PanoTrace: Interactive 3D Modeling of Surround-View Panoramic Images in Virtual Reality

Ehsan Sayyad
Department of Media Arts and Technology
University of California Santa Barbara
Santa Barbara, CA 93106
ehsan@mat.ucsb.edu

Pradeep Sen
Department of Electrical and Computer Engineering
University of California Santa Barbara
Santa Barbara, CA 93106
psen@ece.ucsb.edu

Tobias Höllerer
Department of Computer Science
University of California Santa Barbara
Santa Barbara, CA 93106
holl@cs.ucsb.edu

ABSTRACT

Full-surround panoramic imagery can provide a viewer with a high-resolution visual impression of a pictured real or realistically rendered environment, but it does not provide as high a level of immersion as modeled 3D geometry can, when viewed with virtual reality (VR) headsets or projection-based setups. In this paper, we demonstrate that augmenting panorama images with geometrical models can be done simply in VR itself and can significantly increase the feeling of immersion a viewer experiences. We propose a novel interactive modeling tool that allows users to model geometry depicted in a surround-panoramic scene directly in VR, utilizing projection mapping of the panorama on top of the evolving geometry. The user interface is intuitive and allows novice users to produce geometry that approximates ground truth models sufficiently to enhance a user's VR viewing experience. We designed a user study that compares users' self-reported levels of immersion, scene realism, and discomfort on a set of created models and comparison cases. Our results indicate that our modeled scenes produce a significantly higher sense of immersion than a basic dome geometry for the panorama when viewed in VR with head orientation and position tracking.

CCS CONCEPTS

•**Human-centered computing** →**Virtual reality;** *User studies;*

KEYWORDS

Virtual reality, panorama imaging, geometric modeling, VR modeling tools

ACM Reference format:
Ehsan Sayyad, Pradeep Sen, and Tobias Höllerer. 2017. PanoTrace: Interactive 3D Modeling of Surround-View Panoramic Images in Virtual Reality. In *Proceedings of VRST '17, Gothenburg, Sweden, November 8–10, 2017,* 10 pages.
DOI: 10.1145/3139131.3139158

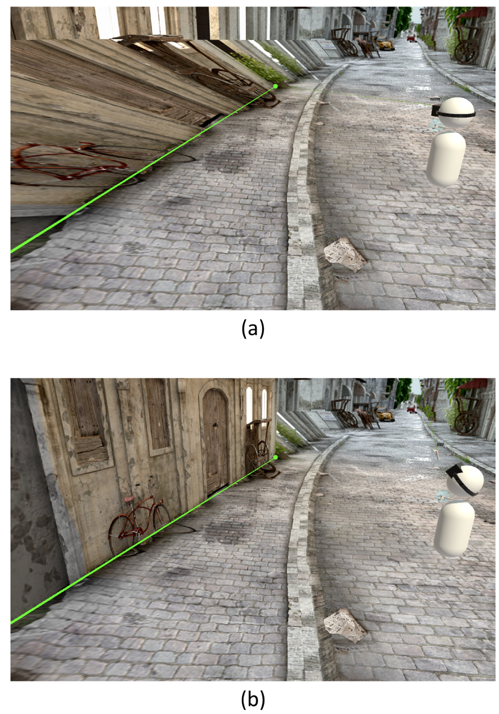


Figure 1: Example modeling action while experiencing a panorama in 6DoF-tracked VR. We see the user represented as an avatar from the side, a view that is only used for illustration purposes. The user is presented with a view corresponding with the avatar's head pose. The user has just placed a simple ground plane and now observes the panorama projected onto this plane. (a) User traces a wall line on the reference plane; (b) User has extruded the wall.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
VRST '17, Gothenburg, Sweden
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-5548-3/17/11…$15.00
DOI: 10.1145/3139131.3139158

1 INTRODUCTION

Panoramic images are ubiquitous today. The desire to capture and share visual experiences has led to the development of many kinds of imaging techniques and devices for the acquisition of visual realities, and surround-scene panoramic images in particular capture the experience of being "present" in a scene much better than just single points of view.

The capture of panoramas can either be done by taking multiple images with a standard, hand-held camera and then stitching them together with software techniques [DiVerdi et al. 2008; Szeliski and Shum 1997; Xiong and Pulli 2010], or directly with specialized panoramic cameras [Barber 2017; Gledhill et al. 2003; Gurrieri and Dubois 2013; Nokia Ozo 2017; Weissig et al. 2012]. Furthermore, many tools and interfaces have been developed to explore and navigate image panoramas, such as QuickTime VR [Chen 1995], YouTube Virtual Reality [YouTube VR 2017], Cardboard Camera VR [Google Cardboard VR 2017], or Facebook 360 [Facebook 2017]. The sense of presence that panorama images provide along with the abundance of capture and visualization tools for them has resulted in a plethora of online repositories for panoramic images produced by users all around the world.

Recently, there has also been a flurry of renewed interest on immersive display technologies such as virtual reality (VR), thanks to the introduction of mass-market systems such as the Oculus Rift, HTC Vive, and Sony PlayStation VR. However, despite their full-surround nature and sense of presence they provide, 2D panoramas have serious limitations when experienced in VR. 2D panoramic imagery simply lacks geometrical depth information necessary for viewing with binocular disparity (for stereo rendering) or for navigation of the scene freely in six degrees of freedom, orientation and position (for motion parallax). When viewing a 2D panorama using VR, the user gets the feeling they are trapped, unable to move inside of a large "bubble" texture-mapped with the panorama.

This problem can obviously be addressed by augmenting the panorama with geometrical data. The idea of augmenting images with geometry to enable free viewing, or providing a sufficient number of alternative viewpoints to enable light field perception, goes back to the beginnings of image-based rendering [Debevec et al. 1996; Gortler et al. 1996; Levoy and Hanrahan 1996] Later, researchers also explored the specific problem of generating geometry for panoramic images. For example, Oh et al. [Oh et al. 2001] showed how image-based modeling for panoramic images would result in an aesthetically pleasing mock-up of the environment. Unfortunately, many of the methods for augmenting panoramas with geometry usually need expensive or complicated setups at acquisition time [Huang and Klette 2010; Peleg et al. 2001].

Other approaches require capturing panoramas at multiple locations in order to employ structure-from-motion techniques to reconstruct the environment [Kwiatek and Tokarczyk 2015].There are also ways to capture depth information by capturing multiple images and using optical flow [Anderson et al. 2016; Peleg et al. 2001]. This often involves special-purpose and often expensive camera rigs and setups and in the end the geometry produced is often affected by noise. We are interested in the problem of generating depth information, or, even better, full 3D geometry, for 2D panoramas that have already been captured with simpler imaging solutions. There have been some great automatic approaches to generate 3D model information from a single image or a small set of images; Some have been successful in generating a 3D context for a single panorama using a well-trained Support Vector Machine [Zhang et al. 2014]. This would suit specific kinds of environments but can be unsuccessful with a broader range of scenarios as you need to train the system using ground truth depth information and manual object annotation.

Finally, it is possible to manually model the scene using traditional 3D modeling tools. However, these tools are usually complex and require a skill set and deep understanding of 3D environments. This makes it difficult for novice users to complete the modeling task.

We observe that VR itself could be a better environment to create 3D models of 2D panoramas. After all, since VR is a natural medium for *viewing* immersive scenes, we hypothesize that it would also be the natural medium for *creating* immersive scene content. Therefore, in this paper we designed and implemented a complete toolset for modeling panorama geometry directly in VR. We use projection mapping as way of tracing 3D geometry to make it easier for novice users to create complex 3D environments.

We designed a user study to examine the effect of augmenting panoramic images with geometrical models on a user's sense of immersion, realism and discomfort. We compared the geometry created with our tool against other approaches, by exposing users to panoramas rendered with various forms of underlying geometry including: no geometry (infinitely-large sphere, which is what the default panorama would be), a simple hemispherical dome, our modeled scenes with the novel VR toolset, and the ground-truth geometry available from rendered scenes and Matterport [Matterport 2017] captures. Our results show that in cases with man-made environments, our modeled scenes had a significantly higher sense of immersion and realism than basic geometries such as the dome. They also tended to cause lower discomfort for the user in those cases.

The study results demonstrate that viewing a 2D panorama (with additional model geometry) using 6DoF (orientation and position) head tracking, can significantly increase the feeling of immersion a viewer experiences. Using our toolset, novice users can model, in as little as 20 minutes, simple scene geometry that leads to a superior 6DoF viewing experience compared to projection onto a skydome and about halfway as effective in terms of perceived immersion and scene realism as ground truth geometry models.

Previous Work

Image based modeling. Debevec [Debevec et al. 1996] showed how by using a sparse set of images and a calibrated camera some basic geometries can be constructed to match the image and generate the shape of buildings. Horry [Horry et al. 1997] presented a way to add depth to a single image based on indicating the vanishing points. Criminisi [Criminisi et al. 1999] later showed how by tracing parallel lines in an image, one can calculate the camera position and reconstruct the 3D model of the image. Zhang et al. [Zhang et al. 2002] demonstrated a way to generate the 3D model of the scene from a single image, based on a sparse set of user-specified constrains on the scene. Van den Hengel et al. [van den Hengel et al.

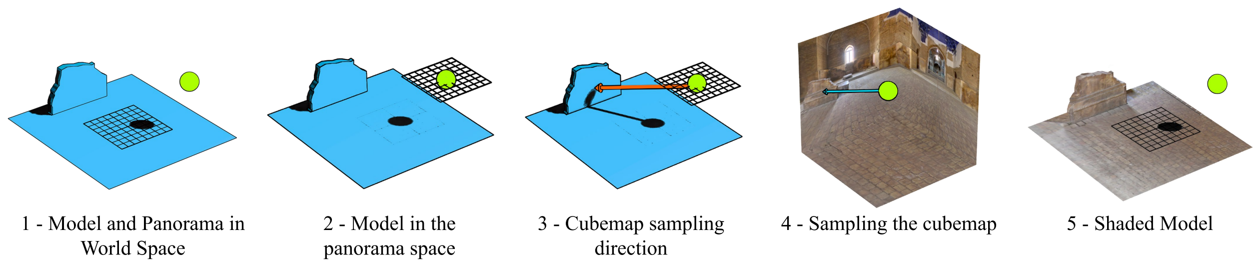


Figure 2: This image shows how the projection mapping works in the shader.

2007] introduced a system to use images in a video as a reference to create complex 3D geometry while tracking camera movement. While these influential works all present methods to interactively create 3D geometry from 2D imagery, none of these operate specifically on surround-view panoramas, and none of these methods utilize interactive VR technologies for modeling.

3D modeling in VR. There have been several early works regarding 3D modeling in VR [Butterworth et al. 1992; Deering 1995; Liang and Green 1994; Whyte et al. 2000] and two-handed interaction in virtual environments [LaViola et al. 2017; Mapes and Moshell 1995]. More recently, Jackson [Jackson and Keefe 2016] generated a creative toolset that lets the user trace curves from images to create 3D objects. There is also a number of recent commercially available 3D modeling tools for VR [Google Blocks 2017; Oculus Medium 2017]. However, to our knowledge no one has yet addressed the problem of modeling geometry in existing 2D image panoramas using VR tools.

Contributions

We posit that a 3D modeling interface in VR can be a very effective option to manually augment 2D panoramas with depth information. In order to create a useful toolset for novice users, we implemented some novel interactions and features (such as our 3D texture snapping, and some 3d interactions as parts of our bi-manual transformation interface). We also performed a user study and the analysis of the results revealed valuable information regarding the effect of depth information on a user's sense of immersion, realism, and discomfort.

2 SYSTEM OVERVIEW

In this project, we have developed a complete interactive system to model panoramas in virtual reality. In the subsections that follow, we shall describe the different components of our system.

2.1 Architecture

The system is implemented using the Unity game engine, selected for its flexibility and general adoption. We have used SteamVR and the "OpenCV for Unity" plugins to control the HTC Vive device and use OpenCV functionality in the Unity environment. The program has been implemented in C# based on the existing Unity classes and data structures.

Data Structure. We used equirectangular Panorama image files as the input. These files are being converted to OpenGL/ DirectX cubemap textures in Unity. The geometry is also being handled by Unity's Mesh class which contains Vertex, Normal, UV and Triangle arrays. In our rendering system UV and Normal arrays are not being used.

Rendering. All the model geometries are rendered using a cubemap projection shader. A Projector object sends the panorama's transformation matrix $T \times R \times S$ to the shader. In the vertex stage we multiply each vertex's world position by the panorama matrix to find the relative vertex position. In the fragment stage, we look up each fragment's color by sampling the cubemap using the normalized relative fragment position (see Figure 2). Shading all the geometries with existing panorama data causes images to duplicate on the occluded surfaces (see Figure 3). This may cause confusion for the user. In order to diminish this effect, we implemented a shadow map algorithm that treats the projector as an omnidirectional light source and avoids shading the occluded fragments. We experimented with different ways of in-painting, i.e. replacing the occluded pixels, and provided a MIP mapping level control, giving the user an adjustment slider to choose a level that fits the scene.

User Interface. The user input is handled by a SteamVR plugin. We used an HTC Vive with six-degree-of-freedom, room-scale tracking. It consists of a head mounted display and two controllers. A layer of user input handling was implemented on top of the raw input from SteamVR. The aim of the UI design was to create a 3D modeling interface for novice users. We focused on avoiding complexity in order to flatten the learning curve for the toolset. We developed a dynamic Pie menu system (see Figure 4) to benefit from the touch interface of the Vive controllers. We were able to select from up to 10 radial menu items effectively. However, for our final design, we limited the menu items to 5. Each menu item could trigger an action or lead to a sub-menu. Users can explore the menu system linearly (they can press the back button to return to the parent menu). Using the HTC Vive tracking functionality, users can move controllers and walk in the space. We also implemented basic 3D user interactions in the form of aiming and selecting, grabbing, shaking, and bimanual interactions.

2.2 Modeling

In order to model a panorama that corresponds to the ground-truth depth, first the transformation of the panoramic camera should

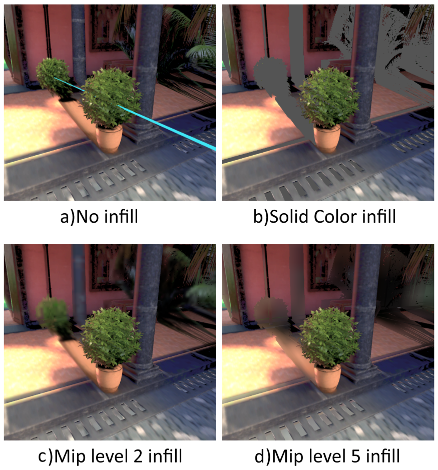


Figure 3: Texture inpainting: The blue ray indicates the direction from the projection center. Note how the colors are being repeated on all the surfaces.

Figure 4: Our pie menu with some 3D geometry to showcase the functionality of each control. The menu appears above the touchpad on the controller.

be known. Unlike normal photographs, panoramas are by default considered to be equivalent to 6 pinpoint cameras with 90° field of view. Therefore, there is no need to calibrate for lens distortion. However, if a panorama is not aligned correctly, the importer needs to know the quaternion representing the panoramic camera's orientation. Users can use existing techniques to realign panoramas before introducing them to the system. A panorama's height can be adjusted while inserting a reference plane. The rest of the modeling, like finding the location of the walls will proceed based on the visual feedback on the reference plane (see Figure 1).

2.2.1 *Traditional modeling tools.*

3D brush tool: the brush tool draws geometry in a freehand style. This gives the user the ability to model more complex and organic shapes (see part (d) in Figure 5)

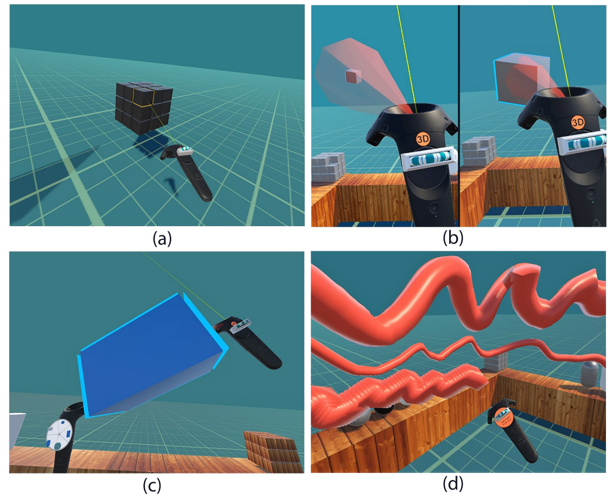


Figure 5: (a) Object selection with laser pointer. (b) Bringing objects closer using a wheel. (c) Freeform bi-manual transformation of a cube. (d) Brush tool with tube and cube strokes with varying brush sizes.

3D bimanual transformation tool: Users can point to and select objects in 3D. Objects can be moved and rotated with one hand. We apply the selecting controller's transformation to the object that we are displacing. Users can also move, rotate and scale the object using two controllers. We generate a transformation matrix that rotates, moves and scales based on the position and rotation of the two controllers in each frame. This results in a widget-less direct manipulation tool that is easy to learn for the novice users. A duplicate button is also placed on the controller to make a copy of the created shapes (see parts (a),(b),(c) in Figure 5)

Extrusion and vertex editing tool: The Extrusion tool works by creating a ground plane and extruding it to make walls. A vertex editing tool is provided to adjust the vertices to their correct locations. As the user points the tool to surface geometry, we cast a ray and find the contact point. The user can choose multiple points to create a polygon and extrude the polygon using the controller (see Figure 1)

Navigation: Our system offers different ways of navigation. The first one is the natural six-degree-of-freedom movement that the VR tracking provides. Users can walk and turn as long as they don't leave the Vive's tracking area. We also give the user the ability to fly around using a fly button. This is useful for users that don't experience motion sickness in VR. We also provide a World in Miniature [Stoakley et al. 1995] experience, which gives the user the ability to resize themselves to have better access to areas that are otherwise hard to reach or to change their precision by focusing on specific points.

2.2.2 *Novel modeling tools.*

Panoramic image snapping: We implemented an image snapping algorithm [Gleicher 1995] to help the user snap the pointer to points of interest on the panorama. We first create a Canny Edge cubemap

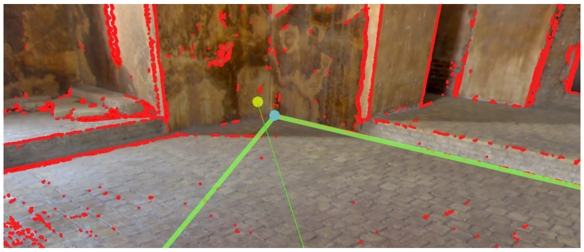


Figure 6: Red lines showing the Canny edges. Laser pointer snaps to the lines to help the user aim.

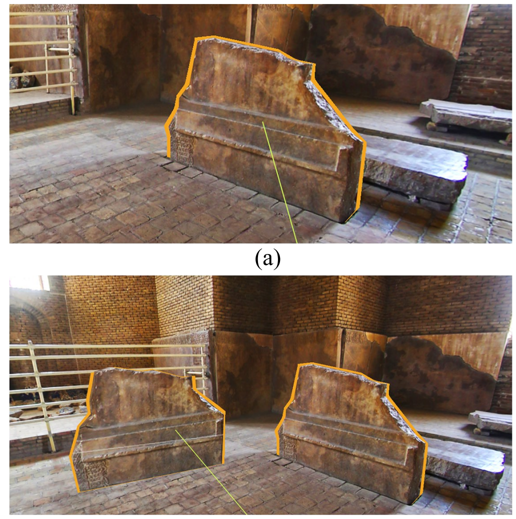
for the panorama that we are modeling. Then, in each frame, we cast a ray to the geometry and find the contact point. Using the direction from the projection center to the contact point, we find the corresponding cubemap face and pixel. We apply the technique to find the snapping point. Based on the snapping point and the cubemap face, we regenerate a new ray as the snapping pointer ray. Canny edge detection could produce some undesired lines that do not represent change in geometry (such as shadow lines). However user can disable the snapping on the areas with large amount of false edge detection. (see Figure 6).

Texture baking and object duplication: A realtime-shader-based texture baking procedure was developed to support object duplication functionality. We generate an unwrapped UV set for the object and render it to a separate buffer using UV coordinates as vertex positions. Then we save the buffer as a texture and assign it to an unlit textured shader and render the object using that shader. Baked textures will be undistorted if they are projected on the correct corresponding 3D geometry. This makes it possible to extract textures for later use directly from the panorama (see Figure 7).

Depth information refinement: Depth data calculated from structure-from-motion or stereo-reconstruction techniques tend to be very noisy. We demonstrate a use case for our modeling tool as a refining and healing tool for existing depth information. This could also benefit cases of missing geometric information; We applied our tool to stereo depth-map panoramas from the Nokia OZO camera [Nokia Ozo 2017]. This high-end stereo real-time panoramic video camera specifically targets VR but it does not provide perfect stereo imagery: for example, it produces stereo imagery only for a field of view of +/−130°(h), +/−65° (v). The rest of the surround field of regard does not have depth information. On these areas, and for noise within the existing depth map, missing depth can be replaced with user generated 3D geometry or corrected with healing tools (see Figure 8).

3 MODELING RESULTS

To test our system we decided on a diverse set of representative panoramas for which some sort of ground truth depth information was obtainable. Ground truth 3D models of panoramic scenes are generally hard to come by, as most panorama capturing techniques won't provide noise-free depth information. Therefore, we decided to include computer-generated photorealistic-looking scenes, as those contain the actual ground-truth geometry implicitly. We



(a)

(b)

Figure 7: (a) 3D geometry of the object is created and the object is shaded using cubemap projection. (b) Pixels are baked onto a texture and object is duplicated.

(a) (b) (c)

Figure 8: (a) Error in depth information retrieved from the Nokia OZO panoramic camera is visible on the books. (b) User is refining the depth using the brush tool. (c) The result (just the stack of books was modified)

created several surround environments from commercially available raytracing resources and rendered them using a bidirectional pathtracer. We also decided to compare our method with scenes we captured with the Matterport 3D capturing system [Matterport 2017]. We selected 6 representative panoramas (3 synthetic outdoor scenes and 3 Matterport-modeled indoor scenes) for our test set. An expert user worked with PanoTrace on 2D panoramas for each of these scenes, for a maximum of 20 minutes to create PanoTrace scene models, not using any ground truth information, just the plain 2D panorama for each scene, and our system.

The Matterport capturing system is much more suitable for indoor scenes than for outdoor scenes and high frequency depth information is challenging for it. We decided to dedicate the rendered scenes to outdoor environments and high-frequency details such as plants and foliage. All 3 Matterport scenes cover indoor

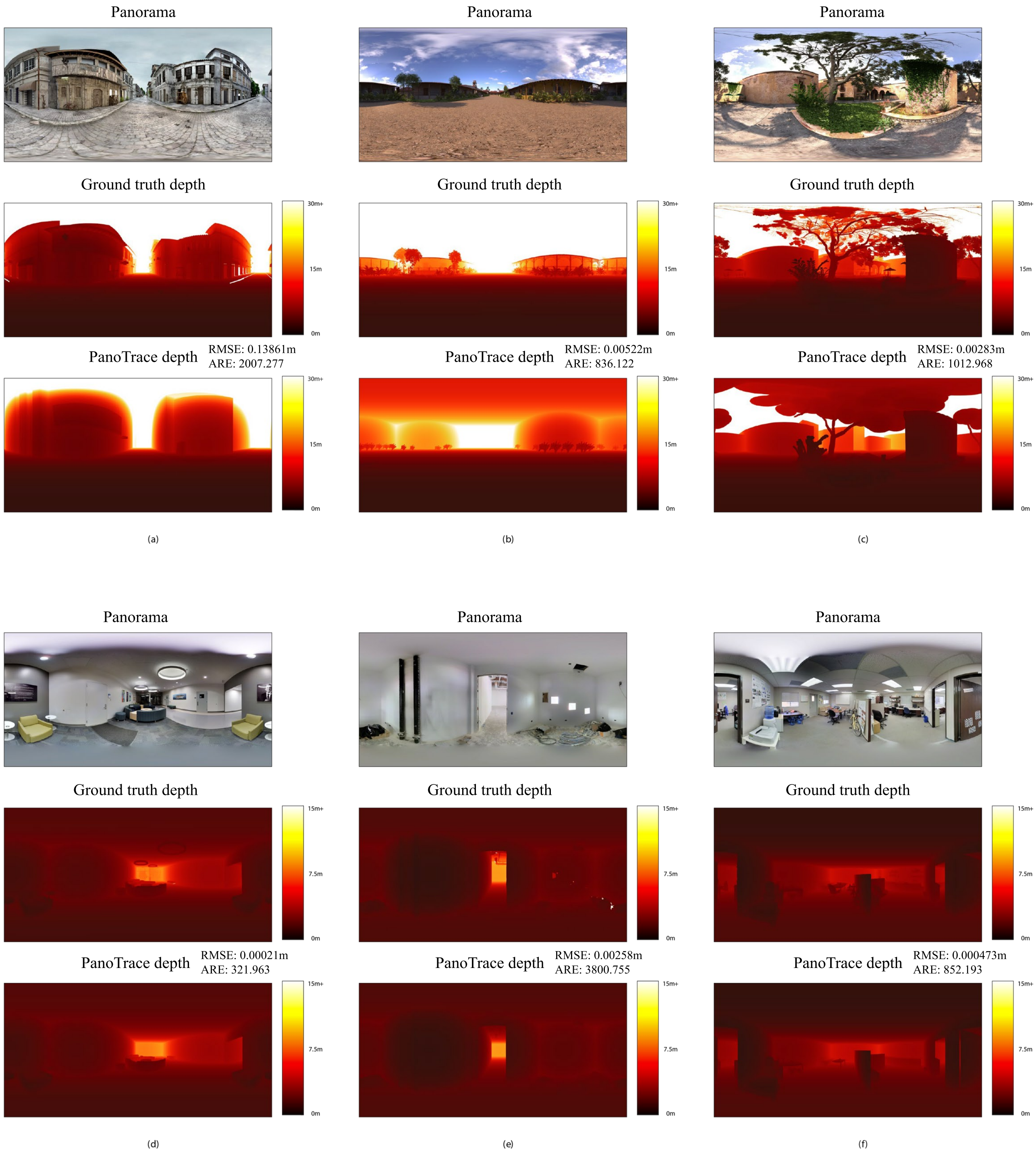


Figure 9: Panoramas along with the ground-truth depth map and the depth map resulting from the expert user's attempt to model the scene with PanoTrace. (a),(b) and (c) are synthetic rendered scenes and (d),(e), and (f) are real scenes captured using the Matterport system. RMSE and ARE errors are provided for each PanoTrace model

environments. Overall, this selection approach ensured variability of test cases in our user study and lets us compare the success of our system in modeling different types of environments.

Figure 9 shows the results of the PanoTrace modeling efforts as equirectangular projections for each panorama, PanoTrace-created depth map, and ground truth depth map. It also lists two common depth image difference metrics [Cadena et al. 2016] for each pair of PanoTrace and ground truth depth images: Absolute Relative Error [Saxena et al. 2009] and RMSE [Li et al. 2010]. The smaller these values, the more similar the PanoTrace panorama models were to ground truth. Several observations can be made here:

In terms of modeling error as compared to ground truth for each depth map stemming from the 20-minute PanoTrace session our reasonably experienced modeler spent on each panorama, the indoor Matterport scenes generally are better than the synthetic outdoor scenes (with the exception of panorama e), which is a panorama of a more confined space than the others, and the ARE metric biases against that to a certain extent - perceptually, the modeled depth and the ground truth are still very close there). The example of panorama c) demonstrates that natural geometry such as tree branches and foliage are difficult to model in a short amount of time, and panorama b) exemplifies a case where visually apparent differences don't factor too badly into the metrics because relevant geometry occurs generally at large distances.

4 USER STUDY

A user study was conducted to compare user experiences with different geometrical representations of the same panoramas, including the results of our PanoTrace modeling system. Panoramas can be presented on VR headsets in different ways. In all cases, the user's head was orientation-tracked, so that he or she could look around naturally in the surround panorama. Head position tracking did only have an effect in our 6DoF conditions (see Study Setup below for conditions). When geometry (a simple skydome, our PanoTrace model, or a ground truth model) was present and 6DoF tracking mode was on, the user could move the head around, and even take a few steps, to perceive the scene with motion parallax. In that case, textures behind objects were simply duplicated (cf. Figure 3a)). We were interested in the question of what advantages a 3D panorama might have over a 2D panorama in terms of perceived realism, immersion, and viewing comfort, and how far a PanoTrace model with a modeling time limit of 20 minutes can get you towards a detailed 3D panorama. In this section, we discuss the details of this study.

4.1 Participants

We iteratively designed the study by first performing pilot studies with 3 users. Then, for the actual study, a total of 11 participants were recruited, ages 18 to 32 years old (average 23.2), 7 male and 4 female. Participants had either normal (7 users) or corrected vision (4 users). We did not have participants with colorblindness or stereoblindness, as determined by standard tests. Of the 11 users, 8 reported themselves as having "only tried out VR a few times," 2 said they were not familiar with VR at all, and 1 user said they "frequently used VR." Each user was compensated with $10 US for the 1.5 hours they spent on the study.

4.2 Study Setup

To do the study, 6 panoramas were chosen, 3 of them synthetic and rendered using a path-tracer, and 3 of them captured with a Matterport 3D capturing system. Each panorama was presented with 7 different conditions (3DoF and 6DoF represent 3 and 6 degrees of freedom, respectively):

(1) 2D: Plain 2D panorama without being projected to 3D geometry (equivalent to an infinitely large sphere, only viewed in 3DoF).
(2) Dome 3DoF: panorama projected on a user-adjusted hemisphere geometry, viewed with 3DoF head orientation tracking.
(3) Dome 6DoF: panorama projected on a user-adjusted hemisphere geometry, viewed with 6DoF head-tracking.
(4) PanoTrace model 6DoF: panorama projected on the model that an expert user prepared in 20 minutes, viewed with 6DoF head-tracking.
(5) PanoTrace model 3DoF: panorama projected on the model that an expert user prepared in 20 minutes, viewed with 3DoF head orientation tracking.
(6) Ground truth with 6DoF: panorama projected on the ground truth geometry, viewed with 6DoF head-tracking.
(7) Ground truth with 3DoF: panorama projected on the ground truth geometry, viewed with 3DoF head orientation tracking.

The user-study framework was developed using the Unity engine. [Unity - Game Engine 2017]. The study system presents the panoramas in random order on an HTC Vive VR head-mounted display (HMD). For each panorama, the user would experience 2 repetitions of each condition, so that we could check on the consistency of responses.

4.3 Task

The 7 conditions listed above with 2 repetitions each resulted in 14 test scenes for each panorama. Each user was shown scenes 1–14 in two cycles. In the first cycle, the user would be given an overview of all 14 scenes, and then in the second cycle they would answer questions about each. Users could move on to the next scene by pushing a button on their on their controller. The scene number would show up on their controller. After experiencing each scene in the second cycle, users had to answer these 3 questions on a Likert scale:

- "How strongly did you feel like actually being in the scene?" (immersion)
- "How realistic was the experience of the scene?" (realism)
- "How much discomfort did you experience while viewing the scene?" (discomfort)

The questions would pop up on the display after the users chose to proceed from the scene.

4.4 Experimental Design

We used a within-subjects design with 3 dependent variables (immersion, realism, discomfort) and 2 independent variables (6 panoramas, 7 conditions). We really considered four different conditions (baseline 2D, Skydome, Model from PanoTrace, and ground truth)

Table 1: Wilcoxon signed-rank test on the dep. variables by DOF. Listed are the Likert question means, critical z, and p values. Scenes were perceived as more immersive, more realistic, and less discomfort-inducing with 6DoF viewing.

Dep. Variable	M 3DoF	M 6DoF	z	p
Immersion	4.39	4.81	5.532	<.0005
Realism	4.42	4.56	3.007	0.003
Discomfort	2.38	1.97	-6.929	<.0005

and two different degrees of freedom for viewing (3DoF and 6DoF) but since it doesn't make sense to experience the plain 2D panorama in 6DoF, we simply enumerated the seven resulting conditions. The order of independent variables were defined randomly for each user.

Our hypotheses about the outcome of the study were as follows:

H1: Users will experience more immersion, more realism, and less discomfort in expert-user-modeled scenes (using our tool) compared to the less detailed dome scene.

H2: Users will not experience more immersion, more realism, and less discomfort in the ground truth model scenes compared to the expert-user-modeled scenes using our tool.

4.5 Procedure

Before the study could begin, each participant was tested for color-blindess using Ishiahara color plates, and for stereoblindness using a VR random dot stereogram. Then they filled out the pre-study questionnaire with their demographics and some background information. Next, the user was asked to stand in the tracking area while wearing the HMD. The test administrator explained how to operate the system and how many scenes are they going to explore, but they did not recommend any option. Also, users were not asked to pay attention to the differences between the scenes. They were told that they could have some limited movement in the tracking space (e.g., walking a few steps).

After finishing the study, users could choose to interact with the modeling system in the remaining time. Administrator explained how to toolset would work and how to use the tooltips to learn the interactions in our tool. Later, users could choose to model one of our modeled scenes from scratch.

4.6 User Study Results

After experiencing each scene, participants were asked to rate the experience based on the amount of immersion, realism and discomfort that they felt, on 7-point Likert scales. The average values are shown in Figure 10.

We performed a Wilcoxon signed-rank test on the dependent variables to check for any statistically significant difference between 3DOF and 6DOF. All three variables, Immersion, Realism, and Discomfort, showed to be significantly affected by DOF (see Table 1), with 6DoF viewing leading to higher reported immersion and realism, and lower discomfort levels.

For each dependent variable, a Friedman test was run to determine if there were differences in participant responses using any of the different 3D Model categories (2D, Dome, PanoTrace, and Ground Truth). Pairwise comparisons were performed with Bonferroni correction for multiple comparisons.

Main Findings. The main findings relating to the performance of our PanoTrace models in terms of perceived immersion, realism, and viewing comfort among the 6DoF viewing conditions can be summarized as follows:

- PanoTrace models and Ground Truth models provided a greater sense of immersion than Dome models when viewed with 6DoF head tracking.
- Both PanoTrace models and Dome models provided a lower sense of realism than Ground Truth models overall across all six panoramas when viewed with 6DoF head tracking.
- For indoor Matterport models, only Dome provided a lower sense of realism than Ground Truth.

Additionally, we observed the interesting effect among the 3DoF viewing conditions that the Dome condition resulted in lower viewing discomfort than the plain 2D panorama condition.

We categorize all detailed results as follows:

6DOF Immersion. Immersion values were significantly different using different model categories, $\tilde{\chi}^2(2) = 31.311$, $p < .0005$. Post-hoc analysis revealed statistically significant differences in Immersion from Dome (Mean = 4.44) to PanoTrace (mean = 4.80) (p = 0.027) and Dome (mean = 4.44) (p = .027) to Ground Truth (mean = 5.19), but not between Ground Truth and PanoTrace models.

This supports both hypothesis H1 and H2.

6DOF Realism. Realism values were significantly different using different model categories, $\tilde{\chi}^2(2) = 21.919$, $p < .0005$. Post-hoc analysis revealed statistically significant differences in Realism from Ground Truth (mean = 4.98) to Dome (mean = 4.24) ($p < 0.0005$) and from Ground Truth (mean = 4.98) (p = .049) to PanoTrace, but not between the Dome and PanoTrace models.

This supports hypothesis H1 but does not support hypothesis H2.

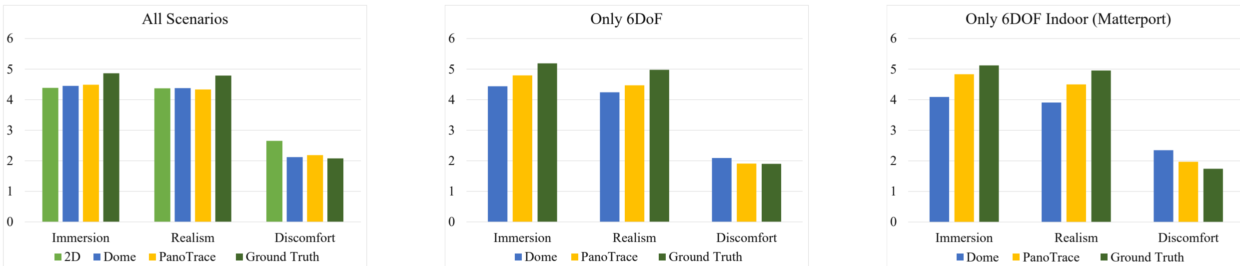
6DOF Discomfort. Discomfort values were significantly different using different model categories, $\tilde{\chi}^2(2) = 31.311$, p = .004 according to the Friedman test, with discomfort highest for the Dome geometry. However, a Bonferroni post-hoc analysis did not reveal statistically significant differences in discomfort from different models.

Based on these results we decided to narrow down the data-set in one more step. We filtered out the panoramas that were computer-generated and focused on the indoor panoramas that were captured by Matterport.

6DOF - Indoor - Immersion. Immersion values were significantly different using different model categories, $\tilde{\chi}^2(2) = 17.495$, p ¡ .0005. Post-hoc analysis showed statistically significant differences in Immersion from Dome (mean = 4.09) to PanoTrace (mean = 4.83) (p = 0.027) and Dome (mean = 4.09) (p = .001) to Ground Truth (mean = 5.12), but not between the Ground Truth and PanoTrace models.

This supports both hypothesis H1 and H2.

6DOF - Indoor - Realism. Realism values were significantly different using different model categories, $\tilde{\chi}^2(2) = 13.624$, p = 0.001.



(a) **(b)** **(c)**

Figure 10: User responses for Immersion, Realism, and Discomfort. Note that only (a) includes a plain 2D panorama condition. (b) and (c) are showcasing results under the 6DOF-head-motion condition, which is not meaningful for 2D panoramas

Post-hoc analysis revealed statistically significant differences in Realism from Ground Truth (mean = 4.92) to Dome (mean = 3.91) (p = 0.006) but not between the Ground Truth and PanoTrace (mean = 4.50) models.

This supports hypothesis H2 but does not support hypothesis H1.

6DOF - Indoor - Discomfort. Immersion values were significantly different using different model categories, $\tilde{\chi}^2(2) = 16.775$, $p < .0005$. Post-hoc analysis showed statistically significant differences in Immersion from Ground Truth (mean = 1.74) to Dome (mean = 2.35) (p = 0.008) but not between the Ground Truth and PanoTrace (mean = 1.97) models.

This supports hypothesis H2 but does not support hypothesis H1.

3DOF Immersion, Realism and Discomfort. We performed the Friedman test using the results from all 3DOF conditions. The test showed significant differences between the models for all 3 dependent variables. A set of Bonferroni post-hoc tests only showed a significant decrease (p = 0.037) of discomfort for dome (mean = 2.15) as compared to 2D (mean = 2.65).

5 DISCUSSION

We expected the result for our expert users's PanoTrace models to fall in between the ground truth and our very basic 3D model, the dome, in many of the scenarios. Across all datasets, we could not find a statistically significant difference between PanoTrace models and the dome. However, for 6DoF viewing, this was the case. The mean for all three dependent variables for our method was always in between the ground truth and the dome and there was significance for immersion between Dome and PanoTrace but not between PanoTrace and Ground Truth.

Most of the results that were aligned with our hypotheses were about the participants' sense of immersion. This may mean that even a modestly realistic environment can feel more immersive if it somehow provides a proper response to the user's movements.

In the 6DoF cases, the realism perceived from PanoTrace models was significantly less than the ground truth. This could be related to the fact that our tool is not really strong at modeling organic and very detailed geometry, such as trees and foliage. This was a motivation for us to also analyze the data regarding indoor Matterport scenes separately.

On the indoor Matterport scenes, our method is shown to be very effective. This could be due to three factors. First, Matterport scenes do not have many organic objects, so it was easier to for the expert user to create a representation of the scene. Second, Matterport models are not the actual ground truth, as the system has artifacts and is an approximation of the true geometry. Third, the Matterport scenes all include geometry close to the user. The skydome approach would not naturally do well with such scenes.

The very significant difference between 6DoF and 3DoF shows the importance of the movement for the users, since users were told they could move if they wished. Many of them did try to walk in the environments, and reported feeling dizzy when the panorama was 'moving with them' as happens with 3DoF viewing.

There appears to be a benefit of using simple dome geometry in the 3DoF viewing case in terms of discomfort, i.e. the Dome condition had lower discomfort than the plain 2D condition. This means that surround panoramic content in VR would likely benefit from this simple viewing adjustment.

6 CONCLUSION AND FUTURE WORK

We designed and developed a system to model 2D panoramas interactively in Virtual Reality, featuring several novel interactions. We demonstrated the tool by modeling several scenes, both reconstructed physical spaces, and virtual computer graphics scenes, for which ground truth geometry was accessible. Then we designed a user study in which participants experienced different versions of six surround-view panoramas and reported their sense of immersion, scene realism, and discomfort. An analysis of the collected data provided statistical evidence on the significance of our tool.

In the end our tool proves to be very effective in certain areas, such as modeling man-made indoor scenes (rooms and furniture) and still of relevance for complex natural outdoor scenes.

However there are a lot of expansions that would seem appropriate for this work. A more automated approach involving additional computer vision constraints would be a worthwhile extension of this system. One might utilize previous existing machine learning [Hoiem et al. 2005; Saxena et al. 2005] or computer vision approaches [Cherian et al. 2009] to generate 3D geometry from a single image. Also, using interactive lighting control could be an effective way of increasing the interactivity, user engagement and immersion. Implementing a relighting algorithm for the projective texture mapping would also benefit this system.

ACKNOWLEDGMENTS

Support for this work is provided by Office of Naval Research grant number N00014-16-1-3002. We thank Ali Sayyad for help with the statistical analysis.

REFERENCES

Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. 2016. Jump: virtual reality video. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 198.

Nick Barber. 2017. Start Caring About VR And 360-Degree Video. (2017). https://www.forbes.com/sites/forrester/2017/01/10/start-caring-about-vr-and-360-degree-video/ accessed on July 1, 2017.

Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc. T. Olano. 1992. 3DM: A Three Dimensional Modeler Using a Head-mounted Display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (I3D '92)*. ACM, New York, NY, USA, 135–138. https://doi.org/10.1145/147156.147182

Cesar Cadena, Yasir Latif, and Ian D. Reid. 2016. Measuring the performance of single image depth estimation methods. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4150–4157. https://doi.org/10.1109/IROS.2016.7759611

Shenchang Eric Chen. 1995. QuickTime VR: An Image-based Approach to Virtual Environment Navigation. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 29–38. https://doi.org/10.1145/218380.218395

Anoop Cherian, Vassilios Morellas, and Nikolaos Papanikolopoulos. 2009. Accurate 3D ground plane estimation from a single image. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2243–2249.

Antonio Criminisi, Ian Reid, and Andrew Zisserman. 1999. Single view metrology. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Vol. 1. IEEE, 434–441.

Paul E Debevec, Camillo J Taylor, and Jitendra Malik. 1996. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 11–20.

Michael F. Deering. 1995. HoloSketch: A Virtual Reality Sketching/Animation Tool. *ACM Trans. Comput.-Hum. Interact.* 2, 3 (Sept. 1995), 220–238. https://doi.org/10.1145/210079.210087

Stephen DiVerdi, Jason Wither, and Tobias Hollerer. 2008. Envisor: Online environment map construction for mixed reality. In *Virtual Reality Conference, 2008. VR'08. IEEE*. IEEE, 19–26.

Facebook 2017. 360 Video. (2017). https://facebook360.fb.com/

Duke Gledhill, Gui Yun Tian, Dave Taylor, and David Clarke. 2003. Panoramic imagingfi!?a review. *Computers & Graphics* 27, 3 (jun 2003), 435–445. https://doi.org/10.1016/S0097-8493(03)00038-4

Michael Gleicher. 1995. Image Snapping. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 183–190. https://doi.org/10.1145/218380.218441

Google Blocks 2017. Create 3D models in VR - Google VR. (2017). https://vr.google.com/blocks/

Google Cardboard VR 2017. Cardboard Camera VR Photo Format. (2017). https://developers.google.com/vr/concepts/cardboard-camera-vr-photo-format

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*. ACM Press, New York, New York, USA, 43–54. https://doi.org/10.1145/237170.237200

Luis E. Gurrieri and Eric Dubois. 2013. Stereoscopic cameras for the real-time acquisition of panoramic 3D images and videos, Andrew J. Woods, Nicolas S. Holliman, and Gregg E. Favalora (Eds.). International Society for Optics and Photonics, 86481W. https://doi.org/10.1117/12.2002129

D. Hoiem, A. A. Efros, and M. Hebert. 2005. Geometric context from a single image. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 1. 654–661 Vol. 1. https://doi.org/10.1109/ICCV.2005.107

Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 225–232.

Fay Huang and Reinhard Klette. 2010. Stereo panorama acquisition and automatic image disparity adjustment for stereoscopic visualization. *Multimedia Tools and Applications* 47, 3 (2010), 353–377. https://doi.org/10.1007/s11042-009-0328-2

Bret Jackson and Daniel F. Keefe. 2016. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Transactions on Visualization and Computer Graphics* 22, 4 (April 2016), 1442–1451. https://doi.org/10.1109/TVCG.2016.2518099

Karol Kwiatek and Regina Tokarczyk. 2015. Immersive photogrammetry in 3D modelling. *Geomatics and Environmental Engineering* 9, 2 (2015).

J.J. LaViola, E. Kruijff, D.A. Bowman, R.P. McMahan, and I.P. Poupyrev. 2017. *3D User Interfaces: Theory and Practice*. Pearson Education, Limited. https://books.google.com/books?id=ilUyjwEACAAJ

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*. ACM Press, New York, New York, USA, 31–42. https://doi.org/10.1145/237170.237199

Congcong Li, Adarsh Kowdle, Ashutosh Saxena, and Tsuhan Chen. 2010. Towards holistic scene understanding: feedback enabled cascaded classification models. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 1351–1359. http://dl.acm.org/citation.cfm?id=2997047

Jiandong Liang and Mark Green. 1994. JDCAD: A highly interactive 3D modeling system. *Computers & graphics* 18, 4 (1994), 499–506.

Daniel P. Mapes and J. Michael Moshell. 1995. A Two-handed Interface for Object Manipulation in Virtual Environments. *Presence: Teleoper. Virtual Environ.* 4, 4 (Jan. 1995), 403–416. https://doi.org/10.1162/pres.1995.4.4.403

Matterport 2017. Immersive 3D for the Real World. (2017). https://matterport.com/

Nokia Ozo 2017. 360 virtual reality camera. (2017). https://ozo.nokia.com/

Oculus Medium 2017. v. 1.1.2. (2017). https://www.oculus.com/experiences/rift/1336762299669605/

Byong Mok Oh, Max Chen, Julie Dorsey, and Frédo Durand. 2001. Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 433–442.

Shmuel Peleg, Moshe Ben-Ezra, and Yael Pritch. 2001. Omnistereo: Panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 3 (2001), 279–290.

Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. 2005. Learning depth from single monocular images. In *NIPS*, Vol. 18. 1–8.

A. Saxena, Min Sun, and A.Y. Ng. 2009. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 5 (may 2009), 824–840. https://doi.org/10.1109/TPAMI.2008.132

Richard Stoakley, Matthew J Conway, and Randy Pausch. 1995. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 265–272.

Richard Szeliski and Heung-Yeung Shum. 1997. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 251–258.

Unity - Game Engine 2017. v. 2017.1. (2017). https://unity3d.com/

Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. 2007. VideoTrace: Rapid Interactive Scene Modelling from Video. *ACM Trans. Graph.* 26, 3, Article 86 (July 2007). https://doi.org/10.1145/1276377.1276485

Christian Weissig, Oliver Schreer, Peter Eisert, and Peter Kauff. 2012. The Ultimate Immersive Experience: Panoramic 3D Video Acquisition. Springer, Berlin, Heidelberg, 671–681. https://doi.org/10.1007/978-3-642-27355-1_72

Jennifer Whyte, N Bouchlaghem, A Thorpe, and R McCaffer. 2000. From CAD to virtual reality: modelling approaches, data exchange and interactive 3D building design tools. *Automation in construction* 10, 1 (2000), 43–55.

Yingen Xiong and Kari Pulli. 2010. Fast panorama stitching for high-quality panoramic images on mobile phones. *IEEE Transactions on Consumer Electronics* 56, 2 (2010).

YouTube VR 2017. Virtual Reality Videos. (2017). https://support.google.com/youtube/answer/6316263?hl=en

Li Zhang, Guillaume Dugas-Phocion, Jean-Sebastien Samson, and Steven M Seitz. 2002. Single-view modelling of free-form scenes. *Computer Animation and Virtual Worlds* 13, 4 (2002), 225–235.

Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. 2014. PanoContext: A Whole-Room 3D Context Model for Panoramic Scene Understanding. In *ECCV: European Conference on Computer Vision*. 668–686. https://doi.org/10.1007/978-3-319-10599-4_43