

LEARNING TO COMPOSE 6-DOF OMNIDIRECTIONAL VIDEOS USING MULTI-SPHERE IMAGES

Jisheng Li*, Yuze He*, Yubin Hu*, Yuxing Han[†], Jiangtao Wen^{†‡}

*Tsinghua University, Beijing, China

[†]Research Institute of Tsinghua University in Shenzhen, Shenzhen, China

ABSTRACT

Omnidirectional video is an essential component of Virtual Reality. Although various methods have been proposed to generate content that can be viewed with six degrees of freedom (6-DoF), existing systems usually involve complex depth estimation, image inpainting or stitching pre-processing. In this paper, we propose a system that uses a 3D ConvNet to generate a multi-sphere images (MSI) representation that can be experienced in 6-DoF VR. The system utilizes conventional omnidirectional VR camera footage directly without the need for a depth map or segmentation mask, thereby significantly simplifying the overall complexity of the 6-DoF omnidirectional video composition. By using a newly designed weighted sphere sweep volume (WSSV) fusing technique, our approach is compatible with most panoramic VR camera setups. A ground truth generation approach for high-quality artifact-free 6-DoF contents is proposed and can be used by the research and development community for 6-DoF content generation.

Index Terms— Omnidirectional video composition, multi-sphere images, 6-DoF VR

1. INTRODUCTION

As a result of the rapidly increasing popularity of omnidirectional videos on video-sharing platforms such as YouTube and Veer, professional content producers are striving to produce better viewing experiences with higher resolution, new interaction mechanisms, as well as more degrees of viewing freedom, using playback platforms such as the Google *Welcome to Light Field*, where the viewer has the freedom to move along three axes with motion parallax. A body of research has been dedicated to composing 6-DoF from footage captured by VR cameras, extending toolsets for the professional content generators to produce more immersive contents. However, most of the proposed systems involve complex procedures such as depth estimation and in-painting, that are both time- and resource- consuming and require tedious hand-optimizations.

Recently, Attal *et al.* proposed MatryODShka [1], which used a convolutional neural network to predict multi-sphere images (MSIs) that can be viewed in 6-DoF. This approach significantly simplified the overall pipeline complexity while producing promising visual results. However, the system requires omnidirectional stereo (ODS) inputs that cannot be acquired directly from widely available VR cameras, as ODS must be produced from raw panoramic VR footage using stitching with annoying visual artifacts. In addition, as in the case for any system design, 6-DoF content production also requires

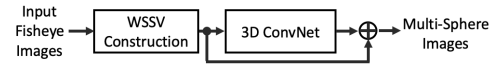


Fig. 1: Our proposed 6-DoF omnidirectional video composition framework.

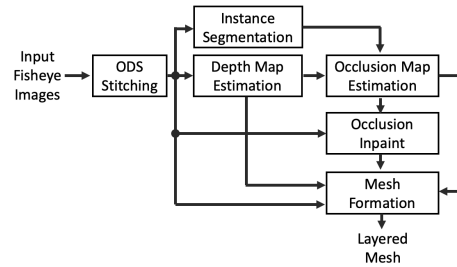


Fig. 2: An example of conventional 6-DoF content generation method

a large volume of high-quality content that can serve as the ground-truth, for both training purposes and performance evaluation. However, because 6-DoF contents are difficult to produce, there is no widely available high-quality 6-DoF content that can be used as a ground-truth dataset for the community.

The contribution of this paper is therefore two-fold: First, we propose a system that can produce large volumes of high-quality artifact-free 6-DoF data for training and performance evaluation that serves as the basis for evaluation of the system proposed in this paper, as well as for future development by the community. Second, we propose an algorithm for predicting MSI from VR camera footage directly, using a modified weighted sphere sweep volume fusing scheme with 3D ConvNet without explicit depth map or segmentation mask, thereby improving quality while lowering complexity.

2. RELATED WORKS

Substantial research has been carried out on the reconstruction and representation of omnidirectional contents [2][3][4][5][6][7]. In this section, we focus on recent research aimed at 6-DoF reconstruction and view synthesis.

Six degrees of freedom content generation requires detailed scene depth information. Dynamic 3D reconstruction and content playback have been extensively studied in the context of free-viewpoint video, with many approaches achieving real-time performance [8][9]. Using the conventional multi-view stereo method, Google *Welcome to Light field* [10] and Facebook manifold system [11] both achieved realistic high-quality 6-DoF content composition, with hardware systems that are substantially more complex

[‡] Denotes corresponding author.

We acknowledge funding from the Shenzhen Science and Technology Program (Grant No.KQTD20190929172829742).

than VR cameras widely available on the market. Lately, studies that used convolutional neural networking (CNN) show promising results for depth estimation and view synthesis [12][13][14]. CNN achieves excellent results for predicting multi-plane images (MPI) and representing the non-Lambertian reflectance [15][16][17][18][19]. Many recent approaches adopt and extend these studies to generate omnidirectional contents [1][20][21]. Broxton *et al.* [21] designed a half-sphere camera rig with GoPro cameras and used the method in [16] to generate pieces of MPIs and later converted it into 360 layered mesh representation for viewing. Lin *et al.* [20] generated multiple MPIs and formed them into a multi-depth panorama using the MPI predicting techniques in [17]. Lai *et al.* [22] proposed to generate panorama depth map for ODS images 6-DoF synthesization. More recently, MartyODShka [1] archived real time performance when using the method in [15] to synthesize MSI from ODS content. In spite of such successes, ODS images relaid by these approaches still require stitching of original VR camera footage, which by itself is still a problem that has not been sufficiently solved. Many existing neural network based approaches were also designed with a fixed number of input images that is not configurable.

In this paper, we propose an approach without stitching pre-processing or depth map estimation. In our approach, a weighted sphere sweep volume is fused directly from camera footage using spherical projection, eliminating artifacts introduced by stitching. By utilizing 3D ConvNet, our framework is applicable to different VR camera designs with different numbers of MSI layers. We also propose a high-quality 6-DoF dataset generation approach using UnrealCV [23] and Facebook Replica [24] engine, so that our, and future systems for 6-DoF content composition can be designed and evaluated qualitatively and quantitatively using our data generation scheme.

3. OMNIDIRECTIONAL 6-DOF DATA GENERATION

Commercially available VR camera models provide a range of selection for omnidirectional content acquisition. A great amount of footage from these VR cameras is captured by professional photographers and enthusiasts. However, it is hard to render a ground truth from these footage, as content captured by such cameras need to be stitched, which by itself is still a problem that has not been completely solved. As a result, it has been extremely difficult to find artifact-free high-quality 360 video datasets in this literature, which is required for the continued development and evaluation of VR and/or 6-DoF content. The few datasets produced by companies like Google or Facebook are limited in size, content scenarios, resolutions etc., as they were constrained by the various factors imposed by the time and equipment used for producing such datasets.

It is therefore highly desirable to be able to generate any number of test clips of any use cases and different resolutions and to continue this test data generation process as new cameras with higher resolutions become available. In this study, we use two CG rendering frameworks UnrealCV [23] and Replica [24] to generate high-quality VR datasets for both training and evaluation. The UnrealCV engine can render realistic images with lighting changes and reflections on handcrafted models. On the other hand, Replica engine uses indoor reconstruction models, and produces rendered images with a similar distribution to real-world textures and dynamic ranges. We use the combination of two datasets to mimic real-world challenges.

To compose an omnidirectional image using the above mentioned CG engines, we first generate six 120° pin-hole images with different poses towards the engine's x-y-z axes and their inverse directions. The optical centers of these virtual cameras are located at

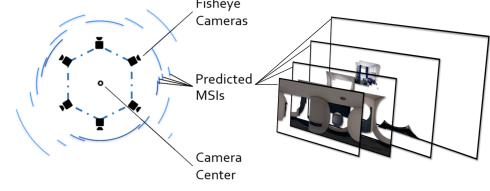


Fig. 3: Illustration of an MSI representation. Blue lines show the opacity at different area of the concentric spheres.

the same point, to avoid any parallax between cameras. We then project six pin-hole images to equirectangular projection (ERP) and blend the overlapping area to avoid aliasing. To simulate camera footage from a n-sensor VR camera, we render ERP footage on the poses of each sensor and mask off the external details according to the lens field of view. In our experiments, we selected 2000 locations to generate 6-DoF contents for network training and evaluation. The generated datasets were split into training subsets of 1600 locations and evaluation subsets with 400 locations. The split was implemented based on virtual camera locations with no overlap across the training and evaluation subsets. Contents generated from UnrealCV and Replica were mixed together during training procedure but evaluated separately. We rendered two resolutions (640×320 and 400×200) on each location for training with different numbers of MSI layers due to GPU memory constraints. Upon that, we rendered two fields of view (190° and 220°) to simulate fisheye images from different VR camera modules.

4. A SIMPLIFIED SYSTEM FOR PRODUCING 6-DOF CONTENT FROM PANORAMIC VR FOOTAGE

Our process of turning panoramic VR camera footage into 6-DoF playable multi-sphere images involves three steps as shown in Fig. 1. We first project the fisheye images onto concentric spheres using the weighted sphere sweep method and generate weighted sphere sweep volume (WSSV) as described in Sec. 4.2. This 4D volume is input into the 3D ConvNet detailed in Sec. 4.3 to predict the α channel. We combine the predicted α channel with WSSV to form the MSI representation. Later inside the render engine, the MSI is calculated following Equ. 2 to infer per-eye views in VR.

4.1. Multi-Sphere Images

Inspired by multi-plane image (MPI) and its performance in view synthesis applications, multi-sphere images (MSIs) is generated for omnidirectional content representation. Following the design of MPI, the MSI representation used in our work consists of N images to represent N concentric spheres. These concentric spheres are warped into planes using equirectangular projection (ERP) for efficient processing. Each image inside an MSI contains 3 channels of color and an additional α channel of transparency information. This character inherited from MPI empowers the MSI to represent scene occlusion and non-Lambertian reflectance. The form of $RGB\alpha$ data representation also allows MSIs to be compressed using standard image compression algorithms.

The differentiable rendering scheme for the MPI also applies on the MSI. As described in Equ. 1, in MPI differentiable rendering, the position $\{p_i\}_{i=1}^N$ where a ray intersects with each layer $\{L_i\}_{i=1}^N$ is calculated and RGB value $\{c_{p_i}\}_{i=1}^N$ and α value $\{\alpha_{p_i}\}_{i=1}^N$ are interpolated and then used to calculate the output color c

$$c = \sum_{i=1}^N c_{p_i} \cdot \alpha_{p_i} \cdot \prod_{j=1}^{i-1} (1 - \alpha_{p_j}). \quad (1)$$

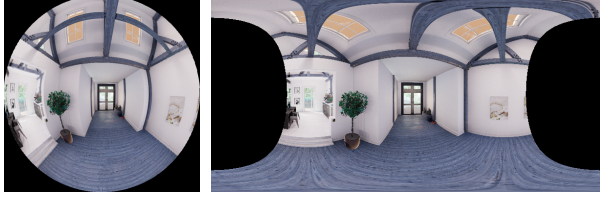


Fig. 4: Left: A fisheye image with 220° FOV. Right: A fisheye image reprojected to ERP format

The rendering procedure for viewing MSI in 6-DoF is similar, as described in Equ. 2. To render a ray inside of the inmost sphere of MSI with viewing angle (θ, ϕ) from camera location (x, y, z) , the intersection point s_i with i -th layers of MSI is determined by ray-plane intersection function $q(\cdot)$ in graphic engines, where d_i is the radius of i -th sphere. The color of the rendered ray $c_{x,y,z,\theta,\phi}$ is computed by color values $\{c_{s_i}\}_{i=1}^N$ and transparency values $\{\alpha_{s_i}\}_{i=1}^N$ in each layer following:

$$s_i = q(x, y, z, \theta, \phi, d_i)$$

$$c_{x,y,z,\theta,\phi} = \sum_{i=1}^N c_{s_i} \cdot \alpha_{s_i} \cdot \prod_{j=1}^{i-1} (1 - \alpha_{s_j}). \quad (2)$$

This procedure can be carried out efficiently on a common CG engine such as Unity or OpenGL.

4.2. Weighted Sphere Sweep Volume Construction

A weighted sphere sweep volume is constructed with N layers of concentric spheres. The radius of these spheres is chosen uniformly in the reciprocal space between the closest object distance and the farthest distance. As images from VR cameras are often captured using fisheye lenses that compresses the field of view and causes chroma aberration on the edges of images, we propose a weighted sphere sweep method to reduce such optical defects.

To this end, we first warp these input fisheye images into the ERP form as shown in Fig. 4. Then M input ERP images are projected onto N layers of sphere in ERP form using the intrinsic and extrinsic parameters of the camera. In each layer, we fuse M input images together on the overlapping area following :

$$c_p = \sum_{i=1}^Q \frac{e^{\gamma_{p,i}} \cdot c_{p,i}}{\sum_{j=1}^Q e^{\gamma_{p,j}}}, \quad (3)$$

where Q is the number of overlapping images at position p , and $c_{p,i}$ is the color of the i -th image on position p . The parameter γ is the optical distortion value. Here we use $\gamma_p = 1 - r_p$, where r_p is the distance of pixel p to the optical center normalized to $[0, 1]$. We can also use the lens MTF data to replace γ values. Then, N projected ERP images are stacked to form a 4D volume so that the first three dimensions are ERP image height (H), width (W) and number of layers (N), while the 4-th dimension is the color channel. As each ERP image contains 3 channels of RGB color, the constructed WSSV has a shape of $[H, W, N, 3]$. The WSSV construction can be represented by Equ. 4, where the weighted and warp function $\mathcal{W}(\cdot)$ takes in a camera pose $\{\omega_j\}_{j=1}^M$ and the camera center pose ω_c , then projects an input image from $\{I_j\}_{j=1}^M$ to a set of concentric spheres with pre-defined radius $\{d_i\}_{i=1}^N$. The $\mathcal{S}(\cdot)$ function stacks the warped results along 4-th dimension to form the weighted sphere sweep volume:

$$WSSV = \mathcal{S}_{j=1}^N (\mathcal{W}_{i=1}^M (I_i, \omega_i, \omega_c, d_j)). \quad (4)$$

Layer	s	d	n	depth	in	out	input
conv1_1	1	1	8	32/32	1	1	WSSV
conv1_2	2	1	16	32/16	1	2	conv1_1
conv2_1	1	1	16	16/16	2	2	conv1_2
conv2_2	2	1	32	16/8	2	4	conv2_1
conv3_1	1	1	32	8/8	4	4	conv2_2
conv3_2	1	1	32	8/8	4	4	conv3_1
conv3_3	2	1	64	8/4	4	8	conv3_2
conv4_1	1	2	64	4/4	8	8	conv3_3
conv4_2	1	2	64	4/4	8	8	conv4_1
conv4_3	1	2	64	4/4	8	8	conv4_2
nnup_5				4/8	8	4	conv3_3+conv4_3
conv5_1	1	1	32	8/8	4	4	nnup_5
conv5_2	1	1	32	8/8	4	4	conv5_1
conv5_3	1	1	32	8/8	4	4	conv5_2
nnup_6				8/16	4	2	conv2_2+conv5_3
conv6_1	1	1	16	16/16	2	2	nnup_6
conv6_2	1	1	16	16/16	2	2	conv6_1
nnup_7				16/32	2	1	conv1_2+conv6_2
conv7_1	1	1	8	32/32	1	1	nnup_7
conv7_2	1	1	8	32/32	1	1	conv7_1
conv7_3	1	1	1	32/32	1	1	conv7_2

Table 1: Our proposed network architecture. The size of all convolution kernels are set to 3. Column **s,d,n** denote the stride, kernel dilation and number of output filters respectively. **depth** column shows the number of layer input/output depth (we use 32 input depth layers as an example), while **in** and **out** are the accumulated stride of layer input/output. **input** names with + mean concatenation along the depth dimension and **Layer** names starting with “nnup” perform 2x nearest neighbor upsampling.

As each layer of sphere sweep volume is formed by multiple input images projected onto the same sphere, the combination of these images is similar to the result of light field refocusing. Because multiple images of the same object are projected to a layer whose radius approximately equals to the distance of that object, the average color of such a combination of the projected images will appear in-focus. The 3D ConvNet in our framework is trained to distinguish such properties and predict the transparency values, which are further combined with WSSV to generate MSI.

4.3. Network Architecture

Our 3D ConvNet architecture is a slight variation of Middle *et al.*’s work [17]. In our design, we modify the network to suit the WSSV input described in Sec. 4.2.

The output vector of the neural network has a shape of $[H, W, D, 1]$, which is converted into the α channel of an MSI by a ReLU function. This α channel is then stacked with WSSV RGB volume to form the MSI. As color information of an MSI comes directly from WSSV, which is the combination of projected input footage, during training, the network is not learning to simulate the correct color but to select the correct layer by inferring the corresponding weight of the α channel to imply an object’s distance.

The training objective is :

$$MSI_\theta = f_\theta(WSSV)$$

$$\mathcal{L} = \mathcal{L}_{L1} + \lambda_{VGG} \mathcal{L}_{VGG}$$

$$\text{argmin}_\theta \sum_{i=1}^M \mathcal{L}(\text{Render}(MSI_\theta, \omega_c, \omega_i), \mathcal{W}_{ERP}(I_i)), \quad (5)$$

where the goal is to minimize the difference between the rendered MSI result and ground truth. In Equ. 5 the 3D ConvNet $f_\theta(\cdot)$ takes

N	Dataset	Resolution	PSNR	SSIM
N=32	UnrealCV	400 × 200	28.28±2.45	0.90±0.03
		640 × 320	28.79±2.40	0.90±0.03
N=64		400 × 200	28.52±2.51	0.90±0.03
		640 × 320	29.23±2.43	0.91±0.03
N=32	Replica	400 × 200	33.48±3.72	0.95±0.03
		640 × 320	33.40±3.87	0.95±0.03
N=64		400 × 200	33.59±3.74	0.95±0.03
		640 × 320	33.80±3.92	0.95±0.03

Table 2: Evaluation results. We examine our framework on the combination of two different number of MSI layers (N=32 and N=64), two input resolution (400×200 and 640×320) and two data generation engine (UnrealCV and Replica).

in $WSSV$ and predicts MSI_{θ} . The $Render(\cdot)$ function follows the differentiable rendering scheme in Equ. 1, where the $\mathcal{W}_{ERP}(\cdot)$ is the fisheye to ERP warp function. The training loss $\mathcal{L}(\cdot)$ is a weighted combination of L1 loss $\mathcal{L}_{L1}(\cdot)$ and VGG loss $\mathcal{L}_{VGG}(\cdot)$ proposed in [25]. We refer readers to Tab. 1 for detailed network architecture.

5. EXPERIMENTS

We trained and evaluated our network performance by generating MSI representations and rendering them in different input positions and computing peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) quality scores against ground truth generated using our proposed approach in Sec 3. In this section, we describe network implementation details, and experiment results.

5.1. Implementation Details

We implement our proposed 3D ConvNet using TensorFlow API following the description in Tab. 1. During training, Automatic Mixed Precision feature was applied to utilize GPU memory by using half precision (FP16) in computation. The network was optimized with an SGD optimizer with the learning rate set to 2×10^{-4} . We employed the VGG loss [25] as the perceptual loss with weight $\lambda_{VGG} = 200$. Using the data generation method described in Sec. 3, we synthesized a set of 6-sensors VR camera footage on 2000 locations and masked the field of view to 190° and 220° randomly. The entire dataset was split into 1600 locations for training subsets and the rest for evaluation. The network was trained on 640×320 resolution with 32 layers of MSI and 400×200 resolution with 64 layers of MSI simultaneously for 400k iterations on an Nvidia RTX 2080Ti GPU.

5.2. Evaluation

We examined the performance of our approach with the evaluation subsets that contains 400 locations with two different resolutions and corresponding color ground truth. We generated MSIs on evaluation subsets using our proposed method and rendered each MSI to novel viewpoints using input poses, then computed PSNR and SSIM of these novel views with the ground truth color. These quantitative results are reported in Tab. 2 and some selected qualitative results are shown in Fig 5.

As demonstrated in Tab. 2, our system can generate high-quality 6-DoF contents from VR camera footage. Comparing with the



Fig. 5: Illustration of system input and output. Left column: Input fisheye images. Right column: Render result from MSI predicted by our system.

ground truth color, results rendered on input views achieve an average PSNR over 31 dB. After carefully examining the quantitative results in Tab. 2, we notice a minor performance difference between two datasets. A small variance between two render engines could be the cause, as UnrealCV engine renders reflectance that varies from different angles while Replica uses static reconstructed object texture acquired from real-world scenes. Overall, results rendered from MSIs show plausible visual details on complex textures like leaves, books, floor textures as shown in Fig. 5. We also notice there is a performance gain by increasing the number of layers in MSI. As each MSI is a set of concentric spheres, a denser set of spheres can represent more detailed depth variances among scene objects. In our experiments, we also confirmed that our network can be trained and utilized among variable image resolution and number of layers of an MSI. We used 640×320 with 32-layer and 400×200 with 64-layer configuration on training and evaluated the network on the combination of both resolution and numbers of layers. Experiment results show that the network is not overfitting on one particular configuration.

6. CONCLUSIONS

In this paper, we present an end-to-end deep-learning framework to compose 6-DoF omnidirectional with multi-sphere images. We use weighted sphere sweep volume to unify inputs from various panoramic camera setups into one constant volume size, solving the compatibility issue in previous work. Combined with our proposed 3D ConvNet architecture, we can process camera footage directly and reduce the systematic artifacts introduced in the ODS stitching process. We propose a high-quality 6-DoF dataset generation method using UnrealCV and Facebook Replica engines for training and quantitative performance evaluations. A series of experiments were conducted to verify our system. Experiment results show our system can operate on variable image resolution and MSI layers, as well as producing high-quality novel views that contain correct occlusion and detailed textures.

7. REFERENCES

- [1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin, “Matryodshka: Real-time 6dof video view synthesis using multi-sphere images,” in *European Conference on Computer Vision*. Springer, 2020, pp. 441–459.
- [2] Venkata N Peri and Shree K Nayar, “Generation of perspective and panoramic video from omnidirectional video,” in *Proc. DARPA Image Understanding Workshop*. Citeseer, 1997, vol. 1, pp. 243–245.
- [3] Richard Szeliski, “Image alignment and stitching: A tutorial,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.
- [4] Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz, “Jump: virtual reality video,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–13, 2016.
- [5] Minhao Tang, Jiangtao Wen, Yu Zhang, Jiawen Gu, Philip Junker, Bichuan Guo, Guansyun Jhao, Ziyu Zhu, and Yuxing Han, “A universal optical flow based real-time low-latency omnidirectional stereo video system,” *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 957–972, 2018.
- [6] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt, “Omniphotos: casual 360° vr photography,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–12, 2020.
- [7] Jisheng Li, Ziyu Wen, Sihan Li, Yikai Zhao, Bichuan Guo, and Jiangtao Wen, “Novel tile segmentation scheme for omnidirectional video,” in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 370–374.
- [8] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan, “High-quality streamable free-viewpoint video,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–13, 2015.
- [9] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al., “Fusion4d: Real-time performance capture of challenging scenes,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–13, 2016.
- [10] Ryan S Overbeck, Daniel Erickson, Daniel Evangelakos, and Paul Debevec, “Welcome to light fields,” in *ACM SIGGRAPH 2018 Virtual, Augmented, and Mixed Reality*, pp. 1–1. 2018.
- [11] Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyce Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral, “An integrated 6dof video camera and system design,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–16, 2019.
- [12] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.
- [13] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi, “Learning-based view synthesis for light field cameras,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–10, 2016.
- [14] Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng, “Learning to synthesize a 4d rgb light field from a single image,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2243–2251.
- [15] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” in *SIGGRAPH*, 2018.
- [16] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker, “Deepview: View synthesis with learned gradient descent,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2367–2376.
- [17] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar, “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.
- [18] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely, “Pushing the boundaries of view extrapolation with multiplane images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 175–184.
- [19] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz, “Extreme view synthesis,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7781–7790.
- [20] Kai-En Lin, Zexiang Xu, Ben Mildenhall, Pratul P. Srinivasan, Yannick Hold-Geoffroy, Stephen DiVerdi, Qi Sun, Kalyan Sunkavalli, and Ravi Ramamoorthi, “Deep multi depth panoramas for view synthesis,” in *European Conference on Computer Vision (ECCV)*. 2020, vol. 12358, pp. 328–344, Springer.
- [21] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec, “Immersive light field video with a layered mesh representation,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 86–1, 2020.
- [22] Po Kong Lai, Shuang Xie, Jochen Lang, and Robert Laquarère, “Real-time panoramic depth maps from omni-directional stereo images for 6 dof videos in virtual reality,” in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2019, pp. 405–412.
- [23] Weichao Qiu and Alan Yuille, “Unrealcv: Connecting computer vision to unreal engine,” in *European Conference on Computer Vision*. Springer, 2016, pp. 909–916.
- [24] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al., “The replica dataset: A digital replica of indoor spaces,” *arXiv preprint arXiv:1906.05797*, 2019.
- [25] Qifeng Chen and Vladlen Koltun, “Photographic image synthesis with cascaded refinement networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1511–1520.