

일반논문 (Regular Paper)

방송공학회논문지 제25권 제2호, 2020년 3월 (JBE Vol. 25, No. 2, March 2020)

<https://doi.org/10.5909/JBE.2020.25.2.265>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

사용자 선택에 따른 자유 시점 비디오 서비스 기반의 통합 플레이어 시스템 구현

양 지 희^{a)}, 송 민 기^{b)}, 박 구 만^{c)*}

Implementation of Integrated Player System based on Free-Viewpoint Video Service according to User Selection

Ji-hee Yang^{a)}, Min-ki Song^{b)}, and Gooman Park^{c)*}

요 약

자유 시점 비디오 서비스는 사용자와의 상호작용을 통해 원하는 임의의 각도나 위치, 거리에서 시청할 수 있게 하는 기술이다. 본 논문에서는 사용자가 선택하여 시청할 수 있는 자유 시점 비디오 서비스를 Inward view, Outward view, 3D object view, First person view의 네 가지 시청 모드로 정의하였으며 하나의 플레이어에서 시청이 가능한 새로운 통합 프로그램을 개발 및 구현하였다. 아이돌 공연과 농구 경기 콘텐츠에 대해 각 시청 모드에 적합한 다시점 카메라를 설치하여 데이터를 확보하였으며, 서버에 저장된 데이터가 네트워크를 통해 스트리밍 됨으로써 시청이 가능하도록 하였다. 사용자는 자유롭게 네 가지의 시청 모드와 공간상의 위치, 각도 등을 선택할 수 있으며, 선택된 시점에 맞는 영상과 음향이 렌더링 되어 통합 플레이어에 표출된다. 이는 기존의 자유 시점 비디오 서비스를 포함한 다양한 시청 형태를 결합함으로써 사용자에게 몰입감과 현장감을 제공할 수 있을 뿐만 아니라 확장성이 있는 자유 시점 비디오 서비스 플레이어가 될 것으로 기대된다.

Abstract

Free-viewpoint video service is a technology that allows users to watch at any angle, location and distance through interaction. In this paper, the free-viewpoint video services are defined in four viewing modes: Inward view, outward view, 3D object view and first person view. And we developed and implemented a new integrated program that plays all the suggested views. In the contents of girl band performances and basketball games, multi-view cameras suitable for each viewing mode are installed to acquire media, and data stored on the server is streamed over the network, making it available for viewing. Users can freely choose four viewing modes, space location, angle and so on, and the media data such as images and sounds are provided to them by rendering appropriately for the selected viewpoint. Our system is expected to be a scalable free-viewpoint video service player as well as provide users with immersion and presence by combining various viewing modes.

Keyword : Free-viewpoint, Immersive media player system, Inward view, Outward view, 3D object view

I. 서론

다시점 비디오 서비스는 n 개의 카메라 장비에서 획득한 데이터를 응용 분야에 맞게 편집, 저장, 전송 및 관리하는 서버와, 받은 데이터를 처리 및 표출하는 단말기로 구성된다. 임의로 배열된 n 개의 카메라 장비로 다시점 영상을 획득하고 편집, 보정 등의 전/후처리 단계를 통해 다시점 콘텐츠를 제작하게 된다. 이후 서버에서 단말기로 전송되어 표출하게 됨으로써 원하는 임의의 각도나 위치, 거리에서 이벤트 시청이 가능하게 된다^[1,2]. 특히 자유 시점 비디오는 많은 수의 카메라를 조밀하게 배치하여 사용자와 상호작용하면서 선택된 시점을 사용자에게 제공한다. 또한 영상 처리 분야에서는 물리적인 카메라에서 획득한 영상 데이터를 합성 또는 3차원으로 재구성하여 시점의 수를 증가시켜 더 넓은 시점 선택의 폭을 제공한다. 이러한 경우 기존의 단일 시점에 비해 시점을 전환할 때 끊김 없이 부드럽게 연결하면서 시청이 가능하며 임의 위치에서 공연이나 경기의 관찰을 통해 실감성을 높여준다^[3-5].

이러한 자유 시점 비디오 서비스를 문화 콘텐츠에 적용함으로써 간단한 장비와 네트워크 기반의 전송 플랫폼만으로 다양한 시청 모드에 대해 뷰 선택이 가능한 새로운 통합 플레이어 시스템을 구성하고자 하였다. 기존의 다양한 자유 시점 비디오 서비스는 자유롭게 공간상의 위치, 각도 및 인칭 시점을 선택할 수 있지만 사용자가 선택한 시점 위치

에 따라 영상과 음향이 동시에 렌더링 되어 단말기를 통해 사용자에게 제공되지 않는다. 또한 자유 시점 영상의 서비스 종류는 많아지고 있으나 하나의 통합된 플레이어로 시청하는데 한계가 있다. 이에 따라 본 논문에서는 다시점 카메라 배열로 표출할 수 있는 자유 시점 비디오 서비스를 네 가지의 뷰로 정의하였으며, 하나의 프로그램에서 시청이 가능한 통합 플레이어 시스템을 개발 및 구현하였다. 제안하는 통합 플레이어 시스템은 아이돌 공연, 스포츠 경기와 같은 문화 콘텐츠에 대해 자유 시점 뷰에 따른 다시점 카메라를 설치하여 영상과 음향을 획득하고, 이를 스트리밍 서버를 통해 전송함으로써 사용자에게 몰입감과 현장감을 제공할 뿐만 아니라 다양한 시청 형태를 추가적으로 확장성이 있게 결합할 수 있는데 목적을 둔다.

본 논문의 2장에서는 통합 플레이어 시스템의 전체적인 구성 및 자유 시점 뷰의 종류와 각 뷰의 표출을 위한 세부적인 구현 과정에 대해 설명한다. 3장에서는 테스트 베드 구축을 통한 시스템 성능 검증과 실험 결과에 대해 논한다. 끝으로 4장에서는 본 논문의 결론 및 향후 연구 계획에 대해 논한다.

II. 본론

1. 시스템 전체 구성

그림 1은 본 논문에서 제안하는 시스템의 전체 구성도이다. 시스템은 크게 표출하고자 하는 콘텐츠에 대해 네 가지의 뷰마다 영상과 음향을 획득하여 네트워크를 통해 송신하는 송신단과 실시간으로 스트리밍 되는 데이터를 전송받아 표출하는 수신단으로 구분하여 모든 수행과 작동이 이루어진다. 송신단에서는 콘텐츠에 대해 네 가지의 뷰로 표출하는데 적합하도록 취득된 영상 및 음향 데이터를 저장하고, 수신단에서는 전송된 데이터를 디코딩하여 각 뷰에 맞게 렌더링 및 표출함으로써 제공한다. PC에 설치된 통합 플레이어에서는 현재 제공하는 콘텐츠 중 시청하길 원하는 콘텐츠를 가장 먼저 선택한다. 선택된 콘텐츠에서 자유 시점 뷰와 오디오 믹싱 채널을 선택적으로 제공받을 수 있으

a) 서울과학기술대학교 나노IT디자인융합대학원 정보통신미디어공학전공(Dept. of Information Technology and Media Engineering, The graduate School of Nano IT Design Fusion, Seoul National University of Science and Technology)

b) 서울과학기술대학교 일반대학원 미디어IT공학과(Dept. of Media IT Engineering, The Graduate School, Seoul National University of Science and Technology)

c) 서울과학기술대학교 전자IT미디어공학과(Dept. of Electronic IT Media Engineering, Seoul National University of Science and Technology)

‡ Corresponding Author : 박구만(Gooman Park)

E-mail: gmpark@seoultech.ac.kr

Tel: +82-2-970-6430

ORCID: <https://orcid.org/0000-0002-7055-5568>

※ This research was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2017 (R2017030041, Experience Maximization Technology for Cultural Contents at Free Selection View).

• Manuscript received February 25, 2020; Revised March 24, 2020; Accepted March 24, 2020.

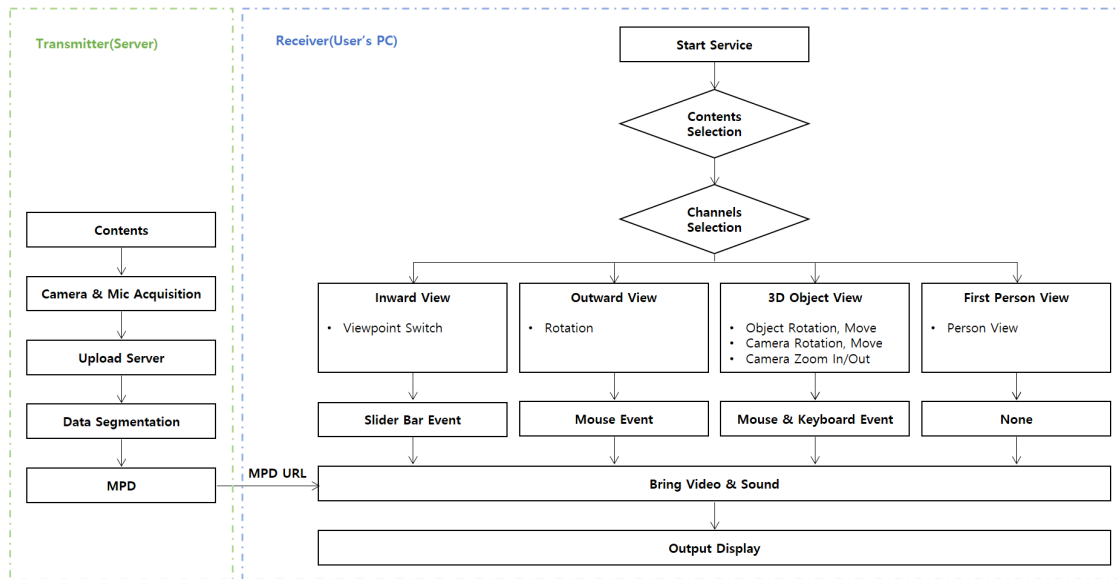


그림 1. 제안 시스템 전체 구성도
Fig. 1. The overall configuration of proposed system

며, 각각의 뷰에서 시점 제어를 통해 실감성이 있는 콘텐츠 시청이 가능하다.

본 논문에서는 시스템 구현을 위해 자유 시점 서비스가 가능한 뷰를 각각 Inward view, Outward view, 3D object view, First person view로 정의하였다. 첫 번째 뷰는 일반적인 자유 시점 비디오에 해당되며 촬영을 위해 선형으로 설

치된 카메라 배열에서 시점을 전환하여 시청 가능하도록 한다. 두 번째 뷰는 360도 카메라를 이용하여 전방위 영상을 볼 수 있다. 그리고 세 번째 뷰는 3차원 객체를 중심으로 자유자재로 회전, 이동하여 관찰하는 시점을 나타내며, 네 번째 뷰는 콘텐츠 내에 있는 한 객체에 장착된 카메라를 이용하여 현재 바라보고 있는 시점을 볼 수 있도록 한다.

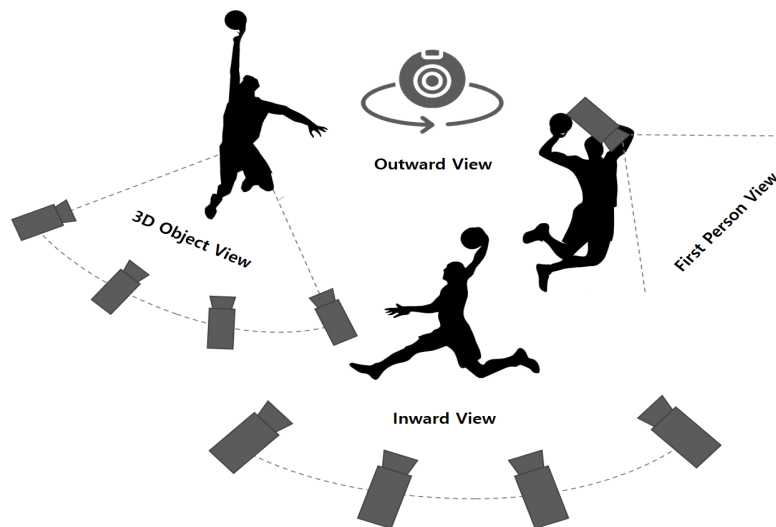


그림 2. 통합 플레이어에서 표시 가능한 자유 시점 뷰의 정의
Fig. 2. Definition of free-viewpoint view that can be displayed in the integrated player

네 가지의 뷰에 대한 정의는 그림 2와 같다.

2. 송신단

본 논문에서 제안하는 시스템의 송신단에서는 콘텐츠의 영상 및 음향을 획득하기 위해 네 개의 뷰를 표출하는데 적합한 카메라를 사용한다. 카메라는 각 뷰의 실감성을 최대한으로 표출할 수 있도록 위치와 각도를 고려하여 배치하고 촬영한다. 획득한 영상 및 음향 데이터는 각 뷰의 표출 방법에 맞게 편집 및 보정 과정을 거쳐 서버에 저장된다. 이때, 각 뷰에서 콘텐츠를 표출하기 위한 렌더링 방식이 다르기 때문에 데이터들은 각 뷰에 따라 구분 지어 관리된다. 실시간으로 스트리밍하기 위해 데이터는 인코딩되며 특정 길이의 DASH(Dynamic Adaptive Streaming over HTTP) 세그먼트를 제작하고 수신단에서 서버로부터 데이터를 불러오기 위한 MPD(Media Presentation Description)를 작성한다^[6]. 그림 3은 실험을 위해 설치한 카메라의 구성 모습과 이를 통해 획득한 입력 영상 결과를 나타낸다.

3. 수신단

본 논문에서 제안하는 시스템의 수신단에서는 송신단으로부터 받은 데이터를 사용자에게 자유 시점 비디오 기반의 콘텐츠로 제공하기 위해 통합 플레이어를 통해 실시간으로 스트리밍 한다. 인터넷망이 구축된 환경에서 MPD 주소를 통해 서버로부터 데이터를 전송받으며 네 개의 시청 모드 중 사용자가 선택한 하나의 뷰로 표출되도록 한다. 통

합 플레이어는 네 개의 시청 모드인 뷰들 간의 선택 전환이 용이하도록 구성한다. 선택된 각각의 뷰 내에서 시점 전환이나 위치, 각도 제어가 가능하도록 하며 이에 따른 영상과 음향이 시점에 맞게 렌더링 되어 표출된다.

3.1 Inward view

Inward view는 초기 영상으로 다시점 카메라 배열에서 정중앙에 설치된 카메라로부터 획득한 시점 영상을 서버로부터 받아 표출한다. 이 시점의 위치는 기준 좌표로서 x축의 원점이 되며, 슬라이드 바 이벤트를 받아 현재 시청하는 시점이 변경 가능하도록 한다. 사용자가 x축 중 특정 위치 x_{user_pos} 에서 시청하고 있다고 가정할 때, 슬라이더 바로 움직인 거리를 이용하여 현재 위치의 x좌표를 이동시킨다. Inward view에서 사용되는 다시점 카메라 배열은 1차원 선형으로 동일한 간격 a 를 갖도록 배치되었기 때문에 원점을 기준으로 우측 또는 좌측으로 이동시킬 경우, 각각 a 씩 증가하거나 감소시켜 사용자의 위치를 조정한다. 변경된 위치는 $x_{user_pos} = x_{pre_user_pos} \pm a$ 가 되며, 이에 해당하는 영상을 서버로부터 전송받아 표출함으로써 시점 간의 전환이 가능하게 한다. 슬라이더 바로 이동이 가능한 범위는 사용된 카메라의 수를 넘지 않는다. 음향도 동일하게 사용자의 현재 위치를 이용하여 해당 시점에 대한 음향을 자동으로 믹싱하여 렌더링함으로써 사용자에게 제공된다^[7,8].

3.2 Outward view

Outward view는 서버에 저장된 Equirectangular 파노라믹 영상을 전송받아 구형 영상으로 맵핑하여 표출한다. 이

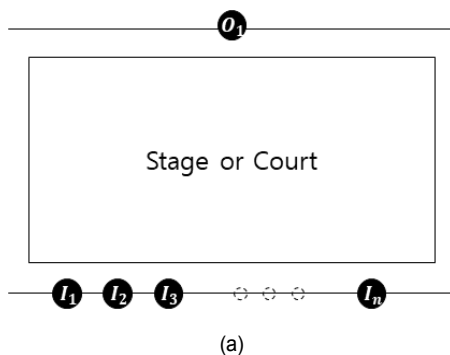


그림 3. 영상 및 음향 획득을 위한 카메라 구성 모습 및 결과 (a) 카메라 구성 (b) 획득 영상 결과

Fig. 3. Configuration and result for video and audio acquisition (a) Configuration of camera (b) The result of video

때, 왼쪽 마우스 버튼 이벤트를 받아 각각 영상 회전이 가능하게 함으로써 사용자의 제어에 따라 시점이 변화하도록 한다. 영상을 z축 방향에서 바라보고 있다고 가정할 때 각각 가로/세로 방향으로 드래그한 거리를 이용하여 x, y축을 회전한다. 마우스 이벤트에 대한 회전은 다음과 같다.

$$x_{angle} = x_{pre_angle} + (\Delta x / width \times 360) \quad (1)$$

$$y_{angle} = y_{pre_angle} + (\Delta y / height \times 360) \quad (2)$$

식 (1)과 (2)에서 $\Delta x, \Delta y$ 는 각각 마우스를 드래그한 x, y거리를 의미하며, 회전각은 영상의 해상도 크기에 비례하도록 계산한다^[9].

영상의 회전 변화에 따라 음향도 같이 렌더링하기 위해서는 x, y축 회전각에 대한 3차원 벡터 값 $listener_{up_vector}$, $listener_{forward_vector}$ 가 필요하다. $listener_{up_vector}$, $listener_{forward_vector}$ 는 x, y축의 회전각으로부터 도출이 가능하며 수도 코드 표 1과 같다.

3.3 3D object view

3D object view는 다른 뷰들과는 다르게 회전 및 이동 등의 제어가 가능한 Unity기반의 응용 프로그램을 제작하여 실행함으로써 3차원으로 복원된 객체를 표출한다. 먼저 콘텐츠 내 객체를 그래픽 기반의 3차원 모델로 복원하기 위해 객체를 중심으로 카메라를 둘러싸아 다시점 영상을 획득한다. 이때, 객체의 동작에 대한 애니메이션을 추가로 획득하여 동적 3D 객체인 FBX(Filmbox)를 생성한다^[10-12]. FBX 파일은 응용 프로그램 실행 즉시 서버로부터 전송되며 응용 프로그램 내 가상공간에서의 자유 시점 서비스를 위해 사용자의 이벤트를 추가한다. 인터페이스는 사용자의 편의성을 위해 마우스와 키보드를 사용하며, 각각의 이벤

트에 따라 객체 또는 공간에 대한 제어가 가능하도록 한다. 표 2는 응용 프로그램에서 3D 객체의 제어가 가능한 마우스 및 키보드 이벤트를 나타낸다.

표 2. 3D object view 제어를 위한 키보드 및 마우스 이벤트
Table 2. Keyboard and mouse event for 3D object view control

Key	Event
Space	Animation Play/Stop
W, A, S, D	Camera Move
Mouse Move	Camera Rotation
Mouse Wheel	Camera Zoom In/Out
Mouse Left Click + Drag	Object Move
Mouse Right Click + Drag	Object Rotation

마우스 이벤트에 의한 제어 부분에서 마우스의 드래그를 통해 가상공간에 표출된 다수의 객체 중 특정 객체가 선택되어 이벤트가 제어 될 수 있도록 구현한다. 객체를 z축 방향에서 바라보고 있다고 가정할 때, 실시간으로 화면상에서의 2차원 마우스 좌표를 전달받아 가로 방향으로 왼쪽 마우스 버튼의 드래그한 거리를 이용하여 객체의 위치를 이동 시키며, 오른쪽 마우스 버튼과 드래그한 방향을 이용하여 객체를 y축 기준으로 회전한다. 이때, 객체의 위치는 3차원 좌표이기 때문에 2차원 화면 좌표로 변환 후 계산한다.

또한 응용 프로그램의 가상공간은 프로그램에 내장된 카메라를 통해 화면으로 표출된다. 이때, 마우스의 이동과 휠 이벤트를 받아 각각 카메라 회전, 줌 인/아웃이 가능하게 한다. 마우스가 상, 하, 좌, 우 방향으로 움직인 범위를 이용하여 카메라의 x, y축을 회전하며, 휠을 위로 당길 경우 확대, 아래로 당길 경우, 축소된다. 각 마우스 이벤트에 대한 수도코드는 표 3과 같다.

표 3에서 Δx_{obj} 는 왼쪽 마우스 버튼으로 드래그한 x 거리를 의미하며, y 좌표 $y_{obj_{pos}}$ 는 고정하여 x축으로만 이동이 가

표 1. 리스너에 대한 up vector와 forward vector 수도 코드
Table 1. Pseudo code for up vector and forward vector of listener

Up Vector	Forward Vector
$x_{up} = -1 \times \sin y_{angle} \times \cos x_{angle}$	$x_{forward} = \cos y_{angle} \times \cos x_{angle}$
$y_{up} = \cos y_{angle}$	$y_{forward} = \sin y_{angle}$
$z_{up} = -1 \times \sin y_{angle} \times \sin x_{angle}$	$z_{forward} = \cos y_{angle} \times \sin x_{angle}$

표 3. 마우스 이벤트에 대한 객체 이동, 회전과 카메라 회전, 줌 인/아웃 수도코드

Table 3. Pseudo code of mouse event for object move, object rotation, camera rotation and camera zoom in/out

Object	Move	Rotation
	$x_{obj_{pos}} = x_{pre_obj_{pos}} + \Delta x_{obj}$ $y_{obj_{pos}} = y_{pre_obj_{pos}}$	$y_{obj_{rot}} = y_{pre_obj_{rot}} + (\Delta y_{obj} \times \alpha)$
Camera	Rotation	Zoom In/Out
	$x_{cam_{rot}} = x_{pre_cam_{rot}} + (\Delta x_{cam} \times \beta)$ $y_{cam_{rot}} = y_{pre_cam_{rot}} + (\Delta y_{cam} \times \beta)$ $\text{where, } -45 \leq x_{cam_{rot}} \leq 45$	$\text{if}(fov \leq 20 \ \&\& \ \Delta z < 0)$ $fov = 20$ $\text{else if}(fov \geq 60 \ \&\& \ \Delta z > 0)$ $fov = 60$ $fov = fov_{pre} - (\Delta z \times \gamma)$

능하도록 한다. 수정된 객체의 2차원 위치 좌표는 3차원 좌표로 다시 변환하여 드래그한 지점으로 이동시킨다. Δy_{obj} 는 오른쪽 마우스 버튼으로 드래그하여 좌, 우로 움직인 이동 범위를 의미하며, 특정 상수 α 를 곱하여 회전 속도를 조정한다. 본 논문에서는 α 를 3으로 두어 사용하였다. 또한 Δx_{cam} , Δy_{cam} 는 마우스로 상, 하, 좌, 우로 이동한 범위를 나타내며, 특정 상수 β 를 곱하여 카메라의 회전 속도를 조정한다. 본 논문에서 β 는 2로 두어 사용하였으며, 마우스가 상, 하로 움직였을 때 카메라가 특정 각도 이상으로 회전이 되면 화면이 뒤집혀서 표출이 되므로 회전 범위를 제한하였다. Δz 는 마우스 휠이 움직인 방향과 크기를 의미하며, 마우스 휠의 방향과 크기에 맞게 γ 속도로 시야각을 조정하도록 구현하였다. 이때 γ 는 0.1로 두었다.

다음으로 키보드에 의한 이벤트 제어 부분에서 키 입력을 받아 카메라 이동이 가능하게 한다. 카메라 z축 방향에

표 4. 키보드 이벤트에 대한 카메라 이동 수도코드

Table 4. Pseudo code of keyboard event for camera move

Move
$\Delta move$ $= (0, 0, -1) \times \Delta v + (-1, 0, 0) \times \Delta h$ $(x_{cam_{pos}}, y_{cam_{pos}}, z_{cam_{pos}})$ $= (x_{pre_cam_{pos}}, y_{pre_cam_{pos}}, z_{pre_cam_{pos}}) + \Delta move$

서 촬영하고 있다고 가정할 때, 각각 수직/수평 방향에 해당하는 키를 이용하여 카메라의 3차원 좌표를 각각 -1~1씩 증가하거나 감소시킨다. 표 4에서 $\Delta move$ 는 카메라가 이동하는 방향과 크기를 의미하며, 각각 수평/수직 방향으로의 이동 범위 $\Delta h, \Delta v$ 를 곱하여 계산한다.

3.4 First person view

First person view는 콘텐츠 내에 특정 객체의 시점을 사

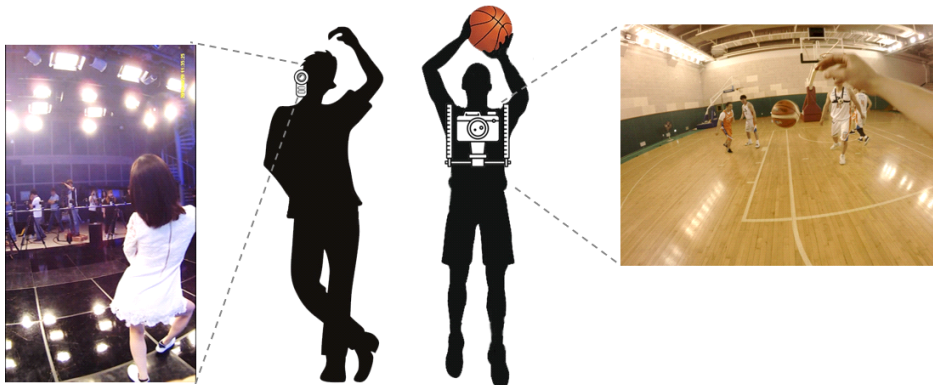


그림 4. First person view를 위한 시점 카메라 구성 모습 예시 및 획득 결과

Fig. 4. Camera configuration and acquisition results for first person view

용자가 간접적으로 시청하는 뷰를 나타내므로 콘텐츠 내의 특정 객체에 장착된 카메라로 획득한 영상을 서버로부터 전송받아 그대로 표출하게 된다. 따라서 별도의 영상과 음향이 렌더링 없이 수행되지만, 고품질의 인칭 시점 뷰를 위해 영상 획득 시 짐벌을 장착하여 안정화된 영상을 획득하고자 하였다. 그림 4는 First person view에 대한 영상 데이터 획득을 위해 일인칭 시점 카메라를 설치한 예시를 나타낸다.

III. 실험 및 결과

제안된 시스템의 성능 검증을 위하여 아이돌 공연과 농구 경기 콘텐츠를 테스트 베드로 설정하고 각 뷰에 대한 영상 및 음향 데이터를 획득하였으며, 앞서 제시한 네 종류의 자유 시점 비디오를 표출하도록 실험 환경 구성 및 프로

그램을 구현하였다. 각각의 콘텐츠 제작 현장에서 설치된 다시점 카메라와 마이크를 통해 영상 및 음향을 획득하였으며, 각 뷰의 표출 방법에 맞게 편집 및 보정 과정을 거쳐 PC에 저장하였다. 이때, 데이터는 스트리밍 할 때 용이하도록 Inward view, Outward view, First person view, 3D object view 순으로 폴더를 생성하여 관리된다. 그림 5와 같이 내부망이 설치된 환경에서 데이터가 저장된 PC에 Nginx를 설치하여 서버 PC로 처리하였으며, 통합 플레이어가 설치된 다른 PC에서 서버 PC에 접속하여 데이터를 받아 자유 시점 비디오가 표출되도록 하였다. 또한 인터넷이 연결되지 않은 환경에서도 프로그램이 실행되도록 두 PC를 P2P (Peer-to-Peer network)로 연결하여 컴퓨터 간의 통신을 통해 실행이 가능하도록 하였다.

프로그램 실행 시 기본 모드인 Inward view가 실행되고, 시청하길 원하는 자유 시점 뷰를 택일하여 수행하거나 중

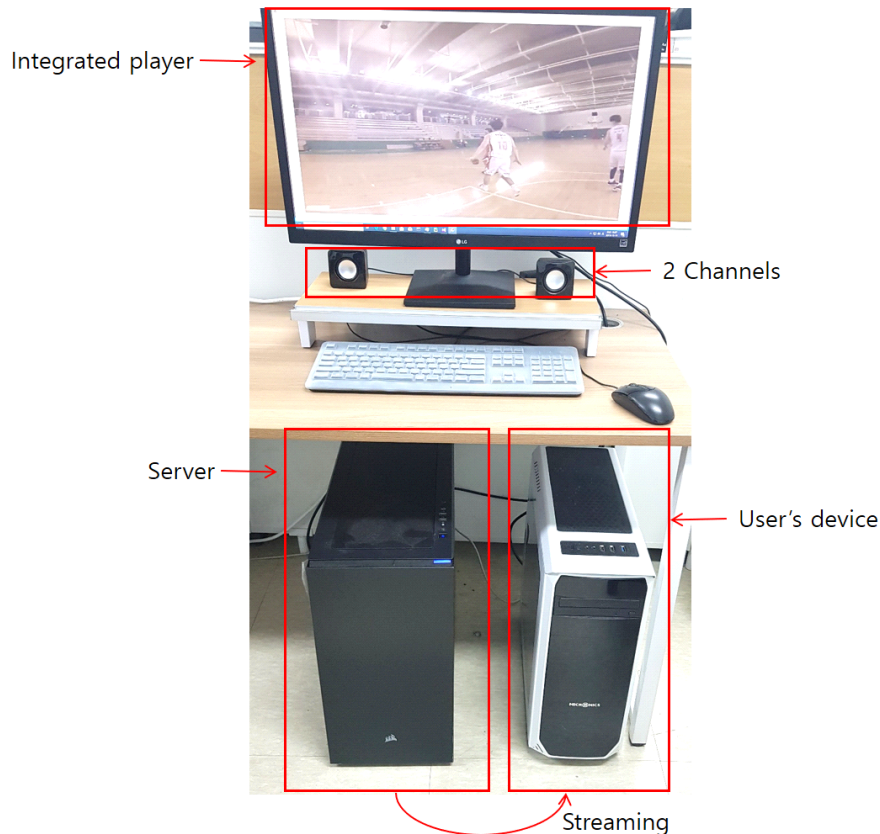


그림 5. 성능 검증을 위한 실험 환경 구성

Fig. 5. Configuration of experimental environment for performance verification

료된다. 또한 실감성이 있는 자유 시점 비디오 서비스를 위해 각각의 뷰 내에서 시점을 제어할 수 있는 기능을 추가하였다. 이는, 시청하길 원하는 뷰를 선택하면 해당 기능이 활성화되어 자유 시점 뷰 내에서 시점 제어가 가능하며 이에 따른 영상을 화면으로 표출한다.

그림 6은 자유 시점 비디오 스트리밍을 위해 구성된 통합 플레이어를 나타낸다. 먼저 시청하고자 하는 콘텐츠의 종류를 콤보 박스로 선택할 수 있다. 아이돌 공연 또는 농구 경기 콘텐츠에 대한 데이터를 받도록 구현하였으며, **play**와 **pause**, **exit** 버튼을 통해 서버로부터 영상 및 음향 데이터 스트리밍 하는 것을 수행하거나 종료한다. 뷰 선택부에서는 네 개의 뷰를 라디오 버튼으로 실행 및 종료할 수 있으며, 결과 영상이 메인 화면에 표출된다. **Inward view**는 슬라이더 바가 활성화되어 시점 간의 전환이 이루어지며 해당 시점에 대한 영상을 화면에 표출함으로써 자유 시점 비디오 제어가 가능하다. **Outward view**는 메인 화면에서 마우스 왼쪽 이벤트를 입력받아 전방위로 콘텐츠 감상이 가능하다. **First person view**는 별도의 시점 제어 기능 없이 콘텐츠 영상이 메인 화면에 출력된다. **3D object view**를 선택하게 되면 별도로 제작한 응용 프로그램 창이 생성되며, 마우스와 키보드 이벤트를 통해 객체를 중심으로 관찰이 가능하다. 스피커 환경 선택부에서는 사용자의 시청 환경에 적합한 스피커 환경을 2채널과 5.1채널 중 선택한다. 각 뷰에서 선택된 채널 수로 음향이 렌더링된다.

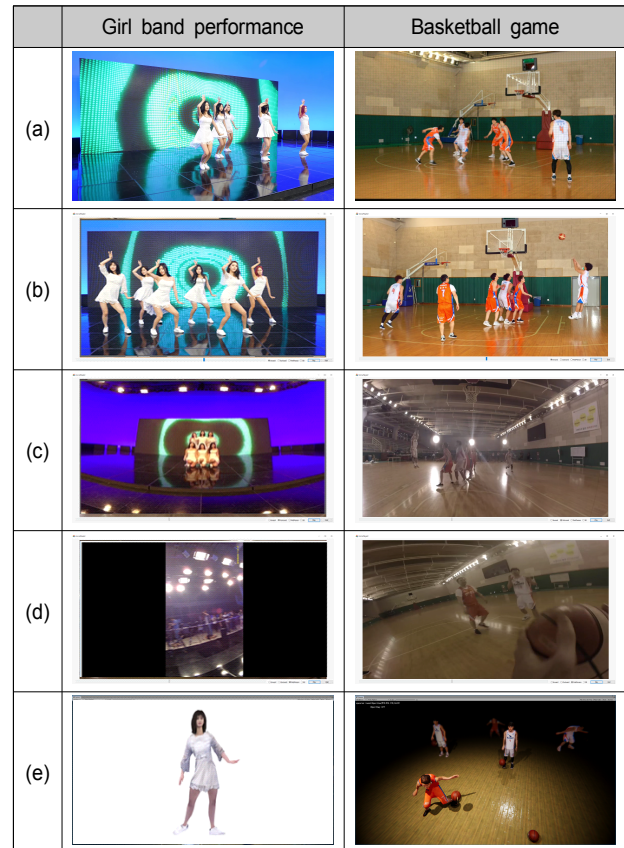


그림 7. 프로그램 실행 결과 (a) 콘텐츠 영상 데이터 (b) Inward view (c) Outward view (d) First person view (e) 3D object view
Fig. 7. Result of program implementation (a) Contents video data (b) Inward view (c) Outward view (d) First person view (e) 3D object view

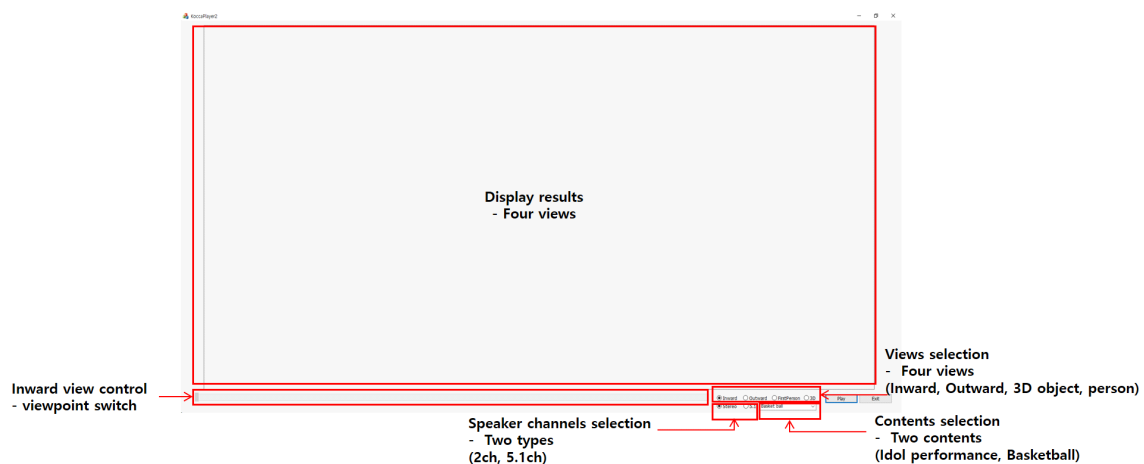


그림 6. 통합 플레이어 구성
Fig. 6. Configuration of integrated player

제안하는 시스템은 앞서 제시한 네 가지의 모드를 사용자의 선택에 따라 각 영상 및 음향을 개별적으로 표출하기 때문에 처리 속도를 줄이는 것이 중요한 문제이다. 이에 본 논문에서는 Thread를 사용하여 병렬 프로그래밍을 수행하였다. 결과적으로 데스크톱 PC에서의 안정적인 스트리밍 서비스를 제공받을 수 있음을 확인하였으며, 네 가지의 자유 시점 비디오 서비스 기반의 통합 플레이어 시스템 구현 성능을 확인하였다.

IV. 결 론

본 논문에서는 다시점 영상 데이터를 활용한 통합 플레이어 프로그램을 구현하였다. 다양한 표출 방법을 갖는 자유 시점 비디오 서비스에서 원하는 자유 시점 뷰를 선택하여 콘텐츠를 시청할 수 있도록 함으로써, 모든 시점을 자유 자재로 볼 수 있는 자유 시점 비디오 서비스의 이점을 최대한 활용하고자 하였다. 또한 각각의 자유 시점 뷰에서 사용자의 시점 제어를 통해 몰입감과 현실감을 제공하고자 하였으며, 개별 플레이어로 시청을 해야만 했던 자유 시점 비디오 서비스를 하나의 플레이어로 통합하여 사용자의 편의성을 제공하도록 하였다.

현재 구현된 통합 플레이어는 시청 가능한 콘텐츠에 대해 미리 확보된 영상 및 음향 데이터에 한해서만 동작하므로 촬영된 데이터를 각 뷰의 표출 방법에 맞게 편집 및 변환 과정을 거쳐 서버에 저장해야 한다. 이를 개선하기 위해서는 서버 플랫폼을 발전시켜 연결된 다시점 카메라로부터 실시간으로 받은 데이터에 대해 보정 과정이 자동화되어 처리되는 기술과 고성능의 하드웨어가 수반되어야 할 것이다. 그러나 다양한 시점과 뷰로 콘텐츠를 시청할 수 있다는 점은 한정된 시점을 이용한 시스템에 비해 강점을 갖기 때문에, 자유 시점 뷰에 대한 정의를 확장하여 추후 속도 개선 및 추가 기능 구현을 통해 편리한 시스템이 구축될 수 있을 것으로 기대된다. 더 나아가 카메라에서 획득한 색상 영상으로부터 가상의 영상을 합성하여 Inward view에서 시점간의 부드러운 전환을 향상시킬 예정이다.

참 고 문 헌 (References)

- [1] M. Domanski, O. Stankiewicz, K. Wegner and T. Grajek, "Immersive Visual Media - MPEG-I:360 Video, Virtual Navigation and Beyond", *Proceeding of International Conference on Systems, Signals and Image Processing(IWSSIP)*, Poznan, Poland, pp.1-9, May, 2017, doi: 10.1109/IWSSIP.2017.7965623
- [2] X. Zhang, L. Toni, P. Frossard, Y. Zhao and C. Lin, "Adaptive Streaming in Interactive Multiview Video Systems", *Transactions on Circuits and Systems for Video Technology*, Vol.29, No.4, pp.1130-1144, April, 2019, doi: 10.1109/TCSVT.2018.2819804
- [3] G. Park, "Free-Viewpoint Content Composition Technology Development Case", *Broadcasting and Media Magazine*, Vol.24, No.3, pp.24-34, July, 2019.
- [4] G. Park, "Free-Viewpoint Video Technology Trends and Free Selection View Development Case", *The Journal of The Korean Institute of Communication Sciences*, Vol.36, No.12, pp.3-9, Nov, 2019.
- [5] G. Lafruit, M. Domanski, K. Wegner and T. Grajek, et al, "New Visual Coding Exploration in MPEG: Super-MultiView and Free Navigation in Free Viewpoint TV", *Electronics Imaging, Stereoscopic Displays and Applications XXVII*, pp.1-9, Feb, 2016, doi: 10.2352/ISSN.2470-1173.2016.5.SDA-426
- [6] S. Song, Y. Park and J. Wee, "Streaming Technology for Free-Viewpoint Service", *The Journal of The Korean Institute of Communication Sciences*, Vol.36, No.12, pp.32-37, Nov, 2019.
- [7] S. Yairi, Y. Iwaya, M. Kobayashi, M. Otani, Y. Suzuki and T. Chiba, "The Effects of Ambient Sounds on The Quality of 3D Virtual Sound Space", *Proceeding of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, Japan, pp.1122-1125, Sept, 2009, doi: 10.1109/IHH-MSP.2009.267
- [8] Fmod API Documentation, <https://www.fmod.com/resources/documentation-api>.
- [9] S. Jeon, C. Kim and G. Park, "Implementation of Omni-directional Image Viewer Program for Effective Monitoring", *Journal of Broadcasting Engineering*, Vol.23, No.6, pp. 939-946, Nov, 2018, doi: 10.5909/JBE.2018.23.6.939
- [10] J. Jeong, M. Yoon, S. Kim and G. Park, "Design and Production of Realtime 3D Animation Viewer based on Motion Capture", *Proceeding of The Institute of Electronics and Information Engineers Conference*, Jeju, Korea, pp.531-535, July, 2019.
- [11] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", *Transactions on Pattern Analysis and Machine Intelligence*, pp.1-14, June, 2018, doi: 10.1109/TPAMI.2019.2929257
- [12] J. Jeon, H. Lim and C. Kim, "A Study of Dynamic 3D Modeling based on Photogrammetry Technology for Object Observation View", *The Journal of The Korean Institute of Communication Sciences*, Vol.36, No.12, pp.38-44, Nov, 2019.

저 자 소 개



양 지 희

- 2012년 2월 : 강릉원주대학교 멀티미디어공학과 공학사
- 2015년 2월 : 서울과학기술대학교 미디어IT공학과 공학석사
- 2015년 3월 ~ 현재 : 서울과학기술대학교 나노IT디자인융합대학원 정보통신미디어공학전공 박사과정
- 2020년 3월 ~ 현재 : 서울과학기술대학교 지능형미디어연구센터 연구원
- ORCID : <https://orcid.org/0000-0003-2776-5487>
- 주관심분야 : 디지털 영상처리, 실감미디어, 자유시점영상



송 민 기

- 2018년 2월 : 수원대학교 전자공학과 공학사
- 2018년 3월 ~ 2020년 2월 : 서울과학기술대학교 미디어IT공학과 공학석사
- ORCID : <https://orcid.org/0000-0002-9914-9993>
- 주관심분야 : 컴퓨터비전, 디지털 영상처리



박 구 만

- 1984년 2월 : 한국항공대학교 전자공학과 공학사
- 1986년 2월 : 연세대학교 전자공학과 공학석사
- 1991년 2월 : 연세대학교 전자공학과 공학박사
- 1991년 3월 ~ 1996년 9월 : 삼성전자 신호처리연구소 선임연구원
- 2016년 1월 ~ 2017년 12월 : 서울과학기술대학교 나노IT디자인융합대학원 원장
- 1999년 8월 ~ 현재 : 서울과학기술대학교 전자IT미디어공학과 교수
- 2006년 1월 ~ 2007년 8월 : Georgia Institute of Technology Dept.of Electrical and Computer Engineering, Visiting Scholar
- ORCID : <https://orcid.org/0000-0002-7055-5568>
- 주관심분야 : 컴퓨터비전, 지능형실감미디어