

시점 추적 기반 Foveated Rendering을 지원하는 360 VR Tiled Video Player 구현

김현욱[†], 양성현^{††}

Implementation of 360 VR Tiled Video Player with Eye Tacking based Foveated Rendering

Hyun Wook Kim[†], Sung Hyun Yang^{††}

ABSTRACT

In these days, various technologies to provide a service of high quality of 360 VR media contents is being studied and developed. However, rendering high-quality of media images is very difficult with the limited resources of HMD (Head Mount Display). In this paper, we designed and implemented a 360 VR Player for high quality 360 tiled video image render to HMD. Furthermore, we developed multi-resolution-based Foveated Rendering technology. By conducting several experiments, We have confirmed that it improved the performance of video rendering far more than existing tiled video rendering technology.

Key words: MPEG-DASH, MPEG-DASH SRD, Live Streaming, 360 VR, Tiled Video, Multi-Resolution, Foveated Rendering

1. 서 론

최근 IT 기술이 발달하고 고화질 콘텐츠 전송이 가능해 지면서 고품질 미디어 콘텐츠 영상 스트리밍(Streaming) 서비스에 대한 수요가 증가하고 있다. 이에 UHD 이상의 고화질 고품질 미디어 콘텐츠 영상을 더욱 효율적으로 스트리밍하고, 렌더링(Rendering)하기 위한 다양한 기술들에 대한 연구들이 빠르게 진행되고 있다[1,2]. 그 중 고품질의 미디어 콘텐츠를 효과적으로 스트리밍하기 위한 대표적인 기술로는 Adobe사의 HTTP Dynamic Streaming, Apple사의 HTTP Live Streaming(HLS), Microsoft

의 Smooth Streaming 등이 있으며, 대표적인 표준화 기술로는 MPEG-DASH(Dynamic Adaptive Streaming over HTTP) 기술이 있다[3]. 최근에는 영상을 공간 단위로 분할하여 HEVC로 인코딩을 수행하고, 인코딩된 영상을 다시 공간과 시간단위로 분할한 타일(Tile)을 생성한다. 그 후 사용자의 관심영역(ROI, Region-of-Interest)에 해당하는 타일은 고품질 타일로, 해당하지 않은 영역의 타일은 저품질의 타일로 비트스트림스티칭(Bitstream Stitching)을 수행하여 네트워크 효율 및 스트리밍 영상의 품질을 선택적으로 향상시킬 수 있는 SRD(Spatial Relationship Description)가 제안되었다[4]. 이는, 비디오

※ Corresponding Author : Sung Hyun Yang, Address: (01897) Kwangwoon-ro 20, Nowon-gu, Seoul, Korea, TEL : +82-2-940-5751, FAX : +82-2-909-5715, E-mail : shyang@kw.ac.kr

Receipt date : Mar. 8, 2018, Revision date : May 31, 2018
Approval date : Jun. 11, 2018

[†] Dept. of Electronic Eng., Graduate School, Kwangwoon University (E-mail : khw@kw.ac.kr)

^{††} Dept. of Electronic Eng., Graduate School, Kwangwoon University

※ This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00307, Development of Tiled Streaming Technology for High Quality VR Contents Real-Time Service)

스트리밍 서비스에서 사용자에게 높은 QoE(Quality of Experience)를 보장하기 위한 메커니즘에 대한 연구가 이루어지고 있다[4,5]. 하지만 아직까지는 360 VR 영상과 같은 4K 이상의 고화질 영상을 제한적인 자원을 갖는 HMD기기로 스트리밍하고 렌더링하기에는 어려움이 있다.

이에 본 논문에서는 제한적인 자원을 갖는 HMD 장비에서 4K 이상의 고화질 영상을 렌더링 하기 위해서 시점 추적(Eye-Tracking)의 관심영역에 해당하는 영역은 고화질로, 나머지 영역은 저화질로 전체 영상을 구성하고 비트율을 낮추어 렌더링 할 수 있도록 MPEG-DASH SRD과 Multi-Resolution Shading 기반의 Foveated Rendering[6,7] 기술을 구현하고, 이를 적용한 360 VR플레이어를 구현하였다. 그리고 실험을 통해 실제 360VR HMD(Head Mount Dis-

play) 장비에 영상을 렌더링 하였을 때 성능이 향상 되는 것을 확인하였다.

본 논문의 구성은 2장에서 플레이어를 구현하기 위한 기반 기술을 소개하고, 3장에서 시점 추적 기반 Foveated Rendering을 지원하는 360VR 플레이어를 설계하고 구현하였다. 4장에서 실험을 통해 렌더링 성능을 측정하고, 5장에서 결론을 맺는다.

2. 사전 연구

2.1 Foveated Rendering

VR기기와 같은 제한적인 공간, 시스템 자원에서 몰입감, 현실감을 증대시키기 위해 대표적인 Display기술 중 하나가 Foveated Rendering이다. Foveated Rendering은 관심 영역(ROI, Region of Interest-



Fig. 1. Eye Tracked Foveated Rendering.

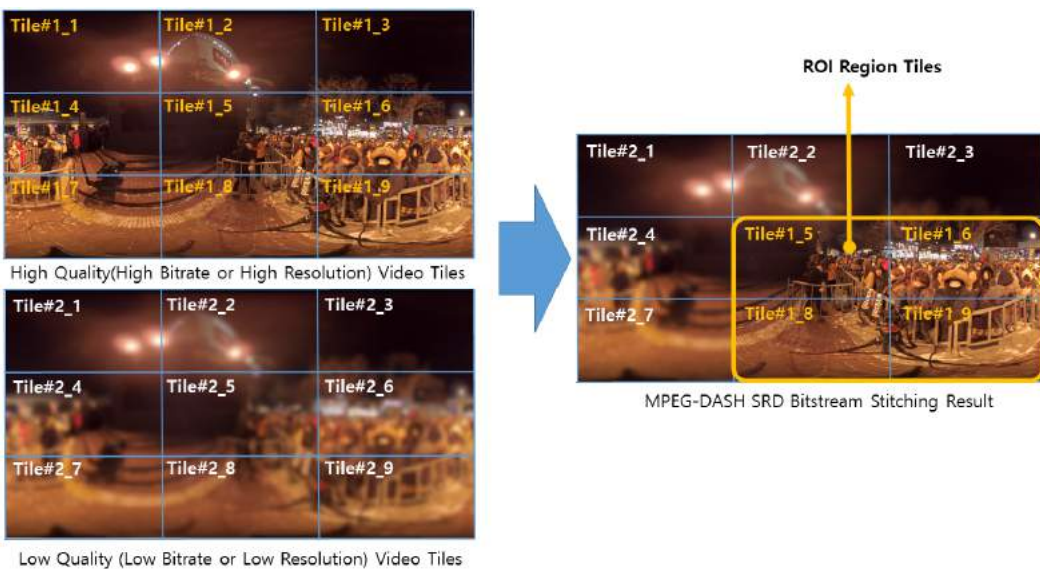


Fig. 2. MPEG-DASH SRD Tiled Video Concept.

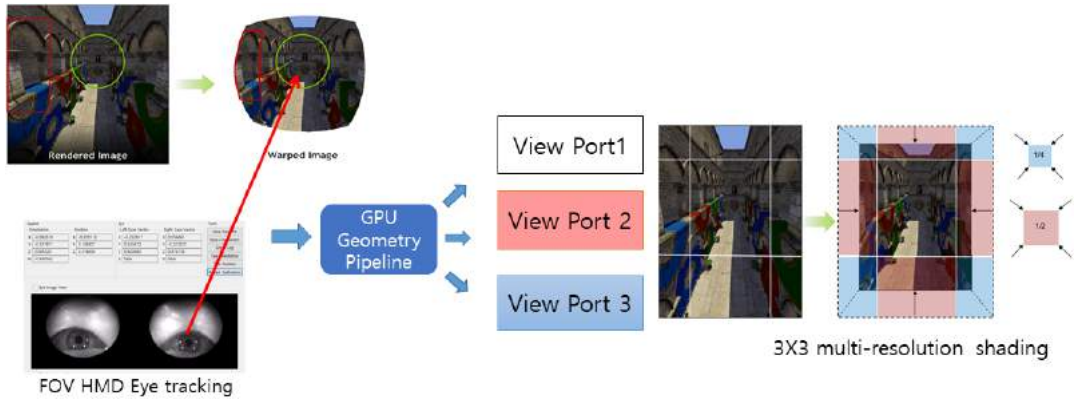


Fig. 3. Eye Tracking based Multi-Resolution Shading Concept[7].

ing)를 중심으로 고화질, 중화질, 저화질로 Layer를 분리하여 위 Fig. 1과 같이 Display하는 기술이다[8].

위와 같이 시점 추적 결과에 따라 FOV를 검출하여 고화질, 중화질, 저화질로 영상을 렌더링 할 경우 HMD 장치의 렌더링 성능을 개선시킬 수 있다.

2.2 MPEG-DASH SRD 분할 영상 기술

MPEG-DASH SRD Tiled Video 기술은 네트워크에 적응적인 MPEG-DASH의 표준 기술로 스트리밍 시 타일을 선택적으로 서비스 할 수 있다는 장점이 있다. 즉, ROI 영역에 해당하는 영역은 고화질 타일들을, 비 ROI 영역은 저화질 타일들을 선택하여 위 Fig. 2과 같이 비트스트림 스티칭을 수행할 수 있다[7]. 이와 같이 타일을 관심영역에 적응적으로 선택하여 스트리밍 서비스를 제공할 경우 더 적은 네트워크 대역폭, 시스템 자원을 사용하여 사용자에게 고화질의 영상을 시청하는 것과 동일한 품질의 스트리밍 서비스를 제공할 수 있다. 물론 적은 수의 타일로 분할할 경우 관심영역에 따라 모두 고화질의 분할 영상 타일을 사용해야 하는 경우가 발생 할 수 있다. 이에 해상도와 관심영역을 고려하여 영상을 타일링(Tiling)하는 것이 중요하다.

2.3 Multi-Resolution Shading

Multi-Resolution Shading은 GPU의 3D 렌더링 파이프라인(Rendering Pipe Line)을 통해 다중 뷰포트(View Port)를 생성하고, 병렬 처리를 통해 다중 뷰포트에 렌더링하는 기술이다. 각각의 뷰포트는 서로 상이한 해상도로 렌더링 할 수 있으며, 각 뷰포트

를 조합하여 최종으로 렌더링 되는 화면을 구성 할 수 있다.

각 뷰포트를 조합 할 때 시점 추적 좌표를 이용하여 해당 추적 좌표에 해당하는 관심 영역은 원본 또는 고화질의 뷰포트를, 검출된 비 관심 영역은 축소된 화질의 뷰포트 화면을 조합하여 위 Fig. 3과 같이 시점 추적을 기반으로 Multi-Resolution Shading 기술을 적용하여 Foveated Rendering을 구현할 수 있다.

3. 본 론

3.1 관심영역(ROI) 검출 기술 설계

시선 추적 기반의 Foveated Rendering을 위해서는 전체 텍스처(Texture) 내 관심영역을 정의해야 한다. 관심영역을 정의하기 위해서는 우선 HMD 장치의 위치 추적(Position Tracking)을 사용하여 고화질 타일을 사용할 시야각 영역(FOV, Field of View)을 아래 Fig. 4와 같이 정의한다.

HMD 장치의 위치 추적 값은 x, y, z 벡터 값으로 수신할 수 있으며, 수신된 벡터 값은 아래 식1과 같이 정규화(normalization) 한다.

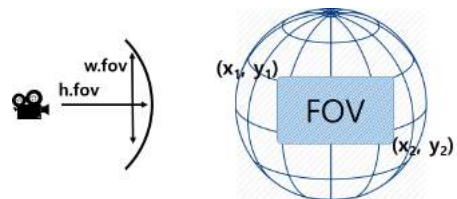


Fig. 4. FOV detecting.

$$res = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

$$x' = \begin{cases} 1.0 - res & \text{if } (res \neq 0 \wedge res \neq 1) \\ x & \text{if } (res = 0 \vee res = 1) \end{cases}$$

$$y' = \begin{cases} 1.0 - res & \text{if } (res \neq 0 \wedge res \neq 1) \\ y & \text{if } (res = 0 \text{ or } res = 1) \end{cases}$$

$$z' = \begin{cases} 1.0 - res & \text{if } (res \neq 0 \wedge res \neq 1) \\ z & \text{if } (res = 0 \text{ or } res = 1) \end{cases}$$

정규화된 HMD 장치의 위치 추적 좌표를 기반으로 위 Fig. 4와 같이 시야각 영역 좌표를 도출 할 수 있다. 좌표 x_1, x_2 는 아래 식 2와 같이, 좌표 y_1, y_2 는 아래 식 3과 같이 도출 하였다.

$$\theta = \text{atan}(z', x') - \frac{\pi}{2} \quad (2)$$

$$h.fov = \frac{\text{camera.fov} \times \text{camera.width}}{\text{camera.height}} \div 2$$

$$x_1 = 1 - (\theta - h.fov + \Pi) \div \frac{\pi}{2} \times \text{width}$$

$$x_2 = 1 - (\theta + h.fov + \Pi) \div \frac{\pi}{2} \times \text{width}$$

$$\theta = \text{atan}(z', y') \quad (3)$$

$$w.fov = \text{camera.fov} \div 2$$

$$y_1 = (\theta - w.fov) \div \pi \times \text{height}$$

$$y_2 = (\theta + w.fov) \div \pi \times \text{height}$$

위와 같이 HMD 장치 위치 추적 기반으로 시야각 영역 내에서 시점 추적 좌표(x, y, z)를 사용하여 시야각 영역 내 관심 영역을 정의한다. 시점 추적 좌표(x, y, z=1)를 시야각 영역 내 위치 좌표로 계산하는 식은 아래 식 4와 같다.

$$\text{Eye}X = \frac{x' \times \text{width}}{2} + \frac{\text{width}}{2} \quad (4)$$

$$\text{Eye}Y = \frac{y' \times \text{height}}{2} + \frac{\text{height}}{2}$$

렌더링 되는 텍스처는 아래 Fig. 5와 같이 뷰포트 영역을 9개의 셀로 나누어 고품질(1.0), 저품질(0.7), 저품질(0.2) 3개의 레벨로 표시하도록 하였다. 시점 추적 좌표가 위치하는 좌표의 셀은 고품질(1.0), 주변의 셀은 중품질(0.7), 그 밖의 셀은 저품질(0.2)로 구성하여 Multi Resolution Texture를 생성하였다.

3.2 360 VR Foveated Tiled Video Rendering Player 설계

본 논문에서 사용한 Player는 기본적으로 MPEG-DASH 표준을 지원하며 그 구조는 아래 Fig. 6과 같다. Player에서는 HMD를 통해 눈 시점 추적 좌표를 수신하여 ROI를 검출하고, 이를 기반 FOV를 정의한다. FOV에 해당하는 영역은 고품질 Tiles로, 해당하지 않는 영역은 저화질 Tiles로 Adaptation Set을 구성한다. 그리고 해당 Adaptation Set의 Segment 파일을 MPEG-DASH Media Streaming Server로 요청하여 수신한다. 기존에 수신하였던 MPD 파일이 없을 경우 ROI 검출 이전에 MPD 파일을 먼저 수신한다.

Segment 파일들을 모두 수신하고 나면 Bitstream Stitching을 수행하고, Decoding을 수행한다. De-



Fig. 5. Multi-Resolution Cell Composition.

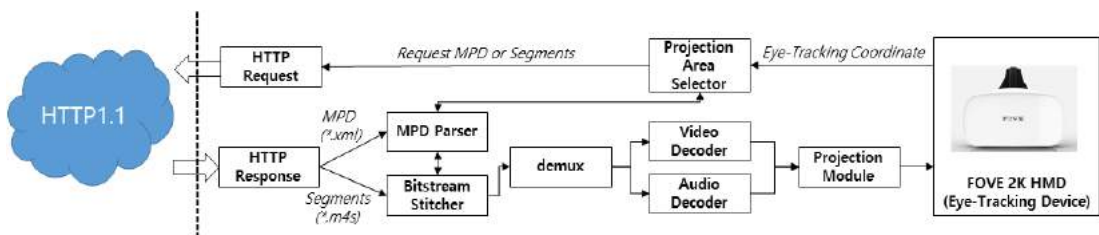


Fig. 6. 360 VR Foveated Rendering Player Structure.

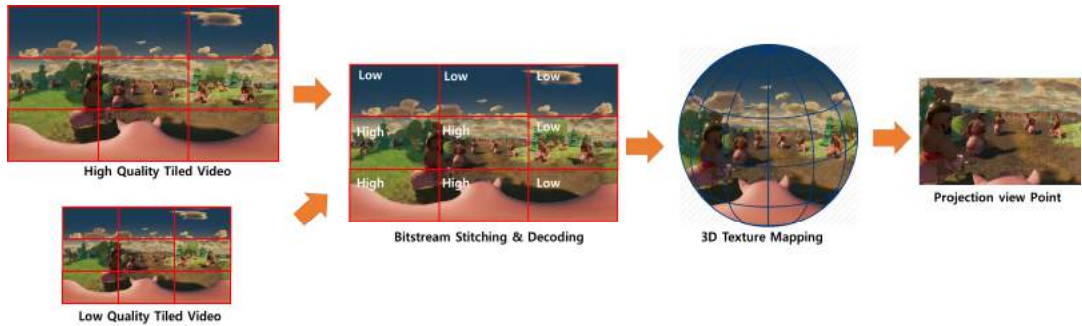


Fig. 7. 360 VR Video Projection Modeling.

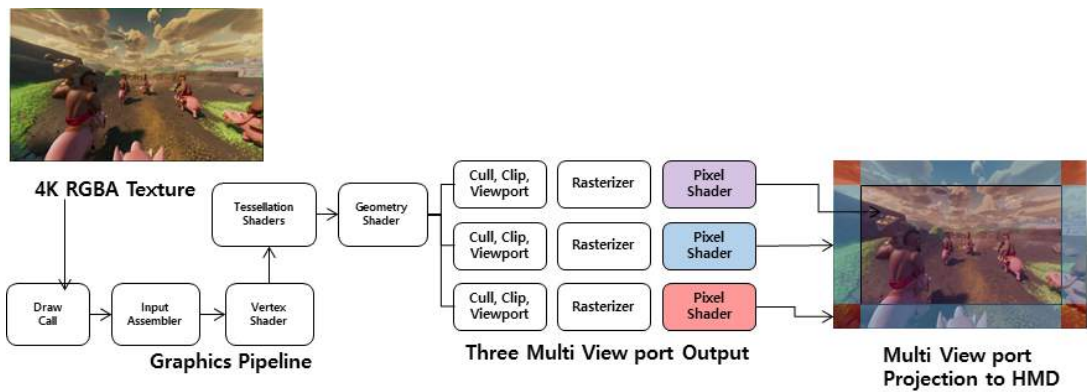


Fig. 8. Multi-Resolution Rendering.

coding을 수행한 후 HMD에 Rendering하기 위해 Decoding된 영상 이미지를 3D Texture에 Mapping한 후 HMD에 Rendering을 위한 Projection View Point를 선택한다.

선택된 Projection View Point 영상 이미지 Single Projection Texture를 마지막으로 Multi-Resolution

Rendering을 사용하여 위 Fig. 8과 같이 Multi View Port Projection Texture를 생성하여 Display장치로 Rendering 한다.

3.3 360 VR Foveated Tiled Video Rendering Player 구현

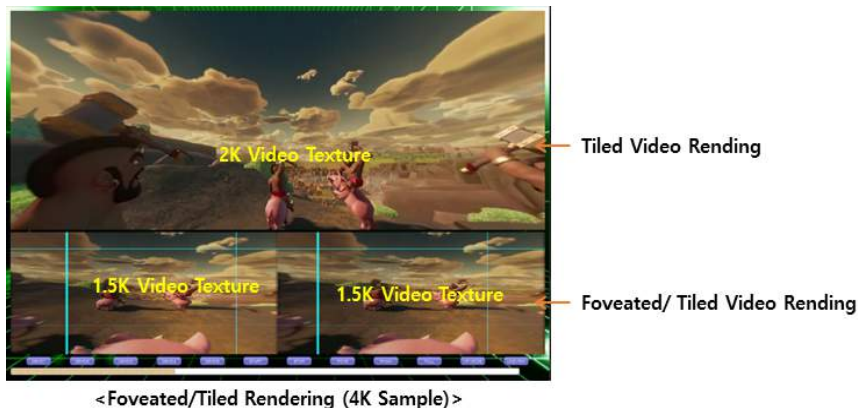


Fig. 9. 360VR Tiled Video Foveated Rendering.

Player는 아래 Fig. 9와 같이 Tiled Video Rendering 화면, HMD 좌/우안에 렌더링되는 Foveated Tiled Video Rendering 화면으로 UI를 구성하였다. Tiled Video Rendering은 2K Texture으로 Downsizing하여 렌더링 하였으며, HMD에 렌더링 되는 영상은 HMD 해상도에 맞추어 1.5K정도 크기의 Texture로 Downsizing하여 렌더링 하여 기능 구현을 검증하였다.

4. 실험 결과

4.1 Foveated Rendering 성능 측정 방법 및 환경

MPEG-DASH SRD를 기반으로 관심영역은 고품질 타일로, 비관심영역은 저화질 타일로 전체 화면을 구성하여 렌더링 하는 기존 Tiled Rendering 방식과, 본 논문에서 제안한 Multi Resolution기술까지 적용한 Foveated Rendering 방식을 사용하여 HMD에 렌더링 하여 성능을 비교 측정하였다.

성능 비교 측정을 위한 스트리밍 서버 및 VR 플레이어 클라이언트 PC는 아래 표와 같이 구성하였으며, HMD 장비는 Eye-Tracking을 지원하는 FOVE를 사용하였다.

성능 측정에 사용한 시료 영상 종류 아래 표 2와

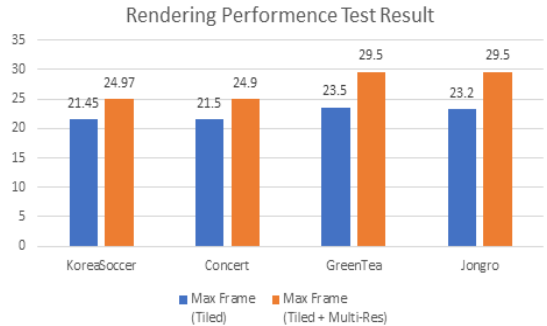


Fig. 10. Foveated Rendering Performance.

같이 8K급 4개의 영상을 사용하였다. 성능 측정 기준은 각 영상을 10회씩 플레이 했을 때, 최대로 측정되는 렌더링 속도를 측정하였다. 영상을 플레이하기 위한 디코더는 OpenHEVC 2.0[9]를 사용하였다.





4.2 Foveated Rendering 성능 측정 결과

성능 측정 결과 위 Fig. 10과 같이 기존의 Tiled Streaming 기술에 Foveated Rendering 기술을 적용하였을 때 Texture Rendering FPS(Frame per Second)성능이 평균 10~15% 정도 향상되는 것을 확인하였다. 8K 영상에는 Eye-Tracking FOV에 해당하는 Tiles Set 구성에 따라 최대 18%까지도 성능

Table 1. Specification of experiment environment

Item	Streaming Server	360 VR Player Client PC
CPU	Intel Xeon E5-2687W v4 @3GHz (12Core, 24Thread)	Intel Core-i7 6900K @3.2GHz (8Core, 16Thread)
Memory	DDR4 32GB	DDR4 16GB
Ethernet	10Gbps	1Gbps
GPU	-	NVIDIA Geforce GTX 1080Ti

Table 2. Sample Video Specification

Item	KoreaSoccer	Concert	GreenTea	Jongro
resolution	8192 × 4096	7680 × 3840	7680 × 3840	7680 × 3840
bitrate	72324kbps	78415kbps	52356kbps	79342kbps
frame/sec	25.00	25.00	29.97	29.97
length	56 sec	1min 12sec	33 sec	1min 18sec
Tiles	6 × 6	6 × 6	6 × 6	6 × 6
Sample Image				

이 향상되는 것을 확인 할 수 있었다. 측정 결과는 아래 Table 1과 같다.

5. 결 론

고화질 360 VR Tiled Video를 MPEG-DASH SRD 기반 Foveated Rendering을 할 수 있는 Video Player 기술을 구현하였다. 이를 통해 제한적인 자원을 갖는 HMD에서 고화질 영상 시청 시 몰입감, 현실감을 높이고, Rendering 부담을 낮출 수 있음을 확인하였다. Video Player가 동작하는 호스트PC의 연산량이 증가하였지만, HMD와 연동되는 최근 PC의 자원으로 충분히 처리할 수 있는 부분이므로 크게 고려해야 하는 사항은 아닌 것 같다. 다만 VR에서 지속적으로 언급되는 이질감, 멀미감소 등을 위해서는 트래킹(Tracking)의 응답 속도, 더 넓은 시야각, 높은 해상도를 지원하는 등 HMD 장비의 개선이 반드시 필요한 실정이다.

앞으로 호스트PC의 연산 부담을 줄이고, SW단에서 성능을 개선할 수 있도록 지속적인 연구를 수행해 나갈 것 이다.

REFERENCE

[1] J.W. Son, Y.I. Ryu, H.J Roh, and E.S. Ryu, "SHVC-Based ROI Tiled Parallel Processing for Mobile Virtual Reality," *Proceeding of the Winter Conference of Korea Institute of Information Scientists and Engineers*, pp. 1715-1717, 2016.

[2] Y.I. Rye, H.J. Roh, and E.S. Ryu, "Tile Partitioning-Based HEVC Parallel Decoding Optimization for Asymmetric Multicore Processor," *Journal of Korea Institute of Information Scientists and Engineers*, Vol. 43, No. 9, pp. 1060-1065, 2016.

[3] J.Y. Lee, "MPEG-DASH Standardization and Commercialization Trend," *Broadcasting and Media Magazine*, Vol. 20, No. 1, pp. 68-74, 2015.

[4] O.A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S.Y. Lim, "MPEG-DASH SRD - Spatial Relationship Descrip-

tion," *Proceedings of the 7th International Conference on Multimedia Systems*, No. 5, pp.5:1-5:8, 2016.

[5] I. Kim, S. Hong, S. Jung, K. Lim, "An Intelligent Video Streaming Mechanism based on a Deep Q-Network for QoE Enhancement," *Journal of Korea Multimedia Society*, Vol. 21, No. 2, pp. 188-198, 2018.

[6] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, "Foveated 3D Graphics," *ACM Transactions on Graphics*, Vol. 31, No. 6, pp. 1-22, 2012.

[7] VRWorks-Multi-Res Shading, <https://developer.nvidia.com/vrworks/graphics/multi-resshading> (accessed Oct., 09, 2017).

[8] A. Patney, D. luebke, and J.H. Kim, "Latency Requirements for Foveated Rendering in Virtual Reality," *ACM Transactions on Applied Perception*, Vol. 14, No. 4, pp. 25:1-25:13, 2017.

[9] OpenHEVC, <http://openhevc.github.io/openHEVC/> (accessed May., 30, 2017)



김 현 욱

2009년 광운대학교 컴퓨터공학과 (공학사)

2013년~2016년 (주)케이사인 정보 보안연구소 선임연구원

2009년~현재 광운대학교 대학원 전자공학과 석박사통합 과정

관심분야: 스마트 홈, 임베디드 시스템, 콘텐츠 응용 기술



양 성 현

1993년 광운대학교 대학원 전기 공학(자동제어) 공학박사

1991년~현재 광운대학교 전자공학과 교수

2005년~현재 광운대학교 Smart H&B Technology Center 센터장

관심분야: 스마트 홈, 디지털 로직, 임베디드 시스템, IoT, 상황인식(행동인식알고리즘) 시스템