Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 203-212
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

View Direction Adaptive 360 Degree Video Streaming System Based on Projected Area

San Kim, Jihyeok Yun, Bokyun Jo, Ji Ho Kim, Ho Gyun Chae, Doug Young Suh

College of Electronics and Information, Kyung-Hee University, Youn-in, Republic of Korea
Email: kimsan0622@gmail.com

How to cite this paper: Kim, S., Yun, J., Jo, B., Kim, J.H., Chae, H.G. and Suh, D.Y. (2018) View Direction Adaptive 360 Degree Video Streaming System Based on Projected Area. *Journal of Computer and Communications*, **6**, 203-212.
https://doi.org/10.4236/jcc.2018.61020

Received: October 30, 2017
Accepted: December 26, 2017
Published: December 29, 2017

Abstract

360 video streaming services over the network are becoming popular. In particular, it is easy to experience 360 video through the already popular smartphone. However, due to the nature of 360 video, it is difficult to provide stable streaming service in general network environment because the size of data to send is larger than that of conventional video. Also, the real user's viewing area is very small compared to the sending amount. In this paper, we propose a system that can provide high quality 360 video streaming services to the users more efficiently in the cloud. In particular, we propose a streaming system focused on using a head mount display (HMD).

Keywords

360 Panoramic Video, Cloud, Split Image, Real-Time Streaming, MPU

1. Introduction

Recently, 360 degree video streaming service has become a very universal service. Many media streaming platforms such as You Tube and Facebook already support 360 degree video [1]. Also, there are many streaming service for head-mount-display (HMD) like Oculus Rift, Gear VR. Equirectangular format is often used for 360 degree video. In fact, most current 360 degree videos are equirectangular format in internet.

Equirectangular format is often used because it has several advantages such as compression efficiency and simplicity in implementation. However, the fatal disadvantage of equirectangular format is that the pixel representation of the polar part is very inefficient. Because a number of pixels used to represent the polar part are equal to a number of pixels used to represent the equator part even though the area of the polar part is smaller than the equator part. For this

reason, it is necessary to send a very high quality equirectangular video in order to improve the image quality of the equator part.

However, only a small part of the full sphere is rendered to a user. Therefore, it is inefficient to send a portion of the image not seen by a user at a high image quality. Therefore, if there is a difference between these areas, it is possible to reduce the size of data to be sent while enhancing the quality of the area viewed by the user. In addition, in the case of HMD, it will be able to differentiate the rendering quality between the focused and non-focused areas because the focused area of human eyes is narrower than the full viewing angle. In this paper, I will divide the full sphere as three-part on focal FOV, FOV of a display device, and invisible area and differentiate image quality of each area.

2. System Framework

To provide network efficiency and improved experience for users, this paper proposes a system that sends different image quality to different areas in the cloud.

Figure 1 shows framework of the proposed system. The scenario of the proposed system is as follows. The server sends a high quality image (4 K) to the cloud. The cloud transcodes the video into a lower quality video to produce multiple quality videos. In spatial, video of each quality is divided by vertical (n), horizontal (2n). In temporal, it is divided by 1 second. The client tells the cloud where user is currently seeing, and the FOV of the device. The cloud uses the information received from the client to determine the area to be sent with high image quality, medium image quality, and low image quality through the decision process, and sends video blocks with additional information about the area. The client plays the corresponding 360 degree video using video block and additional information.

3. Resolution Transcoding and Split Video

The process of resolution transcoding is simple. First of all, a cloud can make low-resolution videos with high-resolution (4 K) video which is received from a server. The interpolation method used in transcoding to generate low-resolution

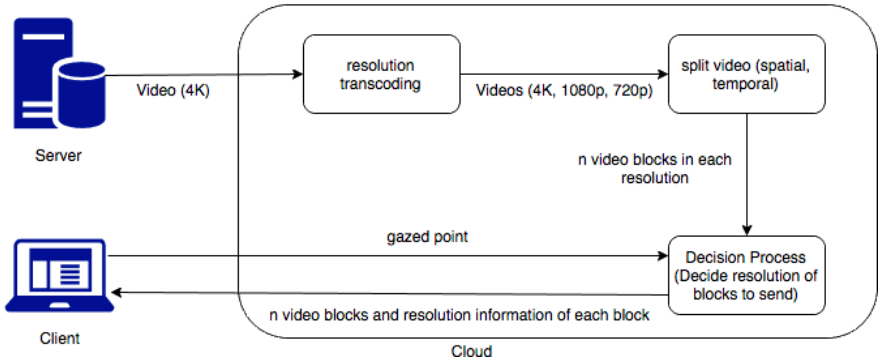

Figure 1. Framework of proposed system.

video is not covered in this paper. Then, the cloud divides all the videos into space and time.

[2] **Figure 2** Shows that the image is cut in space when the horizontal divisor is 4 and the vertical divisor is 4, and the lower part shows the image cut in time. Since truncating the image in space tends to increase the bps to be sent, determining the size of the truncation can also have a significant impact on overall performance. Also, in terms of time, video is cut by Media Process Unit (MPU) unit. The smaller MPU unit is, the more quickly the system can adapt to the sudden change of direction of HMD.

4. Decision Process

This Decision Process is the core of the proposed system. In this process, using information of direction which is user look at, the FOV of the device, and the network bandwidth, and decide which of the divided blocks is to be sent in high quality and low quality. Using the point received from the user, the system calculates the area that user viewed through the device and also obtains the focal FOV of the user. Through this, if a block is included area of focal FOV, the system sends a high-quality video of that block. The system sends an intermediate quality video of that video if the block is a part of device FOV. Otherwise, the system sends low-quality video [3] [4].

4.1. Translation and Rotation and Projection

To ensure that the video block is within the FOV, we need to translate and rotate the camera so that it looks at the -z-direction at the origin. When rendering the

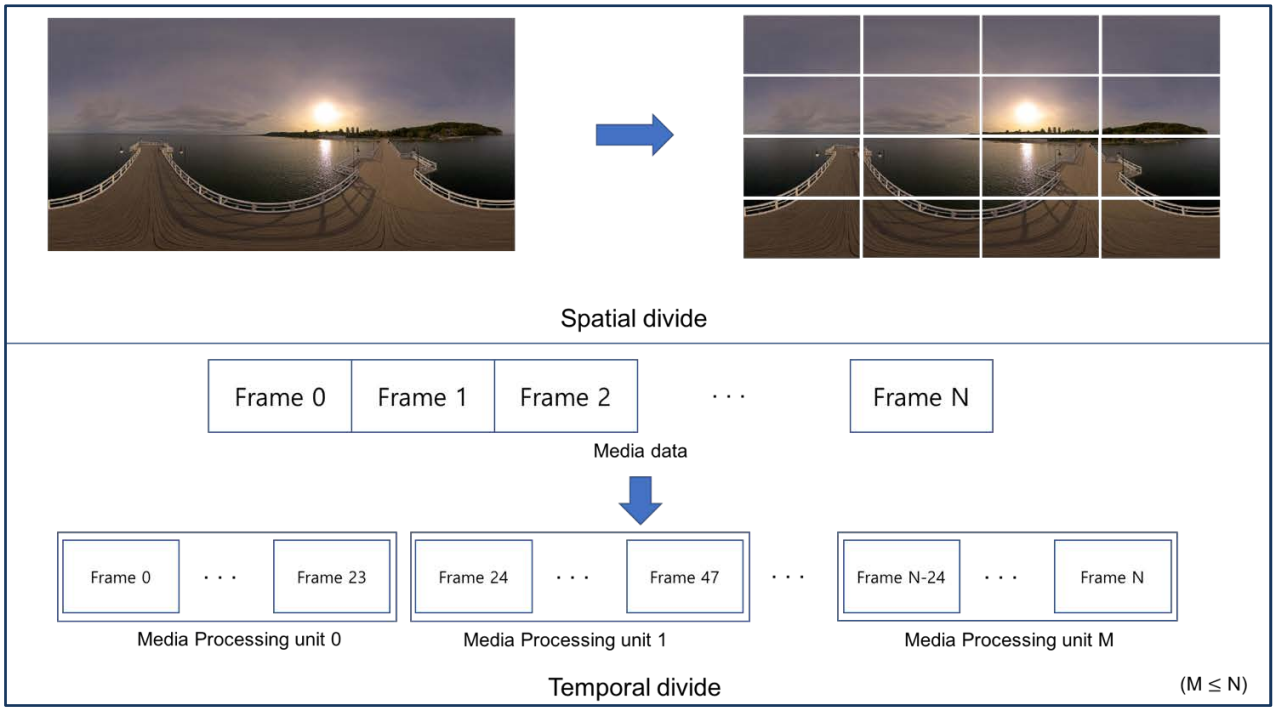

Figure 2. Spatial and temporal divide methods.

360 VR video on the client side, the projection must be done by translating and rotating according to left eye and right eye. However, when judging which part to send in high quality, we can assume that there is only one camera in Origin and decide it using FOV and direction information.

Let $\boldsymbol{V}_{dir} = (yaw, pitch, roll)$ be the direction vector the client is looking at. After receiving the yaw, pitch and roll values from the client, we can create a Rotation and Projection matrix using this information and FOV information [3]. First, let's say that the rotation matrix created by using the yaw, pitch, and roll values received from the client is $\boldsymbol{R}_{rpy}$. This rotation matrix is about the direction the camera is looking at, and what we are actually doing is rotating the video block position on the 3D to match the direction like that the camera is looking at -z-direction. So we use the inverse of $\boldsymbol{R}_{rpy}$ ($\boldsymbol{R}_{rpy}^{-1}$). The inverse matrix of the rotation matrix is the same as transpose, and since there is no movement, the final rotation matrix we obtain is

$$\boldsymbol{M}_{rot} = \begin{bmatrix} & & & 0 \\ & \boldsymbol{R}_{rpy}^{T} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Figure 3 shows the relationship between rotation matrix and invert matrix of rotation matrix. Therefore, multiplying the corresponding rotation matrix and multiplying it by the projection matrix can tell where the video block is located if it is projected on the FOV plane. The projection matrix can be obtained simply by using horizontal FOV and vertical FOV as follows.

As can be seen **Figure 4** and **Figure 5**, the projection matrix is obtained by the following equation.

Near is the minimum distance to be projected, and far is the maximum distance to project. In this case, we use only one sphere, so we can set the radius of

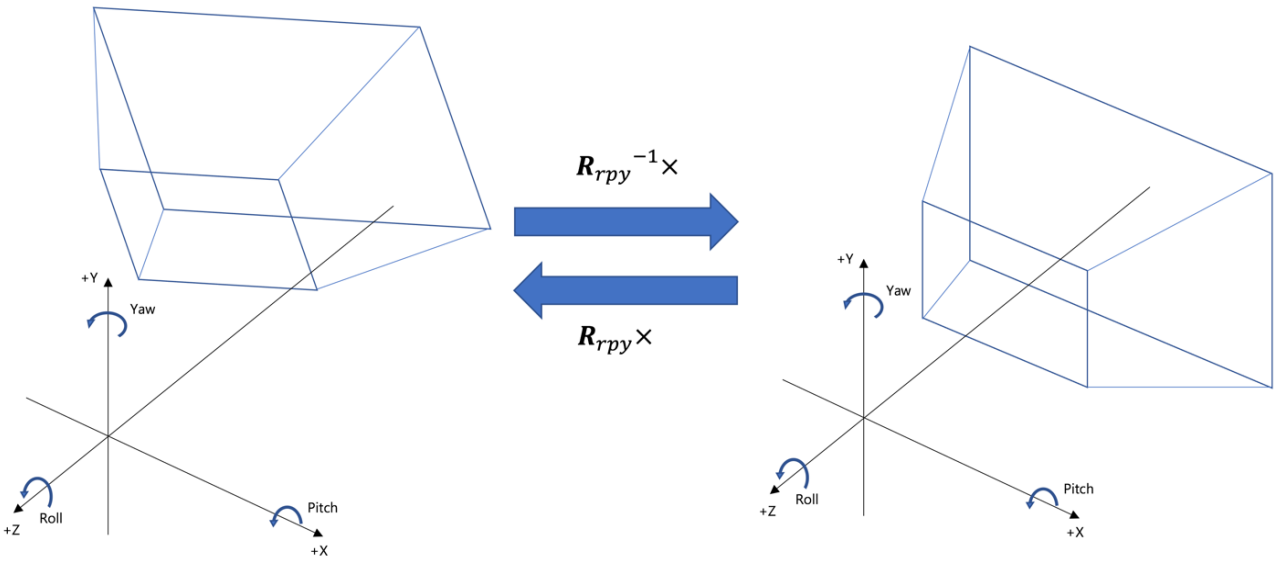

Figure 3. Relationship with rotation matrix and invert matrix of rotation matrix.

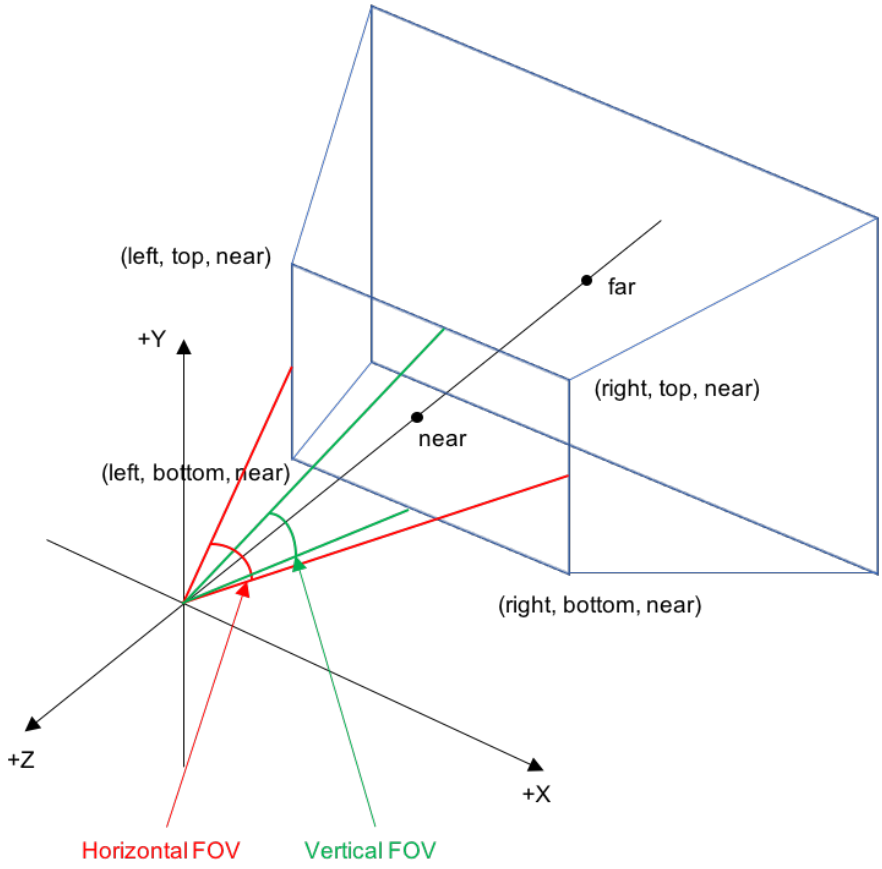

Figure 4. Meaning of horizontal FOV and vertical FOV in projection matrix.

$$\boldsymbol{M}_{proj} = \begin{pmatrix} \frac{2\times near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2\times near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & \frac{-(far+near)}{far-near} & \frac{-2\times far\times near}{far-near} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Figure 5. Projection matrix.

the sphere to be between near and far. Right, left, top, and bottom can be obtained by using horizontal FOV, vertical FOV, and near value.

Once the rotation matrix and the projection matrix have been determined, they can be used to determine if the region falls within the FOV. The following three methods can be used for discrimination.

In the left figure of **Figure 6**, when one of the vertices of the area is inside the FOV, it is judged that the corresponding area is inside the FOV. In the middle figure, when the center point of the four vertices is inside the FOV, It is judged that the corresponding area has entered the FOV. In right figure, when all the vertices are inside the FOV, it is judged that the corresponding area has entered the FOV. In this case, the center point will be used for the determination. Let the

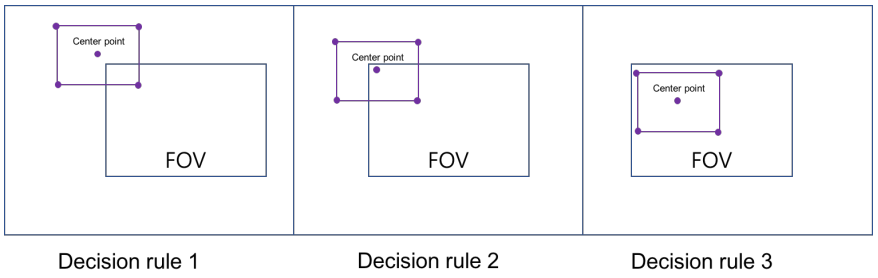

Figure 6. Decision rules.

center point of the i-th face be $\boldsymbol{V}_i = [x\ y\ z\ 1]^T$ and let the inner and outer determination variable of the face be p_i. Also, if p_i is 1 for internal and 0 for external, then the process of discrimination is as follows.

$$\tilde{\boldsymbol{V}}_i = \boldsymbol{M}_{proj} \times \boldsymbol{M}_{rot} \times \boldsymbol{V}_i \tag{2}$$

$$\begin{cases} p_i = 1.\ if\ \left|\dfrac{\tilde{\boldsymbol{V}}_i(1)}{\tilde{\boldsymbol{V}}_i(4)}\right| \le 1, \left|\dfrac{\tilde{\boldsymbol{V}}_i(2)}{\tilde{\boldsymbol{V}}_i(4)}\right| \le 1, \left|\dfrac{\tilde{\boldsymbol{V}}_i(3)}{\tilde{\boldsymbol{V}}_i(4)}\right| \le 1, \tilde{\boldsymbol{V}}_i(4) > 0 \\ p_i = 0.\ otherwise \end{cases} \tag{3}$$

Through the above process, it is possible to judge whether or not the corresponding face falls within the FOV.

4.2. Projected Area

The resolution of the block is determined by using the projected area at the end of the decision process. This is to keep the picture quality rendered on the screen constant. It is meaningless to input a higher resolution because the support resolution of a general HMD device is FHD. Therefore, target resolution is determined for each FOV to ensure quality. The projected area of the face is S_i, the area of the FOV is S_{FOV}, the target resolution is P_{target}, the number of blocks divided by space is n, and the resolutions that can be assigned to the block are P(ex. $P = [P_{720p}\ P_{1080p}\ P_{2k}\ P_{4k}]$, the resolution P_i of the corresponding area is as follows.

$$P_i = \arg\min_{P \in P}\left(P_{target} - \frac{S_{FOV}}{S_i \times n} P\right) \tag{4}$$

4.3. Network Constraint

Finally, there is one more condition in the resolution discrimination formula above. The bitrate required to send the above resolution should be less than the network bandwidth [5]. Therefore, when the network bandwidth is B_N, and the bitrate of the i region with the P_i resolution is B_i, the resolution decision equation has the following constraint.

$$\sum_{i=0}^{n} B_i < B_N \tag{5}$$

Finally, the final resolution determination equation is as follows.

$$P_i = \arg\min_{P \in \boldsymbol{P}}\left(P_{target} - \frac{S_{FOV}}{S_i \times n} P\right) \tag{6}$$

$$s.t.\ \sum_{i=0}^{n} B_i < B_N$$

5. Experiment

In the experiment, the resolution was limited to 4 k, 2 k, 1080 p, 720 p, and the support resolution of the HMD was 1080 p, and the video was cut into 200 pieces and 800 pieces. Because vertical divisor is 10 and the horizontal divisor is 20 when it cut into 200 pieces. And vertical divisor is 20 and the horizontal divisor is 40 when it cut into 800 pieces. In addition, we decided not to measure the resolution adaptation delay according to the viewing direction and not to consider the network delay, so we set the MPU in units of 1 second in time. In addition, the target resolution of focal FOV is set to 1080 p. And we set the vertical focal FOV and horizontal focal FOV as 55 degrees and 60 degrees which is generally because the focal area of the human eye is 55 degrees vertical FOV and 60 degrees horizontal FOV. And the target resolution of device FOV is set to 720 p. And we set the vertical device FOV and horizontal device FOV as 90 degrees and 100 degrees. In addition, due to the nature of the HMD, the image is enlarged as it approaches the edge of the FOV in order to correct distortion of the HMD lens. This is because the edge of the FOV is narrowed when passing through the lens of the HMD. Also, the center point is used to determine if the faces entered the FOV, as shown in **Figure 7**.

The left figure shows how to identify the FOV area when looking upward at 30 degrees, and the right figure shows how to identify the FOV area when looking at the front. The target resolution of yellow and blue areas is 1080 p and the target resolution of red areas is 720 p. The experimental results are as follows. Let's start by cutting it into 200 pieces first.

As shown in the above **Figure 8**, when the video resolution of each block is determined through the decision process, it is obvious that the projected resolution is better than the case of transmitting one QHD video but the projected resolution is lower than the case of transmitting one UHD video. However, in terms of projected resolution which is user actually watch there is almost no

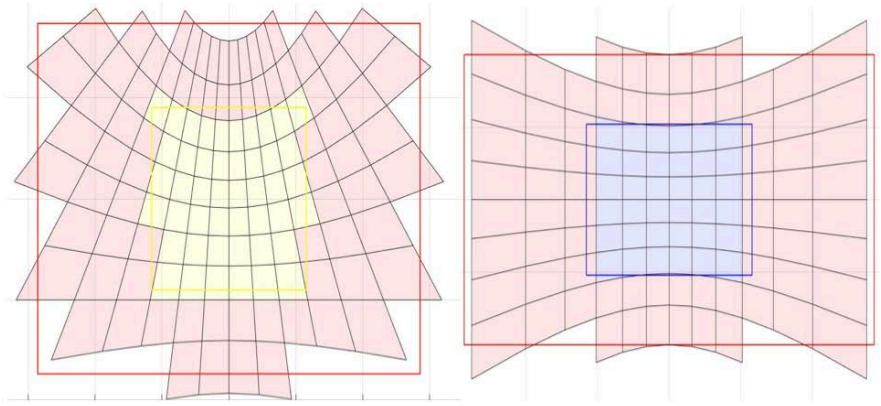

Figure 7. Projected area of each rotation matrix.

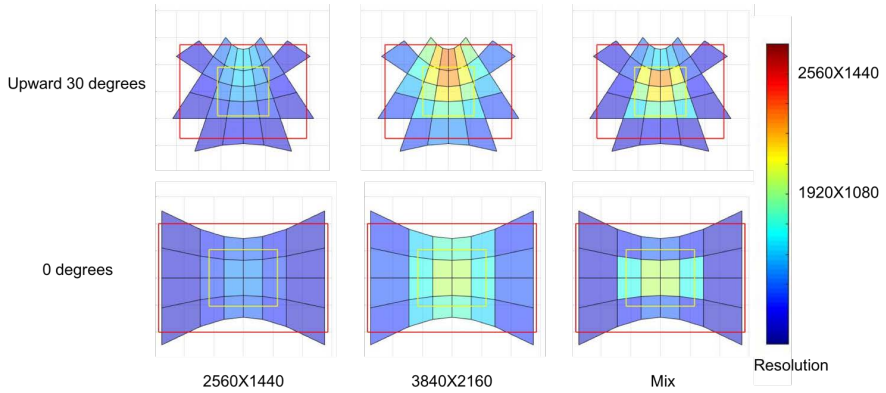

Figure 8. Projected resolution when full sphere cutting into 200 pieces.

difference between the case where the UHD is sent to the actual user and the case where the video is sent using the proposed system. Because the area between the yellow rectangle and the red rectangle is where the user is out of focus. Let's look at the case of cutting to 800 pieces.

As shown in the above **Figure 9**, the result is not much different from the case of cutting into 200 pieces.

Here is a comparison of bit rates.

In the **Figure 10**, it can be seen that bps is smaller than when 2 k is sent regardless of the latitude when 200 pieces and 800 pieces. That is, it shows the higher projected resolution to the user than 2 k, but the bps is lower. In case of 90 degrees, if set the resolution of blocks which fall within yellow FOV as UHD and the resolution of blocks that are included in red FOV as QHD without considering the projected area, bps is higher than 2 k.

In the above **Table 1**, 1 is the number of UHD and QHD blocks when the yellow FOV region is determined as UHD and the red FOV region is determined as QHD. 2 is the number of UHD and QHD blocks when the projected area is taken into account. In this case, if the projected resolution is higher than 1 k, the system is set the resolution of blocks even it is 1080 p or 2 k. That is, it is decision process of the proposed system. If 1 method is used, bps is 11.86 Mbps when cut into 200 pieces, 13.21 Mbps when cutting into 800 pieces. And these are slightly higher than 11.49 Mbps which is the bps when 2 k is used.

6. Conclusion

Through the above experiments, we found that the proposed method improves the video quality and the bps is smaller than that when sending 2 k images. Also, it can be seen that there is no big difference in quality compared to the case of 4 k video transmission, but bps is much smaller. Also, when the video quality of the sending blocks is determined considering only the blocks included in the FOV, there is little difference between the proposed method and the transmitted bps at low latitude, but it is confirmed that the proposed method has large profit at bps in the case of high latitude. In general, since the supported resolution of the HMD is 1 k, the resolution does not need to exceed 2 k.

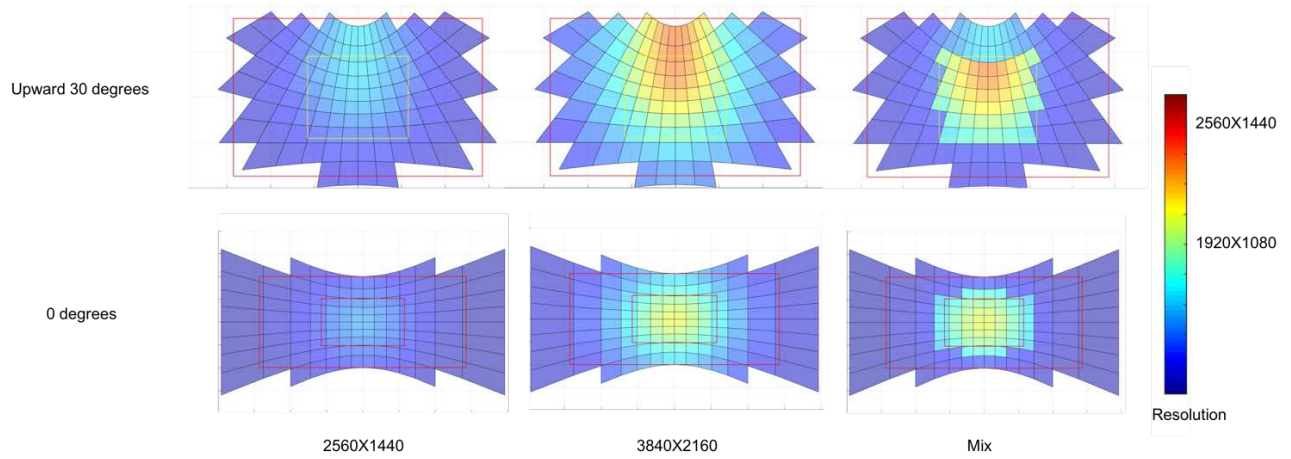

Figure 9. Projected resolution when full sphere cutting into 800 pieces.

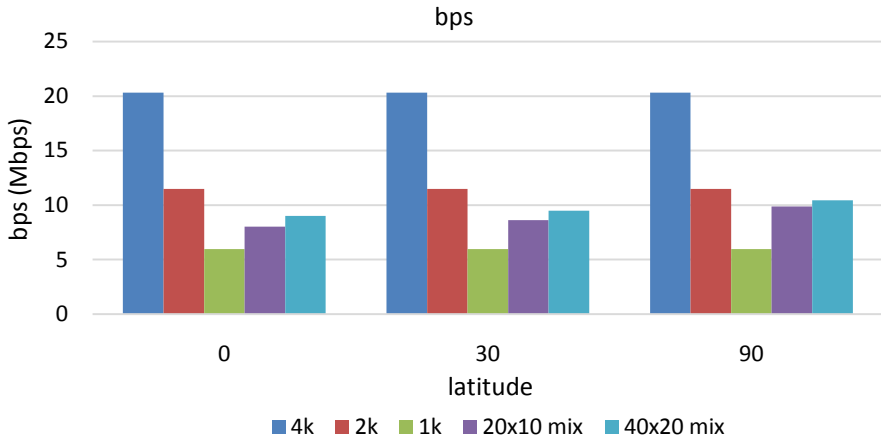

Figure 10. Bps comparison of each transmitting methods.

Table 1. Result of each decition methods.

method	# of pieces	# of UHD pieces	# of QHD pieces
1	200 pieces	40	16
	800 pieces	136	92
2	200 pieces	10	40
	800 pieces	16	164

Acknowledgements

This research was supported by "Software Convergence Cluster Program", through the Ministry of Science, ICT and Future Planning/Gyeonggi Province (20171218).

References

[1] Rhee, T., Petikam, L. and Allen, B. (2017) MR360: Mixed Reality Rendering for 360 Panoramic. *IEEE Transactions on Visualization and Computer Graphics*, **23**, 1379-1388. https://doi.org/10.1109/TVCG.2017.2657178

[2] Nasrabadi, A.T., Mahzari, A. and Beshay, J.D. (2017) Adaptive 360-Degree Video Streaming Using Layered Video Coding. 2017 *IEEE Virtual Reality*, 18-22 March 2017, 2375-5334.

[3] Hosseini, M. (2017) View-Aware Tile-Based Adaptations in 360 Virtual Reality Video Streaming. 2017 *IEEE Virtual Reality*, 18-22 March 2017. https://doi.org/10.1109/VR.2017.7892357

[4] Sreedhar, K.K., Aminlou, A., Hannuksela, M.M. and Gabbouj, M. (2016) Viewport-Adaptive Encoding and Streaming of 360-Degree Video for Virtual Reality Applications. 2016 *IEEE International Symposium on Multimedia*, 11-13 December 2016.

[5] Duanmu, F., Kurdoglu, E., Liu, Y. and Wang, Y. (2017) View Direction and Bandwidth Adaptive 360 degree Video Streaming Using a Two-Tier System. 2017 *IEEE International Symposium on Circuits and Systems*, 28-31 May 2017.