리밍 시스템은 클라이언트가 서버에 필요한 영역의 영상을 요청하면, 서버는 클라이언트에게 파노라마 영상 중 요청된 일부분만을 전송하는 상호작용이 가능한 양방향 시스템이다.

2.2 스트립 영상 압축

본 논문에서는 효율적인 파노라마 동영상 스트리밍을 위한 수정된 MPEG-2 코딩 기법에 대해 설명한다. MPEG-2 코딩 기법으로 연속적인 영상을 압축 했을 때, 압축된 데이터들은 I(Intra)프레임과 P(Predicted)프레임으로 구성된다. I-프레임은 입력 영상을 RGB에서 YCbCr로 바꾸고 이를 DCT와 양자화 과정을 거친 후 가변길이 부호화와 허프만 부호화 과정을 거쳐 생성된다. 반면 P 프레임은 이전의 I 또는 P-프레임을 토대로 시간에 따르는 순방향 예측을 통해 얻어진다. 이는 시간의 흐름에 따르는 연속적인 영상에서 이전 영상과 바로 이후의 영상이 크게 차이가 나지 않는다는 가정에 기인한 것이며, 이전 시간의 기준프레임과 비교하여 움직임 벡터와 오차 값만을 압축하므로 I-프레임보다 훨씬 압축률이 높다. 동영상을 MPEG-2를 이용하여 압축한 결과는 그림 3과 같이 I 와 P-프레임이 시간의 순서에 따라 연속적으로 나열된 형태이다.

그러나, 파노라마 동영상 스트리밍의 경우, 전체 영상을 MPEG-2로 압축하는 것은 비효율적이다. 2.1절에서 언급한 바와 같이 클라이언트는 한 번에 전체 파노라마 영상의 일부분만을 요청하며, 전송 데이터양을 줄이기 위해서는 클라이언트가 요청한 영역만을 압축하여 전송하는 것이 바람직하다. 그런데, 제안하는 시스템이 다수의 클라이언트를 지원하고 파노라마 영상의 시각영역 안에서 시점 이동이 가능한 시스템이라는 점을 감안하면 위 방법은 다음과 같은 문제가 있다. 첫째, 클라이언

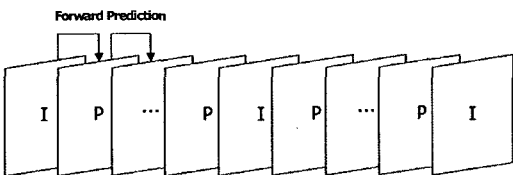
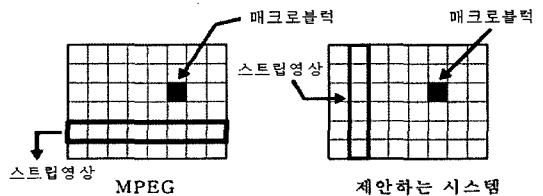


그림 3 MPEG-2를 이용해 압축된 동영상

트에서 시점 이동이 일어날 때마다 새로운 시점에서의 영상을 새롭게 압축하여야 한다면 P-프레임의 높은 압축률의 혜택을 많이 누릴 수 없다. 둘째, 2.1절에서 살펴본 것과 같이 서버는 각 클라이언트가 요구하는 시각영역만큼의 영상을 각각 압축해야 하므로, 실제 서버가 압축해야 하는 데이터의 양은 클라이언트의 수에 비례하여 커지게 되어 서버의 압축 부담이 가중된다.

하지만 사용자가 매우 갑작스럽게 시점 이동을 하는 경우가 아니라면, 대체로 시점 이동이 일어나기 전의 영상과 후의 영상은 중복되는 부분이 많다. 또한 다수의 클라이언트가 요청한 영역들이 서로 겹치는 부분이 있을 수 있으며, 클라이언트의 수가 많아질수록 중복되는 부분이 더 많아질 것이다. 시점 이동 시 중복되는 영역을 P-프레임으로 코딩하면 압축 데이터양이 보다 감소될 것이고, 여러 클라이언트들에 의해 중복 요청된 영역을 서버에서 한 번만 압축한다면, 서버의 압축 부담을 감소시킬 수 있을 것이다. 따라서 본 논문에서는 P-프레임의 높은 압축률의 혜택을 누리면서도, 서버의 압축 부담을 줄일 수 있는 스트립 영상 압축 기법을 제안한다.

스트립 영상 압축은 파노라마 영상을 독립된 스트립 단위로 자르고, 각 스트립을 하나의 독립된 영상으로 보고 스트립 영상단위로 압축 및 전송하는 방법이다. 그림 4에서 보는 바와 같이 MPEG이 하나의 영상을 여러 개의 가로로 된 스트립 영상들로 잘라서 전송하는 것과는 달리, 본 시스템에서는 세로로 된 슬라이스 구조를 가진 스트립 영상이 사용된다. 세로로 된 슬라이스 구조는 원본 영상은 MPEG으로 압축할 때 시계방향으로 90도 회전시킴으로써 쉽게 얻어 질 수 있다. 본 시스템에서는 하나의 스트립 영상은 독립된 영상으로 간주되며, 움직임 정보는 스트립 영상에 국한되어 계산된다. 그러므로 클라이언트가 요청하는 부분에 해당되는 스트립 영상을 독립적으로 압축 및 전송할 수 있기 때문에, 스트립 영상 크기 내에서 시점이동이 가능하다. 또한 카메라의 위치는 항상 고정되어 있으므로, 시점 이동 전후의 영상에서 중복되는 부분은 실제로 카메라의 움직임(motion)이 일어나지 않는 영역이다. 중복되는 영역은 전경물체의 움직임을 제외하면, 시점 이동 이전 영상과 이후 영상의



(a) MPEG 영상구조 (b)제안하는 시스템의 영상구조
그림 4 영상 구조

변화가 매우 작으므로 P-프레임으로 코딩하고, 시점 이동의 결과 새롭게 요청된 부분만 I-프레임으로 코딩하면 압축 데이터의 크기가 훨씬 작아진다. 또한 다수의 클라이언트가 요청한 영역들 사이에 겹치는 부분이 있는 경우에도, 그 부분의 스트립 영상을 한 번만 압축하면 되므로 서버의 압축 부담을 줄일 수 있다.

본 시스템에서 구현된 스트립 영상 압축 방법을 요약하면 다음과 같다. 그림 5와 같이 클라이언트가 서버에 접속하여 영상을 요청하면 서버는 파노라마 영상을 스트립 단위로 자르고 스트립 단위로 압축한다. 즉 서버는 접속 초기에 입력된 파노라마 영상을 I-스트립으로 코딩하고, 다음 시간부터 들어온 영상은 이전 시간의 I-스트립 영상을 참조하여 P-스트립으로 코딩한다. 본 논문에서는 다른 영상을 참조하지 않고 자기 자신만으로 코딩할 수 있는 스트립 영상을 I-스트립 영상으로, 이전 시간에 존재하는 I-스트립 또는 P-스트립 영상을 참조하여 이를 시간의 흐름에 따라 순방향 예측하여 얻어진 스트립 영상을 P-스트립 영상으로 정의한다. I-스트립 영상과 P-스트립 영상의 코딩과정은 각각 MPEG-2의 I-프레임과 P-프레임의 코딩과정과 같다.

3. 적응성 스트립 압축

본 논문에서 제안하는 스트립 영상 압축 기법을 사용하면 각각의 스트립 영상을 상황에 따라 I 또는 P-스트립 영상으로 독립적으로 압축할 수 있으므로, 클라이언트가 시점을 이동하거나 다수의 클라이언트가 접속하여 다양한 시각영역에 대한 요구를 하는 상황에도 서버가 빠르게 대응할 수 있으며, 효율적인 데이터 전송이 가능하다. 본 장에서는 이와 같이 서버가 클라이언트의 요구에 따라 각 스트립 영상의 코딩 방식을 달리 적용하는 적응성 스트립 압축 기법을 클라이언트의 시점 이동시와 새로운 클라이언트 접속 시의 두 가지 상황을 예로 들어 설명한다.

3.1 시점 이동 시

그림 6에서 보는 바와 같이 제안된 시스템에서 서버는 클라이언트가 접속하면, 입력 영상을 스트립 단위로 분할하고, 클라이언트가 요청한 영역에 해당하는 스트립 영상을 모두 I-스트립 영상으로 코딩하여 클라이언트에게 전송한다. 만약 클라이언트 측에서 다음 시간에도 이전 시간과 같은 영역의 영상을 요청한다면, 다음 시간 입력되는 영상은 이전 시간의 영상을 참조하여 I-스트

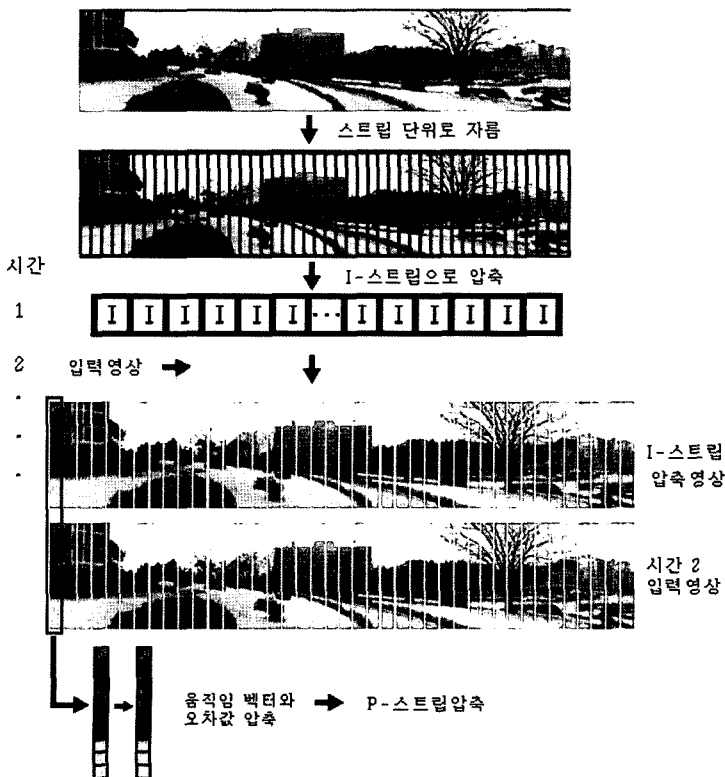


그림 5 스트립 영상 압축

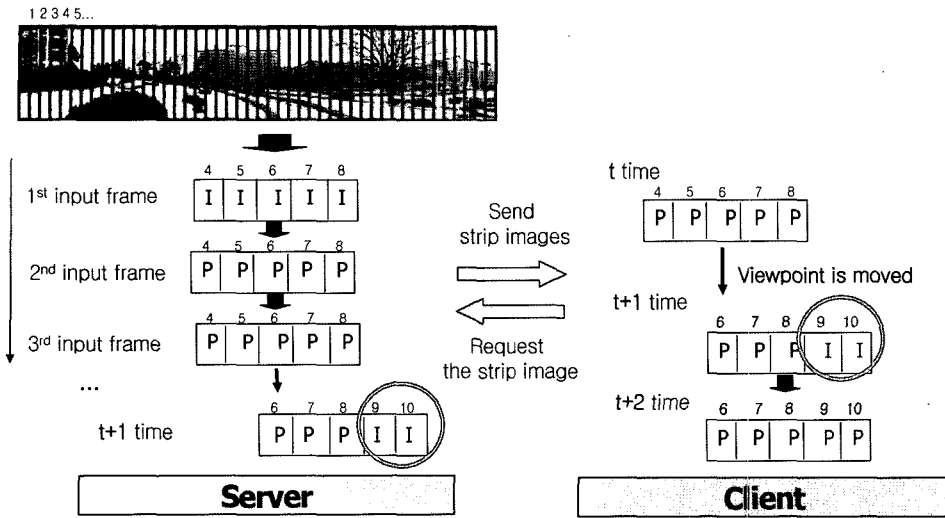


그림 6 시점 이동 시

립 영상보다 데이터양이 훨씬 적은 P-스트립 영상으로 코딩하여 보냄으로써 압축 및 전송 데이터양을 줄일 수 있다.

이 때 클라이언트 측에서 시점을 이동하여 새로운 영역을 요청한 상황을 살펴보자. 예를 들어, 그림 6과 같이 접속 시부터 4, 5, 6, 7, 8 스트립 영상을 요청하던 클라이언트가 어느 시간에 시점을 이동하여, 6, 7, 8, 9, 10 스트립 영상을 요청했다고 가정하자. 이 때 이전 시간부터 서버로부터 전송 받아 재생하고 있던 6, 7, 8에 해당하는 스트립 영상은 P-스트립 영상으로 코딩하여 전송하더라도 클라이언트 측에서 바로 재생이 가능하지만, 클라이언트는 새롭게 요청한 9, 10 스트립 영상에 대한 데이터는 가지고 있지 않다. 만약 이를 고려하지 않고 모든 스트립 영상을 P-스트립 영상으로 압축하여 보낸다면, 클라이언트 측에서는 9, 10 스트립 영상에 대한 부분을 재생할 수 없다. P-스트립으로 코딩된 영상을 재생하기 위해서는 P-스트립을 생성하기 위해 참조했던 이전 영상들이 모두 클라이언트 측에 있어야만 재생이 가능하기 때문이다. 반면에 만약 모든 스트립 영상들을 I-스트립으로 코딩하여 보낸다면, P-스트립을 전송할 수 있는 6, 7, 8 스트립 영상까지도 불필요하게 I-스트립으로 코딩하게 되므로 압축 효율이나 전송 데이터 양 측면에서 비효율적이다.

이러한 경우에 제안된 시스템의 서버는 이전에 재생되고 있었던 부분은 이전 스트립 영상을 참조하여 P-스트립으로 코딩하고, 새롭게 요청된 부분은 (그림 6에서는 9, 10 스트립 영상)은 I-스트립으로 코딩하여 전송함으로써 효율적인 압축 및 전송이 가능하다. 물론 클라이언트가 새로운 영역에 대한 I-스트립 영상을 전송받음

다음부터는 또 다른 시점 이동이 일어나기 전까지 P-스트립 영상을 전송받는다.

3.2 새로운 클라이언트의 접속 시

기존의 클라이언트가 서버로부터 데이터를 전송받고 있는 중에 새로운 클라이언트가 접속할 때에도 새로운 스트립 영상의 압축이 요구된다. 예를 들어 그림 8에서처럼 클라이언트 A가 t 시간에 4, 5, 6, 7, 8에 해당하는 P-스트립 영상을 전송받고, 바로 이후인 t+1 시간에도 시점 이동 없이 계속해서 4, 5, 6, 7, 8의 P-스트립 영상을 전송받고 있으며, 시간 t+1에 클라이언트 B가 새로 접속하여 6, 7, 8, 9, 10의 스트립 영상을 전송받기를 요청한다고 가정하자. 클라이언트 B에는 자신이 요청하는 6, 7, 8, 9, 10을 참조 할 스트립 영상이 존재하지 않으므로, 서버는 클라이언트 B에게 6, 7, 8, 9, 10의 I-스트립 영상을 최초로 전송해 주어야 한다. 그러나 이 때, 서버가 6, 7, 8, 9, 10을 모두 I-스트립으로 코딩하여 모든 클라이언트에게 전송하게 되면, P-스트립 영상을 전송할 수 있는 클라이언트 A에게도 I-스트립으로 코딩된 데이터를 전송하게 되므로 전송 데이터 양 측면에서 비효율적이다.

따라서 제안된 시스템에서는 그림 7과 같이, 새로운 클라이언트가 접속하는 순간에는 기존의 클라이언트를 위한 P-스트립 영상을 생성함과 동시에, 새로운 클라이언트를 위한 I-스트립 영상도 생성한다. 그래서 그 순간 기존의 클라이언트와 새로운 클라이언트 간에 중복되는 영역(그림 7에서는 6, 7, 8 스트립 영상)에 대해서는 서버가 P-스트립과 I-스트립 영상을 모두 가지고 있게 된다. I-스트립 영상이 필요한 클라이언트 B로는 I-스트립 영상, P-스트립 영상만으로 재생이 가능한 클라이언트

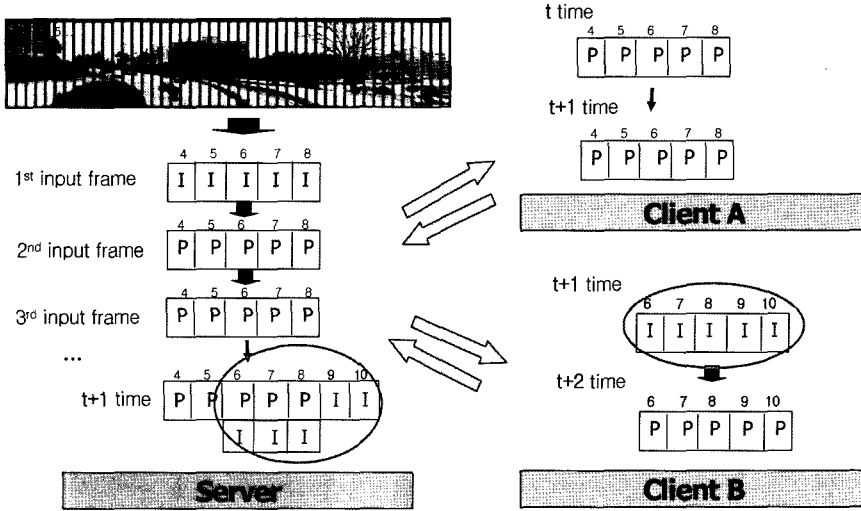


그림 7 새로운 클라이언트 접속 시

인트 A로는 P-스트립 영상을 제공해 준다. 물론 다음 시간부터는 클라이언트 B도 P-스트립 재생을 위해 참조할 이전 영상이 있으므로, 서버는 4, 5, 6, 7, 8, 9, 10에 대해 모두 P-스트립 코딩만 하면 된다.

위에서 살펴본 바와 같이 제안된 시스템에서는 클라이언트의 요청 및 상태에 따라 각 스트립 영상의 중복 여부 및 코딩 타입을 고려하여 I 또는 P-스트립으로 적절히 압축, 전송함으로써 클라이언트가 시점 이동 했을 때와 새로운 클라이언트가 접속 하는 상황에 적절히 대응하여 효율적인 전송이 이루어질 수 있도록 하였다.

또, 서버가 클라이언트가 요청하는 시각영역만을 정확히 압축하고 전송하는 것이 아니라, 그림 8과 같이 요청된 영역에서 양쪽 방향으로 약간 더 확장된 영역의 스트립 영상을 전송함으로써 시점 이동 시에도 끊기는 현상 없이 자연스럽게 영상이 연결되도록 배려하였다.

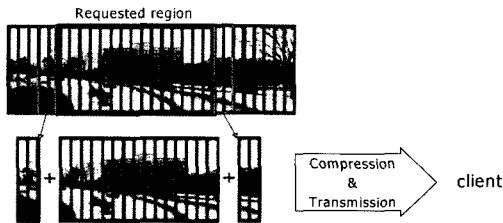


그림 8 스트립 영상 패딩(padding)

4. 실험 및 결과

실험을 위해 사용된 파노라마 영상의 해상도는 640×240이며, 총 600장의 연속적인 파노라마 영상으로 이루어

표 1 압축 데이터 양

압축 방법	평균 영상 크기	
입력 파노라마 영상 (BMP)	460,800 (bytes)	460,800 (bytes)
입력 파노라마 영상 (JPEG)	47,104 (bytes)	48,164 (bytes)
MPEG-2	26,422 (bytes)	23,142 (bytes)
제안된 시스템	28,244 (bytes)	25,051 (bytes)
C. Grunheit의 시스템	51,463 (bytes)	51,824 (bytes)

어진 파노라마 동영상을 가지고 실험하였다. 스트립 영상의 크기는 16×240으로써 하나의 파노라마 영상은 총 40개의 스트립 영상으로 구성된다.

표 1은 서버에서 다양한 압축방법에 의해 압축된 파노라마 영상의 평균 크기를 보여주고 있다. 입력으로 들어오는 파노라마 영상의 평균 크기는 460,800바이트이다. MPEG-2가 압축률이 가장 좋게 나타나며, 제안된 시스템의 경우 MPEG-2보다 압축 영상의 크기가 약간 크다. 그 이유는 제안된 시스템에서는 스트립영상들이 하나의 독립적인 영상으로 간주되므로, 움직임 정보가 독립적인 스트립 내에서만 계산되어, 스트립간의 움직임 정보를 이용하지 못하기 때문이다. 그러나 MPEG-2로 압축할 경우엔 클라이언트에게 영상의 일부분만을 전송할 수 없으나, 제안하는 방법은 파노라마 영상의 일부분만을 제공할 수 있다. Grunheit의 시스템도 영상의 일부분을 제공할 수 있으나 스케일러블 코딩 방식으로 압축하는 관계로 제안된 시스템 보다 약 2배정도 압축 데이터의 크기가 크다.

압축성능을 평가하기 위해, 우리는 PSNR(Peak-Signal-To-Noise Ratio)을 사용하여 원본 영상과 압축된 영상과의 차이를 실험하였다. 평가에 사용된 수식은

다음과 같다.

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \hat{x}_{ij})^2}$$

여기에서 x_{ij} 와 \hat{x}_{ij} 는 원본 영상과 압축 영상을 의미하여 M, N 은 영상의 높이와 넓이다. 그림 9는 PSNR을 사용한 제안된 방법, MPEG-2 그리고 Grunheit의 압축방법으로 압축했을 때 PSNR값을 비교한 그래프이다. 제안된 방법이 MPEG-2보다 압축성능이 좋다는 것을 보여준다.

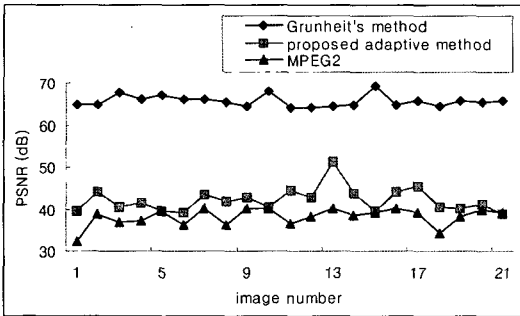


그림 9 압축 성능 비교

그림 10, 11, 12는 각각 서로 다른 장소에서 촬영된 파노라마 동영상 2개에 대한 다음과 같은 실험결과를 보여 준다. 각 그림에서 왼쪽은 대학교의 비교적 한적한 도로에서 촬영된 파노라마 동영상, 오른쪽 그래프는 박물관 정문을 촬영한 동영상에 대한 실험 결과 그래프이다.

그림 10은 단일 클라이언트 접속 시, 서버가 압축하여 클라이언트로 전송하는 데이터의 양을 비교한 것이다. 하나의 클라이언트만 접속한 경우, 제안된 시스템은 하나의 클라이언트가 요청하는 영역만 압축하여 전송하므로, 평균 압축 데이터양보다 훨씬 적은 양을 압축하고 전송할 수 있다. 그림 10의 그래프에서 두드러지게 압축 데이터양이 증가하는 부분은 시점 이동 때문에 I-스트림 압축이 필요한 때를 나타낸다.

그림 11은 10명의 클라이언트가 접속했을 때, 서버에서의 압축 데이터양을 비교한 것이다. 단일 클라이언트가 접속했을 때보다 압축 데이터양의 변화가 심해진 하지만, 전체 파노라마 영상을 압축했을 때의 평균 압축 데이터양인 28,244 바이트 보다는 압축 데이터양이 작다는 것을 알 수 있다. 제안하는 시스템은 다수의 클라이언트가 접속하여 시점 이동을 할 때 중복되는 부분을 계속해서 재 압축할 필요 없이 새롭게 요청하는 부분의 스트림 영상만을 압축하여 전송함으로써 서버의 압축 부담을 감소시켰기 때문이다.

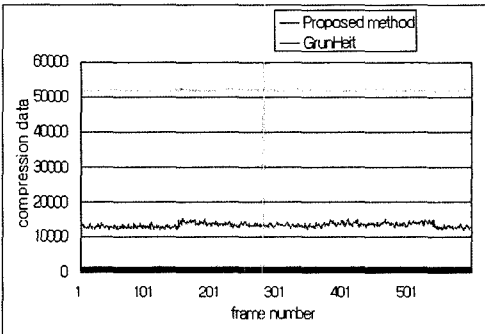
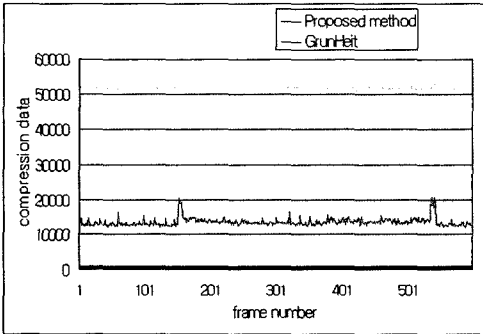


그림 10 단일 클라이언트 접속 시 압축 데이터 양

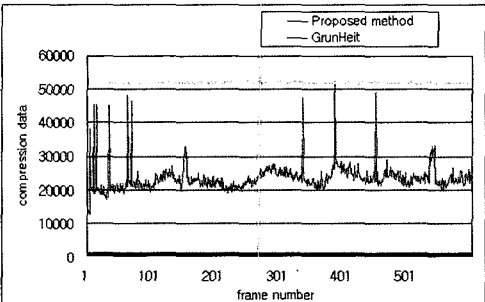
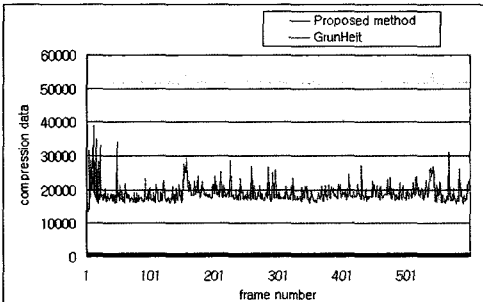


그림 11 다수(10개) 클라이언트 접속 시 압축 데이터 양

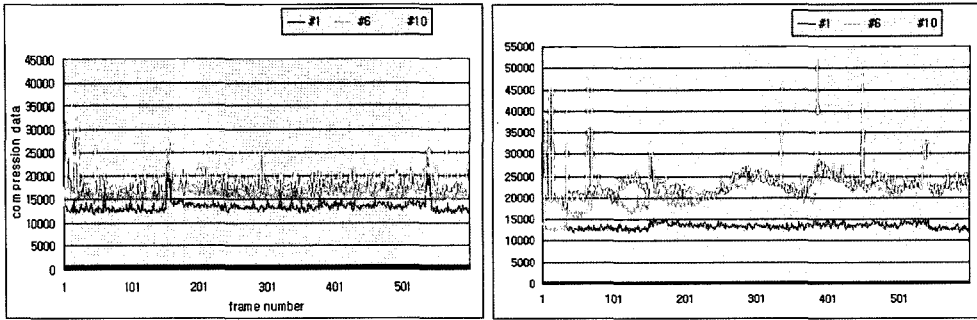


그림 12 클라이언트 수에 따른 압축 데이터 양

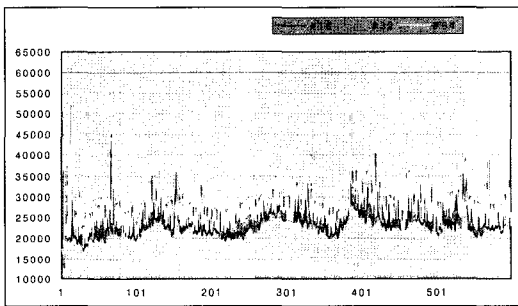
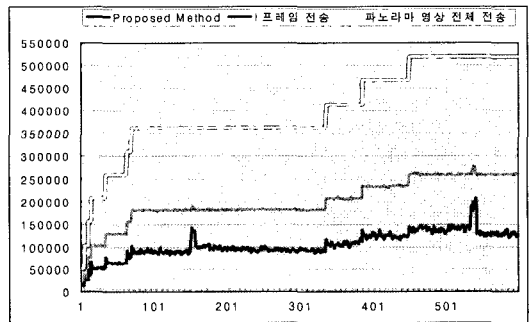
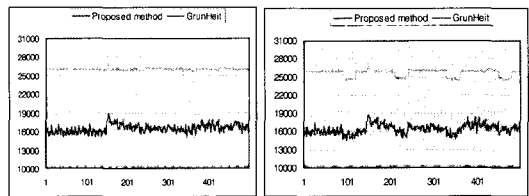


그림 13 스트립 영상 크기에 따른 압축 데이터 양



(a) 전체 전송 데이터 양



(b) 클라이언트 각각에 대한 전송 데이터 양

그림 14 전송 데이터 양

그림 12는 서버에 연결된 클라이언트 수에 따른 서버 측에서의 압축 데이터양 보여주고 있다. 클라이언트 수가 증가함에 따라 압축해야 할 데이터의 양은 증가하지만, 클라이언트의 수가 많아져도 압축 데이터의 양은 표 1의 전체 영상에 대한 평균 압축 데이터양을 넘지 않는다.

그림 13은 스트립 영상의 크기의 변화에 따른 압축 데이터양의 변화를 보여준다. 본 시스템에서 사용한 스트립 크기인 16×240를 사용했을 때가 보다 큰 스트립 크기인 32×240이나 64×240을 사용할 때보다 훨씬 효율적이다. 스트립 크기가 커질수록 움직임이 있을 때 I-프레임과 P-프레임으로 동시에 코딩(coding)되는 영역이 커지게 되기 때문이다.

그림 14(a)는 시간에 따른 전송 데이터의 양의 변화를 보여준다. 10개의 클라이언트가 순서대로 접속하였으며, 파노라마 영상 전체에 대한 그래프의 계단현상은 새로운 클라이언트의 접속 시 데이터양의 급격한 변화를 보여준다. 물론, 그림 11과 12에서 보는바와 같이 때때로 서버에서 높은 압축 데이터량을 보이지만, 그림 14(b)에서 보는 바와 같이 실제 클라이언트 전송 시 각각의 클라이언트로 전송되는 데이터의 양은 GrunHeit의 방법에 비해 항상 적은 수의 데이터를 클라이언트로 전송함을 알 수 있다. 이 실험을 통해 우리는 클라이언트가 요청한 영역에 대해 I-프레임만으로 압축한 경우보

다 제안된 방법이 항상 데이터 전송량이 적음을 알 수 있다.

5. 결론

본 논문은 다수의 사용자에게 시점 이동이 가능한 동영상을 제공해줄 수 있는 파노라마 동영상 스트리밍 시스템을 제안하였다. 제안된 시스템은 서버와 클라이언트 사이에 상호작용이 가능한 양방향 스트리밍 시스템으로서, 서버는 실시간으로 원격의 사용자에게 각 클라이언트가 요구한 시각영역에 해당하는 부분영상을 각각 전송한다. 본 논문에서는 대용량의 파노라마 동영상 데이터를 효율적으로 전송하도록 하기 위해 새로운 스트림 압축 기법을 제안한다. 전체 파노라마 영상을 쬐은 스트

림 영상들로 분할하고, 클라이언트 측에서 요구하는 시각영역에 해당하는 스트림 영상들만을 독립적으로 압축, 전송하는 스트림 압축 기법을 사용함으로써, 압축 및 전송 데이터양, 서버의 압축 부담을 모두 감소시킬 수 있었다. 또한 서버 측에서 모든 스트림 영상을 동일한 코딩 방식으로 압축하는 것이 아니라, 각각의 스트림 영상을 경우에 따라 독립적으로 I 또는 P-스트림 영상으로 적절히 압축, 전송함으로써 사용자의 시각영역 변화 및 새로운 사용자의 접속에도 빠르게 대응할 수 있을 뿐만 아니라, 실험결과로부터 시점이동을 제공하기 위해 I-스트림만을 전송하는 방법에 비해 클라이언트로 전송되는 데이터양을 효과적으로 감소시킬 수 있었다.

참고 문헌

- [1] P. Peixoto and J. Batista and H. Araujo, "A Surveillance System Combining Peripheral and Foveated Motion Tracking," ICPR-98, pp.574-577, August, 1998.
- [2] <http://www.canon.com/>
- [3] <http://www.icantek.com/>
- [4] K.H. Jang, S.K.Jung and M. L., "Constructing Cylindrical Panoramic Image using Equidistant Matching," IEE Electronics Letters, IEE, England, ISSN: 0013-5194, Vol. 35, No. 20, pp. 1715-1718, September 1999.
- [5] S. E. Chen, "QuickTime VR - An Image-Based Approach to Virtual Environment Navigation," Proc. ACM SIGGRAPH, pp. 29-38, August 1995.
- [6] H.-Y. Shum and L.-W. He. "Rendering with Concentric Mosaics," Proc. ACM SIGGRAPH, pp. 299-306, August 1999.
- [7] C. Zhang and J. Li, "Interactive Browsing of 3D Environment over the Internet," VCIP 2001, SPIE Vol. 4310, No. 51, San Jose, CA, January 2001.
- [8] C. Zhang and J. Li, "Compression and rendering of Concentric Mosaics with Reference Block Codec (RBC)," VCIP 2000, Perth, Australia, June 2000.
- [9] C. Grunheit, A. Smolic, and T. Wiegand, "Efficient Representation and Interactive Streaming of High-Resolution Panoramic Views," ICIP-2002, Rochester, NY, USA, Vol. 3, pp. 209-212, September 2002.
- [10] A. Puri, "MPEG Video Coding Standards," Invited Tutorial: International Society for Circuits and Systems, April 1995.
- [11] A. Puri, "Video Coding Using the MPEG-2 Compression Standard," Proc. SPIE Visual Communication and Image Processing, Boston, November 1993.



김보연

2001년 동서대학교 컴퓨터 공학과 졸업(공학사). 2004년 경북대학교 컴퓨터 공학과 졸업(공학석사). 2004년 9월~현재 한국전자통신연구소. 관심분야는 Virtual Reality, Computer Graphics, Computer Vision



장경호

1998년 경상대학교 전산통계학과 졸업(이학사). 2000년 경북대학교 컴퓨터공학과 석사 졸업(공학석사). 2005년 경북대학교 컴퓨터공학과 박사 졸업(공학박사) 2005년~현재 경북대학교 BK21 박사후과정. 관심분야는 Image-Based Modeling and Rendering, Virtual Reality, Computer Graphics, Computer Vision



구상욱

2001년 경북대학교 컴퓨터공학과 졸업(공학사). 2003년 경북대학교 컴퓨터공학과 석사 졸업(공학석사). 2004년 3월~현재 경북대학교 컴퓨터공학과 박사 과정 관심분야는 Computer Graphics, Computer Vision, Information Visualization, Virtual Reality



정순기

1990년 경북대학교 컴퓨터공학과 졸업(공학사). 1992년 한국과학기술원 전산학과(이학석사). 1997년 한국과학기술원 전산학과(공학박사). 1997년~1998년 University of Maryland, Research Associate. 2001년~2002년 IRIS, University of Southern California, Research Associate. 1998년~현재 경북대학교 컴퓨터공학과 부교수. 1999년~현재 (주)아이디스 기술고문. 관심분야는 Virtual Reality, Artificial Intelligence, Computer Vision, Image Processing, Computer Graphics