Cinematic Virtual Reality With Motion Parallax From a Single Monoscopic Omnidirectional Image

Grégoire Dupont de Dinechin, Alexis Paljic

Cinematic Virtual Reality With Motion Parallax From a Single Monoscopic Omnidirectional Image

Grégoire Dupont de Dinechin, Alexis Paljic
MINES ParisTech, PSL Research University, Centre for Robotics
Paris, FRANCE
Email: {gregoire.dupont_de_dinechin, alexis.paljic}@mines-paristech.fr

***Abstract*—Complementary advances in the fields of virtual reality (VR) and reality capture have led to a growing demand for VR experiences that enable users to convincingly move around in an environment created from a real-world scene. Most methods address this issue by first acquiring a large number of image samples from different viewpoints. However, this is often costly in both time and hardware requirements, and is incompatible with the growing selection of existing, casually-acquired 360-degree images available online. In this paper, we present a novel solution for cinematic VR with motion parallax that instead only uses a single monoscopic omnidirectional image as input. We provide new insights on how to convert such an image into a scene mesh, and discuss potential uses of this representation. We notably propose using a VR interface to manually generate a 360-degree depth map, visualized as a 3D mesh and modified by the operator in real-time. We applied our method to different real-world scenes, and conducted a user study comparing meshes created from depth maps of different levels of accuracy. The results show that our method enables perceptually comfortable VR viewing when users move around in the scene.**

I. Introduction

Using the growing online collection of 360-degree images to generate VR environments representing real-world scenes is tempting for a great number of applications, such as virtual reality tours and immersive cultural heritage experiences. These real-world-based applications are often grouped under the name *cinematic VR*. Such experiences are all the more convincing if they provide a sense of *motion parallax*, i.e. if, when users move around in the environment (translation, not just head rotation), novel viewpoints are rendered in which the different objects in the scene are shifted differently depending on their respective depths.

Representing a scene with motion parallax using a single image as input, however, is not an easy task. One notably cannot estimate depth information using standard methods based on feature detection and matching in multiple stereo pairs. Nor does one have access to color or structure information for objects occluded or not seen in the original view. A single image with a restricted field of view offers in fact very limited information on the scene.

This has led many in the literature to discard the problem of generating a scene from a single image, and focus instead on the less-constrained problem of using multiple images taken from different viewpoints. We argue, however, that the complementary development of consumer VR HMDs and low-cost 360-degree cameras breathes new life into the issue. One can now acquire images containing full 360-degree information on a scene as easily as as one would acquire a regular photo, i.e. instantaneously, with a simple click of a button. Consumers now also easily have access to immersive devices for viewing virtual environments. We believe that these developments are game-changing, and that studying how to create 3D environments for VR from a single 360-degree image is a research avenue with great potential: it would open up the possibility of viewing any such image found online in VR with full motion parallax. To our knowledge, this is a question that has not yet been addressed in the literature.

Our contributions to the problem are threefold. Firstly, we present a novel method for manually generating a depth map from a 360-degree image, a process we propose to do interactively in VR to enhance spatial visualization. The problem of recovering depth from a single image is frequently examined in the literature, yet few works study the specific case of omnidirectional images, and to our knowledge none propose to do so using a VR interface. We provide insights on the specific strengths of 360-degree images that facilitate the depth assignment process, and explain the advantages of doing so in VR. Secondly, we provide a detailed account of the strengths and limits of the monoscopic panorama plus depth representation, which creates a scene mesh from a single 360-degree depth-color image pair. Few papers have examined this representation in detail despite the growing number of images acquired by consumer monoscopic 360-degree cameras, and fewer still have proposed using this representation to enable VR viewing with motion parallax. Thirdly, we conduct and discuss a user study comparing, for different real-world scenes, how different meshes (i.e. different depth maps) of the scene are perceived in terms of user comfort. The results of this study show that different levels of depth map accuracy are perceived as providing significantly different levels of comfort, and are not perceived in the same way under different viewing conditions. We conclude by discussing the implications of these results, as well as potential paths for future work.

II. Related Work

A. Capturing Real-World Scenes for VR Applications

Many recent works in the field of image-based modeling and rendering have examined the question of rendering novel views from a set of image samples for viewing in VR. We

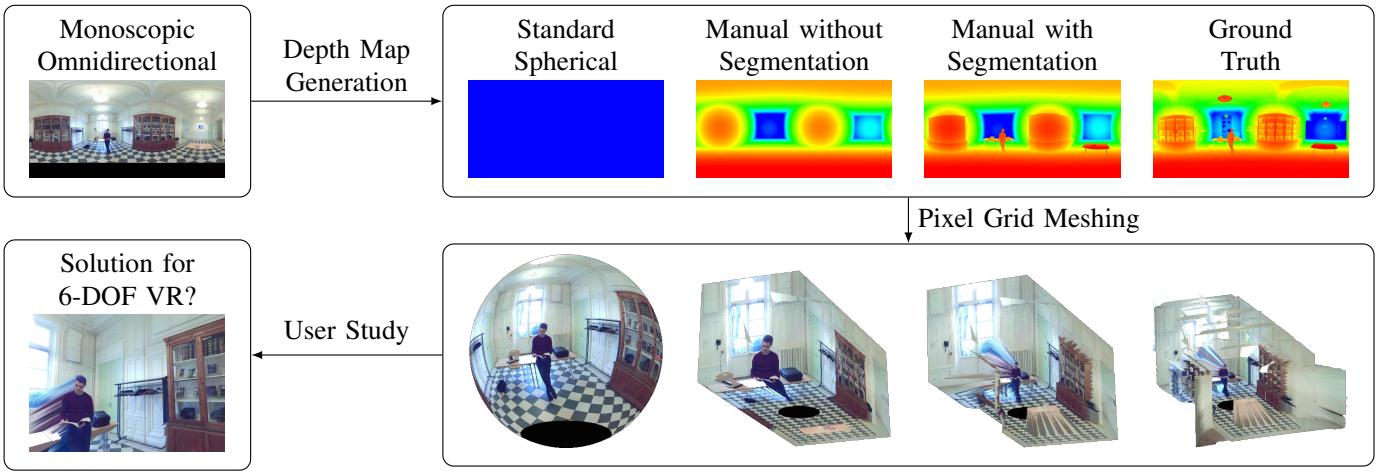

Fig. 1. A summary of our approach.

discuss in the following paragraphs how our approach can be compared with these state-of-the-art methods.

A first group of methods seeks to render novel views without recovering explicit geometry for the scene. This includes light field rendering techniques, seen as having great potential for VR content creation [1] and recently showcased in Google's Welcome to Light Fields VR experience [2]. Although they also enable convincing VR viewing with motion parallax in a given area, these works require the use of expensive camera arrays and are based on very large sets of image samples, all in which they thus differ from our approach. Other such works seek instead to generate omnistereo images and videos, presenting innovative camera rigs and novel algorithms to obtain high-quality outputs at lower costs [3][4]. Although some of these works are also based on the use of consumer hardware, they only seek to provide a sense of binocular parallax, whereas our approach aims to enable motion parallax.

Another popular group of techniques explicitly reconstructs textured 3D meshes. This includes methods referred to as photogrammetry or videogrammetry, based on structure-from-motion and multi-view stereo algorithms, and aiming to create 3D meshes or volumetric videos. Many recent works thus present the use of state-of-the-art methods to explicitly generate meshes for static scenes [5][6] and dynamic objects [7][8] for VR applications. Our approach differs from these works in that we seek to use only a single image for our reconstruction, not a large dataset. Contrarily to approaches requiring the use of converging multi-camera rigs, we also only require the use of a consumer monoscopic 360-degree camera. That being said, our approach resembles these works in that we generate a textured 3D mesh as output. A strong resemblance can notably be drawn with the approach presented in [5], as we both use low-cost, consumer hardware to produce textured meshes of real-world scenes from recovered depth maps.

A related group of methods is concerned with representing environments by recovering a dense depth map for each input view, enabling the creation of a point cloud which can then be reprojected or used as an image warping proxy to generate novel views. The method presented in [9] thus renders VR video with motion parallax using a single moving consumer monoscopic omnidirectional camera. A similar low-cost approach is presented in [10], based on computing optical flow between two displaced monoscopic 360-degree cameras to enable VR motion parallax in a small zone. Our work resembles these approaches by the use of consumer hardware and recovered depth information to enable VR viewing with motion parallax. However, we do not rely on multiple views of the same scene, nor do we use image warping to produce novel views.

A final set of papers we believe is important to cite as a related work is the work of J. Thatte, concerned with developing novel scene representations that enable motion parallax in a given area with only a small number of omnistereo panoramas plus depth [11][12]. Although these works focus on reducing visual artifacts, whereas our constraint is using only a single image, our approaches converge in seeking to enable VR viewing with motion parallax using an omnidirectional color-depth image pair. Moreover, [11] presents an interesting comparison with the monoscopic panorama plus depth representation, a representation we expand on in this paper.

B. Manual 2D-to-3D Conversion

Another related field of work is concerned with developing methods that leverage human perception of monocular depth cues to add depth to existing color images. The typical manual workflow consists in three steps: segmentation/rotoscoping, depth assignment, and inpainting [13]. Each step can be automated to some degree and facilitated by the user interface, yet by definition leverages human input at some point, often for reasons of accuracy. Our approach recovers depth based on such a manual pipeline, with the particularity of using a VR user interface for interactive real-time 3D visualization. We believe that such a user interface is particularly adapted for the 2D-to-3D conversion of omnidirectional images.

Note that many recent works instead present semi-automatic methods, typically using human input in the form of sparse

scribbles on a color image [14][15] or a pre-segmentation of said image [16] to help guide depth generation algorithms. Our approach may be augmented to integrate such methods in future work, if they prove to generate sufficiently convincing results for viewing in VR. The same can be said for methods that generate depth maps using deep convolutional neural networks [17][18], which may be used to complement our approach provided the necessary datasets are available.

III. Our Approach

A. Enabling Motion Parallax in VR from a Single Image

The standard solution for displaying a single 360-degree image in a VR HMD is to project the image on an inside-facing sphere of large radius surrounding the user (we refer to this as the *standard spherical* representation). A drawback of this method is that it fails to provide a sense of motion parallax. This causes an uncomfortable sensation when users perform translations, due to the incoherence between the respective stimuli received by the visual and vestibular systems.

In this paper, we propose to solve this problem by generating novel information in the form of a dense depth map for the image, i.e. a per-pixel encoding of the distance between elements in the scene and the central viewpoint. The meshed model created from the obtained depth-color image pair is referred to as the *monoscopic panorama plus depth* representation. It can be seen as an extension of the standard spherical representation, where the uniform depth map is replaced by a more accurate one, deforming the sphere in a radial fashion. This paper intends to show the relevance of creating such a mesh from a single 360-degree image, in the sense that this process (1) requires a single image as input, (2) entails little to no additional cost, and (3) creates a representation significantly more comfortable than the standard spherical alternative.

B. Depth Map Generation With Human Input

For the representation to be relevant, one must therefore be able to generate perceptually convincing depth maps rapidly, at a low cost, and based solely on a single omnidirectional image. Our approach aims to solve that problem by leveraging the strengths of both 360-degree panoramas and VR interfaces.

An efficient pipeline can indeed be defined for any 360-degree image containing a planar surface of easily-estimated depth, e.g. the ground. Namely, one can take the following steps (visualized in Figure 2):

1) The operator defines a maximum depth, used to initialize a uniform depth map. This creates a sphere in 3D space.
2) The operator estimates the distance between ground and acquisition camera center. The bottom of the sphere is horizontally flattened at the corresponding depth.
3) The operator enters VR. The sphere created in (1) and (2) is visualized as a static 3D mesh, around which the operator can move. On the wall of the virtual room is also an image representation of the depth map.
4) Using the controllers, the operator selects an object in the image (segmented beforehand) to enhance in depth. The operator determines where the object is in 3D space,

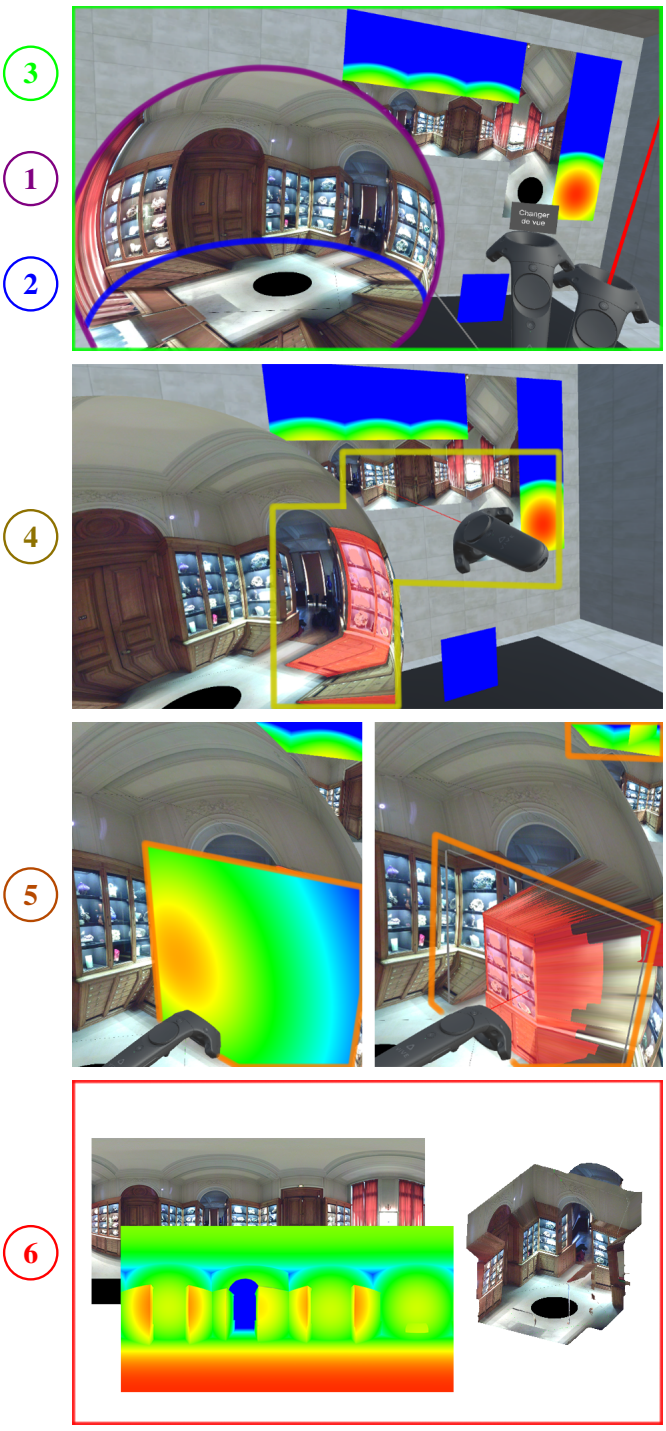

Fig. 2. Main steps of the manual creation process, demonstrated here using our prototype implementation. The numbering corresponds to the steps described in the text of Section III-B.

by looking at the intersection in the image of the object and the ground (placed in 3D space in (2)).

5) Using the controllers, the operator takes a 3D primitive in hand. The operator places the primitive inside the sphere at the position determined in (4). Upon release of the operator's hand, the program computes the projection of the segmented part of sphere onto the primitive. The depth map and 3D visualization (deformed sphere) are updated in real-time.
6) The operator iterates steps (4)-(5) for all objects segmented in the image. The operator then exits VR and saves the output depth map.

Note that our implementation projects the image based on computed in-engine depth for each pixel, without the need for a parameterization of the projection primitives. We can thus estimate every object in the scene using planes (billboard representation), but can also use any other primitive, and even 3D models if need be. For simplicity, the pre-segmentation part of the process was not done in VR, but using consumer image editing software, as it would have been time-consuming to obtain similar results in our prototype.

Also note that one simply has to provide a maximum depth (in step (1)) and an estimated distance from the ground (in step (2)) to subsequently obtain metric position information for all objects in the scene, provided they are connected to the ground (at least transitively, i.e. provided an unbroken path to the ground can be drawn in the image).

The choice of implementing this process as a VR application allows for a very intuitive positioning of primitives in 3D space. Using our implementation, an operator simply has to select a segment of the image, choose a projection primitive, and manually place the primitive in 3D space at the estimated position for the real-world object. Since the operator can see the applied changes in real-time and in VR, and given that the underlying goal is to use the reconstructed environment for viewing in VR, this creation tool thus generates a what-you-see-is-what-you-get output, which we believe is particularly relevant. The operator can also choose to activate an inside view of the scene for additional precision.

As an output of this depth generation step, we thus obtain an omnidirectional depth map corresponding to the input color image. With our implementation, this comes at the cost of only about twenty minutes for an experienced operator on a moderately complex scene. This paper does not aim to evaluate our particular prototype implementation, which was developed only as an illustration of the potential of these methods and for the purpose of the subsequent user study. As discussed in Section II-B, semi-automatic approaches are likely to be quicker, and may be examined in future work. We do however also encourage the development of novel tools centered on the specificities of omnidirectional images and VR interfaces described above, and hope that our insights on the subject can help developers adapt their tools when relevant.

We now give more details on how we transform the obtained color-depth image pair into a textured mesh, and outline the strengths and limits of this representation.

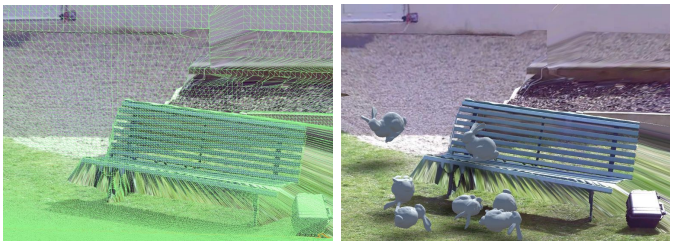

Fig. 3. The monoscopic panorama plus depth mesh immediately enables effects such as collision handling and shading.

C. The Monoscopic Panorama Plus Depth Representation

Monoscopic panoramas plus depth are very practical on multiple levels. Concerning data storage and manipulation, they are composed of images, a format which is compact, easy to modify using consumer-available editing tools, and already well-integrated as a data format in modern game engines. As for acquisition, monoscopic color panoramas are easy to acquire at a low cost using modern consumer pocket-sized cameras. As shown in Section III-B, the complementary per-pixel depth information can then be generated post-acquisition with relative ease.

Another strength of this representation is that its conversion to the 3D space is conceptually simple. In the equirectangular format for example, an omnidirectional depth map can be interpreted directly as a 3D point cloud, where each (w, h) pixel ($0 \leq w, h \leq 1$) of 8-bit value v in the depth map (where v encodes the real-world depth d) is a 3D point defined by the spherical coordinates $\{d, 2\pi w, \pi h\}$. Note that if the depth map contains only 256 values, a staircase effect may appear, very visible in VR. In our implementation, we thus encoded the depth maps such that $d = R + 0.01G + 0.0001B$ (in meters). The colored maps shown in this paper, where red is close and blue is far, are only used for better visual representation, not mesh creation.

Going even further, monoscopic panoramas plus depth provide a natural mesh for the point cloud, based on the pixel grid. Indeed, one can intuitively form a 3D mesh by defining each pixel in the image as a mesh vertex and each quad of pixels as a pair of mesh triangles. Projecting each vertex into 3D space can then be done efficiently on GPU using vertex shaders. Mesh deformation can thus be done in real-time, meaning that movement in the scene can be accounted for without the need for any animation software, and that our approach can immediately be transposed to video panoramas. Note that the large number of vertices (e.g. 7372800 for a 3840x1920 panorama) may require building multiple meshes, but can easily be handled by standard game engines provided one has the modern hardware required to run a roomscale VR head-mounted display (e.g. in our case, HTC Vive in Unity with NVIDIA GeForce GTX 1070 GPU), and thus does not hinder real-time performance (e.g. above 120 FPS with our implementation).

A consequence of this is that monoscopic panoramas plus depth easily enable user interaction, in particular moving

around in the scene. Indeed, since rendering is done using the conventional graphics pipeline, users can move in this virtual environment exactly as they would in a computer-generated one, within the boundaries defined by the maximum depth encoded in the depth map. Moreover, real-time interactive elements such as shading from virtual objects and collision handling can immediately be accounted for (see Figure 3). This means that this representation has strong potential for interactive cinematic VR experiences.

D. Visual Artifacts: a Potential Source of Discomfort

Why are monoscopic panoramas plus depth not more popular then? A blatant weakness of this representation is the large amount of visual artifacts it produces at the edges of occluding objects. Indeed, since all the available color information is contained in the single monoscopic panorama, no hole-filling step can be performed to recover occluded objects. One therefore has to make do with the interpolation naturally performed by the fragment shader between distant vertices, causing a stretching effect. This effect can be observed in other recent works demonstrating depth-based mesh reconstruction [5], and was referred to in the seminal work of T. Kanade as *phantom surfaces* [19]. This representation therefore does not provide the most visually faithful representation of the scene, and gains in relevance only when one wishes to use a single image as input: typical use cases could be immersing friends in a 360-degree photo casually taken during a tourist trip (e.g. oneself in front of a historic monument), or visualizing an omnidirectional image found online in a VR HMD. Those are the typical use cases we target in this paper.

In order to determine to what extent our method is able to enhance users' sense of comfort despite the presence of visual artifacts, we thus led a user study, measuring reactions to different levels of depth map accuracy under different viewing conditions.

IV. User Study

A. Independent and Dependent Variables

A first independent variable for the study was the type of scene representation, i.e. the type of depth map used to represent the scene. Four levels of this variable were considered:

- SCAN: the omnidirectional color image and depth map were acquired by LiDAR; this served as a ground truth reference (ideal depth map), and provided the color images for the other representations.
- NONE: no depth was recovered; the depth map was uniform at 10m, creating a standard spherical projection.
- MMIN: depth was added manually but without the time-expensive segmentation step; the depth map thus accounted only for the walls and ground in the scene.
- MMAX: a segmentation step was performed before the manual depth assignment step; the depth map thus accounted for all close objects in the scene, including walls and ground.

A second independent variable was the viewing condition, which we separated into two levels, VIEW and MOVE. This variable was chosen as a between-subjects factor to avoid a learning effect due to users understanding the difference between the distinct scene representations. Two user groups were thus formed before the study. The VIEW group conducted the experiment sitting on a rotating chair, able only to rotate body and head, without translation. The MOVE group on the other hand conducted the experiment with room-scale movement enabled, being encouraged to naturally walk around the 2.5x2.5m testing space.

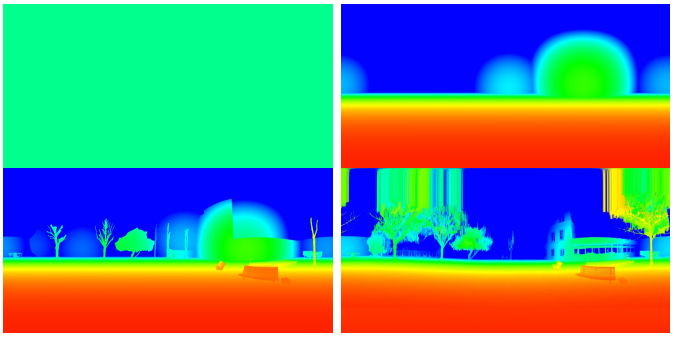

Fig. 4. The four different levels of depth map accuracy, defining four levels of scene representation. From left to right, top to bottom: NONE, MMIN, MMAX, SCAN. Note that the ground truth depth map was vertically filled where the laser scanner was unable to acquire depth values.

As a dependent variable, we chose to measure users' sense of comfort on a 5-point Likert scale. Users were immersed in a scene representation, tasked either with observing or moving in the space, and asked to describe how comfortable they felt whenever ready to do so. Users were told that 1 corresponded to feeling very uncomfortable / disturbed by the representation, and 5 to feeling very comfortable / able to spend much more time within it. Note that our user study thus derives results based on user responses to a subjective Likert-scale question, and is therefore to be interpreted as seeking to determine general trends, not individually meaningful values [20].

B. Hypotheses

Our hypotheses for the experiment were as follows:

- Hypothesis 1 (H1): In the room-scale viewing condition, users will feel most comfortable when there is added depth (depth-augmented representations). Comfort is likely to be linked to how close the depth map is to the ground truth depth map, and the ground truth is likely to have the highest comfort scores.
- Hypothesis 2 (H2): In the seated viewing condition, users will feel most comfortable when there is no added depth (spherical representation). Indeed, seated users are likely to notice depth-induced visual artifacts, but not the absence of motion parallax.

C. Experimental Setup

The user study was conducted on 25 participants (6 female). The median age was 25 (standard deviation: 3.36).

Four different omnidirectional images were used. These images were captured using a FARO laser scanner, to obtain

Fig. 5. The four environments used in the user study. From left to right, top to bottom: Garden, Bookshelves, Snow, Museum. Each environment was depth-augmented using four different depth maps.

the ground truth depth maps. The captured images had a resolution of 8192x3414 representing a 360x150 field-of-view, which we padded with black pixels at the nadir to obtain 8192x4096 equirectangular images representing a 360x180 field-of-view. The scenes were chosen to be diverse in terms of location, size and illumination: one was taken in a sunlit garden with benches, one in a snow-covered garden with a table, one in a small room with a person and two bookshelves, and one in a museum room filled with various minerals.

The images were depth-augmented by the experimenter in a pre-processing phase. For simplicity, depth assignment was done by projecting exclusively on plane primitives. Two depth maps were thus manually created for each of the four image, in addition to the ground truth and uniform depth maps. Each participant therefore gave comfort scores to $4*4 = 16$ different environments. The order in which the environments were presented to users was randomized, to prevent biases (due to user fatigue for example). In total, $16*25 = 400$ observations were thus recorded. Of the 25 participants, 13 were tasked with moving around the room, and 12 with observing from a seated position. 208 observations were thus recorded in the MOVE condition and 192 in the VIEW condition. The experiment lasted on average 463 seconds, i.e. 7-8 minutes.

D. Results

Our results were obtained by conducting an ANOVA on the comfort scores given by the participants, with scene representation as a within-subjects factor and viewing condition as a between-subjects factor. Statistical testing was computed in R.

Interaction between viewing condition and scene representation was found to be a significant factor ($p < 0.001$). We refined this result by conducting Tukey's HSD test to determine significantly different means for comfort scores under the different conditions. In this way, under the MOVE viewing condition, user appreciation was found to be significantly different for the scene representations NONE and SCAN ($p < 0.01$) and NONE and MMAX ($p < 0.01$). Additionally, the NONE scene representation was appreciated in a significantly different manner based on whether the viewing condition was VIEW or MOVE ($p < 0.05$). No other result was found to be of statistical significance.

TABLE I
CONFIDENCE INTERVALS - COMFORT SCORES

Viewing	Depth	Mean	CI (95%)
MOVE	SCAN	3.48	[3.26; 3.70]
MOVE	MMAX	3.42	[3.15; 3.70]
MOVE	MMIN	2.94	[2.66; 3.22]
MOVE	NONE	2.58	[2.24; 2.92]
VIEW	SCAN	3.23	[2.89; 3.56]
VIEW	MMAX	3.00	[2.68; 3.32]
VIEW	MMIN	3.42	[3.14; 3.70]
VIEW	NONE	3.52	[3.24; 3.80]

Means and 95% confidence intervals for the comfort scores can be found in Table I.

E. Hypothesis Validation

H1 was proven for the SCAN and MMAX depth-augmented representations, as compared to the NONE spherical representation. Although the same cannot be said for MMIN, note that it also benefited from higher comfort scores than NONE in our sample group. These results validate the idea that the monoscopic panorama plus depth representation is a relevant choice when users are encouraged to perform room-scale movement. They are also consistent with the idea that user comfort is increased as the depth map is refined to grow closer to the ground truth (although this statement cannot be validated statistically beyond our sample group).

H2 cannot be confirmed with this user study, since no statistical significance was established between scene representations under the VIEW condition. That being said, the spherical representation was established to be significantly more appreciated under this condition than when users could perform full room-scale movement. In a non-statistically significant manner, this representation also obtained the highest comfort score in our sample user group. The H2 hypothesis is therefore consistent with the results of our user study.

V. DISCUSSION

A. Viewing Conditions

Under the constraint of creating an environment from a single omnidirectional image, these results validate the use of monoscopic panoramas plus depth for applications where users are to move around in the scene. The output scene representation enables perceptually comfortable viewing, with increasing levels of comfort as the depth map grows increasingly accurate. Depth-augmented meshes are preferred in that sense to the standard spherical representation.

However, our results encourage us to recommend to content creators *not* to use this representation when the VR application does not require enabling more than just head rotation. Indeed, depth assignment remains an operation with non-null cost, and our user study fails to show that it improves user comfort when users cannot translate in the scene. In fact, users in our sample group preferred when no depth was added, since in that case the lack of motion parallax was less a factor of discomfort

than the presence of visual artifacts. The standard spherical projection may therefore be more relevant if the environment is to be viewed in a seated position or without positional tracking.

B. Depth Map Accuracy

If the monoscopic panorama plus depth representation is used, we recommend performing a segmentation step before manually assigning depth. Indeed, estimating depth for only the walls and ground seems to enhance user comfort much less than if other objects are depth-augmented as well. The cubic representation for a walled room for example is therefore probably not the best solution if there are also other visible objects in the scene. On the other hand, depth-enhancing segmented elements seems to produce results perceptually close to those obtained when working with the ground truth. Our manual depth maps, which were created using only planes as the projection primitives, thus obtained comfort scores similar to those given to the ground truth depth map. This validates the idea of generating depth in a post-acquisition step: precise depth values (i.e. close to the ground truth) are not required to create a comfortable VR experience.

C. Extending Our Approach

A first possible extension of this paper could consist in exploring semi-automatic depth assignment methods. Deep convolutional neural networks notably have been a popular approach in past years, yielding good results when trained correctly on the right datasets. Implementing these methods could thus be interesting, provided we have access to large datasets of 360-degree depth-color image pairs for different types of scenes. Whether or not depth maps generated by such methods are convincing during VR viewing is to be studied.

A second path for future work could be extension to video. If one moves the monoscopic 360-degree camera during filming, approaches such as [9] offer hole-filling solutions, and are thus to be preferred. If the video is taken with a fixed camera however, the constraints are similar to those considered in this paper, in which case the approach presented here is relevant as well. Extending our current approach and implementation for video seems quite feasible, e.g. by replacing segmentation with rotoscoping, and by interpolating the depth assignment step between a set of keyframes.

VI. Conclusion

In this paper, we presented a low-cost approach enabling cinematic VR experiences with motion parallax from a single monoscopic omnidirectional image. Specifically, we presented a method enabling the generation of a 360-degree depth map by projection on 3D primitives manually placed in VR by a human operator, and demonstrated the generation of a scene mesh from the omnidirectional color-depth image pair. We validated the strength of this scene representation in terms of viewing comfort by means of a comparative user study. We believe our approach can be used by content creators of all kinds to enhance novel or existing omnidirectional images for viewing in virtual reality with motion parallax.

References

[1] J. Yu, "A light-field journey to virtual reality," *IEEE MultiMedia*, vol. 24, no. 2, pp. 104–112, 2017.

[2] "Welcome to light fields on steam." [Online]. Available: http://store.steampowered.com/app/771310/Welcome_to_Light_Fields/

[3] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz, "Jump: virtual reality video," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 198, 2016.

[4] K. Matzen, M. F. Cohen, B. Evans, J. Kopf, and R. Szeliski, "Low-cost 360 stereo photography and video capture," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 148, 2017.

[5] P. Hedman, S. Alsisan, R. Szeliski, and J. Kopf, "Casual 3d photography," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 234, 2017.

[6] B. J. Fernandez-Palacios, D. Morabito, and F. Remondino, "Access to complex reality-based 3d models using virtual reality solutions," *Journal of cultural heritage*, vol. 23, pp. 40–48, 2017.

[7] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou *et al.*, "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 741–754.

[8] T. Ebner, I. Feldmann, S. Renault, O. Schreer, and P. Eisert, "Multi-view reconstruction of dynamic real-world objects and their integration in augmented and virtual reality applications," *Journal of the Society for Information Display*, vol. 25, no. 3, pp. 151–157, 2017.

[9] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-dof vr videos with a single 360-camera," in *Virtual Reality (VR), 2017 IEEE*. IEEE, 2017, pp. 37–44.

[10] S. Pathak, A. Moro, H. Fujii, A. Yamashita, and H. Asama, "Virtual reality with motion parallax by dense optical flow-based depth generation from two spherical images," in *System Integration (SII), 2017 IEEE/SICE International Symposium on*. IEEE, 2017, pp. 887–892.

[11] J. Thatte, J.-B. Boin, H. Lakshman, and B. Girod, "Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[12] J. Thatte, T. Lian, B. Wandell, and B. Girod, "Stacked omnistereo for virtual reality with six degrees of freedom," in *Visual Communications and Image Processing (VCIP), 2017 IEEE*. IEEE, 2017, pp. 1–4.

[13] A. Smolic, P. Kauff, S. Knorr, A. Hornung, M. Kunter, M. Muller, and M. Lang, "Three-dimensional video postproduction and processing," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 607–625, 2011.

[14] R. Phan and D. Androutsos, "Robust semi-automatic depth map generation in unconstrained images and video sequences for 2d to stereoscopic 3d conversion," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 122–136, 2014.

[15] H. Yuan, S. Wu, P. An, C. Tong, Y. Zheng, S. Bao, and Y. Zhang, "Robust semiautomatic 2d-to-3d conversion with welsch m-estimator for data fidelity," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[16] Q. Zeng, W. Chen, H. Wang, C. Tu, D. Cohen-Or, D. Lischinski, and B. Chen, "Hallucinating stereoscopy from a single image," in *Computer Graphics Forum*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 1–12.

[17] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[18] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[19] T. Kanade, P. Rander, and P. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE multimedia*, vol. 4, no. 1, pp. 34–47, 1997.

[20] M. Slater and M. Garau, "The use of questionnaire data in presence studies: do not seriously likert," *Presence: Teleoperators and Virtual Environments*, vol. 16, no. 4, pp. 447–456, 2007.