일반논문 (Regular Paper)
방송공학회논문지 제23권 제6호, 2018년 11월 (JBE Vol. 23, No. 6, November 2018)
https://doi.org/10.5909/JBE.2018.23.6.876
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

360 VR E-Sports 중계를 위한 실시간 360 VR 3D Stereo 게임 영상 획득에 관한 연구

김 현 욱[a], 이 준 석[a)‡], 양 성 현[b)]

Study of Capturing Real-Time 360 VR 3D Game Video for 360 VR E-Sports Broadcast

Hyun Wook Kim[a)], Jun Suk Lee[a)‡], and Sung Hyun Yang[b)]

요 약

최근 VR(Virtual Reality) 기반의 e-sports 중계 시장이 자리를 잡고 있다. 하지만 국내의 경우 시장경쟁력 확보를 위한 기술 개발이 미비한 실정이다. 이미 해외의 SLIVER TV, Facebook에서는 e-sports를 4K 30FPS급의 360 VR으로 중계할 수 있는 기술을 개발하고 사업화까지 진행하고 있다. 하지만 360 VR 영상으로 활용하기에는 2D영상으로 몰입감이 저하되고, 해상도가 낮고, 멀미현상이 있다. 이에 본 논문에서는 e스포츠 VR 중계방송을 위해 게임 내 공간을 4K 3D 60FPS으로 360영상으로 획득 할 수 있는 가상카메라 기술을 제안하고, 구현하였다. 그리고 게임엔진 샘플게임과, 상용게임에 가상카메라를 설치하고, 실험을 통해 최대 4K/60FPS Stereo 360 영상 획득이 가능함을 검증하였다.

Abstract

Although e-sports broadcasting market based on VR(Virtual Reality) is growing in these days, technology development for securing market competitiveness is quite inadequate in Korea. Global companies such as SLIVER and Facebook already developed and are trying to commercialize 360 VR broadcasting technology which is able to broadcast e-sports in 4K 30FPS VR video. However, 2D video is too poor to use for 360 VR video in that it brings less immersive experience and dizziness and has low resolution in the scene. this paper, we not only proposed and implemented virtual camera technology which is able to capture in-game space as 360 video with 4K 3D by 60FPS for e-sports VR broadcasting but also verified feasibleness of obtaining stereo 360 video up to 4K/60FPS by conducting experiment after setting up virtual camera in sample games from game engine and commercial games.

Keyword : 360 VR Game, esports, 360 Video Capture, CUDA, 360 VR Video Live Streaming

a) 전자부품연구원 콘텐츠응용연구센터(Contents Convergence Research Center, KETI)
b) 광운대학교 전자공학과(Department of Electronic Engineering, Kwangwoon University)
‡ Corresponding Author : 이준석(Jun Suk Lee)
E-mail: jslee@keti.re.kr
Tel: +82-2-6388-6634
ORCID: https://orcid.org/0000-0002-7028-9271

※본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2017년도 문화기술 연구개발 지원사업으로 수행되었음(R2017030018).
※This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2017 (R2017030018).
· Manuscript received September 10, 2018; Revised October 16, 2018; Accepted October 22, 2018.

Ⅰ. 서 론

최근 e스포츠(e-sports)와 VR(Virtual Reality) 환경을 접목한 중계서비스 시장이 형성되어 활발하게 확장되고 있다. SLIVER TV의 경우 인터넷 TV 스트리밍 플랫폼으로 League of Legend, Dota2, Counter Strike 등 다양한 게임 e스포츠 중계영상을 360 VR 영상으로 제공하고 있으며, 이미 2016년도에는 e스포츠 경기를 실시간으로 VR로 시청할 수 있는 LiveVRCast 기술 등을 발표하고 상용화에 성공하였다. 최근 GDF2018에서는 VR-e스포츠 쇼 케이스를 통해 VR e스포츠의 방향을 제시하는 등 VR 기반의 e스포츠 시장이 계속하여 발전할 것으로 예상되고 있다[1-2].

하지만 국내 e스포츠 시장의 경우 e스포츠의 종주국이란 단어가 무색하게도 VR-e스포츠 중계 기술에 대한 확보가 이루어지지 않고 있다.

이미 해외에서는 VR e스포츠 중계를 위한 활발한 투자가 이루어지고 있으며, 대표적으로 SLIVER TV에서는 620만 달러정도의 투자를 받아 게임 내 360 VR 영상 생성을 위한 가상 카메라 어레이 기술인 “Auto Contents Generation Technology”를 개발하였으며, Facebook에서는 Unity 기반의 4K 30FPS 정도의 360 VR 영상을 생성할 수 있는 SDK를 개발하여 공개 하였다[3]. 대부분 현재까지 공개된 360 VR 영상을 생성하는 SDK, 즉 가상 카메라들은 2D 4K 30FPS의 성능으로 360 VR 영상으로서 해상도가 낮고, 몰입감 등이 부족하다. 이에 본 논문에서는 몰입감을 높이고, 멀미현상을 감소시키기 위해 4K Stereo 60FPS급의 게임 내 공간을 360 VR 영상으로 획득할 수 있는 위한 가상 카메라 기술을 설계하고 제안한다. 그리고 360 VR 게임 영상 획득용 라이브러리 구현을 통해 실험을 통해 그 성능을 검증하였다. 본 논문의 구성은 2장에서 3D 게임 내 360도 공간을 실시간으로 촬영할 수 있는 360 Stereo 가상 카메라를 설계하였고, 3장에서 설계하고 제안하는 가상카메라를 구현하여 실험을 통해 성능을 확인하였다. 마지막으로 4장에서 결론을 맺는다.

Ⅱ. 360도 Stereo 게임영상 획득을 위한 가상 카메라 설계

본 절에서는 게임 내 공간을 360도 스테레오 영상으로 획득하기 위해 가상카메라를 배치하는 방법을 제안하고, 큐브 맵으로 촬영된 영상을 실시간으로 ERP방식으로 변환하기 위한 연구를 수행하였다. 그리고 최종적으로 획득한 영상을 기반으로 고정 Frame rate으로 구성되는 영상 생성을 위한 프레임을 삽입하는 방법을 설계하였다.

1. 360도 Stereo 영상 획득을 위한 가상 카메라 배치 요구 사항 분석

대부분의 게임 엔진은 6방향을 동시에 렌더링 하는 Cubemap 카메라를 제공하지만 이를 통해 획득하는 영상은 Stereo 영상으로 사용할 수 없다. 사람의 눈과 마찬가지로

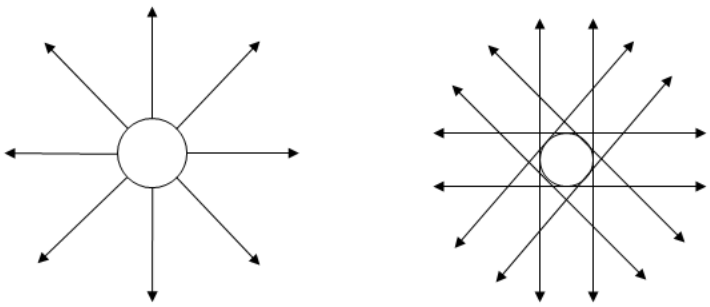


그림 1. 모노 및 스테레오 360촬영을 위한 카메라 배치
Fig. 1. Mono & Stereo Omni-Directional Camera Position

Stereo 영상을 획득하기 위해서는 같은 방향을 보는 두 대의 카메라를 회전하여 각 방향의 영상을 획득할 필요가 있다[4].

Mono 360영상을 획득하는 경우 카메라 회전 중심이 같아 Top, Down 영상을 한 번씩 획득해도 각 방향의 세로 방향 FOV(Field of View)를 180° 확보하는데 문제가 없지만 Stereo 360 영상을 획득하는 경우 각 방향마다 Top, Down영상을 추가적으로 획득할 필요가 있다. 따라서 각 카메라의 가로 방향 FOV가 90°인 경우 Mono의 경우 6대의 카메라, Stereo의 경우 24대의 카메라를 필요로 하게 된다.

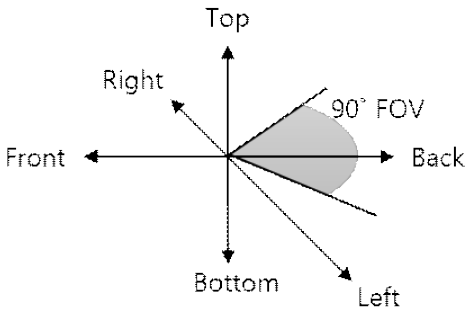
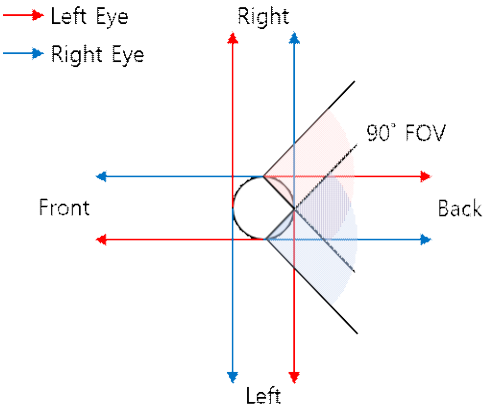


그림 2. 90° FOV 카메라를 이용한 모노 배치 방법
Fig. 2. Virtual mono camera placement for 90° FOV

2. 360도 Stereo 게임 영상 획득을 위한 가상 카메라 배치 설계

Stereo의 경우 카메라 회전 중심이 어긋나 있기 때문에 최종적으로 만들어지는 360°Stereo 영상에 다소의 왜곡이 발생하게 된다. 가로 방향 FOV를 줄이고 가상 카메라의 수를 늘림으로써 360° 영상과 실제로 사람이 인식하는 공간과의 차이를 줄일 수 있다. 이를 위해서, 카메라를 적절한 위치에 배치시켜야 하며, 각 카메라마다 적절한 변환 연산을 적용한다. 해당 연산은 다음 과정을 통해 얻을 수 있다.



(a) Top view of stereo camera placement
(b) Side view of stereo camera placement

그림 3. 90° FOV 카메라를 이용한 스테레오 배치 방법
Fig. 3. Virtual stereo camera placement for 90° FOV

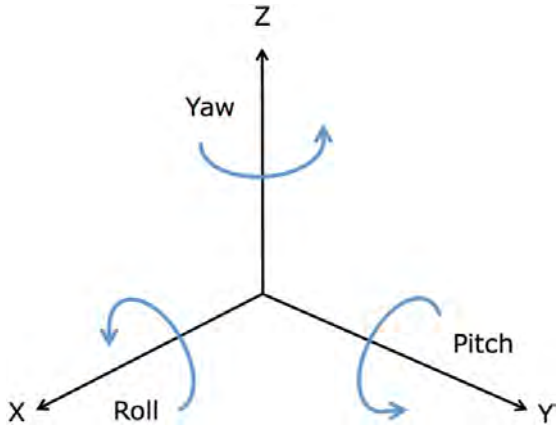


그림 4. 3D 공간 좌표 축
Fig. 4. 3D spatial coordinate axes

3D 공간에서 각 좌표축에 대한 회전을 *yaw*, *pitch*, *roll* 이라고 하고 각각을 다음과 같이 회전 행렬로 표현할 수 있다.

$$
\begin{aligned}
yaw(\alpha) &= \begin{bmatrix} \cos\alpha, & 0, & \sin\alpha, & 0 \\ 0, & 1, & 0, & 0 \\ -\sin\alpha, & 0, & \cos\alpha, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix} \\
pitch(\beta) &= \begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & \cos\beta, & -\sin\beta, & 0 \\ 0, & \sin\beta, & \cos\beta, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix} \\
roll(\gamma) &= \begin{bmatrix} \cos\gamma, & -\sin\gamma, & 0, & 0 \\ \sin\gamma, & \cos\gamma, & 0, & 0 \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}
\end{aligned} \tag{1}
$$

카메라 rig에 대한 회전은 다음과 같이 yaw, $pitch$, $roll$의 행렬 곱으로 정의된다.

$$
rig(\alpha,\beta,\gamma) = yaw(\alpha)pitch(\beta)roll(\gamma) \tag{2}
$$

각 카메라는 카메라 rig에 연결되어 움직이게 되므로 각 카메라는 기존 3D공간상의 x,y,z축이 카메라 rig 회전과 동일하게 회전된 새로운 x',y',z'축에서 회전하게 된다.

각 카메라의 z'축에 대한 회전각을 θ라고 정의할 때, 회전각 θ에 대한 top, $front$, $down$방향의 카메라의 회전을 다음과 같이 정의할 수 있다.

$$
\begin{aligned}
cam_{top}(\theta) &= yaw(\theta)pitch(\frac{\pi}{2}) \\
cam_{front}(\theta) &= yaw(\theta) \\
cam_{down}(\theta) &= yaw(\theta)pitch(\frac{3\pi}{2})
\end{aligned} \tag{3}
$$

Stereo 영상 획득을 위해 좌/우안 카메라를 다른 좌표에 배치해야 하며 이는 y'축에 대한 이동으로 정의가 가능하다. 이 때 이동하는 거리는 미리 정의된 상수 Eye Distance의 절반이다. 이를 행렬로 표현하면 다음과 같다.

$$
\begin{aligned}
Eye_{left} &= \begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 1, & 0, & -\frac{EyeDistance}{2} \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix} \\
Eye_{right} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0, & 1, & 0, & \frac{EyeDistance}{2} \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}
\end{aligned} \tag{4}
$$

카메라 rig의 3D 공간상에서의 좌표를 x,y,z 각 좌표축에 대한 회전각을 α,β,γ 라고 할 때, $(0,0,0)$좌표에서 x축을 바라보고 있는 각 카메라에 대한 변환 행렬은 다음과 같다.

θ값의 단위는 배치되는 카메라 방향 수 n에 따라 달라지며, 이 때 각 카메라의 촬영 범위는 아래 그림 5에서 원 안에 내접한 직선의 범위와 같다.

3. 가상카메라 Cubemap 촬영 영상 3D Stereo Equirectangular 변환

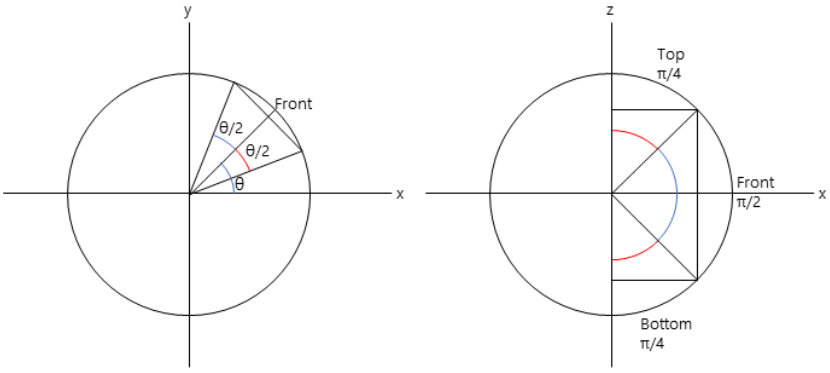


그림 5. 가상 카메라 배치 및 촬영 방향
Fig. 5. Virtual camera placement and shooting directions

$$
\begin{aligned}
&eyedir = \{left,\ right\} \\
&camdir = \{top,\ front,\ down\} \\
&cam_{eyedir,\ camdir}(\theta,x,y,z,\alpha,\beta,\gamma) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} Eye_{eyedir} cam_{camdir}(\theta) rig(\alpha,\beta,\gamma)
\end{aligned} \tag{5}
$$

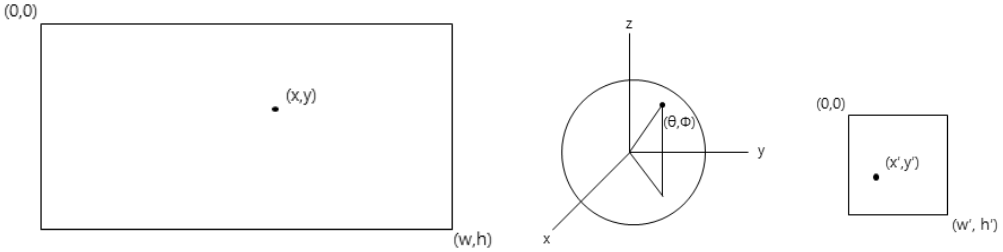


그림 6. 360 VR 파노라마 이미지 변환
Fig. 6. 360 VR Panorama Equirectangular Projection Convert

가상 카메라가 촬영한 이미지를 360 파노라마 이미지로 변환하기 위해 파노라마 이미지의 x,y좌표에 대한 카메라가 촬영한 이미지의 x',y'좌표를 구하는 공식을 필요[5]로 하며, 본 절에서 제안한다.

파노라마는 (ERP)(Equirectangular Projection) 방식으로 변환되며 평면 좌표를 구의 특정 좌표에 매핑한 후 해당 좌표를 다시 카메라 이미지의 특정 좌표에 매핑 해야 한다.

파노라마 이미지의 가로 및 세로 길이를 각각 w,h라고 하고 파노라마 이미지의 특정 좌표의 값을 (x,y)로 표현할 때 해당 좌표의 값을 구 표면상의 좌표 (θ,ϕ)로 변경하는 공식은 다음과 같다.

$$\begin{aligned}\theta &= \frac{2x}{w}\pi \\ \phi &= \frac{y}{h}\pi\end{aligned} \tag{6}$$

이 때 파노라마 영상의 중심이 실제 중심과 일치하도록 좌표계를 이동하여 다음과 같은 식을 도출한다.

$$\begin{aligned}\theta &= \frac{2x-w}{w}\pi \\ \phi &= \frac{2y-h}{2h}\pi\end{aligned} \tag{7}$$

계산된 θ값에 따라 해당 좌표에 해당하는 카메라를 선택할 필요가 있다. 배치된 카메라의 방향이 n방향인 경우 다음을 만족하는 카메라 번호 i를 계산한다.

$$\begin{aligned}&cam_{start}(i) = \frac{\pi}{n}(i-1) \\ &cam_{end}(i) = \frac{\pi}{n}(i+1) \\ &cam_{start}(i) \leqq \theta < cam_{end}(i)\end{aligned} \tag{8}$$

선택된 카메라 번호 i에 따라 θ를 각 카메라 촬영 범위의 중심에서 상대적인 각도 θ'으로 변환할 필요가 있다. 각도 θ'으로 변환 식은 아래 식 9와 같다.

$$\theta' = \theta - cam_{start}(i) \tag{9}$$

θ'값에 따라 $top,\ bottom$을 결정하기 위한 ϕ_{thr}를 계산한다. ϕ_{thr} 계산식은 아래 식 10과 같다.

$$\begin{aligned}\mathrm{atan2}(y,x) &= \begin{cases}\arctan(\frac{y}{x}) & x>0 \\ \arctan(\frac{y}{x})+\pi & y \geqq 0, x<0 \\ \arctan(\frac{y}{x})-\pi & y<0, x<0 \\ +\frac{\pi}{2} & y>0, x=0 \\ -\frac{\pi}{2} & y<0, x=0 \\ undefined & y=0, x=0\end{cases} \\ \phi_{thr} &= \mathrm{atan2}(1,\frac{1}{\cos\theta'})\end{aligned} \tag{10}$$

ϕ'값에 따라 $top,\ front,\ bottom$카메라를 선택한다.

$$\begin{cases}front & (\phi \leqq \phi_{thr}, \phi \geqq -\phi_{thr}) \\ top & (\phi > \phi_{thr}) \\ bottom & (\phi \leftarrow \phi_{thr})\end{cases} \tag{11}$$

결정된 방향에 따라 ϕ를 각 카메라의 촬영 범위의 중심에서 상대적인 각도 ϕ'으로 변환할 필요가 있다.

$$\phi' = \begin{cases}\phi + \frac{\phi_{thr}}{2} & (front) \\ \phi - \phi_{thr} & (top) \\ \phi + \phi_{thr} & (bottom)\end{cases} \tag{12}$$

각 카메라 이미지 크기가 w',h'이라고 할 때, 각 카메라 이미지 내의 좌표 x',y'은 다음과 같이 정의될 수 있다.

$$x' = \frac{w'}{2}(\frac{\sin\theta'}{\cos\theta'}+1)$$
$$y' = \frac{h'}{2}(\frac{\sin\phi'}{\cos\phi'\cos\theta'}+1)$$

빠른 계산을 위해 (x,y)=>(x',y')에 대한 Mapping Table을 생성하고, 병렬 연산을 위해 (x',y')=>(x,y)에 대한 Reverse Mapping Table로 변환하여 사용한다.

4. 고정 Video Frame Rate를 위한 프레임 삽입

게임 영상을 구성하는 텍스쳐의 복잡도에 따라 렌더링 성능 저하가 발생한다. 때문에, 복잡도가 높은 텍스쳐로 구성된 게임 공간을 가상 카메라를 통해 원하는 영상 FPS로 획득하기 어렵게 된다. 때문에 고정적인 Video Frame rate를 위해서는 프레임 보정 작업을 필요로 한다. 이에 본 논문에서는 다음과 같이 프레임 삽입 방법을 제안한다.

우선 매 프레임마다 프레임이 생성된 시간을 기록한 후 1초 단위로 적용하고, 지난 1초 동안 생성된 프레임 개수가 60개 미만인 경우 아래 그림 7과 같이 Frame Copy를 수행한다. 그리고 시간 단위가 16.6ms인 60개의 슬롯을 준비하고 생성된 프레임을 해당 프레임이 기록된 시간에 따라 맞추어 각각의 슬롯에 정렬한다. 즉 빈 슬롯에 가장 가까운 프레임을 복제하는 것이다. 이를 통해 고정 프레임을 갖는 Video 영상을 최종적으로 획득 할 수 있다. 이때 획득되는 영상은 실제 영상을 획득하는 장치에 표시되는 화면과 그 끊김의 정도가 동일하나, 영상을 조금 더 자연스럽게 만들기 위해 프레임을 단순 복제하는 것이 아니라 여러 프레임 보간 알고리즘을 적용하여 조금 더 자연스러운 영상을 획득할 수 있다. 다만 이 경우 알고리즘의 성능에 따라 실시간으로 처리를 할 수 없을 수 있으며, 때문에 본 논문에서는 단순 복사를 사용한 프레임 삽입을 수행한다.

Ⅲ. 구현 및 실험

90°FOV를 가지는 가상 카메라를 이용하여 Stereo 360° 게임 영상을 획득하기 위해서는 총 24대의 가상 카메라를 사용해야 하지만, 방송을 위한 FPS(Frame Per Second, 24FPS film)를 확보하기 위해 추가적인 왜곡을 감수하고 Top, Down 카메라를 각각 두 대를 사용해 총 12대의 카메라를 이용하였다. 이 때 영상의 핵심은 Front 방향 카메라가 획득하는 영상이기 때문에 Top, Down영상은 Front 방향의 영상을 사용하여 Front 방향의 왜곡을 최소화하였다. 이 때 발생하는 왜곡의 차이는 그림 8에서 확인할 수 있다.

전제척인 360 VR 영상 획득을 위한 프로세스는 아래 그림 9와 같이 구현하였다. Unreal Engine에 적용 가능한

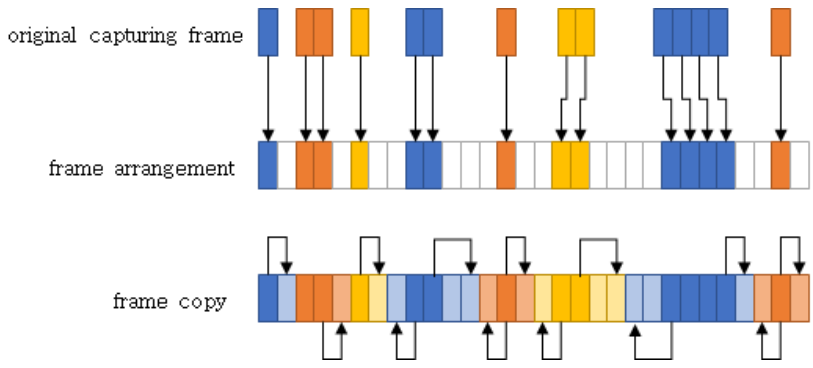


그림 7. 고정 Frame Video 생성을 위한 Frame Copy
Fig. 7. frame copy for fixed frame video

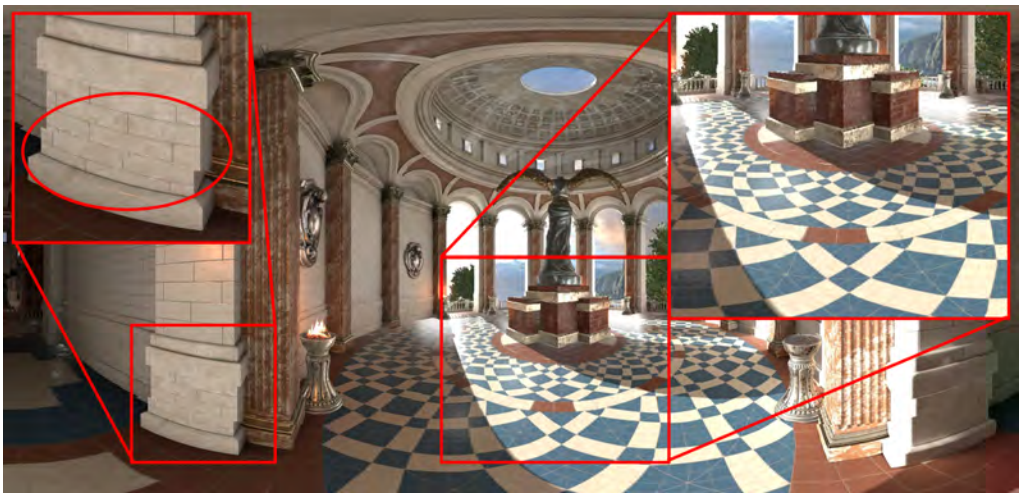


그림 8. 영상의 전방과 후방에서 발생하는 왜곡 비교
Fig. 8. Distortion comparison between front and rear image

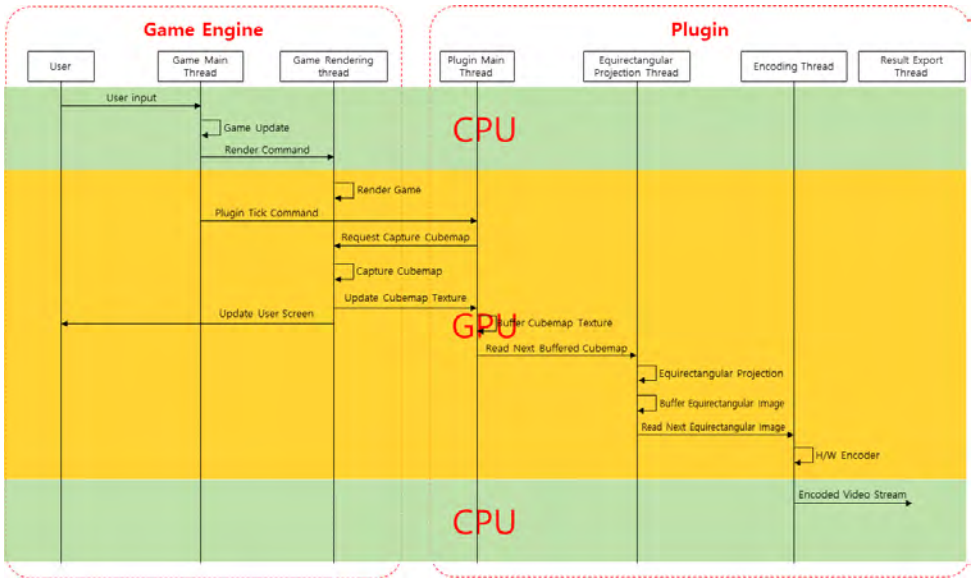


그림 9. Stereo 360 게임 영상 획득 프로세스 시퀀스다이어그램
Fig. 9. Stereo 360 Game Video Capture Process Sequence Diagram

Native Code로 구현하고 Plugin 형태로 적용하였다.

위 그림과 같이 게임 엔진에서 게임이 실행되고 화면이 렌더링 되면, Plugin에서는 가상카메라를 사용하여 영상 이미지를 획득하고, 변환, 인코딩 과정을 거쳐 최종 영상을 획득하게 된다.

영상 획득은 중계 영상 송출을 위해 일정 시간 단위로 Encoding되어 출력되게 된다. 이때, 실시간 처리를 위해 변환 영상에 대한 인코딩하는 과정은 그림 10과 같다.

본 논문에서 제안한 게임 360 VR 영상 생성 플러그인 성능 측정을 위해, 게임 엔진 샘플 프로젝트 중 Cinematic 영상 재생이 지원되는 샘플 프로젝트[6]인 1개를 이용하여 Cinematic 영상을 Stereo 360° VR Video로 획득하였고, 현재 상용화되어 운영 중인 FPS 슈팅 게임인 Infinite Fire[7]에 라이브러리를 적용하여 게임을 플레이 옵저빙 화면을 기준

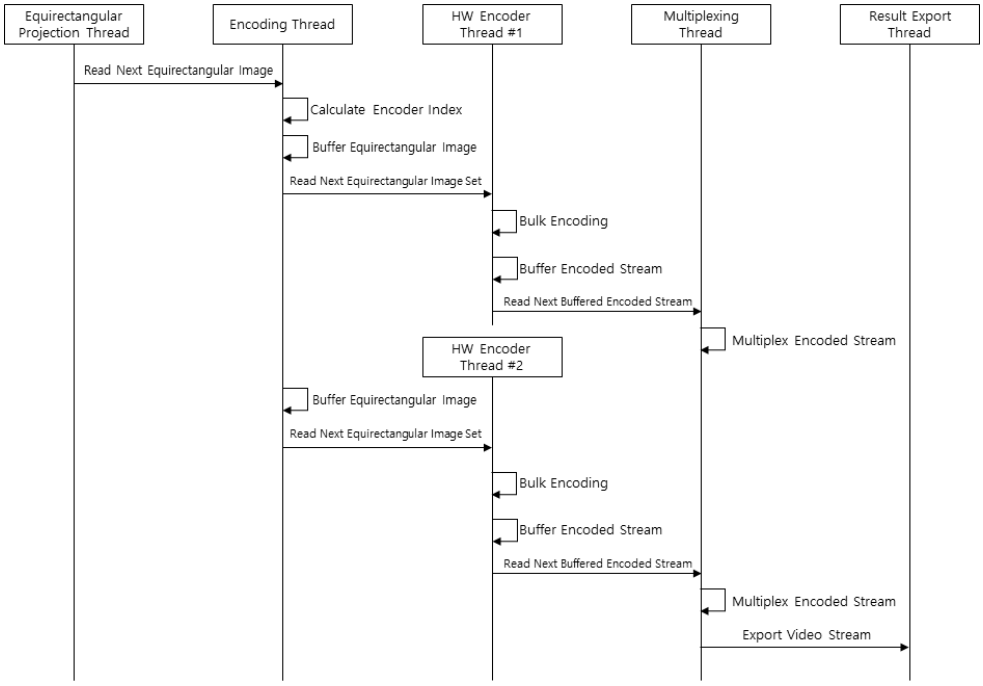


그림 10. 가상 카메라 획득 영상 병렬 인코딩 과정
Fig. 10. Virtual Camera capture video parallel encoding process

으로 Stereo 360 Stereo 360° VR Video을 획득하였다. 이 때, 각 가상 카메라의 렌더링 해상도는 1K(1024 x 1024px)로, 최종 생성되는 Stereo 360영상은 좌/우 각각 4K(4096 x 2048px)로 설정하였고 최종적인 영상은 Top-Down 방식으로 저장하여 4096 x 4096px로 영상을 출력하였다. Equirectangular Projection은 빠른 속도를 위해 CUDA로 구현하였

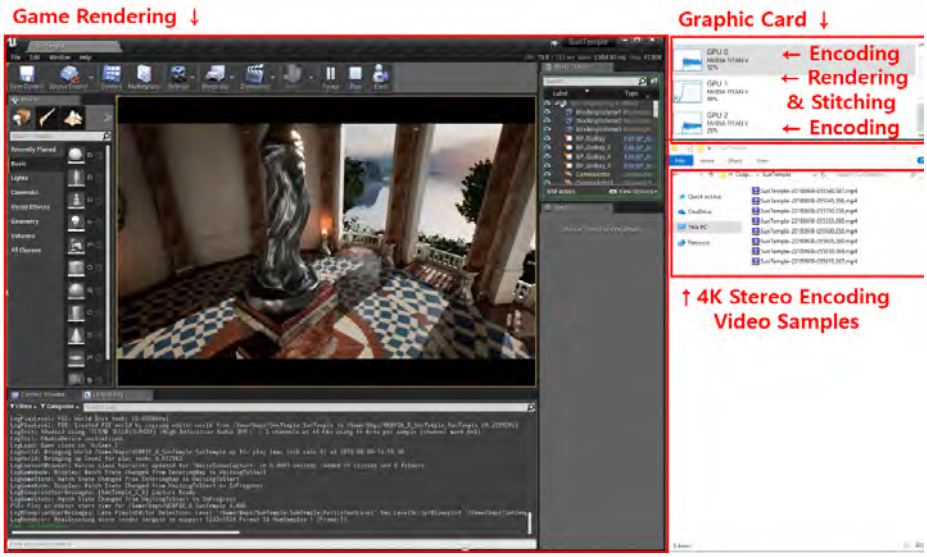


그림 11. 360 VR 영상 획득 테스트
Fig. 11. Test for 360 VR Video Capture

표 1. SunTemple, Infinite Fire 렌더링 및 캡처 성능 측정 결과
Table 1. Result of SunTeample and Elemental Demo rendering and capture performance

	#1 (SunTemple)	#2 (Infinite Fire)
Game Rendering (FPS)	651 (FHD)	62 (FHD Cap)
Game Rendering with 360 Capture (FPS)	73 (FHD)	50
4K 360 Stereo Capture (FPS)	60	50
Capture Result		

고 h264로 인코딩하여 위 그림 9, 10과 같이 360 VR Stereo Video를 생성하였다. 성능 실험을 위해 게임의 렌더링 프레임 성능을 측정하였고, 가상카메라가 동작 중일 때의 게임 렌더링 프레임 성능을 측정하였다. 그 결과는 아래 표 1과 같다.

상용 게임인 Infinite Fire의 경우 기본 렌더링 성능을 모니터 성능에 맞추어 렌더링 하도록 제어가 되어있어 어느 정도 저하가 발생하는지 확인은 어려웠지만, Unreal Game Engine에서 제공하는 SunTemple으로 확인한 결과 캡쳐를 수행할 경우 기본 게임 렌더링 성능 대비 약 90%의 렌더링 성능 저하가 있는 것으로 확인되었다. 이에 방송 중계를 위해 필요로 하는 25FPS ~ 30FPS 영상을 획득하기 위해서는 기본적으로 270FPS 이상의 렌더링을 지원하는 시스템을 구축해야 한다.

Ⅳ. 결 론

본 논문에서는 e스포츠 VR 중계방송을 위해 게임 내 공간을 360으로 획득 할 수 있는 가상카메라 기술을 설계하고 제안하였고, 프레임 보정 기술을 적용하여 고정 Frame rate를 지원하는 가상 카메라 라이브러리를 구현하였다. 그리고 게임엔진 샘플게임과, 상용게임에 가상카메라를 설치하고, 실험을 통해 최대 4K/60FPS Stereo 360 영상 획득이 가능함을 검증하였다. 다만 게임 렌더링 성능 저하를 감소하기 위한 연구를 추가적으로 필요로 한다.

앞으로 지속적인 연구를 통해 가상 카메라의 성능을 고도화하여 고사양, 고품질 게임을 360 VR로 중계할 수 있도록 4K급 3D으로 중계할 수 있도록 성능을 개선해 나갈 것이다. 또한 기존 중계 시스템들과의 연동을 통해 실제 스트리밍이 될 수 있도록 기술을 확장해 나갈 예정이다.

참 고 문 헌 (References)

[1] H. Kim, J. Yang, Y. Kim, S. Yoon and W. Park, “Virtual Camera desing for 360degree game image acquistion,” Proceeding of Korea Multimedia Society Winter Conference, Busan, Korea, pp. 317-318, 2017.

[2] KOCCA, “The apperance of 'SLIVER.tv', the E-Sports broadcasting platform via VR”, Monthly Industry Trend, Vol.06, pp29-31, 2016.

[3] FBCAPTURE SDK 2.25, https://github.com/facebook/360-Capture-SDK (accessed Sep. 01, 2018).

[4] Google Inc. - Rendering Omni-directional Stereo Content, https://develo-pers.google.com/vr/jump/rendering-ods-content.pdf (accessed Sep. 20, 2017)

[5] Under the hood: Building 360 video, https://code.fb.com/video-engineering/under-the-hood-building-360-video/, (accessed Oct. 15, 2015)

[6] UE4 Sun Temple, https://developer.nvidia.com/ue4-sun-temple, (accessed Oct, 2017).

[7] INFINITE FIRE ARENA, https://realitymagiq.com/projects/?lang=ko, (accessed Sep. 17, 2018).

저 자 소 개

김 현 욱

- 2009년 2월 : 광운대학교 컴퓨터공학과 (공학사)
- 2013년 1월 ~2016년 9월 : ㈜케이사인 정보보안연구소 선임연구원
- 2016년 10월 ~ 현재 : 전자부품연구원 연구원
- 2009년 3월 ~ 현재 : 광운대학교 대학원 전자공학과 석박사통합과정 재학 중
- ORCID : https://orcid.org/0000-0001-9725-6050
- 주관심분야 : 스마트 홈, 임베디드 시스템, 콘텐츠 응용 기술

이 준 석

- 2014년 8월 : 건국대학교 컴퓨터공학과 공학사
- 2014년 4월 ~ 2017년 12월 : (주)케이사인 정보보안연구소 선임연구원
- 2018년 1월 ~ 현재 : 전자부품연구원 콘텐츠응용연구센터 연구원
- ORCID : https://orcid.org/0000-0002-7028-9271
- 주관심분야 : VR, GPGPU, 데이터베이스

양 성 현

- 1993년 2월 : 광운대학교 전지공학(자동제어) 공학박사
- 1991년 3월 ~ 현재 : 광운대학교 전자공학과 교수
- 2005년 7월 ~ 현재 : 광운대학교 Smart H&B Technology Center 센터장
- ORCID : https://orcid.org/0000-0001-8856-764X
- 주관심분야 : 스마트 홈, 디지털 로직, 임베디드