FoVA-Depth: Field-of-View Agnostic Depth Estimation for Cross-Dataset Generalization

Daniel Lichy[†,◇] Hang Su[◇] Abhishek Badki[◇] Jan Kautz[◇] Orazio Gallo[◇]
[†]University of Maryland [◇]NVIDIA

Abstract

Wide field-of-view (FoV) cameras efficiently capture large portions of the scene, which makes them attractive in multiple domains, such as automotive and robotics. For such applications, estimating depth from multiple images is a critical task, and therefore, a large amount of ground truth (GT) data is available. Unfortunately, most of the GT data is for pinhole cameras, making it impossible to properly train depth estimation models for large-FoV cameras. We propose the first method to train a stereo depth estimation model on the widely available pinhole data, and to generalize it to data captured with larger FoVs. Our intuition is simple: We warp the training data to a canonical, large-FoV representation and augment it to allow a single network to reason about diverse types of distortions that otherwise would prevent generalization. We show strong generalization ability of our approach on both indoor and outdoor datasets, which was not possible with previous methods.

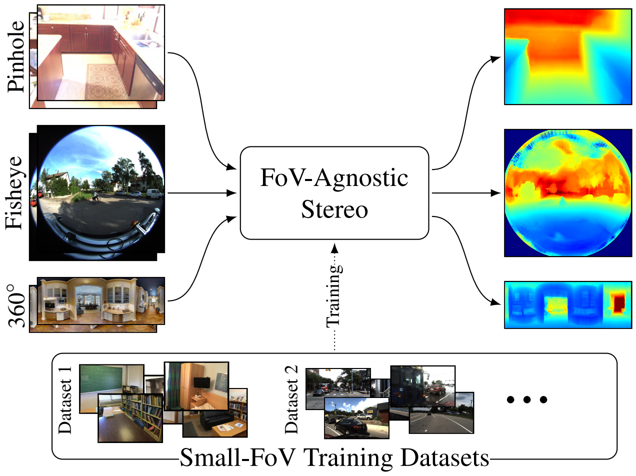


Figure 1. Our FoV-agnostic MVS model can be trained on small-FoV pinhole data and generalizes to images of various FoVs and camera models at inference time.

1. Introduction

Multi-view stereo (MVS), the task of estimating depth from multiple overlapping images, has applications in autonomous driving, robotics, real estate capture, and others. Large field-of-view (FoV) images, *e.g.*, fisheye or 360° equirectangular projection (ERP) images, and the corresponding depth estimates, capture larger portions of the scene with fewer images compared with pinhole images, making them attractive for automotive and real estate applications. The challenge, however, is the scarcity of datasets with ground truth depth for large-FoV images needed to train the depth estimation models.

Can we use the abundant small-FoV data and generalize to large-FoV fisheye and ERP data instead?

Distortion is one main challenge for generalization across FoVs. This is because the image of a given object is distorted based on the location at which the corresponding rays intersect the image plane, and the prominence of this effect is a function of the FoV, with larger FoVs inducing larger distortion. Intuitively, distortion makes it harder to learn generalizable image features and hinders matching across different images. As a result, existing methods either introduce datasets for the specific cameras they target, or can only be applied to cameras for which public datasets with GT data are available.

Assume we want to estimate depth for a fisheye image, but we only have a model trained on pinholes. We could extract several rectified pinholes from the fisheye, compute depth for each, and combine the estimates back into a large-FoV depth map, as done by Rey-Area *et al*. for monocular depth estimation [33]. This requires running inference multiple times and optimizing the results together. A more efficient solution would be to define a representation that minimizes distortion, such as a cubemap [4, 37], and train a model using only pinhole images appropriately mapped to this representation. In this paper, we describe a surprisingly simple data augmentation strategy that enables this strategy, and show that it works even for representations that do not minimize distortion, such as ERP.

https://research.nvidia.com/labs/lpr/fova-depth/

We note that most common image models, including pinhole, fisheye, and ERP are central [31], *i.e.*, they capture rays arriving at a single point, the *center of projection*. We call these Generalized Central Cameras (GCCs). This property allows us to conduct *extrinsic rotation augmentation* (ERA), which simulates rotating the camera about its center of projection at training time. ERA warps the original images to different locations of the large-FoV representation (cubemap, ERP, fisheye, *etc.*), forcing the network to learn to reason about distortions from only pinhole data.

To leverage this insight, we adapt sphere-sweeping stereo to use large-FoV representations (see Figure 3 for the case of cubemap). Specifically, we warp the input images to a canonical representation independent of the original camera model (Figure 2(b)) and train it with ERA. This procedure works for different target representations, provided that the warped pinhole images span the whole target surface during training. That is, ERA must cover all locations on the cubemap (including those that straddle multiple cube faces), or all areas of the ERP (including across its borders), which introduces the need for padding. We show that properly dealing with padding is critical, and propose convolution operators for cubemap (Section 4.4) and ERP (Section 4.3) for processing cost volumes.

We demonstrate improved cross-FoV generalization of our model with respect to the state-of-the-art in both indoor and outdoor scenarios. For the indoor case, we train on the small-FoV dataset ScanNet [9] and test on 360° ERP images from Matterport360 [3]. For outdoor, we train on the small-FoV DDAD dataset [17] and test on 180° fisheye images in the KITTI360 dataset [29].

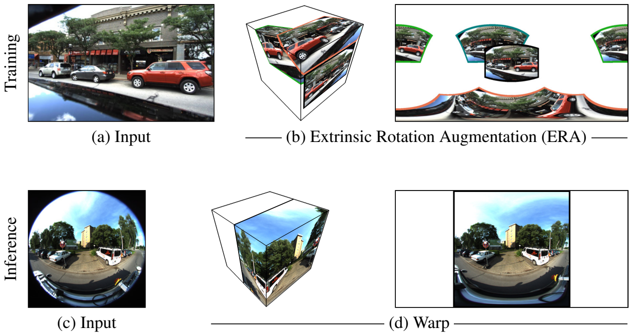
In summary, our contributions in this work include

- A generalized framework for MVS that works for arbitrary GCC images;
- The introduction of extrinsic rotation augmentations, which allow us to train on pinhole images and generalize to arbitrary GCC images, even with significantly larger FoVs;
- The necessary modifications to the convolution operations needed to perform MVS on cubemap and ERP.

2. Related Work

We discuss prior works studying small and large-FoV stereo and MVS, and spherical data representations used for other tasks. We will also discuss how some of these works fit into our generalized framework described in Section 3, their limitations, and how we resolve them.

Small-FoV MVS. Depth estimation from stereo and MVS pinhole cameras is one of the most widely studied topics in computer vision. We discuss a few inspirational works here and point the reader to surveys of traditional [13, 34] and learning-based [40, 47] approaches. Most modern learning-based approaches are based on the ideas proposed by GC-Net [21], which uses a learning-based cost-volume filtering approach for stereo depth estimation. MVSNet [46] and DeepMVS [20] extended this idea by allowing fusing information from multiple cameras. Our framework is inspired by MVSNet and can be seen as generalizing it by adapting convolution operations to work with arbitrary GCCs.



(a) Input

(b) Extrinsic Rotation Augmentation (ERA)

(c) Input

(d) Warp

Figure 2. To estimate FoV-agnostic depth, we warp the inputs to a target representation (*e.g.*, cubemap or ERP). We introduce Extrinsic Rotation Augmentations so that images are warped to all areas of this representation at training time (b). This forces the model trained on pinhole data to learn to reason about distortions in other types of images.

Many works extend and improve upon the basic idea of MVSNet, which can similarly apply to our approach. Several methods [5, 16, 30, 45] increase efficiency and accuracy by adopting a pyramidal approach, which can be seen as applying MVSNet iteratively. Others improve the feature fusion stage [18] or adopt transformer architecture at various stages of the pipeline [2, 10, 41].

Large-FoV MVS. With the widespread availability of large-FoV fisheye and 360° cameras, there has been significant interest in using them for multi-view depth estimation. Most previous learning-based approaches take advantage of a particular geometry structure like fixed stereo [28, 38] or multi-view [22, 27, 27, 38, 42–44] rigs.

Wang *et al.* [38] and Li *et al.* [28] use ERP and hybrid cubemap-ERP [37] representations, respectively, for the stereo setting where cameras are placed on top of each other. MODE [27] shows that for any two ERP images there are extrinsic rotations to simulate the cameras being on top of each other and performs image rectification. Although this can be done for general two-camera configurations, they only study it for a fixed side-by-side ERP configuration. Some multi-view works [42–44] do sphere sweeping from a central location between a fixed four-fisheye camera rig and perform cost-volume filtering using standard CNNs. Komatsu *et al.* [22] also perform sphere sweeping from a central view, however they extract features by projecting images onto an icosahedron. On the other hand, Li *et al*. [26] extract features on a spherical mesh. These works either rely on fixed camera rigs or specialized convolution operations that are non-trivial to extend. A notable exception is [6], which adapts a standard MVS architecture, CasMVSNet [16], by simply replacing the pinhole model with the ERP model.

These works rely on the availability of large-FoV GT depths for training. In contrast, Lee *et al*. [25] use semi-supervised training on real images. However, their network architecture is the same as [42], designed for a fixed rig. In this work, we outline a general MVS framework and a training strategy that works for arbitrary GCCs and can be trained using small-FoV datasets.

Spherical Data Representations. Several methods have studied the problem of applying neural networks to spherical data. The most relevant to our work are those that demonstrate applying convolutional networks to cubemap [4, 37] and ERP [39, 48] images. To handle discontinuities while applying 2D convolutions, 2D padding operations were introduced for cubemap [4, 37] and ERP [39]. We extend these padding operations for different stages of MVS pipelines and show that they are critical.

Orthogonal methods use ERP representations of spherical data, but deform the convolution shape on the ERP such that it is less deformed in the spherical domain [8, 35, 36]. Other works proposed representing spheres and spherical convolutions with spherical harmonics [7, 12]. However, both approaches require significant modifications of convolution operations and have limited support from existing libraries, making them non-trivial to extend to for MVS pipelines. For a more detailed survey on spherical data representations, we refer the reader to [14].

3. Preliminaries

We introduce generalized central cameras (GCCs), state their properties, and define generalized sphere sweeping, which is then used to describe a general MVS pipeline.

3.1. Generalized Central Camera (GCC)

An image is a measurement of the light field, where the camera intrinsics and extrinsics describe how points on the camera sensor relate to physical rays. Mathematically, light field is a function $L : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}^c$, such that $L(o, \omega)$ is the c color channel observed at position o from direction ω. We model the generalized sensor as a 2D surface, U, its intrinsics as an injective function $\phi : U \rightarrow \mathbb{S}^2$, and its extrinsics as rotation and translation, (R, t). The image captured by this camera is then defined as $I(u) = L(t, R\phi(u))$. The tuple (U, ϕ) defines the GCC. Similar definitions can be found in [15, 31]. Note that GCCs may not correspond to a physical camera, hence the term *generalized*. A number of examples fitting the GCC definition are given in Table 1.

GCC	$u \in U$	$\phi(u)$
Pinhole K	$(u_x, u_y) \in [-1, 1] \times [-1, 1]$	$K^{-1}(u_x, u_y, 1)^\intercal / \lVert K^{-1}(u_x, u_y, 1)^\intercal \rVert$
ERP	$(u_x, u_y) \in [0, 2\pi] \times [0, \pi]$	$(\sin(u_y)\sin(u_x), \cos(u_y), \sin(u_y)\cos(u_x))$
Fisheye	$(u_x, u_y) \in [-1, 1] \times [-1, 1]$	various (see Supplementary)
Cubemap	$u \in \mathbb{C} = \{x \in \mathbb{R}^3 : \lVert x \rVert_\infty = 1\}$	$u / \lVert u \rVert$
Sphere	$u \in \mathbb{S}^2 = \{x \in \mathbb{R}^3 : \lVert x \rVert = 1\}$	u

Table 1. GCC Examples. Note that the pinhole camera is actually a family of GCCs parameterized by intrinsics K.

3.2. Image Warping and Extrinsic Rotations

Given an image I from GCC (U, ϕ) and extrinsics (R, t), we can warp I to synthesize an image I' taken with a different GCC (U', ϕ') and a different extrinsic rotation R', but the same translation t, using the formula:

$$I'(u) = I(\phi^{-1}(R^{-1}R'\phi'(u)). \qquad (1)$$

Note that ϕ is surjective only when the FoV covers the full sphere, otherwise some pixels in I' are not defined. We call surjective GCC models *universal* because they capture images that allow us to synthesize the entire image of any other GCC at the same location. Examples of universal GCCs include ERP, cubemap, and sphere. In practice, images are discrete, and thus we need interpolation for u in Equation 1.

3.3. Generalized Sweeping

The foundation of all sweeping-based MVS methods is warping source images onto a reference image using depth or distance hypotheses. We focus on distance as it more naturally lends itself to FoVs larger than 180°. Specifically, consider two images I^0 and I^i, with GCCs (U^0, ϕ^0) and (U^i, ϕ^i)[1]. Let (R^i, t^i) be the transformation that takes points in the coordinate system of camera i to that of camera 0. A pixel $u \in U^0$ maps to a locus in U^i, which we can identify by lifting u to 3D using a hypothesis distance d, and project it back onto U^i. This locus is a generalized epipolar line, which for GCCs is not necessarily straight, as we show in the Supplementary. Formally, the reprojected point E^i is given by:

$$E^i(u, d) = (\phi^i)^{-1} \frac{(R^i)^{-1}\left(d\phi^0(u) - t^i\right)}{\|(R^i)^{-1}\left(d\phi^0(u) - t^i\right)\|}. \qquad (2)$$

Equation 2 allows us to test distance hypotheses. That is, $\hat{d}$ is the correct depth for u if $I^i(E^i(u, \hat{d}))$ and $I^0(u)$ are the image of the same 3D point, barring occlusions. This warping is analogous to homography warping in plane-sweeping [46]. Finally, Equation 2 can easily be adapted to have the distance hypotheses depend on u, making this formulation suitable also for multi-stage methods [6, 16, 30].

[1] We use I^i rather than I^1 to be consistent with the following sections.

3.4. General MVS Pipeline

We can then proceed to describe a general pipeline that is shared by most sweeping-based MVS methods.

Inputs. The inputs are a set of N images from known GCCs and extrinsics. Assume a standard GCC is given. We can warp all images to it using Equation 1, yielding N images I^i, with shared intrinsics (U_s, ϕ_s), and with extrinsics (R^i, t^i), where $i \in \{0 \dots N-1\}$. Rotation during warping is not strictly necessary, however, we later show it is critical for generalization (Section 4.1). We call I^0 the reference image and $I^1, \dots, I^{N-1}$ the source images.

Cost Volume. A cost volume is a data structure for facilitating pixel matching between images. We first extract features F^i for each image I^i. We then select D distance hypotheses d_j, $j \in \{0, \dots, D-1\}$. For each distance hypothesis, we warp the source features to the reference view as described in Section 3.3. The warped source features for each hypothesis distance are then stacked to construct a feature volume given by:

$$\text{Vol}^i(u, j) = F^i\left(E^i(u, d_j)\right). \tag{3}$$

Interpolation is required for evaluating F^i in Equation 3. Therefore the ability to allow efficient interpolation becomes critical, as we further discuss in Section 4.2. The reference feature volume $\text{Vol}^0(u, j)$ is created by just repeating F^0 along the distance hypothesis dimension. Next, we can fuse the features volumes from all views to form a cost volume:

$$\text{CV}(u, j) = f_{\text{fusion}}(\text{Vol}^0(u, j), \dots, \text{Vol}^{N-1}(u, j)). \tag{4}$$

The intuition is that if d_j is the correct distance for pixel u and the features are view-invariant, then the value (u, j) of all feature volumes should be the similar. f_{fusion} is a fusion function that measures such feature similarity. Many MVS methods [16, 46] use variance as the fusion function, where the variance should be low across volumes when d_j is the correct depth at pixel u. Another choice, which we adopt in our experiments, is Group-Wise Correlation [18, 30].

Distance Regression. Next, a 3D network is applied to the cost volume followed by a softmax operation to form a probability volume PV, where $\text{PV}(u, j)$ denotes the probability that the distance of u is d_j among all distance hypotheses. We can then generate the distance estimation with a weighted sum: $d^*(u) = \sum_j \text{PV}(u, j) d_j$.

Monocular Refinement. Stereo matching can fail due to various reasons, *e.g.*, flat textureless regions, occlusions, and dynamic objects. Furthermore, estimations are often output at reduced resolutions due to efficiency limitations. A 2D network can take the initial estimation, and optionally the reference image, and output a refined final estimation.

4. Method

Our goal is to train an end-to-end MVS network that can operate on images taken by any GCCs. We use readily available small-FoV datasets for training, and design our model to generalize to large-FoV datasets at inference time. To accomplish this, we propose warping the input images to a canonical representation independent of their original representations. Although many choices exist for the canonical representation, we develop our approach with ERP (Section 4.3) and cubemap (Section 4.4) due to their desirable properties outlined in Section 4.2. We also introduce a data augmentation strategy at training time that is critical to generalization (Section 4.1). After warping, we can apply the general MVS pipeline described in Section 3.4 adapted to the canonical representation (Figure 3).

4.1. Extrinsic Rotation Augmentation

The first step of our approach is to warp each image to the canonical representation using Equation 1, where (U, ϕ) is the original GCC of the image and $(U', \phi') = (U_s, \phi_s)$. Since we train on small-FoV data, if we only warp the images with $R' = R$, the resulting images may not exhibit the types of distortion seen in wide-FoV images, *e.g.*, on edges and corners of cubemap or near the top and bottom of ERP. This hinders the ability of the network to generalize. To mitigate this issue, we propose *extrinsic rotation augmentation* (ERA) as a type of data augmentation during training. In a nutshell, we warp the image to the canonical representation using a random rotation, R', so as to span all the regions of the canonical representation, forcing the network to learn about distortion (Figure 2).

4.2. Canonical Representations

Any GCC can be used as a canonical representation. However, we identify three properties that an ideal canonical representation for the MVS pipeline should possess:

1. It should be universal (Section 3), so we can represent images with FoVs up to full 360°;
2. It should be compatible with existing deep networks to allow the use of strong architectures and enable transfer learning;
3. It should allow for interpolation at arbitrary locations efficiently for fast construction of cost volumes.

Property 1 requires our representation to be bijective to the sphere, and therefore any GCC can be warped to it without cropping any pixels. An intuitive option is some meshing of the sphere, *e.g.*, an icosphere [22]. Instead, we turn to the ERP and cubemap representations. ERP and cubemap cover 360° FoVs, thus satisfying Property 1. Moreover, standard

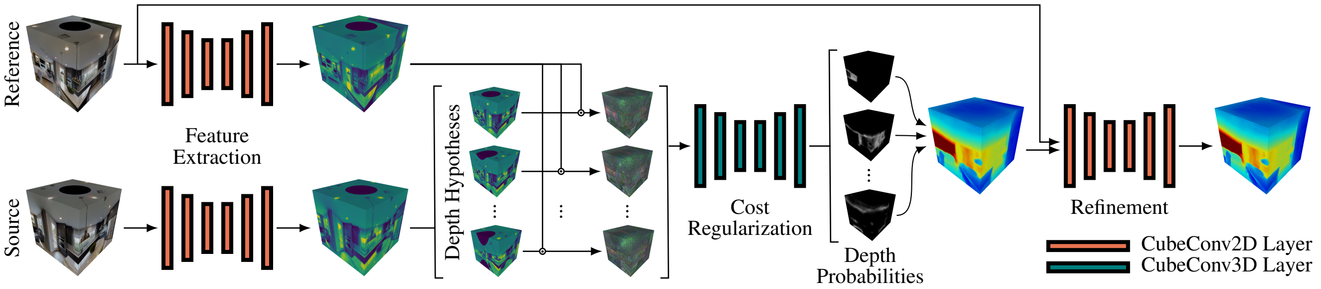


Figure 3. Our MVS pipeline. Here we only show the architecture for cubemap, but the same pipeline can be used for ERP by simply switching the convolution operations.

CNN architectures can be trivially adapted to operate efficiently on ERP [37, 48] and cubemap [4, 37]. Finally, these two GCCs lend themselves to efficient GPU-accelerated interpolation.

In principle, low distortion is another desirable property for a GCC. This would favor cubemap, which incur minimal distortion due to the small FoV of each face. However, empirically we found that the ERA strategy allows our models to perform well even for ERP despite the larger distortion.

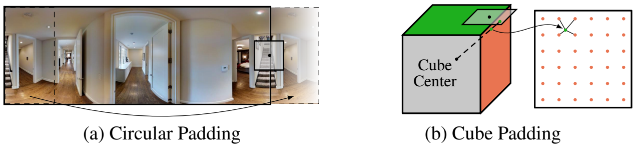
4.3. Equirectangular Projection (ERP)

ERP is widely used for 360° panoramas in the industry, because it is intuitive to visualize and easy to work with. Since ERP data are represented with standard 2D images, we can simply adopt standard 2D CNN operations and architectures without any modifications. However, without proper care for padding, the quality of the results degrades measurably as the errors along the sides increase.

Circular Convolution. ERP captures the full 360° FoV and we can leverage this property to mitigate the issue above. Because of their circular nature, the left and right boundaries of ERP images connect to form a continuous horizon, which allows us to pad at the image boundaries with content from the opposite sides. Note that to really achieve a continuous horizon, circular padding needs to be done at every convolutional layer, not just at the input layer. While it is possible to preemptively pad the input up to the width of the receptive field of the CNN, this can incur substantial overhead. Instead, we use circular padding for each layer in isolation, and only pad the necessary amount. We call the convolution layer that uses this padding strategy Circular Convolution, or CircConv (Figure 4(a)). Formally, given an input feature map F of height H and width W, we define the horizontally padded feature map as:

$$\hat{F}[i,j] = F[i,(j-P) \bmod W], \tag{5}$$

where $i,j \in [0,H) \times [0,W+2P)$. We use zero padding for the vertical sides only, though strategies for the top and bottom are also possible [39].



(a) Circular Padding (b) Cube Padding

Figure 4. (a) We pad a side of the ERP by replicating pixel values from the opposite side. (b) We pad the green cube face with the interpolated value of the green point projected on to the orange face. Transparent squares indicate the convolution filters.

CircConv3D. For the cost-regularization network, we need the 3D analog of CircConv. To achieve that, we simply split the volume into a list of ERP images, one for each depth hypothesis, and apply CircConv to them individually. Last, we can concatenate the list to form a volume and apply a standard 3D convolution.

4.4. Cubemap Representation

Standard CNN architectures can work for the cubemap data in the MVS pipeline by simply operating separately on each face. In this way, though, the relationship between faces is lost, causing excess artifacts along the cube edges and overall more restrictive receptive fields. Similar to CircConv in the case of ERP, we need a mechanism to bring back the continuity along boundaries.

Cube Convolution. We resolve this using a similar strategy to the ERP case: we define a convolution operator on the cube that wraps the filters around the cube edges, in a similar manner that CircConv wraps filters around the sides of the ERP. Just like for ERP, this is implemented using padding followed by standard convolution. In particular, we adapt the spherical padding of Wang *et al.* [37], but replace bilinear interpolation with nearest neighbor interpolation for an efficient implementation. This is equivalent to the cube padding of Cheng *et al.* [4] for the case of 3 × 3 convolution, which is used in all layers in our architecture except the first one of the feature extractor.

Concretely, assume a c-channel cubemap image I_{cube}

taken by a camera, $(\mathbb{C}, \phi_{\text{cube}})$, is represented as an array F of shape $(6, c, W, W)$. Let us consider a $(2P+1)\times(2P+1)$ CubeConv to the 0-th face, $F_0 = F[0, :, i, j]$. To pad F_0 with padding of size P, we first extend the sampling locations beyond the cube face to $i, j \in [-P, W+P) \times [-P, W+P)$. Then we project the extended sample locations back onto cube and interpolate their values to produce the padded 0-th face, denoted as $\hat{F}_0$:

$$\hat{F}_0 = I_{\text{cube}}(\phi^{-1}_{\text{cube}}(\phi_0(\frac{2i}{W} - 1, \frac{2j}{W} - 1))), \tag{6}$$

where ϕ_0 is the intrinsics of a pinhole camera corresponding to the 0-th cube face. Figure 4(b) shows this process. $I_{\text{cube}}(\cdot)$ is evaluated at subpixel locations with interpolation. With the padding in place, each face can be filtered individually and together generates an output of shape $(6, c', W, W)$.

CubeConv3D. The 3D case is handled by treating the volume as a list of cubes analogous to CircConv3D.

Face Culling. In practice, when training on small-FoV images, some cube faces will be empty. We skip these faces to reduce time and memory consumption. These skipped faces may still be queried for cube padding but can simply be treated as all zeros.

4.5. Reciprocal Tangent Sampling

Constructing and processing 3D cost volumes is expensive, so the number of distance hypotheses is limited, and it is critical to sample the distance range efficiently. We therefore propose using *reciprocal tangent sampling*, defined as

$$\{d_j\} = f_{\text{RT}}(\mathcal{U}(f^{-1}_{\text{RT}}(d_{\min}), f^{-1}_{\text{RT}}(d_{\max}), D)), \tag{7}$$

where $f_{\text{RT}}(x) = \frac{2}{\pi \tan(\frac{\pi}{2}x)}$, and $\mathcal{U}(v_{\min}, v_{\max}, D)$ is a function that uniformly samples D points between $v_{\min}$ and $v_{\max}$. The intuition is that the angular disparity between two images is roughly related to distance by the inverse tangent function. We show in Section 5.4 that this sampling strategy works better than the commonly used inverse distance sampling for unbounded scenes in the datasets we evaluate. We discuss this sampling strategy and the intuition behind it in greater detail in the Supplementary.

5. Evaluation and Results

5.1. Datasets

Our primary goal is to train our network on small-FoV datasets and evaluate on large-FoV datasets. For the indoor scenario, we train on ScanNet [9] (small-FoV) and test on Matterport360 [33] (large-FoV). For the outdoor scenario, we train on DDAD [17] (small-FoV) and test on KITTI-360 [29] (large-FoV). Due to the unavailability of GT depths aligned with the fisheyes in KITTI-360, and the presence of dynamic objects in multi-view images selected from different time instances, we only test generalizability for outdoor scenarios qualitatively.

ScanNet consists of 94,212 stereo pairs from 1,201 indoor scenes. The images are all captured with pinhole cameras with FoVs $< 60°$. We use the same data split as [23].

Matterport360 consists of 9,684 RGB-D ERP images from 90 building-scale scenes. We only use the 18 test scenes in the official split. It was originally designed for 360° single-image depth estimation, so we choose any two images within 2 meters of each other as stereo pairs. The images capture the entire 360° view, except for the regions around the poles. We choose Matterport360 over existing large-FoV stereo datasets such as Deep360 [27] and Stanford2D3D [1] in order to analyze the large-baseline settings, where distortion across views differs significantly. The stereo image pairs in Deep360 and Stanford2D3D are nearby and have a fixed orientation relative to each other.

DDAD consists of 200 driving sequences captured from 6 pinhole cameras. We use the first 150 scenes for training and scenes 150–159 for validation. We use every image as a reference image and select the frame forward in time that has a camera displacement closest to 1 meter as the source view. For three-view experiments, we additionally select the frame backward in time with a camera displacement closest to 1 meter as the second source view.

KITTI-360 consists of 11 driving sequences captured using two 180° fisheye cameras on the sides of the vehicle and a front-facing perspective stereo camera. We build image pairs or triplets in the same manner as in DDAD.

5.2. Implementation Details

All models are implemented in Pytorch. We use NVDiffRast [24] for fast interpolation of the cubemap. We apply $L1$ loss on log depth for the estimated depth maps both before and after monocular refinement. We use 48 distance hypotheses and a cube size of $6 \times 256 \times 256$ or an ERP size of 384×1024.

Networks. For the feature extractor, we use the first three blocks of ResNet34 [19] with all convolution layers replaced with CircConv or CubeConv layers, depending on the chosen canonical representation. We then use transposed convolution layers to upsample all feature maps to 1/4 input resolution and concatenate them to form the final image feature. For the cost regularization network, we use the MVSNet architecture [46] with all 3D convolution layers replaced with CircConv3D or CubeConv3D layers. Finally, we use the MiDaS [32] architecture for the refinement network, again with all convolutions replaced with CircConv

Method	AbsRel ↓	SqRel ↓	RMSE ↓	$\delta 1$ ↑	$\delta 2$ ↑	$\delta 3$ ↑
MODE [27]	0.459	0.873	1.292	0.23	0.494	0.763
360MVSNet-FCN [6]	0.477	1.256	1.191	0.579	0.745	0.839
360MVSNet-ResNet	0.367	0.821	0.994	0.654	0.794	0.869
360MVSNet-ResNet-ERA	0.236	0.39	0.779	0.724	0.846	0.907
Ours Cube	0.232	0.445	0.763	0.745	0.857	0.911
Ours ERP	0.236	0.465	0.813	0.736	0.853	0.911
Ours Cube+R	0.186	0.289	**0.665**	0.78	0.879	0.925
Ours ERP+R	**0.170**	**0.249**	0.668	**0.791**	**0.895**	**0.944**

Table 2. Comparison of methods on Matterport. +R = with monocular refinement

or CubeConv. More implementation details can be found in the Supplementary.

Metrics. We use standard metrics widely used for depth estimation [11]. The metrics include AbsRel (absolute relative error), RMSE (root mean square error), and percentage measures $\max(\frac{\hat{y}}{y}, \frac{y}{\hat{y}}) < \delta$ for $\delta = 1.25, 1.25^2, 1.25^3$.

5.3. Baseline Evaluations

Quantitative Comparison for Indoor Scenes. Since we are the first to tackle the problem of training on small-FoV stereo images and generalizing to large-FoV images, there are no off-the-shelf baselines. Therefore, we propose two baselines. The first is the state-of-the-art rectified ERP stereo method, MODE [27], which we retrain using ScanNet. Since MODE is only designed to operate on rectified ERP images, we warp the ScanNet images to rectified ERP images using Equation 3. More details of this procedure can be found in the Supplementary. The second is the state-of-the-art 360 MVS method, 360MVSNet [6], which uses ERP as its underlying representation. Since code is not available, we reimplement it ourselves and train on ScanNet. We also improve 360MVSNet with an upgraded ResNet feature extractor, 360MVSNet-ResNet, and our ERA, 360MVSNet-ResNet-ERA. We compare both of these baselines to our method using the ERP representation and our method using the cubemap representation, both with and without monocular refinement. The results can be found in Table 2.

Surprisingly, we observe similar quality between our method with the cubemap and the ERP, despite the larger distortion of the ERP. This demonstrates the power of the ERA in learning to deal with distortion even from pinhole images. Once we upgrade 360MVSNet's feature extractor to the same ResNet as our models and apply our proposed ERA we obtain similar performance to our ERA model without refinement. This is not surprising since this is basically the same as our ERP model with the additional cascade processing. Moreover, we observe that monocular refinement actually performs better than the cascade strategy on the evaluated datasets. Note that, due to the required image rectification, training MODE with ERA is not trivial.

Method	Ours Cube	Ours Cube+R	Ours ERP	Ours ERP+R	360MVSNet-ResNet
w/ ERA	**0.232**	**0.186**	**0.236**	**0.17**	**0.235**
w/o ERA	0.413	0.375	0.417	0.376	0.367

Table 3. Comparison of models with and without extrinsic rotation augmentation. AbsRel numbers are reported here.

Method	Ours Cube	Ours Cube+R
Full model	0.232	0.186
w/o pre-training	0.279	0.243
w/o cube padding	0.381	0.466
w/o recip. tangent sampling	0.262	0.220

Table 4. Ablations for the cubemap model. AbsRel numbers are reported here.

Qualitative Comparison for Outdoor Scenes. To demonstrate the advantages of training on real small-FoV data versus synthetic large-FoV data, we finetune our ScanNet-trained model with images from DDAD for up to 100 epochs. We compare it to the MODE model pre-trained by the authors on their large-FoV synthetic driving dataset, Deep360 [27]. We evaluate both models qualitatively on the large-FoV KITTI360 dataset. Qualitative results are shown in Figure 5.

5.4. Ablation Studies

Role of Extrinsic Rotation Augmentation. We show that ERA is essential for generalization for both ERP and cubemap. Table 3 shows the comparison of models with and without the augmentation. ERA helps all models generalize both with and without monocular refinement.

Role of CubeConv. To investigate the benefit of CubeConv, we train a version of our model with standard convolution layers and zero padding. There is a drop in performance both with and without monocular refinement as shown in Table 4. Qualitatively, we see large seams form around the edges of the cubemap (see Supplementary).

Role of Pre-training. One advantage of CubeConv is that it is compatible with standard network architectures. This enables transfer learning, which allows training high-performing models with limited training data. To demonstrate this, we train a variant of our model with random initialization of the feature extractor and monocular refinement network instead of using ImageNet pre-trained weights. We again see a large drop in performance both with and without monocular refinement (Table 4).

Role of Reciprocal Tangent Sampling. We compare our cubemap model trained with reciprocal tangent sampling versus inverse distance sampling. We observe a significant benefit of adopting reciprocal tangent sampling (Table 4).

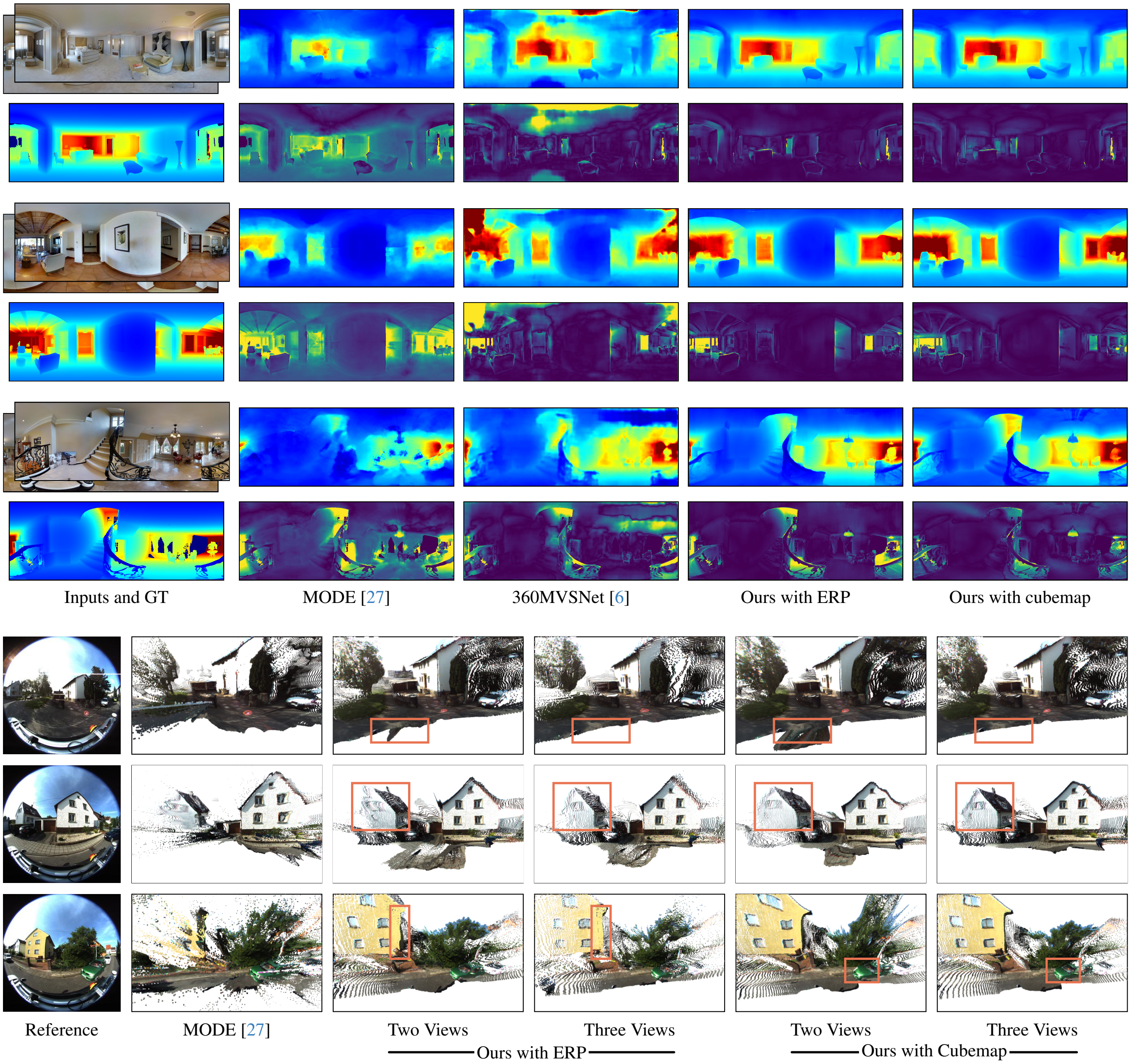


Figure 5. Generalization results of our approach on Matterport360 (top) and KITTI-360 (bottom). For indoor scenes, our approach trained with ERA for both ERP and cubemap representations outperform competing approaches [6, 27]. For outdoor scenes, our approach generalizes better than MODE [27] trained only on large-FoV synthetic data. Our approach can naturally use additional views. Our 3-view stereo shows better reconstructions (see the highlighted regions) for both ERP and cubemap representations.

6. Conclusions

In this work, we introduce a multi-view stereo framework for Generalized Central Cameras that can be trained on small-FoV pinhole data and generalize to any cameras, including ones with large-FoV. We show that a surprisingly simple data augmentation strategy, extrinsic rotation augmentation, is the key to enabling this generalization capability. We adapt our MVS framework for ERP and cubemap representations by introducing efficient padding operations for convolutions for different stages of an MVS pipeline. We see utility for this model in automotive and real-estate applications. Furthermore, the method can be easily extended to leverage improvements proposed for standard pinhole sweeping methods, *e.g.*, multi-scale and self-supervised techniques.

References

[1] Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 6

[2] Chenjie Cao, Xinlin Ren, and Yanwei Fu. MVSFormer: Multi-view stereo by learning robust image features and temperature-based depth. *Transactions of Machine Learning Research*, 2023. 2

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision*, 2017. 2

[4] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3, 5

[5] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[6] Ching-Ya Chiu, Yu-Ting Wu, I-Chao Shen, and Yung-Yu Chuang. 360MVSNet: Deep multi-view stereo network with 360° images for indoor scene reconstruction. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023. 3, 7, 8

[7] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018. 3

[8] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[10] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. TransMVSNet: Global context-aware multi-view stereo network with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 7

[12] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[13] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Found. Trends. Comput. Graph. Vis.*, 2015. 2

[14] Shaohua Gao, Kailun Yang, Hao Shi, Kaiwei Wang, and Jian Bai. Review on panoramic imaging and its applications in scene understanding. *IEEE Transactions on Instrumentation and Measurement*, 2022. 3

[15] Michael D. Grossberg and Shree K. Nayar. A general imaging model and a method for finding its parameters. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[16] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4

[17] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6

[18] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[20] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[21] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[22] Ren Komatsu, Hiromitsu Fujii, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama. 360° depth estimation from multiple fisheye images with origami crown representation of icosahedron. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020. 2, 4

[23] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[24] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 2020. 6

[25] Jaewoo Lee, Daeul Park, Dongwook Lee, and Daehyun Ji. Semi-supervised 360 depth estimation from multiple fisheye cameras with pixel-level selective loss. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022. 3

[26] Ming Li, Xuejiao Hu, Jingzhao Dai, Yang Li, and Sidan Du. Omnidirectional stereo depth estimation based on spherical deep network. *Image and Vision Computing*, 2021. 3

[27] Ming Li, Xueqian Jin, Xuejiao Hu, Jingzhao Dai, Sidan Du, and Yang Li. MODE: Multi-view omnidirectional depth estimation with 360° cameras. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 6, 7, 8

[28] Taili Li, Yali Xue, and Zhi Xiong. Panoramic stereo matching network based on bi-projection fusion. In *China Automation Congress (CAC)*, 2022. 2

[29] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 2, 6

[30] Zhenxing Mi, Chang Di, and Dan Xu. Generalized binary search network for highly-efficient multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4

[31] Srikumar Ramalingam and Peter F. Sturm. A unifying model for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 2, 3

[32] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 6

[33] Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 360MonoDepth: High-resolution 360° monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 6

[34] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 2002. 2

[35] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[36] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[37] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. BiFuse: Monocular 360 depth estimation via bi-projection fusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 5

[38] Ning-Hsu Wang, Bolivar Solarte, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 360SD-Net: 360° stereo depth estimation with learnable cost volume. *arXiv preprint arXiv:1911.04460*, 2019. 2

[39] Tsun-Hsuan Wang, Hung-Jui Huang, Juan-Ting Lin, Chan-Wei Hu, Kuo-Hao Zeng, and Min Sun. Omnidirectional CNN for visual place recognition and navigation. In *International Conference on Robotics and Automation (ICRA)*, 2018. 3, 5

[40] Xiang Wang, Chen Wang, Bing Liu, Xiaoqing Zhou, Liang Zhang, Jin Zheng, and Xiao Bai. Multi-view stereo in the deep learning era: A comprehensive review. *Displays*, 2021. 2

[41] Xiaofeng Wang, Zheng Zhu, Guan Huang, Fangbo Qin, Yun Ye, Yijia He, Xu Chi, and Xingang Wang. MVSTER: Epipolar transformer for efficient multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[42] Changhee Won, Jongbin Ryu, and Jongwoo Lim. OmniMVS: End-to-end learning for omnidirectional stereo matching. *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3

[43] Changhee Won, Jongbin Ryu, and Jongwoo Lim. SweepNet: Wide-baseline omnidirectional depth estimation. *International Conference on Robotics and Automation (ICRA)*, 2019.

[44] Changhee Won, Jongbin Ryu, and Jongwoo Lim. End-to-end learning for omnidirectional stereo matching with uncertainty prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 2

[45] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[46] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 4, 6

[47] Qingtian Zhu. Deep learning for multi-view stereo via plane sweep: A survey. *arXiv preprint arXiv:2106.15328*, 2021. 2

[48] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. OmniDepth: Dense depth estimation for indoors spherical panoramas. In *European Conference on Computer Vision (ECCV)*, 2018. 3, 5