On the possibility to achieve 6-DoF for 360 video using divergent multi-view content

Bappaditya Ray[1,2], Joel Jung[1], and Mohamed-Chaker Larabi[2]
[1]Orange Labs, [2]CNRS, Univ. Poitiers, XLIM, UMR 7252, Poitiers, France
{bappaditya.ray, joelb.jung}@orange.com, {bappaditya.ray, chaker.larabi}@univ-poitiers.fr

***Abstract*—With the rapid emergence of various 360 video capturing devices and head-mounted displays, providing immersive experience using 360 videos is becoming a topic of paramount interest. The current techniques face motion sickness issues, resulting in low quality of experience. One of the reason is that they do not take advantage of the parallax between the divergent views, which may be helpful to provide the correct view according to the user's head motion. In this paper, we propose to get rid of the classical ERP representation, and to synthesize arbitrary views using different divergent views together with their corresponding depths. Thus, we can exploit the parallax between the divergent views. In this context, we assess the feasibility of the depth estimation and the view synthesis using state-of-the-art techniques. Simulation results confirmed the feasibility of such a proposal, in addition to possibility to achieve sufficient visual quality for a head motion up to 0.1m from the rig, when using generated depth map for view synthesis.**

I. Introduction

The popularity of the 360 video format and virtual reality technologies is steadily rising in the recent days, with the aim of providing users the sense of immersion. Many capturing devices are already on the market. They consist of multiple cameras with divergent optical axes to capture the views in different orientations, from a central position. 360 video uses spherical representation, obtained by stitching the different divergent views, with a mapping on a 2D plane, thanks to projection schemes such as the equirectangular (ERP) or the cubemap (CMP), for instance. The projection only considers two degrees of freedom (DoF): yaw and pitch. Due to the low quality of experience, it can be seen as a very preliminary step to achieve immersion. The main issue with the approach is the stitching stage, which restricts the content to a single view and prevents from handling the parallax between divergent views [1]. As a consequence, usage remains limited due to motion sickness issues. Nevertheless, this approach fulfills short-term needs of the industry and also sets some initial basis for 360VR.

Based on the above issues, MPEG-I launches Phase 1b activities, known as 3DoF+, aiming to provide parallax information allowing the user to change the viewpoint in a limited space [2]. Dore et al. propose to address motion parallax at the system level by extending OMAF V1 with metadata, containing depth information, alpha key and patch of residual video [3]. In [4], an omnidirectional video plus depth format (OVD) is proposed. It uses an omnidirectional scene and its corresponding depth to synthesize arbitrary omnidirectional views according to the user's translation and subsequently extracting viewport according to the user's orientation [5]. Depth is estimated from omnidirectional stereo pair when the captured depth is not available [6]. In [7], Kroon investigates the possibility of fulfilling the parallax requirement by utilizing depth maps: a synthetic ERP omnidirectional view (mono/stereo) and its corresponding computer generated depth(s) are used. These investigations, based on ERP omnidirectional views augmented with auxiliary information or metadata, bring a significant improvement over MPEG-I phase 1a, yet might not be sufficient to provide acceptable visual experience, especially considering that relevant information from the divergent views is lost during the stitching.

The quality of experience is an important issue. For instance, motion sickness results from the conflict between the expectation of the brain and the perceived motion. Current Head-Mounted Displays (HMD) manage to estimate users head motion precisely. However, the correct view or even more precisely the correct pixels, according to the user's movement, is not displayed, as the current scheme lacks the information of parallax, necessary for correct view rendering. Indeed, every motion, even small, is a mixture of both translations and rotations. For example, a simple head rotation by the user also consists of little translations. It leads to the conclusion that, even for a small motion, 6DoF is required. 6DoF is not intended to allow a user to move freely to any position in the scene but rather to freely rotate to 360° with small translations, referred to omnidirectional 6DoF [8].

In this paper, we explore the ability to provide omnidirectional 6DoF by getting rid of the 2D omnidirectional (panoramic) representation. The parallax between different divergent views is used in association with the 360 video capture. The paper is organized as follows. In section II, we present the recent progress in MPEG-I activities, the motivation of our work and the proposed method. Section III presents the experimental results and assesses the efficiency for different kind of motions, and different camera rig setups. Finally, section IV concludes the paper and provides directions for further explorations.

II. Motivation and proposed method for achieving omnidirectional 6DoF

The activities in [4] and [7] utilizes OVD format as a tentative to achieve omnidirectional 6DoF. The views are stitched in order to generate an ERP frame, loosing the parallax information between divergent views. The ERP depth is either generated or estimated. It is fed together with the ERP view to the coder. At the decoder side, an arbitrary ERP view is first synthesized and then the viewport extraction is performed to generate the 2D synthesized view. However, there are several issues associated with this approach. Firstly, although the parallax information is

ISBN 978-90-827970-1-5

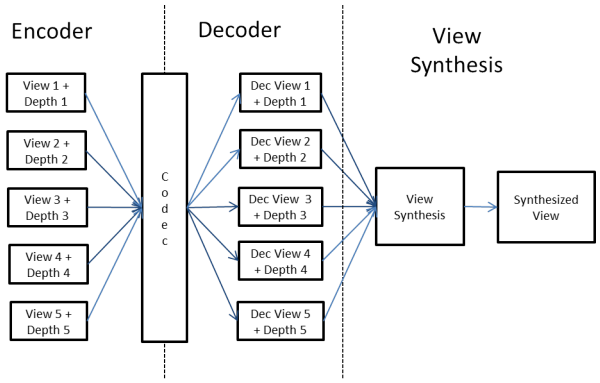


Fig. 1: Illustration of the proposed method to achieve omnidirectional 6DoF, in the case of five divergent views.

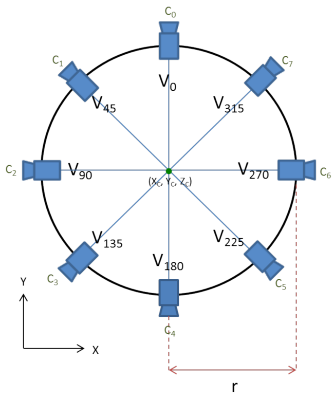


Fig. 2: Illustration of a circular rig with eight cameras.

provided by the ERP depth, it does not account for the parallax between divergent views. As the camera centers of the different views are not the same, this parallax is very important in order to perform precise view synthesis. Secondly, it is dependent on the projection scheme. It currently considers ERP projection, however, for other projections, the scheme should be adapted to the inherent geometry. Thirdly, ERP projection suffers from geometrical distortions, particularly at the polar regions. It decreases the efficiency of depth estimation and view synthesis. Finally, it performs synthesis of the whole ERP frame irrespective of the users view orientation. So, the computation is partially redundant.

To cope with the issues described above, we propose in this paper, to use divergent views and the corresponding depths to synthesize arbitrary views. Accordingly, the captured views and the corresponding depth are encoded without any projection, and synthesis of arbitrary views is performed using decoded views and depths. To assess the feasibility of the proposed scheme, we perform the synthesis with original (uncompressed) views and depths. However, in a realistic scenario, the coding of the views and the depths should also be involved, as illustrated in Fig. 1.

The divergent views are typically captured by a camera rig, such as the one shown in Fig. 2. Part of the scene is viewed by several cameras, so neighboring views are overlapping. As the cameras have a fixed distance between their optical centers, this allows estimating the depth of the scene using both disparity and distance of camera centers. Similarly, views at arbitrary positions can be synthesized using camera views and the corresponding depths with depth image based rendering (DIBR) method. Thus, the parallax between the divergent views (originating from the distance between optical centers) can be exploited to do the depth estimation and the view synthesis.

III. Experimental results and discussion

Two different synthetic contents called *WhiteRoom* and *Kitchen* (static scene, single time instance), generated with Blender software, are used in the following experiments. The choice of synthetic content instead of real data is made because currently, there is no publicly available dataset of divergent views, that can apply to our study. Moreover, there is no way to have pristine views at arbitrary positions and orientations to carry out a reliable objective quality assessment using full reference quality metrics for the interpretation of the results.

For these experiments, we denote the camera position and orientation respectively as $(camX, camY, camZ)$ in meters and $(rotX, rotY, rotZ)$ in degrees. Both contents are generated using a set of cameras having a focal length of 16mm, and a field of view (FoV) of 90°. The resolution of each divergent view is 1024×1024. Each camera is placed on the equator plane and there is no camera in the polar regions. The radius of the rig is called r (in meters) and is set to 0.1. N_{cam} cameras are placed at the vertices of a regular polygon (inscribed to the circle) with optical axes on the equator plane. Fig. 2 illustrates the adopted camera arrangement for $N_{cam} = 8$. The center of the camera i is defined as C_i. The optical axis of the camera 0 is parallel to the Y-axis, also indicated by $rotZ = 0$. The angle between the camera i and the camera 0 can be expressed as $i * 360/N_{cam}$. The resulting view of camera i is denoted as $V_{(i*360/N_{cam})}$ and the corresponding generated depth is denoted as $D_{(i*360/N_{cam})}$. The coordinates of the rig center are (X_c, Y_c, Z_c). According to the rig geometry, for $V_{(i*360/N_{cam})}$, the camera position, and orientation are expressed using Eq. 1. Note that, $camZ$, $rotX$, and $rotY$ are the same for all the cameras, as all of them are on the equator plane.

$$\begin{cases} camX = X_c - r * sin(i * 360/N_{cam}), & rotX = 0 \\ camY = Y_c + r * cos(i * 360/N_{cam}), & rotY = 0 \\ camZ = Z_c, & rotZ = i * 360/N_{cam} \end{cases} \quad (1)$$

One of the aims of the experiments is to explore different camera arrangements. Therefore, six different ones are tested, with $N_{cam} = \{5, 6, 8, 10, 12, 16\}$. The overlap between the views increases as the camera arrangement becomes denser (higher values of N_{cam}). An example of captured views (V_0 and V_{45}) of *WhiteRoom* and *Kitchen* is shown in Fig. 3.

A. Depth Estimation

In this section, we explore the possibility of estimating the depth of different views for different N_{cam} values. We use DERS-6.1, the MPEG depth estimation software [9] and denote the estimated depth as $\hat{D}_{rotZ,N_{cam}}$. For every camera arrangement, depth estimation of each view

ISBN 978-90-827970-1-5 © EURASIP 2018

Fig. 3: Divergent views of *WhiteRoom*: V_0 (top-left), V_{45} (top-right), and *Kitchen*: V_0 (bottom-left), V_{45} (bottom-right).

is performed using the current view and its two neighbor views. For example, $\hat{D}_{0,8}$ is estimated using V_{45}, V_0 and V_{315}. Fig. 4 gives a visual illustration of estimated depth maps and shows poor results for *WhiteRoom* (PSNR: 13dB) and even worse for *Kitchen* (PSNR: 6dB), when compared with the generated depth. This is because the latter has homogeneous background area, where depth estimation typically fails. On the contrary, *WhiteRoom* content has more features (edges, textures, etc) in the background, so depth estimation is comparatively better. To check the performance of depth estimation, we present the synthesis results with estimated depth in the next section.

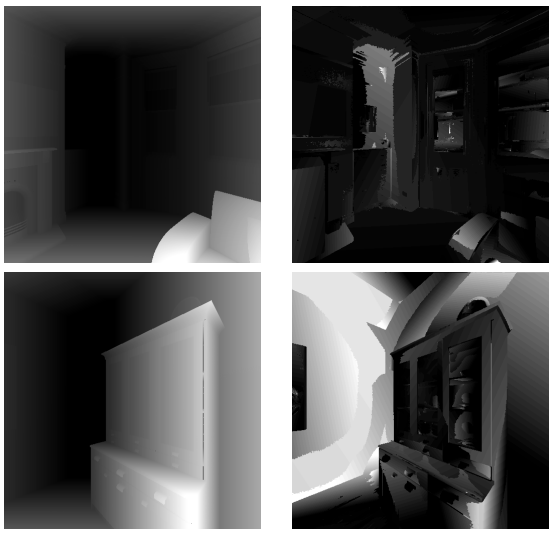
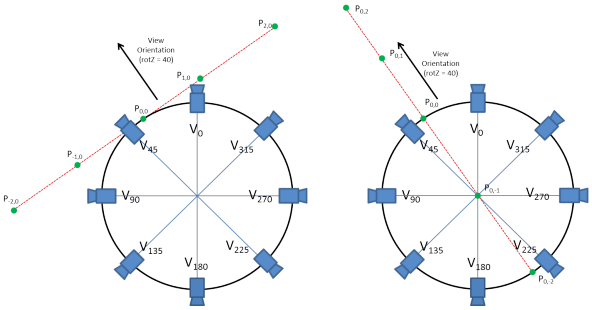


Fig. 4: Comparison between generated D_{45} (left) and estimated $\hat{D}_{45,8}$ (right) depth for *WhiteRoom* (top) and *Kitchen* (bottom).

B. View Synthesis

The view synthesis process is performed using the original captured views and their corresponding depths. To do so, we use VSRS version 4.2, the MPEG view synthesis reference software [10]. The camera parameter files, obtained from Blender software, are provided to VSRS. In order to perform the view synthesis, VSRS requires two input views (referred to as "left" and "right" neighbor views, corresponding to the closest $rotZ$) and the corresponding depths. For convenience, we denote the synthesized view as $S_{position,rotZ}$. In Table I, left and right neighbor views are listed for synthesized view orientation $rotZ$ = 40. A "reference" is generated using Blender software for each synthesized view for quality comparison. For objective quality evaluation, PSNR and MS-SSIM metric are used [11]. We consider a user's position at $P_{0,0}$ = $(X_c - 0.1 * sin(40), Y_c + 0.1 * cos(40), Z_c)$ on the rig and oriented at $rotZ = 40$, with the resulting view $S_{P_{0,0},40}$, as shown in Fig. 5. An illustration of the obtained views after synthesis is given in Fig. 6. In the following, we present two selected scenarios for view synthesis and their corresponding results.



Scenario 1: move horizontally Scenario 2: step-in / step-out

Fig. 5: Illustration for the two scenarios for eight camera arrangements.

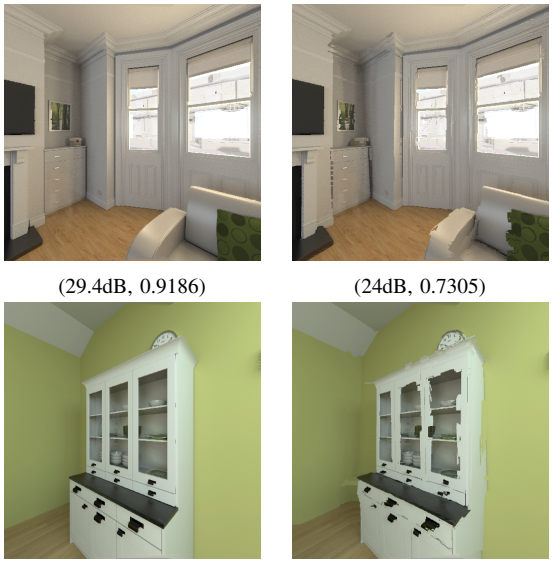
(33dB, 0.9691) (28.3dB, 0.9324)

(36.6dB, 0.9952) (28.4dB, 0.9523)

Fig. 6: Visual illustration of synthesized view $S_{P_{0,0},40}$, with generated (left) and estimated (right) depth for *WhiteRoom* and *Kitchen* (N_{cam} = 8). (PSNR, MS-SSIM) values are given for objective quality evaluation.

1) Scenario 1 (move horizontally): from position $P_{0,0}$, the user performs a horizontal movement, as illustrated in Fig. 5. Position $P_{1,0}$ and $P_{2,0}$ correspond to 0.1m and

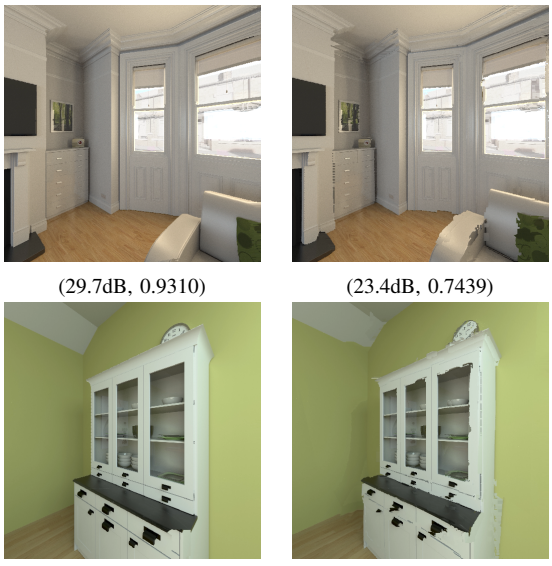
ISBN 978-90-827970-1-5 © EURASIP 2018



(29.4dB, 0.9186) (24dB, 0.7305)

(33.6dB, 0.9732) (21dB, 0.6792)

Fig. 7: Visual illustration of synthesized view $S_{P_{1,0},40}$ (scenario 1), with generated (left) and estimated (right) depth for *WhiteRoom* and *Kitchen* (N_{cam} = 8). (PSNR, MS-SSIM) values are given for objective quality evaluation.



(29.7dB, 0.9310) (23.4dB, 0.7439)

(29.2dB, 0.9597) (21dB, 0.7276)

Fig. 8: Visual illustration of synthesized view $S_{P_{0,1},40}$ (scenario 2), with generated (left) and estimated (right) depth for *WhiteRoom* and *Kitchen* (N_{cam} = 8). (PSNR, MS-SSIM) values are given for objective quality evaluation.

TABLE I: Orientation of left (V_{left}) and right (V_{right}) neighbor views chosen for synthesized view with orientation $rotZ = 40$.

N_{cam}	5	6	8	10	12	16
V_{left}	V_{72}	V_{60}	V_{45}	V_{72}	V_{60}	V_{45}
V_{right}	V_0	V_0	V_0	V_{36}	V_{30}	$V_{22.5}$

0.2m of movement on the right side, and position $P_{-1,0}$ and $P_{-2,0}$ correspond to 0.1m and 0.2m of movement on the left side. The orientation of the user remains the same i.e. $rotZ$ = 40. Fig. 10 and 11 respectively provide PSNR results for the synthesized views using generated and estimated depths, compared with corresponding pre-calculated references for six different camera rig configurations. For the synthesis with estimated depths, the quality of the synthesized picture decreases gradually with the movement from the user's central position $P_{0,0}$. This result was expected since the position is on the camera rig. Thus, it is very close to the captured views, resulting in higher quality of the synthesized view. The PSNR drops below 25dB for 0.1m of movement and severe artifacts are visible, as shown in Fig. 7. The synthesis quality of *Kitchen* is worse compared to *WhiteRoom*, due to the poor quality of estimated depth.

For the synthesis with generated depth, a similar phenomenon is observed. However, the quality of the synthesis is much better compared to the use of estimated depth, due to high quality depth. For movement longer than 0.1m in both directions, the PSNR drops below 30dB and rendering artifacts appear in the resulting view. With regards to the N_{cam} parameter, Fig. 10 shows that a higher number of cameras (i.e, denser camera arrangements) can ensure a better rendering quality. Obviously, this result is also content-dependent. For both scenes, the quality of the synthesized view $S_{P_{0,0},40}$ for N_{cam} = 10 has superior performance compared to other arrangements (except for *Kitchen*, where N_{cam} = 6 outperforms N_{cam} = 10). This is because the "right" neighbor view for this configuration (Table I), V_{36}, is the closest captured view. Moreover, for the case of reflectance (such as reflection of the painting on the glass), the synthesis quality is poor. The error map for the aforementioned view for $S_{P_{0,0},40}$ is given in Fig. 9, where higher errors can be observed for the glass encapsulated region.

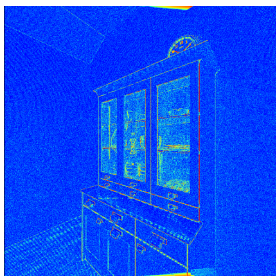


Fig. 9: Error map of $S_{P_{0,0},40}$, blue color indicates lower value and red color indicates higher value. Higher error energy is observed in glass encapsulated region.

2) Scenario 2 (step-in/ step-out): from position $P_{0,0}$, the user performs a step-in/step-out, as illustrated in Fig. 5. Position $P_{0,1}$ and $P_{0,2}$ correspond to 0.1m and 0.2m of step-in, and position $P_{0,-1}$ and $P_{0,-2}$ correspond to 0.1m and 0.2m of step-out. The orientation of the user remains the same, as in scenario 1, i.e. $rotZ$ = 40. Fig. 10 and 11 respectively provide PSNR results for the synthesized views using generated and estimated depths. Corresponding visual illustration is presented in Fig. 8.

Similar to scenario 1, the quality of the synthesized

ISBN 978-90-827970-1-5 © EURASIP 2018

views decreases gradually with the amount of step-in / step-out. For synthesis with generated depth, the PSNR of the synthesized views drops below 30dB, for a step-in/ step-out of more than 0.1m. For synthesis with estimated depth, the quality is much lower. Due to the lower depth quality of *Kitchen*, the synthesis quality is worse than that of *WhiteRoom*. As illustrated by the Fig. 8, severe artifacts can be observed, particularly at the edges and near objects. Similar to scenario 1, the average quality is comparatively higher for higher N_{cam} values.

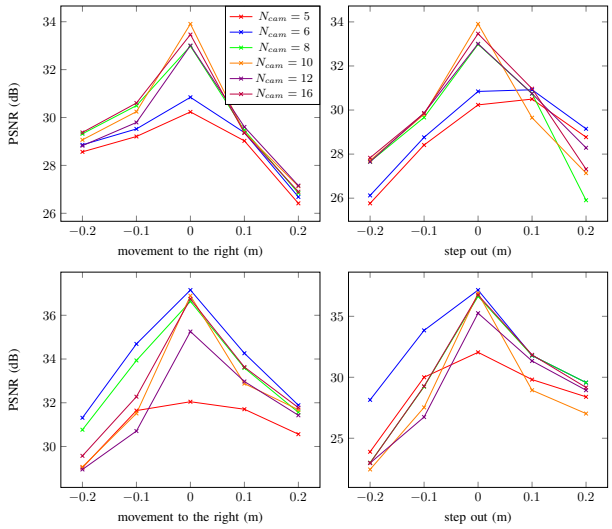


Fig. 10: PSNR of synthesized frames with generated depth for *WhiteRoom* (top) and *Kitchen* (bottom) in scenario 1 (left), scenario 2 (right).

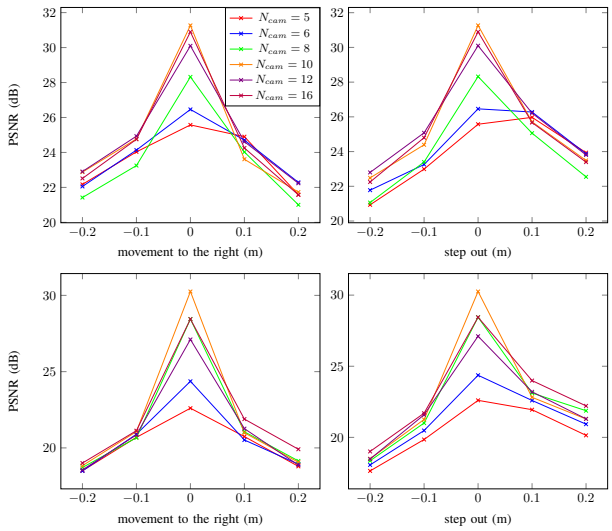


Fig. 11: PSNR of synthesized pictures with estimated depth for *WhiteRoom* (top) and *Kitchen* (bottom) in scenario 1 (left) and scenario 2 (right).

IV. Conclusion and Future Work

The current MPEG phase 1b activity on 3DoF+ might not be sufficient to provide an acceptable visual experience for immersion. 6DoF is required to tackle this issue. In this paper, we explore a novel way to achieve omnidirectional 6DoF by using the parallax between different divergent views. We show that the depth estimation is possible using this parallax, but unfortunately, state-of-the-art depth estimation techniques provide insufficient quality and even fails in the presence of homogeneous areas. View synthesis with the estimated depth provides acceptable quality when the position of the user is on the camera rig. Any movement out of the rig results in a significant drop in synthesis quality, thus making the synthesized picture visually unacceptable. Synthesis with generated depth provides much better quality, and a movement of 0.1m from the rig can be performed while maintaining an acceptable quality. However, the view synthesis still fails in the presence of reflection and transparent objects. In general, the average quality improves with the increase of the number of cameras in the rig. Besides, it can be observed for both scenarios that the resulting quality reduces gradually as the user moves away from the rig. A further study is required for the improvement of the depth estimation and the view synthesis.

References

[1] Fan Zhang and Feng Liu, "Parallax-tolerant image stitching," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June 2014, pp. 3262–3269.

[2] Rob Koenen and Mary-Luc Champel, "Requirements MPEG-I phase 1b," in *MPEG-requirements of ISO/IEC JTC1/SC29/WG11*. MPEG, 2018, pp. 1–7.

[3] Renaud Dore and Mary-Luc Champel, "Information on 3-DoF+ interactive parallax impelementation," in *MPEG-systems of ISO/IEC JTC1/SC29/WG11*. MPEG, 2018, pp. 1–12.

[4] Krzystof Wegner, Olegard Stankiwicz, and Marek Domanski, "Considered format for omnidirectional 6-DoF/3-DoF+," in *MPEG-I visual of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2017, pp. 1–3.

[5] Krzystof Wegner, Olegard Stankiwicz, Adrian Dziembowski, Dawid Mieloch, and Marek Domanski, "Omnidirectional 6-dof/3-dof+ rendering," in *MPEG-I visual of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2017, pp. 1–6.

[6] Krzystof Wegner, Olegard Stankiwicz, Tomasz Grajek, and Marek Domanski, "Depth estimation from circular projection of 360 degree 3D video," in *MPEG-I visual of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2017, pp. 1–7.

[7] Bart Kroon, "Adding depth maps to fulfill MPEG-I phase 1b parallax requirement," in *MPEG-I of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2017, pp. 1–4.

[8] Mary-Luc Champel, Rob Koenen, Gauthier Lafruit, and Madhukar Budagavi, "Technical report on architectures for immersive media," in *MPEG-I of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2017, pp. 1–24.

[9] Krzystof Wegner, Olegard Stankiwicz, Masayuki Tanimoto, and Marek Domanski, "Enhanced depth estimation reference software (DERS) for free-viewpoint television," in *MPEG-FTV of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2014, pp. 1–7.

[10] Krzystof Wegner, Olegard Stankiwicz, Masayuki Tanimoto, and Marek Domanski, "Enhanced view synthesis reference software (VSRS) for free-viewpoint television," in *MPEG-FTV of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. MPEG, 2017, pp. 1–4.

[11] Zhou Wang, Eero P Simoncelli, and Alan C. Bovik, "Multiscale structural similarity for image quality assessment," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference*, Pacific Grove, CA, USA, 2003, vol. 2, pp. 1398–1402.

ISBN 978-90-827970-1-5