LFSphereNet: Real Time Spherical Light Field Reconstruction from a Single Omnidirectional Image

Manu Gond
Mid Sweden University
Sundsvall, Sweden
manu.gond@miun.se

Emin Zerman
Mid Sweden University
Sundsvall, Sweden
emin.zerman@miun.se

Sebastian Knorr
Ernst-Abbe University of Applied Sciences
Jena, Germany
sebastian.knorr@eah-jena.de

Mårten Sjöström
Mid Sweden University
Sundsvall, Sweden
marten.sjostrom@miun.se

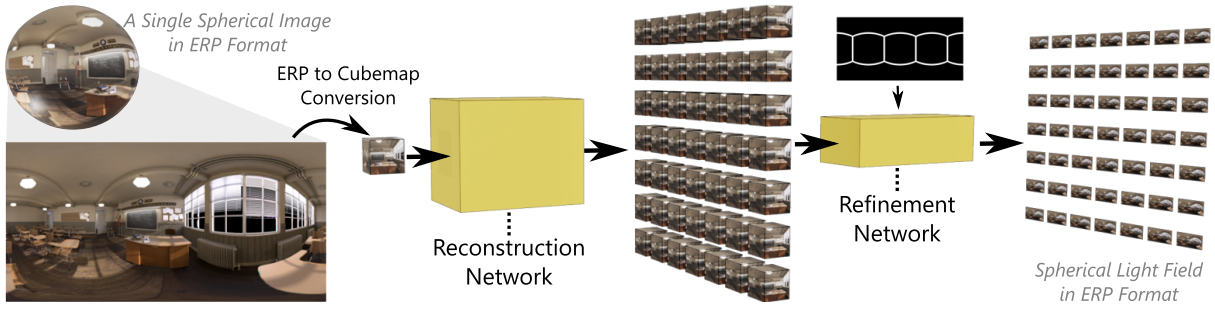

Figure 1: LFSphereNet consists of a network for cubemap light field reconstruction and a refinement network for cube borders.

ABSTRACT

Recent developments in immersive imaging technologies have enabled improved telepresence applications. Being fully matured in the commercial sense, omnidirectional (360-degree) content provides full vision around the camera with three degrees of freedom (3DoF). Considering the applications in real-time immersive telepresence, this paper investigates how a single omnidirectional image (ODI) can be used to extend 3DoF to 6DoF. To achieve this, we propose a fully learning-based method for spherical light field reconstruction from a single omnidirectional image. The proposed LFSphereNet utilizes two different networks: The first network learns to reconstruct the light field in cubemap projection (CMP) format given the six cube faces of an omnidirectional image and the corresponding cube face positions as input. The cubemap format implies a linear re-projection, which is more appropriate for a neural network. The second network refines the reconstructed cubemaps in equirectangular projection (ERP) format by removing cubemap border artifacts. The network learns the geometric features implicitly for both translation and zooming when an appropriate cost function is employed. Furthermore, it runs with very low inference time, which enables real-time applications. We demonstrate that LFSphereNet outperforms state-of-the-art approaches in terms of quality and speed when tested on different synthetic and real world scenes. The proposed method represents a significant step towards achieving real-time immersive remote telepresence experiences.

CVMP '23, November 30–December 01, 2023, London, United Kingdom

ACM ISBN 979-8-4007-0426-0/23/11.
https://doi.org/10.1145/3626495.3626500

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; **Computer graphics**.

KEYWORDS

Light Field, View Synthesis, Deep Learning, 360 Degree Image, Omnidirectional image, Immersive Imaging, 6DoF

ACM Reference Format:
Manu Gond, Emin Zerman, Sebastian Knorr, and Mårten Sjöström. 2023. LFSphereNet: Real Time Spherical Light Field Reconstruction from a Single Omnidirectional Image. In *European Conference on Visual Media Production (CVMP '23), November 30–December 01, 2023, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626495.3626500

1 INTRODUCTION

A telepresence system enables a representation of a remote scene in real time [Dima and Sjöström 2021]. This makes it a viable tool in industrial remote operation of heavy machinery to avoid safety risks. Examples of such industries include construction, mining, forestry, and underwater exploration [Li et al. 2018]. Such a system should allow a natural interaction with the remote scene.

Telepresence systems use the current developments in Virtual Reality (VR), Head-Mounted Display (HMD), network technologies

(like 4G and 5G), and consumer cameras to provide direct video feed of the environment. View Synthesis methods can be used to further enhance the awareness of the on-site environment. The new views generated using view synthesis methods has been investigated in tasks like forestry [Brunnström et al. 2020], and underwater operations [Bruno et al. 2018].

Six degrees-of-freedom (6DoF) is required for full immersive experience. The 6DoF includes forward-backward, left-right, and up-down motion in addition to the 3DoF motions which includes yaw, roll and pitch. Most telepresence methods [Dima and Sjöström 2021; Tripicchio et al. 2017; Yun et al. 2020] is limited to 3DoF but can be extened to 6DoF with the help of view synthesis approaches. The view synthesis approaches use reference views to render new views from different viewpoint, therefore achieving 6DoF.

Recent works in novel view synthesis include different approaches that can be categorized in i) depth-image-based methods that use input images with corresponding depth maps to generate new views [Xu et al. 2021; Zioulis et al. 2019], ii) multi-layer-image-based that use a layer of multiple images at different depths [Flynn et al. 2019; Li and Kalantari 2020; Zhou et al. 2018], and iii) fully-learning-based methods that learn to synthesize new views directly from input images without depth information [Gu et al. 2022; Han and Xiang 2022; Mildenhall et al. 2021]. Depth-image-based methods commonly suffer from shape distortion and pixel misalignment when the depth maps are inaccurate or noisy. The correction of these errors increases the inference time of the pipeline, i.e. making it slower and limiting the usage in real-time applications. Multi-layer-image-based methods are not limited by these issues but instead require a high number of layers to achieve good visual quality, which can increase inference time too. Learning-based methods can be trained to learn the geometrical properties of a scene and can infer final views directly, making them suitable for cases where inference time is a crucial factor. However, learning-based methods might suffer from poor visual quality if the amount of training data is not sufficient.

To date, there is no publicly available dataset for spherical LF. Methods have been proposed for capturing spherical LFs [Overbeck et al. 2018] or extending the wide field of view (FoV) to spherical cases [Broxton et al. 2020]. These methods require significant resources for capturing and processing the dataset. To meet the need of spherical LF datasets, we have rendered a synthetic spherical LF dataset by using open-source virtual scenes. Our dataset contains 120 spherical LF images and is suitable for use in training and testing spherical LF reconstruction models. We also created a small scale real world spherical LF dataset consisting of 7 horizontally placed, photographic ODIs to test the generalization of our model on real world scenes.

In this paper, we build upon learning-based methods and propose an approach to reconstruct a spherical light field (LF) from a single omnidirectional image (ODI). Although ODIs already provide 3DoF, spherical LF reconstruction extends these to 6DoF. As illustrated in Fig. 1, the pipeline consists of two networks. The proposed approach takes an Equirectangular Projection (ERP) image and converts it to a cubemap (CMP) as an intermediate data representation. Cubemaps are tangent images of a sphere with only minor distortions. Re-rendering of such tangent images comply to linear re-projection, which enables the system to utilize traditional convolution kernels without any modification in the network for cubemap re-rendering. For a better generalization of the first network we add a pre-trained ResNet-152 [He et al. 2016] as feature extractor block for leveraging the feature representations [Johnson et al. 2016; Xu et al. 2020] learned by it. The reconstructed images in the cubemap format are then converted back to the ERP format, which quality is refined in the second network that corrects the pixel mis-alignments at cube borders, and thus produces a $N \times N$ spherical LF. Our method is limited to the network inference time itself, and it can run with low inference time of 0.06 seconds on our testbed making it suitable for real time application. In summary our main contributions are:

- The fully learning based LFSphereNet that has an Encoder-Decoder architecture, which can reconstruct a $N \times N$ spherical light field given a single omnidirectional image.
- Our proposed LFSphereNet can render light fields with low inference times of around 0.06 seconds on our testbed, making it suitable for real-time applications. We also present various network variations that prioritize inference speed at the cost of image quality.
- We propose the usage of a pre-trained feature extractor block as part of the LFSphereNet to produce much sharper images given low amount of training data by utilizing the feature representations extracted from it.
- A spherical light field dataset has been rendered in Blender based on 5 open source scenes, for training and evaluation of the LFSphereNet. The dataset contains 120 spherical light fields of size 1024x2048x7x7[1].
- A real world spherical light field dataset has been captured to evaluate the generalization of LFSphereNet. The small scale dataset contains 6 spherical light fields of size 1024x2048x1x7[1].

The remainder of the paper is organized as follows. In Section 2, we review the state-of-the-art methods related to works in the domain of light field (LF) reconstruction and spherical view synthesis. Next, we introduce our proposed LFSphereNet in Section 3, including implementation details. We then modify and evaluate the proposed method for planar LF reconstruction in Section 4. Subsequently, we evaluate the performance of LFSphereNet for spherical LF reconstruction in Section 5 and discuss the results. Finally, we summarize our conclusions in Section 6.

2 BACKGROUND

The aim of this work is to reconstruct a spherical light field (LF) from a single omnidirectional image (ODI). The LF represents the spatial and angular information of a scene [Levoy and Hanrahan 1996], which can be represented by a HxWxNxN image, where HxW is the spatial resolution and NxN is the angular resolution. This allows to solve problems like viewpoint change [Zhou et al. 2020] and refocusing [Ng et al. 2005]. LF reconstruction methods extend view synthesis methods by producing a dense NxN LF from a single or a few input views. Therefore, LF reconstruction method eliminates the requirement of using a multi-camera rig to capture the LF.

We classify different methods for novel view synthesis and LF reconstruction into three categories established on recent works: depth-image-based (Sec. 2.1), multi-layer-image-based (Sec. 2.2),

[1] Dataset can be found at: https://doi.org/10.6084/m9.figshare.24219337

and sampling & learning-based methods (Sec. 2.3). Relevant LF reconstruction methods are developed on planar images. Since there is no prior work on NxN spherical LF reconstruction from a single ODI, we only review methods for LF reconstruction from a single or few input views and methods for spherical view synthesis.

2.1 Depth-Image-Based Methods

Depth-image-based methods are derived from the Depth-Image-Based-Rendering (DIBR) technique. A straightforward approach for LF reconstruction is using a convolutional neural network (CNN) to estimate the depth [Srinivasan et al. 2017] followed by a warping operator to infer target views, and a seperate refinement network to handle warping artifacts [Zhou et al. 2020, 2021]. The approach of forcing a network to learn the depth estimation without ground truth depth data can be traced back to stereo prediction using Deep3D [He et al. 2016]. Other approaches use depth images predicted by a separate network or their own end-to-end frameworks for estimating optical flow [Cun et al. 2019], appearance flow [Ivan et al. 2019] or both [Bae et al. 2021] to synthesize the LF. In case of view synthesis with ODIs, the methods in [Xu et al. 2021; Zioulis et al. 2019] use a forward splatting operator to generate target views.

The approaches to reconstruct a light field in [Xu et al. 2021; Zioulis et al. 2019] can theoretically be extended to spherical LF reconstruction. However, the reconstruction of a NxN spherical LF would increase the inference time by N^2. Additionally, these methods are limited in the sense that the warping operators (forward splatting) designed for ODI require the estimation of intermediate depth maps and will suffer from shape distortion and pixel misalignment in case of inaccurate or noisy depth maps.

2.2 Multi-Layer-Image-Based Methods

Multi-layer-image-based methods use more than one plane to describe the images. There are multiple representations to do so, e.g. Multi Plane Image (MPI), Multi Sphere Image (MSI), Multi Depth Panorama (MDP), and Multi Cylinder Image (MCI).

MPI representations use a set of parallel planes at fixed depths containing a corresponding image and alpha map. The parallel planes allow methods to render novel views built on a fixed [Flynn et al. 2019; Li and Kalantari 2020; Tucker and Snavely 2020; Zhou et al. 2018] or variable [Li and Kalantari 2020] number of planes. For view synthesis with ODIs, Serrano et al.[Serrano et al. 2019] introduced a three-layer scene representation, while other methods use custom formats designed for such images, namely MDP [Lin et al. 2020], MCI [Waidhofer et al. 2022], and MSI [Attal et al. 2020].

Multi-layer-image based reconstruction methods have limitations which stems from the requirement of using multi-camera rigs [Attal et al. 2020; Broxton et al. 2020; Lin et al. 2020] to generate their layered image representations. Another limitation is demand of a high number of layers [Waidhofer et al. 2022] to achieve good visual results which can increase inference time. Furthermore, only one view is synthesized at a time, which implies the inference time to increase by N^2 when we target real time NxN spherical LF reconstruction.

2.3 Sampling-Based and Fully Learning-Based Methods

Sampling-based methods reconstruct LFs from sparse inputs by optimizing sparsity in the Fourier spectrum [Shi et al. 2014], using epipolar-plane images with CNNs [Wu et al. 2019, 2017], and using the shearlet transform [Vagharshakyan et al. 2018]. Some methods use pseudo 4D CNNs [Chen et al. 2022; Wang et al. 2018] to reconstruct the LF. Learning-based methods try to estimate the novel views directly from the input views. Here, the warping and occlusion handling operators are learned by the network. To reconstruct LFs, various methods have used neural networks like generative-adversarial network (GAN) [Chandramouli et al. 2020; Chen et al. 2020] and auto-encoders [Han and Xiang 2022]. Learning-based methods have drawbacks when used with ODIs because the convolution kernel cannot learn the distortion pattern of ODIs without having a significant high amount of training data.

Neural Radiance Fields (NeRF) [Mildenhall et al. 2021] belong to learning-based methods which learn a function to render complex scenes. NeRF was extended to ODIs, as proposed by Gu et al. [Gu et al. 2022]. However, NeRF is limited in terms of generalization as it needs to be trained for each scene separately, which is not applicable in real-time applications.

In summary, the depth-image-based methods, multi-layer-image-based methods, and learning-based methods are possible candidates for view synthesis from a single ODI to a spherical LF. We have chosen to base our proposed solution on learning-based methods because they have characteristics for high quality rendering in a fast inference time.

2.4 Formats for Omnidirectional Image Data

Equirectangular Projection (ERP) is a common way to represent ODIs. However, due to its nature of having high distortion when moving to the polar regions, filter kernels of traditional convolutional neural networks (CNN) cannot be utilized as they cannot learn this distortion pattern. There has been work done with spherical convolution kernels [Cohen et al. 2018; Esteves et al. 2018; Su and Grauman 2019] which increase in width when moving to the polar regions, but their effectiveness diminishes when the networks become deep. Furthermore, each region of an ERP image has different kernels, which do not share any information with each other. These different kernels between each overlapped region require the use of multi-scale alignment field to produce a single consistent feature map.

The limitations of ERP can be overcome using the cubemap projection (CMP), which allows the use of standard CNN kernels.

3 METHOD

This work aims to create a spherical LF from a single ODI using two different networks as depicted in Fig. 1, which allows for 6DoF viewing. The first network performs the spherical LF reconstruction in the CMP format. The second network performs a spherical LF refinement in the ERP format and is responsible for concealing artifacts near the image borders of the cube faces from the first network.

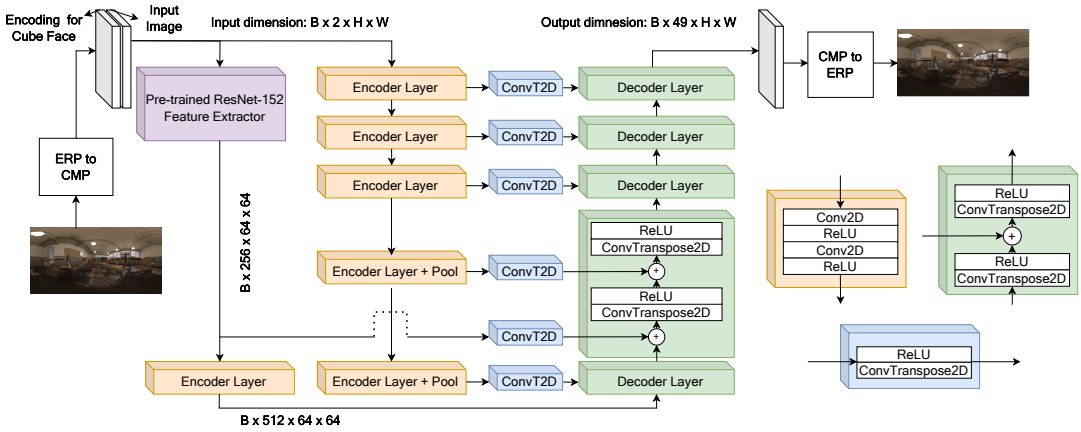

Figure 2: Reconstruction Network Architecture: the input has two information channels, one for the scene, one for the encoded cube face. Network components consists of Encoder layers (orange, certain with pooling), Feature extractor (purple), Decoder layers (green) and Convolution transpose (blue). B = Batch size, H = Height, W= Width.

The following sub-sections describe the proposed reconstruction network (Sec. 3.1), refinement network (Sec. 3.2), loss functions (Sec. 3.3) and implementation details (Sec. 3.4).

3.1 Reconstruction Network

The reconstruction network is used to reconstruct the LF in CMP format given a single ODI as input. The input ODI in ERP format, $L_{ERP}(\mathbf{x}, 0)$, is the central image of a spherical $N \times N$ LF, where 0 indicates the center and $\mathbf{x}$ are the spatial coordinates (x, y). Given the limitations as described in Sec. 2.4, the ODI in ERP format is transformed into the CMP format with its six cube faces L^i_{CMP} with $i \in \{(\theta_1, \phi_1), ..., (\theta_6, \phi_6)\}$. Here, θ and ϕ are the longitude and latitude angles in ERP.

A cube face with angle i is then reconstructed into an array of cube faces $\hat{L}^i_{CMP}(\mathbf{x}, \mathbf{u})$ with:

$$\hat{L}^i_{CMP}(\mathbf{x}, \mathbf{u}) = r(L^i_{CMP}(\mathbf{x}, 0), i), \tag{1}$$

where r represents the reconstruction function of the LF for each cube face $L^i_{CMP}(\mathbf{x}, 0)$, and $\mathbf{u}$ are the angular coordinates (u, v) of the reconstructed cube face array. The reconstruction function is modeled by an encoder-decoder network with an architecture similar to U-Net [Ronneberger et al. 2015], as depicted in Fig. 2, due to its recent usage in reconstruction tasks [Zhou et al. 2020, 2021]. The network learns to model the geometric features implicitly.

Convolutional operations without any pooling are used in the first three encoder layers of this reconstruction network. The pooling layers are not used in order to keep both local and global features of image without dimension reduction which has been shown to perform better in [Cun et al. 2019; Zhou et al. 2021] for reconstruction tasks.

The **Pre-Trained Feature Extractor** block is used in the network to address the lack of training data. The network utilizes a pre-trained ResNet-152 [He et al. 2016], which was trained on the ImageNet dataset and can be used to leverage the feature representation learned by it. The features extracted by pre-trained networks in [Johnson et al. 2016; Liu et al. 2019; Tian et al. 2021; Wang et al. 2021; Xu et al. 2020] have shown to achieve better visual reconstruction results. The reconstruction network uses the first two layers of ResNet-152 and extracts 256 feature maps of 64x64 spatial dimension. The network then further passes these feature maps to an additional encoder layer with 512 filters resulting in 512 feature maps of size 64x64 pixels. Therefore, the network not only learns from the training data, but also uses the information extracted from the ResNet.

The **Upsampling** of feature maps extracted from the ResNet-152 block and the last two encoder layers is required since the low spatial dimension of feature maps needs to be upscaled to the original resolution of the image. When upsampling these feature maps at the decoder layers, the network uses a learning-based upsampling [Ren et al. 2017] to upsample the feature maps without explicitly defining the upsampling method like e.g. bilinear or bicubic interpolation. In Sec. 5.5, we compare the learning-based upsampling against bilinear and bicubic upsampling in an ablation study.

Finally, the network splits the input RGB image into separate channels and processes one channel at a time. The output for the R,G and B channels are generated after 3 forward passes and then merged to the final RGB image. Applying a 2D convolution on a RGB image in a single pass could be avoided as the same kernel is then responsible for handling both cross-channel correlations and spatial correlations as described in [Chollet 2017]. In addition, separating the RGB image into distinct channels serves as a form of data augmentation, tripling the amount of training data.

3.2 Refinement Network

A refinement network is introduced in LFSphereNet to resolve artifacts at the intersection of the cubemap faces. These artifacts are the results of discontinuities at the cubemap boundaries.

The refinement network therefore transforms the cube faces $\hat{L}^i_{CMP}(\mathbf{x}, \mathbf{u})$ in CMP format to $\hat{L}_{ERP}(\mathbf{x}, \mathbf{u})$ in ERP format. The pixels close to the borders of the cube faces are particularly distorted after the transformation back to ERP format. A binary mask M is created for these pixel locations and is used to guide the refinement network to improve the quality of pixels in border regions.

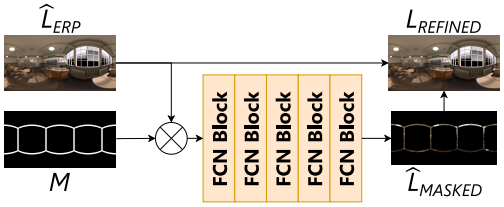

Figure 3: Refinement Network masks out and refines borders of the cubemap.

The refinement network takes the mask M, masks off the whole ERP image and only works with a specified width of 40 pixels at border regions. The refinement step can be expressed as:

$$\hat{L}_{MASKED}(\mathbf{x}, \mathbf{u}) = e(\hat{L}_{ERP}(\mathbf{x}, \mathbf{u}) \odot M) \tag{2}$$

where e denotes the refinement network. As illustrated in Fig. 3, it is a fully convolution network (FCN) built with the same encoder layers as the reconstruction network. The final refined ERP image is created by replacing the pixels of $\hat{L}_{ERP}$ with the pixels of the refined masked image $\hat{L}_{MASKED}$ as follows:

$$L_{REFINED}(\mathbf{x}, \mathbf{u}) = \begin{cases} \hat{L}_{ERP}(\mathbf{x}, \mathbf{u}), & if\ M(\mathbf{x}, \mathbf{u}) = 0 \\ \hat{L}_{MASKED}(\mathbf{x}, \mathbf{u}) & otherwise. \end{cases} \tag{3}$$

A standard convolution kernel can be applied in this network as it only focuses on a narrow width of pixels instead of the whole ERP image. Hence, the network does not need to learn the distortion pattern of ERP images. The resulting improvement in overall visual quality is shown in Sec. 5.

3.3 Loss Functions

The two networks of the proposed LFSphereNet are trained using two different loss functions. The reconstruction network learns the function r by minimizing the error with respect to its parameters β_r:

$$min_{\beta_r} \sum_{b} \sum_{i=1}^{6} \left\| L^i_{CMP} - \hat{L}^i_{CMP} \right\| \tag{4}$$

where b is the number of training samples.

The refinement network works with the masked image and outputs the refined masked image $\hat{L}_{MASKED}(\mathbf{x}, \mathbf{u})$. For training the refinement network, the ground truth masked ERP $L_{MASKED}(\mathbf{x}, \mathbf{u})$ can be generated by applying the same mask M from Sec. 3.2 as:

$$L_{MASKED}(\mathbf{x}, \mathbf{u}) = L_{ERP}(\mathbf{x}, \mathbf{u}) \odot M \tag{5}$$

The refinement network then learns the function e by minimizing the error with respect to its parameters β_e:

$$min_{\beta_e} \sum_{b} \left\| L_{MASKED} - \hat{L}_{MASKED} \right\| \tag{6}$$

The visual results of applying different loss functions and assuming photometric consistency was studied in [Zhao et al. 2016; Zhou et al. 2021]. As described by Zhao et al. [Zhao et al. 2016], the L1 loss is more robust to outliers compared to L2 loss because it does not heavily penalize large errors. Therefore, L1 loss is used for both networks.

3.4 Implementation

The input of LFSphereNet is a single channel of a RGB image at a time along with an extra channel that contains the direction of the cube face. The direction of the cube face is denoted by angles (θ, ϕ) which are normalized between 0 to 1. The value of θ is stored in the first half of the extra channel, [1:Height]x[1:Width/2] and ϕ in the second half. LFSphereNet was implemented in PyTorch and trained with the following hyperparameters: batch size of 16, learning rate of 0.003, adam optimizer, L1 loss and L2 as a regularizer. After each 20 epochs, the learning rate was decreased by a factor of 0.5. The network was trained for 150 epochs on 4 GPUs (Nvidia A100) within a computing cluster with Intel Xeon Gold 6338 CPUs. The training duration was 42 hrs. For the dataset split, 75% of the data was used for training, 12.5% for validation and 12.5% for testing. The entire details of the network layers are listed in our supplementary materials.

4 EXPERIMENTS FOR PLANAR LF

We first want to demonstrate that the reconstruction network of LFSphereNet, although designed for ODIs, also achieves comparable results to the state-of-the-art in planar LF reconstruction from a single or few input images. Planar LF reconstruction does neither require the refinement network nor the encoding of the rotational camera angles in the input because the virtual camera only faces the frontal direction. Therefore, the proposed reconstruction network was modified for this experiment such that the positional encoding was removed from the input.

4.1 Experimental Design

4.1.1 Datasets. The publicly available datasets Lytro Flowers [Srinivasan et al. 2017], Stanford [Raj et al. 2016], and JPEG-Pleno [Rerabek and Ebrahimi 2016] were used for testing LFSphereNet and the state-of-the-art methods NoisyLFRecon [Zhou et al. 2021], DGLF [Zhou et al. 2020], DALF [Cun et al. 2019], and IR-V [Han and Xiang 2022]. The LF images extracted from these datasets were of size 352x512x7x7. 75% of the images from the datasets Flowers and Stanford were used for training. The JPEG-Pleno dataset was only used for testing and not for training.

4.1.2 Evaluation Metrics. The output was tested against ground truth data by employing the metrics mean absolute error (MAE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [Wang et al. 2004], multi-scale structural similarity index measure (MS-SSIM) [Wang et al. 2003], features similarity index matrix (FSIMC) [Zhang et al. 2011], pixel-based visual information fidelity (VIFP) [Sheikh and Bovik 2006], deep image structure and texture similarity (DISTS) [Ding et al. 2020], and learned perceptual image patch similarity (LPIPS) [Zhang et al. 2018].

4.2 Results

The reconstruction of planar LF by the proposed LFSphereNet was compared against state-of-the-art methods NoisyLFRecon [Zhou et al. 2021], DGLF [Zhou et al. 2020], DALF [Cun et al. 2019], and IR-V [Han and Xiang 2022]. All models were trained from scratch using the same datasets. The publicly available codes were used for [Han and Xiang 2022; Zhou et al. 2020, 2021], whereas DALF was

Table 1: Planar LF Reconstruction: Quality, best values in bold, second best in *italics*. Arrows indicate the better direction

Dataset	Method	MAE ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	FSIMC ↑	VIFP ↑	DISTS ↓	LPIPS ↓
Flowers	NoisyLFRecon	**0.0095**	*39.9500*	**0.9763**	**0.9932**	**0.9847**	**0.9285**	**0.0389**	**0.0190**
Train Images: 2181	DGLF	0.0489	35.6194	0.8773	0.9301	0.8947	0.6021	0.1556	0.1431
Test Images: 363	DALF	0.0172	37.3006	0.8941	0.9589	0.9295	0.7362	0.1033	0.0911
	IR-V	0.0163	37.9034	0.9122	0.9645	0.9415	0.7324	0.0994	0.0707
	LFSphereNet	*0.0109*	**41.3719**	*0.9461*	*0.9868*	*0.9680*	*0.9060*	*0.0812*	*0.0512*
Stanford	NoisyLFRecon	0.0185	35.4917	*0.9572*	0.9771	0.9620	0.8380	0.0718	0.0395
Train Images: 264	DGLF	0.0161	35.6509	**0.9673**	**0.9897**	*0.9683*	*0.8655*	0.0685	*0.0321*
Test Images: 45	DALF	0.0143	38.2996	0.9172	0.9750	0.9416	0.7829	0.0667	0.0402
	IR-V	*0.0141*	*39.4909*	0.9440	*0.9852*	0.9619	0.8358	*0.0628*	0.0390
	LFSphereNet	**0.0118**	**40.9830**	0.9488	0.9797	**0.9721**	**0.8842**	**0.0556**	**0.0300**
JPEG Pleno	NoisyLFRecon	0.0234	36.6275	**0.9575**	0.9687	*0.9610*	*0.8549*	**0.0718**	*0.0435*
Train Images: 0	DGLF	0.0489	32.3387	0.7207	0.8166	0.8223	0.4246	0.1556	0.1431
Test Images: 10	DALF	0.0282	35.0285	0.8257	0.9291	0.8828	0.6462	0.1095	0.0937
(Trained on Flowers)	IR-V	*0.0208*	*37.2085*	0.9122	*0.9719*	0.9357	0.7878	0.0894	0.0605
	LFSphereNet	**0.0168**	**39.2624**	*0.9429*	**0.9879**	**0.9618**	**0.9069**	*0.0737*	**0.0425**

re-implemented in PyTorch according to the information presented in the paper [Cun et al. 2019].

The quality of the rendered light field for each network is presented in Table 1. NoisyLFRecon [Zhou et al. 2021] performs best while LFSphereNet is second best for the Lytro Flowers [Srinivasan et al. 2017] dataset. However, NoisyLFRecon uses nine images as input while LFSphereNet only takes a single image as input. Both DGLF [Zhou et al. 2020] and DALF [Cun et al. 2019] also take a single image as input, but DGLF generates the full NxN LF at once, whereas DALF outputs one sub-aperture at a time. For the Stanford dataset, which contains significantly less number of training images compared to the Flowers dataset, LFSphereNet outperforms the remaining methods in each metric except SSIM and MS-SSIM. For JPEG-Pleno [Rerabek and Ebrahimi 2016], which was not used for training, LFSphereNet outperforms the remaining methods, i.e. proving better generalization.

The measured inference times for reconstructing a single LF image of size 352x512x7x7 on a GTX 1080Ti GPU were: 1.8130 seconds for NoisyLFRecon, 1.5009 seconds for DGLF, 0.0583 seconds for DALF, 0.1429 for IR-V, and **0.0008** seconds for LFSphereNet. Although DALF is built on a similar encoder-decoder architecture, the usage of an additional depth estimation network increases its overall inference time.

5 EXPERIMENTS FOR SPHERICAL LF

5.1 Synthetic Spherical Light Field Dataset

No spherical LF dataset was available prior to our investigation. Therefore, a spherical LF dataset was created to enable both training and evaluation of the proposed system. The dataset was rendered in Blender with 1024×2048 spatial resolution and 7×7 angular resolution. We modified the python code written by Gu et al. [Gu et al. 2022] to match the uniform camera array requirement. To create this spherical LF dataset, we used five freely available virtual scenes on the Blender demo website, including two outdoor scenes (*Lone Monk* and *Barcelona*) and three indoor scenes (*Classroom*, *Barbershop* and *Italian Flat*). We placed the spherical camera array at different locations in the scenes to capture multiple images from each scene. The number of different locations varies from 10 to 30 depending on the size of the scene, resulting in a total of 120 spherical LFs. The lens type of each spherical camera was set to *panoramic* and the panorama type was set to *equirectangular*. The baseline between each subsequent sub-aperture ODI was set to 1 cm.

5.2 Real Spherical Light Field Dataset

A real spherical LF dataset was created to verify the generalization of our method on real world scenes. The acquisition of such real spherical LF dataset with 7x7 ODI camera grid is challenging as each ODI camera will occlude the other one due to their 360° FoV nature. In order to overcome this limitation of occlusion, a single *Insta 360 X3* camera was mounted on an *Atlas 200* dolly which was programmed to move the camera 1 cm at a time in horizontal direction. This way, we acquired a 1x7 spherical LF. Capturing the 7x7 LF was not feasible as the vertical movement was not precise enough. In total, we captured 6 different spherical LFs with 1x7 angular resolution.

5.3 Experimental Design

No other spherical LF reconstruction methods exist to our knowledge. Thus, we are bound to only test LFSphereNet against the ground truth data and existing 360 view synthesis networks which are modified to enable spherical LF reconstruction. We use the publicly available code from 360ViewSynth[Zioulis et al. 2019] and PanoSynthVR[Waidhofer et al. 2022] and modify the inference step for creating a spherical LF of size 1024x2048x7x7. These methods output a single view, i.e. sub-aperture ODI, at a time, hence, the inference requires NxN steps. For the objective evaluation, we use the same metrics as described in Section 4.1.2. All models were trained on the same training dataset which included *Lone Monk*,

Barcelona, and *Italian Flat* scenes. We use data augmentation to further increase the amount of training images, resulting in a total of 150 LFs for training. For testing and validation, we use *Barbershop* and *Classroom* scenes which were not used in the training. In total, 25 LFs were used for validation and another 25 LFs were used for testing.

To test the model on the real spherical LF dataset, the model was first trained on the synthetic dataset with angular resolution set to 1x7, followed by retraining the model again on 4 out of 6 LFs from the real spherical LF dataset. The remaining 2 spherical LFs were used for testing.

5.4 Results

5.4.1 Quantitative Results. The comparison results of PanoSynthVR [Waidhofer et al. 2022], 360ViewSynth [Zioulis et al. 2019] and LFSphereNet with and without the refinement network are presented in Table 2 for the synthetic and in Table 3 for the real dataset, respectively.

LFSphereNet outperforms these two methods in all metrics except DISTS and LPIPS where PanoSynthVR shows better results. The 360ViewSynth shows worse results compared to PanoSynthVR because it uses a splatting operator on predicted depth, and for the part of scene where depth is at infinity, it performs worse as shown in the qualitative results in Fig. 4. The last column in Table 2 and Table 3 shows the improvement when using the refinement network. The improvement seems very small in terms of numerical values as refinement is only performed for a very small amount of pixels located at the stitched border regions. However, these improvements are visible in the image as described in Sec. 5.4.2.

The inference times to reconstruct the 7 ODIs are shown in the last row of Table 2. LFSphereNet performs much faster since it reconstructs the whole LF at once. However, the memory footprint of LFSphereNet is much higher with around 19 million parameters. The values presented for the real world dataset in Table 3 show slightly worse results when compared to the synthetic dataset in Table 2. This decrease in performance is related to the size of training data used. The real world dataset has only 4 spherical LF images to learn from. In Table 3, 360ViewSynth shows better DISTS, LPIPS, PSNR, and VIFP compared to the other two methods. However, the visual results indicate that it fails to produce geometrically correct ERP images. Only the objects at the front and back of the sphere move slightly while the remaining regions remain fixed at the same place, i.e. do not have parallax compared to the input view. Hence, these synthesized views are similar to the input view which explains the better DISTS, LPIPS and VIFP values for 360ViewSynth.

Table 2: Spherical LF Reconstruction (1024x2048x7x7) on Synthetic Dataset: Quality and Average Runtime (s) compared with 360ViewSynth (denoted 360VS) and PanoSynthVR (denoted PSVR), best values in bold.

Metrics	LF Reconstruction Methods			
	360VS	PSVR	LFSphereNet*	LFSphereNet
MAE↓	0.0922	0.0265	0.0130	**0.0125**
DISTS↓	0.1215	**0.0531**	0.0877	0.0880
LPIPS↓	0.1935	**0.0670**	0.0860	0.0825
PSNR↑	32.89	34.76	37.14	**37.45**
SSIM↑	0.6495	0.7878	0.9052	**0.9121**
MS-SSIM↑	0.7690	0.8687	0.9660	**0.9691**
FSIMC↑	0.8443	0.9148	0.9573	**0.9573**
VIFP↑	0.3710	0.4978	0.8005	**0.8035**
Runtime↓	12.4845	2.7077	**0.0589**	0.0606

LFSphereNet* is without refinement network

Table 3: Spherical LF Reconstruction (1024x2048x1x7) on Real Dataset compared with 360ViewSynth (denoted 360VS) and PanoSynthVR (denoted PSVR): best values in bold.

Metrics	LF Reconstruction Methods			
	360VS	PSVR	LFSphereNet*	LFSphereNet
MAE↓	0.0414	0.0746	0.0369	**0.0365**
DISTS↓	**0.0221**	0.0906	0.0848	0.0848
LPIPS↓	**0.0728**	0.2381	0.1573	0.1572
PSNR↑	**34.14**	30.20	32.42	32.44
SSIM↑	0.7044	0.5682	0.7361	**0.7402**
MS-SSIM↑	0.8052	0.5784	0.8561	**0.8565**
FSIMC↑	0.8767	0.7412	0.8944	**0.8957**
VIFP↑	**0.5294**	0.2204	0.4909	0.4916

LFSphereNet* is without refinement network

5.4.2 Qualitative Results. Fig. 4 shows visual comparisons between the selected methods for *Classroom* and *Barbershop*, including the corresponding error maps for synthetic scenes. Fig. 6 shows similar comparisons for real world scene. The excerpts show that the proposed method has the smallest error. Nonetheless, both the proposed method and PanoSynthVR produce visually acceptable results for both scenes. 360ViewSynth on the other hand has artifacts at the far distant windows of the classroom scene. However, for the *Barbershop* scene, which has objects in close proximity, the results are deemed acceptable. LFSphereNet has difficulties to render highly reflective surfaces correctly as visible in the error map excerpt, in particular in some paintings, of the barbershop scene in Fig. 4. It also struggles to render objects with fine detail like the plants' leaves in the real world scene as shown in Fig. 6.

Visual results of the classroom scene are shown in Fig. 5 to observe the impact of the refinement network. The artifacts are visible near the zoomed excerpt of the window region, when the refinement network is not applied. These highlighted artifacts get fixed in the refined ODI which is shown in the right excerpt.

5.5 Ablation Study

An ablation study was carried out to understand the impact of the different design choices, in particular for the reconstruction network of LFSphereNet. We trained different variations of the reconstruction network combined with the same refinement network to compare and learn the effect of each component. These configurations are: **LFSphereNet-NoResNet:** Network without any pretrained feature extractor, **LFSphereNet-Bicubic and LFSphereNet-Bilinear:** Networks with bicubic or bilinear upsampling instead of

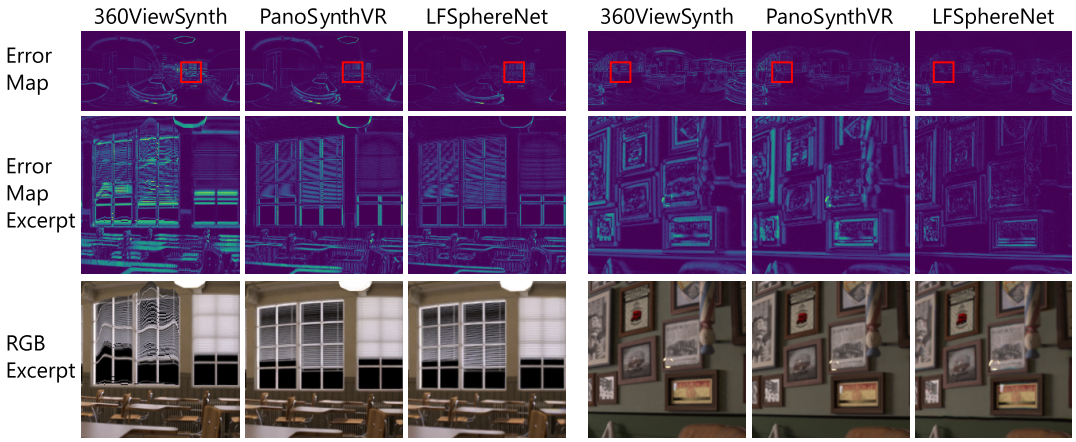

Figure 4: Spherical LF Reconstruction (1024x2048x7x7) on Synthetic Dataset: Visual Quality comparison between 360ViewSynth, PanoSynthVR and LFSphereNet of excerpts from *Classroom* and *Barbershop* and their error maps.

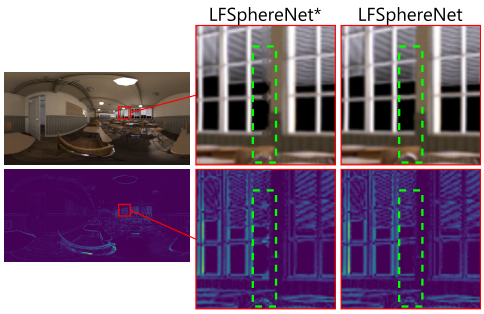

Figure 5: Effect of Refinement network: Visual Quality of *Classroom* scene. Highlighted region in green (dashed) color shows the artifact which gets resolved when refinement network is used. LFSphereNet* is without refinement network.

Table 4: Ablation Study for Light Field Reconstruction (1024x2048x7x7): Quality, Average Runtime (sec.) and #Network Parameters, best values in bold.

Metric	LFSphereNet-(Variation)				
	NoResNet	Bicubic	Bilinear	RGB	[Main]
MAE ↓	0.0160	0.0127	0.0126	0.0153	**0.0125**
DISTS ↓	0.1043	0.0891	0.0886	0.0972	**0.0880**
LPIPS ↓	0.0939	0.0827	0.0832	0.0858	**0.0825**
PSNR ↑	34.55	37.33	37.29	34.76	**37.45**
SSIM ↑	0.8483	0.9110	**0.9121**	0.8580	**0.9121**
MS-SSIM ↑	0.9186	**0.9692**	0.9688	0.9229	0.9691
FSIMC ↑	0.9529	0.9569	0.9570	0.9566	**0.9573**
VIFP ↑	0.7257	0.8024	0.8028	0.7499	**0.8035**
Runtime ↓	**0.0050**	0.0626	0.0551	0.0131	0.0606
#Param.	13.4E6	19.6E6	19.6E6	19.3E6	19.2E6

the proposed learning-based one, **LFSphereNet-RGB:** Network with RGB input instead of single color channels at a time, and **LFSphereNet:** Our proposed network. Note that we keep the architecture of the refinement network unchanged as it is not affected by any of these design choices, but it is necessary to be trained for each case to make it a fair comparison.

Each variation of the network is trained with the same hyperparameters and on the same dataset (see Sec. 5.3). The quantitative results are shown in Table 4. These results are discussed in detail below. Due to space limitation, the visual results can be found in the supplementary materials.

Pre-trained Feature Extractor: When the pre-trained feature extractor based on ResNet-152 [He et al. 2016] is removed in the "NoResNet" case, we can see from Table 4 that it leads to worst results compared to any other variation of the network. Without the feature extractor block, some object features cannot be extracted when a new scene is given to the network, resulting in worse generalization.

Upsampling: When "bilinear" and "bicubic" upsampling modes are used, i.e. in "LFSphereNet-Bilinear" & "LFSphereNet-Bicubic", the results show a small decrease in quality compared to the proposed LFSphereNet as suggested by the metrics in Table 4. Visually, the quality of the output does not differ by a noticeable amount.

RGB Mode: In LFSphereNet, a single color channel at a time is used as input to avoid using a single convolution kernel for handling both cross-channel correlation and spatial correlations as described in [Chollet 2017]. To support this design choice, we test the network with passing full RGB images in "LFSphereNet-RGB" for validation. As seen in Table 4, it gives overall worse results, similar to the "LFSphereNet-NoResNet". Hence, we conclude that the shared weights in standard 2D convolutions can lead to a loss of discriminative power between channels, and as a result, the learned filters may not be able to distinguish between different features in the input image. However, this design choice can be used if the amount of training data is significantly high as shown with some planar LF reconstruction networks [Han and Xiang 2022; Zhou et al. 2021].

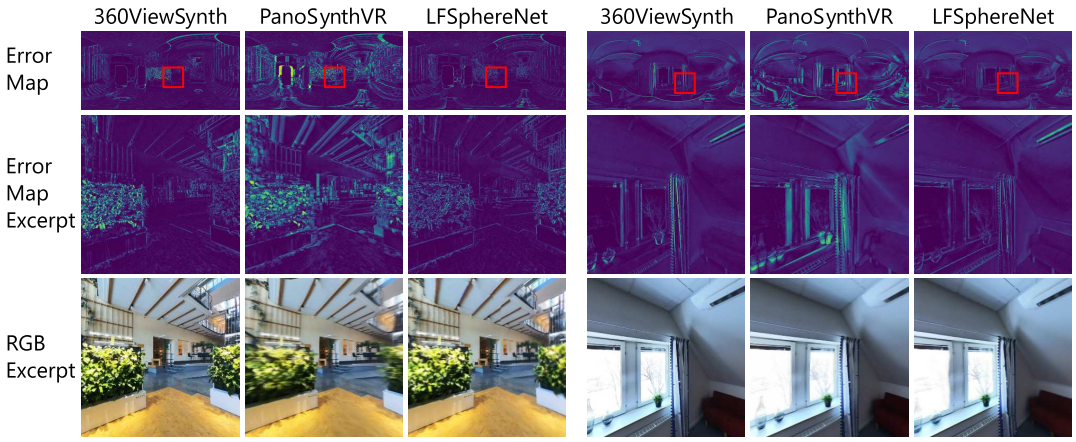

Figure 6: Spherical LF Reconstruction (1024x2048x1x7) on Real Dataset: Visual Quality comparison between 360ViewSynth, PanoSynthVR and LFSphereNet of excerpts from scene and their error maps.

Inference Time and Visual Quality: Comparing the inference times, the LFSphereNet-NoResNet performs fastest since it removes the ResNet-152 layers from the architecture reducing the number of parameters from 19 Million to 13 Million. LFSphereNet-RGB has the second best inference time as it takes just one forward pass to reconstruct the results. These two variations of networks suit well for applications that might require very low inference time on the cost of slightly degraded visual quality. For future work a small scale specialized feature extractor could be investigated for achieving better visual quality and low inference time.

Networks of different upsampling modes have similar inference times close to our proposed network. These upsampling modes are LFSphereNet-Bicubic and LFSphereNet-Bilinear. Hence, as a design choice, it is well suited to use the default network as there is no significant gain in visual quality and inference time when using different upsampling modes.

6 CONCLUSION

This paper proposes a spherical light field reconstruction method: LFSphereNet, which utilizes a learning-based approach. We use a U-Net based reconstruction network which acts as universal function approximator for local LF reconstruction. The spherical LF reconstruction is performed in the CMP format to allow the usage of traditional convolution kernels. We demonstrate that the proposed refinement network is able to correct misaligned pixels in the final ODI.

A synthetic spherical LF dataset is created for training and testing, and a small real world spherical LF dataset for testing the generalization. The proposed LFSphereNet achieves state-of-the-art performance when compared to spherical view synthesis methods which were modified to reconstruct a spherical LF. We show that our method can achieve real time LF reconstruction while keeping good visual quality. Moreover, our reconstruction network also achieves state-of-the-art performance compared to planar LF reconstruction methods. From the visual results, our network as well as the state-of-the-art methods struggle to produce accurate visual quality in areas with reflective surfaces. For future work, a quality of experience study (QoE) can be conducted to evaluate the subjective visual quality of results. In addition, network compression techniques can be applied to further reduce the number of parameters in the proposed LFSphereNet. Finally, a real world spherical LF dataset can be created and used for training and testing spherical LF reconstruction on real data.

ACKNOWLEDGMENTS

The work was supported by the European Joint Doctoral Programme on Plenoptic Imaging (PLENOPTIMA) through the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 956770.

REFERENCES

Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. 2020. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I.* Springer, 441–459.

Kyuho Bae, Andre Ivan, Hajime Nagahara, and In Kyu Park. 2021. 5d light field synthesis from a monocular video. In *2020 25th International Conference on Pattern Recognition (ICPR).* IEEE, 7157–7164.

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive Light Field Video with a Layered Mesh Representation. *ACM Trans. Graph.* 39, 4, Article 86 (aug 2020), 15 pages.

Kjell Brunnström, Elijs Dima, Tahir Qureshi, Mathias Johanson, Mattias Andersson, and Mårten Sjöström. 2020. Latency impact on quality of experience in a virtual reality simulator for remote control of machines. *Signal Processing: Image Communication* 89 (2020), 116005.

Fabio Bruno, Antonio Lagudi, Loris Barbieri, Domenico Rizzo, Maurizio Muzzupappa, and Luigi De Napoli. 2018. Augmented reality visualization of scene depth for aiding ROV pilots in underwater manipulation. *Ocean Engineering* 168 (2018), 140–154.

Paramanand Chandramouli, Kanchana Vaishnavi Gandikota, Andreas Goerlitz, Andreas Kolb, and Michael Moeller. 2020. A generative model for generic light field reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 4 (2020), 1712–1724.

Bin Chen, Lingyan Ruan, and Miu-Ling Lam. 2020. LFGAN: 4D Light Field Synthesis from a Single RGB Image. *ACM Trans. Multimedia Comput. Commun. Appl.* 16 (2 2020). Issue 1. https://doi.org/10.1145/3366371

Yangling Chen, Shuo Zhang, Song Chang, and Youfang Lin. 2022. Light Field Reconstruction Using Efficient Pseudo 4D Epipolar-Aware Structure. *IEEE Transactions on Computational Imaging* 8 (2022), 397–410.

François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition.*

1251–1258.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. 2018. Spherical CNNs. In *International Conference on Learning Representations*.

Xiaodong Cun, Feng Xu, Chi-Man Pun, and Hao Gao. 2019. Depth-Assisted Full Resolution Network for Single Image-Based View Synthesis. *IEEE Computer Graphics and Applications* 39 (2019), 52–64. Issue 2. https://doi.org/10.1109/MCG.2018.2884188

Elijs Dima and Mårten Sjöström. 2021. Camera and Lidar-Based View Generation for Augmented Remote Operation in Mining Applications. *IEEE Access* 9 (2021), 82199–82212.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. 2020. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence* 44, 5 (2020), 2567–2581.

Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. 2018. Learning SO(3) Equivariant Representations with Spherical CNNs. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View Synthesis With Learned Gradient Descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kai Gu, Thomas Maugey, Sebastian Knorr, and Christine Guillemot. 2022. Omni-NeRF: Neural Radiance Field from 360° Image Captures. (2022), 1–6.

Kang Han and Wei Xiang. 2022. Inference-Reconstruction Variational Autoencoder for Light Field Image Reconstruction. *IEEE Transactions on Image Processing* 31 (2022), 5629–5644. https://doi.org/10.1109/TIP.2022.3197976

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Andre Ivan, In Kyu Park, et al. 2019. Synthesizing a 4D spatio-angular consistent light field from a single image. *arXiv preprint arXiv:1903.12364* (2019).

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 694–711.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 31–42.

Qinbo Li and Nima Khademi Kalantari. 2020. Synthesizing Light Field from a Single Image with Variable MPI and Two Network Fusion. *ACM Trans. Graph.* 39 (11 2020). Issue 6.

Xiao Li, Wen Yi, Hung-Lin Chi, Xiangyu Wang, and Albert P.C. Chan. 2018. A critical review of virtual and augmented reality (VR/AR) applications in construction safety. *Automation in Construction* 86 (2018), 150–162.

Kai-En Lin, Zexiang Xu, Ben Mildenhall, Pratul P Srinivasan, Yannick Hold-Geoffroy, Stephen DiVerdi, Qi Sun, Kalyan Sunkavalli, and Ravi Ramamoorthi. 2020. Deep multi depth panoramas for view synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII*. Springer, 328–344.

Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. 2019. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5904–5913.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* 65, 1 (dec 2021), 99–106.

Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. 2005. *Light Field Photography with a Hand-held Plenoptic Camera.* Research Report CSTR 2005-02. Stanford university. Stanford University Computer Science Tech Report pages.

Ryan S Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. 2018. A system for acquiring, processing, and rendering panoramic light field stills for virtual reality. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

Abhilash Sunder Raj, Michael Lowney, Raj Shah, and Gordon Wetzstein. 2016. Stanford lytro light field archive.

Haoyu Ren, Mostafa El-Khamy, and Jungwon Lee. 2017. Image super resolution based on fusing multiple convolution neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 54–61.

Martin Rerabek and Touradj Ebrahimi. 2016. New light field image dataset. In *8th International Conference on Quality of Multimedia Experience (QoMEX)*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

Ana Serrano, Incheol Kim, Zhili Chen, Stephen DiVerdi, Diego Gutierrez, Aaron Hertzmann, and Belen Masia. 2019. Motion parallax for 360 RGBD video. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 1817–1827.

Hamid R Sheikh and Alan C Bovik. 2006. Image information and visual quality. *IEEE Transactions on image processing* 15, 2 (2006), 430–444.

Lixin Shi, Haitham Hassanieh, Abe Davis, Dina Katabi, and Fredo Durand. 2014. Light field reconstruction using sparsity in the continuous fourier domain. *ACM Transactions on Graphics (TOG)* 34, 1 (2014), 1–13.

Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. 2017. Learning to Synthesize a 4D RGBD Light Field From a Single Image. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yu-Chuan Su and Kristen Grauman. 2019. Kernel Transformer Networks for Compact Spherical Convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiaoyang Tian, Jie Shao, Deqiang Ouyang, and Heng Tao Shen. 2021. Uav-satellite view synthesis for cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 7 (2021), 4804–4815.

Paolo Tripicchio, Emanuele Ruffaldi, Paolo Gasparello, Shingo Eguchi, Junya Kusuno, Keita Kitano, Masaki Yamada, Alfredo Argiolas, Marta Niccolini, Matteo Ragaglia, et al. 2017. A stereo-panoramic telepresence system for construction machines. *Procedia Manufacturing* 11 (2017), 1552–1559.

Richard Tucker and Noah Snavely. 2020. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 551–560.

Suren Vagharshakyan, Robert Bregovic, and Atanas Gotchev. 2018. Light Field Reconstruction Using Shearlet Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 1 (2018), 133–147. https://doi.org/10.1109/TPAMI.2017.2653101

John Waidhofer, Richa Gadgil, Anthony Dickson, Stefanie Zollmann, and Jonathan Ventura. 2022. PanoSynthVR: Toward Light-weight 360-Degree View Synthesis from a Single Panoramic Input. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 584–592.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1905–1914.

Yunlong Wang, Fei Liu, Zilei Wang, Guangqi Hou, Zhenan Sun, and Tieniu Tan. 2018. End-to-end View Synthesis for Light Field Imaging with Pseudo 4DCNN. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

Gaochang Wu, Yebin Liu, Lu Fang, Qionghai Dai, and Tianyou Chai. 2019. Light Field Reconstruction Using Convolutional Network on EPI and Extended Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 7 (2019), 1681–1694. https://doi.org/10.1109/TPAMI.2018.2845393

G Wu, L Zhao, L Wang, Q Dai, T Chai, and Y Liu. 2017. Light field reconstruction using deep convolutional network on epi, IEEE. In *CVF Conference on Computer Vision and Pattern Recognition*.

Han Xu, Jiayi Ma, Junjun Jiang, Xiaojie Guo, and Haibin Ling. 2020. U2Fusion: A unified unsupervised image fusion network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 1 (2020), 502–518.

Jiale Xu, Jia Zheng, Yanyu Xu, Rui Tang, and Shenghua Gao. 2021. Layout-guided novel view synthesis from a single indoor panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16438–16447.

Yeohun Yun, Seung Joon Lee, and Suk-Ju Kang. 2020. Motion recognition-based robot arm control system using head mounted display. *IEEE Access* 8 (2020), 15017–15026.

Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. 2011. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing* 20, 8 (2011), 2378–2386.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2016. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging* 3, 1 (2016), 47–57.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis Using Multiplane Images. *ACM Trans. Graph.* 37, 4, Article 65 (jul 2018), 12 pages.

Wenhui Zhou, Gaomin Liu, Jiangwei Shi, Hua Zhang, and Guojun Dai. 2020. Depth-guided view synthesis for light field reconstruction from a single image. *Image and Vision Computing* 95 (2020), 103874.

Wenhui Zhou, Jiangwei Shi, Yongjie Hong, Lili Lin, and Ercan Engin Kuruoglu. 2021. Robust dense light field reconstruction from sparse noisy sampling. *Signal Processing* 186 (9 2021). https://doi.org/10.1016/j.sigpro.2021.108121

Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. 2019. Spherical View Synthesis for Self-Supervised 360° Depth Estimation. *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019* (2019), 690–699. https://doi.org/10.1109/3DV.2019.00081