Spherical View Synthesis for Self-Supervised 360^o Depth Estimation

Nikolaos Zioulis[1, 2], Antonis Karakottas[1], Dimitrios Zarpalas[1], Federico Alvarez[2], and Petros Daras[1]

[1]Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), Greece
[2]Signals, Systems and Radiocommunications Department (SSRD), Universidad Politécnica de Madrid (UPM), Madrid, Spain

Abstract

Learning based approaches for depth perception are limited by the availability of clean training data. This has led to the utilization of view synthesis as an indirect objective for learning depth estimation using efficient data acquisition procedures. Nonetheless, most research focuses on pinhole based monocular vision, with scarce works presenting results for omnidirectional input. In this work, we explore spherical view synthesis for learning monocular 360^o depth in a self-supervised manner and demonstrate its feasibility. Under a purely geometrically derived formulation we present results for horizontal and vertical baselines, as well as for the trinocular case. Further, we show how to better exploit the expressiveness of traditional CNNs when applied to the equirectangular domain in an efficient manner. Finally, given the availability of ground truth depth data, our work is uniquely positioned to compare view synthesis against direct supervision in a consistent and fair manner. The results indicate that alternative research directions might be better suited to enable higher quality depth perception. Our data, models and code are publicly available at https://vcl3d.github.io/SphericalViewSynthesis/.

1. Introduction

Data-driven approaches are producing impressive results in a variety of vision related tasks. Convolutional neural networks (CNNs) are trained to match – and even surpass – human perception, managing to infer three-dimensional (3D) information solely from monocular images. However, their performance is closely related to the availability of high quality training samples, which for certain tasks is tedious, expensive or even outright impossible. While landmark annotations can be crowd-sourced, densely annotating images with ground truth depth values fits the latter category. As a result, fully supervised depth learning has only been demonstrated in small scale datasets [31], usually without pixel perfect depth measurements [33], or otherwise in generated synthetic datasets [45] that, nonetheless, need to overcome the synthetic-to-real domain gap.

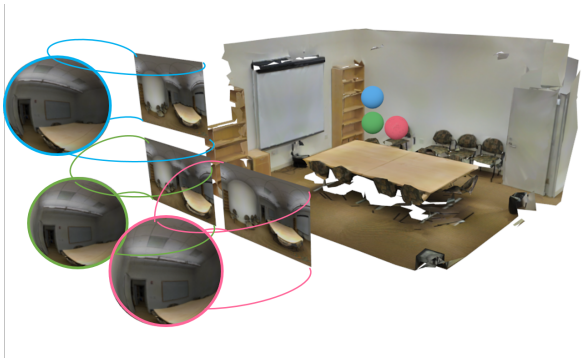

Figure 1. Spherical view synthesis for self-supervised depth estimation using 360^o stereo images. Considering spherical viewpoints within a 3D scene, we render color images from consistent baselines. Starting from a central viewpoint (green), we explore both vertical (cyan) and horizontal (pink) setups, as well as the trinocular case. Indicative equirectangular projection images as observed by the 3 spherical viewpoints are presented on the left, while the 3D scene and viewpoint positions within it, on the right.

Naturally, a great body of work has identified this challenge and focused on overcoming it with self-supervision, using an indirect objective to infer depth, namely, view synthesis. Accurately explaining imaged content from a different viewpoint relies on 3D information, and by extension, accurate depth. Even though view synthesis supervision relies on a set of assumptions (diffuse materials, absence of occlusions, static scenes) that do not necessarily hold for real world acquired data, convincing depth estimation results have been presented without using any ground truth. Earlier efforts relied on synchronized stereo cameras capturing static scenes [9, 11], introducing view synthesis via inverse image warping and paving the way follow-up works.

Circumventing the need for stereo data acquisition, more recent works [47] only rely on video input for learning to in-

fer depth. To achieve this they learn to estimate the camera's motion jointly with estimating the observed scene's depth, and are supervised by synthesizing future and/or past views. While an abundance of data are readily available for these structure-from-motion learning methods, they still need to overcome the violation of the static scene assumption.

The absolute majority of this line of research has focused on traditional pinhole cameras, disregarding self-supervised depth estimation for omnidirectional input, apart from [41], which, however, uses a cube map (*i.e.* pinhole) representation. Spherical view synthesis is relatively unexplored as most works focused on catadioptric or cylindrical cameras. It is also challenging due to the inherent distortions when applied to two-dimensional images, which in turn manifest into severe self-occlusions. Further, due to the content's spherical nature, the irregular disparity patterns that it exhibits hinder efficient learning, especially for horizontal baselines, while the singularities at the epipoles prevent coherent gradient flows when using inverse image warping.

In this work, we explore spherical view synthesis and demonstrate its applicability for self-supervised spherical depth estimation. Summarizing, our contributions are:

- The full spherical disparity model is presented using a purely geometric derivation.
- A robust supervision scheme is developed for spherical view synthesis using depth-image-based rendering (DIBR) and spherical attention.
- Unlike inefficient and resource consuming spherical learning approaches, our network design incorporates a straightforward way to make our model aware of its spherical nature.
- Besides offering a large 360^o stereo dataset, our work is uniquely posed to compare the effectiveness of view synthesis and direct supervision. We perform a fair and consistent evaluation and present its results.

2. Related Work

Learning with spherical content: Applying CNNs to spherical content is accomplished by warping it to a regular grid. MPEG-OMAF [34] defines two projection formats for 360^o images, the cubemap and equirectangular (ERP) projections.

While cubemaps can be straightforwardly fed into a CNN, and then re-merged back into 360^o as in [24], they still suffer from cubemap distortion and discontinuity artifacts. For the latter, cube padding [3] can explicitly aid the network into connecting the cube faces, enabling global reasoning. Similarly, circular padding [42] has been used when applying convolutions directly to the ERP image.

A novel direction is to bypass learning on spherical data and instead, adapt models trained on perspective images to the 360^o domain. Initially, [36] regressed per row rectangular filters from the pre-trained ones, at the expense of increasing the model's size and complexity (multiple filters for a single activation map) and suffering from regression approximation. It was recently extended [37] to transfer 2D CNN models by producing functions that map weights to each row, while preserving inter-channel information exchange, and overcoming some of the previous disadvantages, albeit still taking a model size hit (even though significantly reduced). Another approach is adapting the input data to the 360^o domain [27], yet it was not demonstrated for full spherical images, but rather only for panoramic ones.

Another direction is training rotation equivariant CNNs either using graph-based learning [16] or employing spectral learning approaches, with two notable works using spherical harmonics [7] and spherical cross correlation with Fast Fourier Transforms (FFTs) [4] to achieve expressive training on the sphere. Still, their high memory footprint hinders applicability due to limited input resolutions.

As a result, more efficient approaches resorted to kernel distortion [38], tangent plane kernels [5], kernel resampling [46] or ERP specific dilations [8]. However, as presented in [37], all these approaches are valid only for the first layers, as the CNN's non-linearity distorts the pure spherical representation as the network deepens, breaking the assumptions they are designed for (i.e. the features' spherical smoothness). In addition, inefficient implementations [46] introduce problems during training (very small batch size and low run-time performance). Instead, we resort to a more explicit and efficient solution to make the network aware of the data spherical nature, by exploiting recent research related to CNNs' capacity to self-localize their features, and also utilize spherical attention to allow for distortion aware supervision in the ERP domain.

Monocular self-supervised (spherical) depth: The seminal works of [9] and [11] first demonstrated that view synthesis can serve as the supervisory signal for monocular depth estimation. This has attracted a lot of attention from the research community given the difficulty in obtaining high quality real world depth measurements. Both [9] and [11] used perspective horizontal stereo data and employed either approximately [9] or locally [11] differentiable image warping [14] to synthesize the reconstructed views.

A novel solution was introduced by [47] that extended view synthesis supervision to unstructured video datasets by simultaneously predicting inter-frame pose. However, learning to estimate depth purely from video breaks the static scene assumption and necessitates the use of an attention mechanism for foreground motion between consecutive frames. More recent iterations of this direction added scale normalization and removal of the separate pose estimation branch [40], 3D geometric constraints between the pre-

dicted depths [23], epipolar constraints [29], additional feature reconstruction supervision [44], stereo matching constraints [43] or explicitly used two consecutive frames as input [28].

Prevalent for all the above methods is the reconstruction loss of synthesized views via inverse warping through a stereo disparity model or explicit 3D transformations and projections. Disregarding the challenging Lambertian surfaces assumption, inverse warping does not gracefully handle occlusions, which are only implicitly addressed (e.g. explainability/visibility masks, left-right consistencies). This has a detrimental effect for spherical images as occlusions are magnified due to distortion. Instead, we rely on a soft rendering approach to synthesize the supervisory views.

While a large body of work exists for traditional perspective images, scarce research has addressed depth estimation from spherical panoramas. The most apparent issue is the unavailability of data, and thus, two concurrent works addressed 360^o depth estimation by generating data via rendering existing 3D datasets. Two baseline models were presented in [49] after creating a large dataset of color and depth pairs using a mix of synthetic and real scenes. Further, [41] utilized the more recent advances in depth estimation from videos and rendered videos from a purely synthetic 3D dataset. Still, [49] simply applied a CNN on ERP images while [41] explicitly used a cubemap representation and relied on previous works on perspective depth video learning, but with cubemap constraints.

Indirect supervision through spherical view synthesis has not been explored yet for learning monocular 360^o depth estimation. Previous works mainly focused on estimating depth from fisheye [19] or cylindrical [48] stereo setups and utilized the corresponding disparity models. For the full spherical setting a complete disparity model has not been considered as prior work only focused in extracting depth measurements and not synthesizing views. Consequently, 360^o vertical stereo setups [17] were preferred due to their simpler disparity model that requires no rectification. Works using 360^o horizontal stereo [20, 16] relied only on horizontal disparity modeling which is sufficient to triangulate depth values after rectification. On the other hand, horizontal spherical view synthesis introduces distortions which manifest as vertical disparity. In this work we present and explore the complete spherical disparity model for both stereo placements under a view synthesis, self-supervised 360^o depth estimation learning context.

3. Self-supervised Spherical Depth

3.1. Spherical Disparity Model

We define a spherical image through its ERP on a 2D grid as shown in Fig. 2. Each image's local 3D coordinate system in spherical $\boldsymbol{\rho} = (r, \phi, \theta)$ and Cartesian $\mathbf{v} = (x, y, z)$

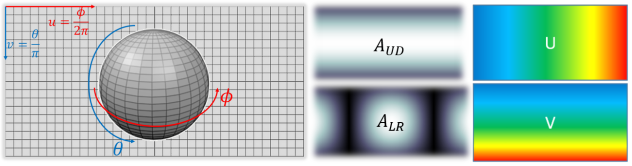

Figure 2. Spherical $\boldsymbol{\rho} = (\phi, \theta)$ longitudinal and latitudinal coordinates aligned with the image grid's equirectangular coordinates $\mathbf{p} = (u, v)$ respectively (**left**). The spherical attention masks A_{UD} for vertical and A_{LR} for horizontal stereo placements as defined in Eq. 5 respectively (**middle**). Both attenuate towards the singularities, while the A_{LR} also includes distortion related attenuation. Image grid coordinate feature maps for the horizontal (u) and vertical (v) image grid directions (**right**).

coordinate systems are given in Eq. 1. An ERP image's width w and height h span $w \times h := 2\pi \times \pi$ radians at the $[0, 2\pi]$ and $[0, \pi]$ ranges respectively, covering a complete spherical view with $\varphi = 2\pi/w$ the horizontal and $\vartheta = \pi/h$ the vertical angular resolutions respectively. Columns correspond to constant longitude/azimuth (ϕ) angles, while rows to constant latitude/elevation (θ) angles. Each pixel $\mathbf{p} = (u, v)$ can be mapped to angular spherical coordinates $\boldsymbol{\rho} = (\phi, \theta)$ as $(u\varphi, v\vartheta)$ and vise versa. This linear mapping between image domain pixels $\mathbf{p}$ and spherical domain angular coordinates $\boldsymbol{\rho}$ allows for straightforward transitions between image and spherical based operations. We will therefore omit any explicit conversions between them in the following text. Contrary to perspective images, 360^o depth is defined as the 3D Euclidean distance to a point, which corresponds to the radius r in spherical coordinates.

$$\begin{bmatrix} r \\ \phi \\ \theta \end{bmatrix} = \begin{bmatrix} (x^2 + y^2 + z^2)^{1/2} \\ \arctan(x/z) \\ \arccos(y/r) \end{bmatrix}, \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\sin(\phi)\sin(\theta) \\ r\cos(\theta) \\ r\cos(\phi)\sin(\theta) \end{bmatrix} \quad (1)$$

Spherical stereo considers physical displaced spherical viewpoints that image the same scene. They are positioned with a known baseline in either horizontal or vertical placements. Fig. 3 shows both of these placements as well as the projection of a 3D point on each displaced viewpoint. Disparities $\boldsymbol{\gamma} = (\gamma_\phi, \gamma_\theta)$ correspond to angular differences in the angular spherical coordinates (ϕ, θ) measured in radians. Spherical disparities $\boldsymbol{\gamma}$ can be analytically derived from a source viewpoint $\mathbf{v}_{src}$ with respect to an unrotated target viewpoint $\mathbf{v}_{tgt}$ according to their baseline $\mathbf{b} = \mathbf{v}_{src} - \mathbf{v}_{tgt}$ by calculating the partial derivatives of the spherical coordinates with respect to the Cartesian ones:

$$\begin{bmatrix} \partial r \\ \partial \phi \\ \partial \theta \end{bmatrix} = \begin{bmatrix} \sin(\phi)\sin(\theta) & \cos(\theta) & \cos(\phi)\sin(\theta) \\ \frac{\cos(\phi)}{r\sin(\theta)} & 0 & \frac{-\sin(\phi)}{r\sin(\theta)} \\ \frac{\sin(\phi)\cos(\theta)}{r} & \frac{-\sin(\theta)}{r} & \frac{\cos(\phi)\cos(\theta)}{r} \end{bmatrix} \begin{bmatrix} \partial x \\ \partial y \\ \partial z \end{bmatrix} \quad (2)$$

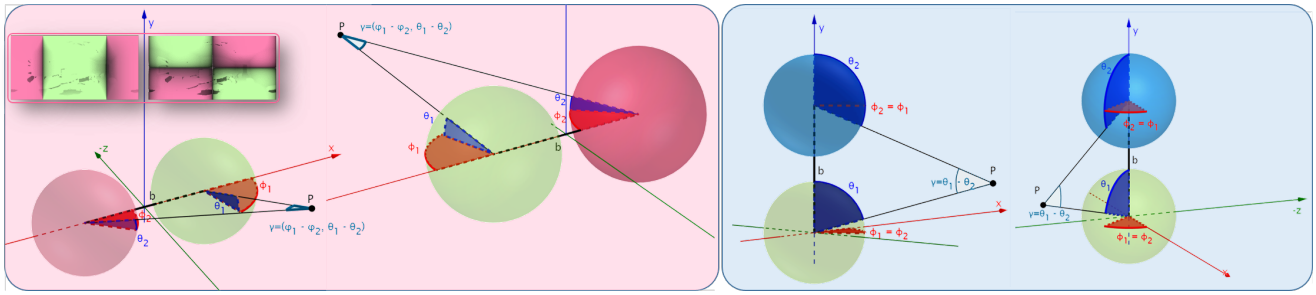

Figure 3. The spherical disparity γ model for a horizontal (pink region) and vertical (cyan region) baselines. Besides longitudinal disparity, the horizontal placement introduces latitudinal disparity as well, with both being a function of the estimated depth according to Eq. 2. For the vertical placement scenario, a simpler model that only includes latitudinal disparity simplifies spherical view synthesis and depth estimation. The top left inset illustrates the irregular sign patterns of the disparities in the horizontal stereo placement setting (negative – green and positive – pink). The left image corresponds to longitudinal (ϕ), and the right to latitudinal (θ) disparity.

These link Cartesian displacements, *i.e.* the baseline $\mathbf{b} = (\mathbf{d}x, \mathbf{d}y, \mathbf{d}z)$, to angular displacements on the sphere, *i.e.* disparity $\boldsymbol{\gamma} = (\mathbf{d}\phi, \mathbf{d}\theta)$, through the radius, *i.e.* depth, r.

For horizontal stereo along the x axis with a baseline $\mathbf{b}_x{=}(\mathbf{d}x, 0, 0)$ it is $\boldsymbol{\gamma_{horiz}} = \mathbf{d}_x(\partial\phi/\partial x, \partial\theta/\partial x)$ while for vertical stereo with a baseline $\mathbf{b}_y{=}(0, \mathbf{d}y, 0)$ along the y axis it is $\boldsymbol{\gamma_{vert}} = \mathbf{d}_y(\partial\phi/\partial y, \partial\theta/\partial y)$. Evidently, the disparity model for vertical placements is simpler, as there is no longitudinal disparity and thus, the pixels reproject to the same vertical scan path. However, for horizontal placements the reprojected pixels lie on epipolar curves on the ERP domain which are sinusoidal, resulting in a more complex disparity model with displacements along both angular directions.

3.2. Depth-image-based rendering

As presented in Sec. 3.1, the angular disparity $\boldsymbol{\gamma}$ for an ERP pixel $\boldsymbol{\varrho} = (\phi, \theta)$ is a function of its depth r and the baseline $\mathbf{b}$ between the viewpoints. Consequently, we can transform pixel coordinates from a source ERP image I_s to a target ERP image I_t given the source's depth map D_s using an angular pixel displacement function Γ:

$$\boldsymbol{\varrho}_t = \Gamma_{s\to t}(D_s, \boldsymbol{\varrho}_s, \mathbf{b}_{s\to t}) = \boldsymbol{\varrho}_s - \gamma(D_s, \boldsymbol{\varrho}_s, \mathbf{b}_{s\to t}). \quad (3)$$

It should be noted that for horizontal stereo, the longitudinal disparity wraps around the sphere. This corresponds to a modulo operation, which is omitted to simplify notation.

Under a traditional inverse warping approach, the target image would be bilinearly sampled to synthesize the source view and supervise learning through the reconstructed source view. Yet, this approach cannot easily handle occluded regions or non-linear mappings which are prevalent in the sphere. Indeed, the ERP distortions are responsible for many-to-one as well as one-to-many pixel mappings, a fact that is more pronounced in wider baselines that are a necessity for higher accuracy in farther depths. Furthermore, wider baselines produce noticeable occlusions, especially for spherically imaged content.

In order to enable learning through view synthesis for spherical stereo we use a soft locally differentiable rendering approach (DIBR) that involves splatting the contributions of each source image pixel to an empty target canvas $\hat{\mathbf{I}}_t$ (Fig. 4). The splatted coordinates are derived by Γ and are a function of the source depth map D_s. Local differentiability is ensured by neighborhood based bilinear splatting, while soft rendering relies on weighted contribution accumulation in the target image [39].

In more detail, each source pixel $\boldsymbol{\varrho}_s$ contributes to four target pixels $\boldsymbol{\varrho}_t^{\mathcal{N}} : \{\boldsymbol{\varrho}_t^{tl}, \boldsymbol{\varrho}_t^{tr}, \boldsymbol{\varrho}_t^{bl}, \boldsymbol{\varrho}_t^{br}\}$ comprising a neighborhood $\mathcal{N}$ created through floor and ceiling operations on the target pixel's $\boldsymbol{\varrho}_t$ coordinates. A bilinear weight $\beta(\boldsymbol{\varrho}_t^{\mathcal{N}}, \boldsymbol{\varrho}_t)$ is associated with each of them. The contributions of all source pixels are accumulated on the target image via scattering operations and additionally weighted by a depth attenuation factor $\alpha(\boldsymbol{\varrho}_s, D) = e^{-D(\boldsymbol{\varrho}_s)/d_{max}}$, with d_{max} a pre-selected maximum depth value. Each source pixel's contribution to the target image canvas $\hat{\mathbf{I}}_t$ is weighted by $w(\boldsymbol{\varrho}_s) = \alpha(\boldsymbol{\varrho}_s, D)\beta(\boldsymbol{\varrho}_t^{\mathcal{N}}, \boldsymbol{\varrho}_t)$. Additionally, the weights themselves are also splatted in a target weight canvas $\hat{W}_t$.

In this way, soft z-buffering is enforced and the target view $\tilde{\mathbf{I}}_t$ is synthesized, after a normalization operation that divides the splatted color canvas with the splatted weight canvas in an element-wise fashion: $\tilde{\mathbf{I}}_t = \hat{\mathbf{I}}_t \oslash (\hat{W}_t + \epsilon)$, ϵ being a small numerical stability constant. This allows for backpropagation to the occluded areas whose view synthesis contributions and gradients are weighted according to a viewpoint proximity criterion. Besides gracefully handling occlusions, this splatting based view synthesis can accommodate many-to-one pixel mappings. While one-to-many pixel mappings are not supported, they do not need to be explicitly handled as the canvas will be empty in those regions where no source pixel contribution landed. This way, a binary mask $M_t = \hat{W}_t < \varepsilon$ can be calculated that masks empty canvas areas. On the contrary, when using inverse

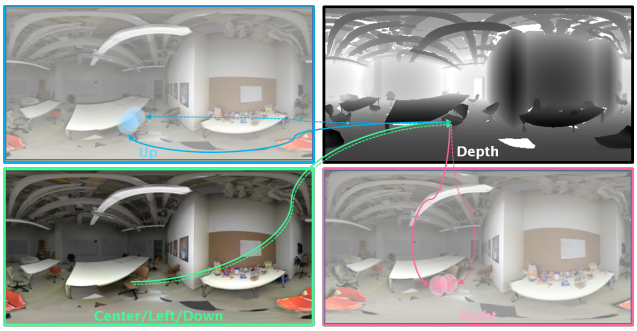

Figure 4. Depth-image-based rendering view synthesis. For each pixel, its reprojection to another viewpoint (in horizontal or vertical stereo placement) is calculated via the estimated depth map. For each pixel a weighted splat is added to an empty canvas around its immediate reprojection neighborhood. Subsequent to their accumulation, a normalization step produces the final rendering. In this way, occlusions and irregular pixels mappings are handled gracefully, allowing for the use of view synthesis as a supervision objective, even for severe distortion areas.

warping, either ground truth depth for z-testing, or an attention mechanism is required to prevent false supervision and destabilizing gradient backpropagation.

3.3. CoordNet

Architecture: Our network, CoordNet, illustrated in Fig. 5, is designed to be efficient in learning with spherical data, minimizing memory consumption and maximizing inference speed compared to other approaches for 360^o learning. Our lightweight backbone architecture is inspired by [15] but we replace traditional residual blocks with pre-activated ones [12] and utilize ELU [32] activations instead of RELU [25] and batch normalization [13].

We introduce 360^o awareness implicitly within our model by utilizing the recently introduced coordinate convolutions [21]. Each input feature map is concatenated with two additional feature maps that represent its grid coordinates in the two dimensional grid. These extra features allow the network to learn the spatial context, which in our case is the ERP domain. CoordNet has minimal memory overhead compared to spectral or model transference approaches, which only scales with feature resolution and the number of convolutional layers. Additionally, in terms of run-time performance, the processing overhead is lower than kernel based approaches that involve trigonometry calculations for warping features or weights.

Unlike other stereo self-supervised learning approaches, we resort to predicting depth directly instead of disparities, and use Eq. 2 to calculate them for view synthesis. This allows for a more general spherical view synthesis model that can facilitate both vertical and horizontal stereo placements. For the vertical case, a direct disparity estimation is equivalent to depth estimation, but for the horizontal one, it touches on an important weakness of CNNs: their inability to simultaneously regress spatially varying positive and negative values. Longitudinal disparities for horizontal stereo are of opposing signs at the front and back looking directions. Moreover, latitudinal disparities, in the same placement, follow spatially varying sign patterns, depicted in Fig. 3, further magnifying the problem. While a solution would be to predict absolute values and explicitly enforce correct signs this was not the case in our experiments as training did not manage to converge. Since the longitudinal and latitudinal disparities are correlated, directly predicting the first would make the estimation of the second possible, but only after transitioning to depth, yet this only strengthens the choice of regressing depth directly.

Supervision: CoordNet is self-supervised by a depth driven photometric image reconstruction loss as well as a depth smoothness prior:

$$\mathcal{L}_{total} = \lambda_{recon}\mathcal{L}_{recon} + \lambda_{smooth}\mathcal{L}_{smooth}, \tag{4}$$

where λ_{recon} and λ_{smooth} are weights that sum up to one. Our reconstruction loss uses a standard photometric loss as presented in [11], which is also used in most self-supervised monocular depth estimation methods:

$$\mathcal{L}_{photo}(\mathbf{p}) = \eta\mathcal{L}_D(\mathbf{I}_t^M(\mathbf{p}), \tilde{\mathbf{I}}_t^M(\mathbf{p})) + (1-\eta)\left|\mathbf{I}_t^M(\mathbf{p}) - \tilde{\mathbf{I}}_t^M(\mathbf{p})\right|.$$

It combines the L1 penalty function with structural dissimilarity $\mathcal{L}_D$, under a relative weighting factor η. The superscript M denotes multiplication with the binary mask M_t.

While previous ERP domain learning approaches [49] used uniform supervision on the ERP image, such an approach will greatly bias higher quality predictions towards the more distorted areas. Instead, we explicitly use a spherically weighted attention mechanism to uniformly aggregate errors and gradients on the sphere, instead of on the distorted ERP image. We use an attention weight matrix A defined on the ERP domain in two different variants:

$$A(\boldsymbol{\varrho}) = \begin{cases} |\sin(\theta)|, & \text{for vertical stereo,} \\ |\sin(\phi)||\sin(\theta)|, & \text{for horizontal stereo.} \end{cases} \tag{5}$$

These weight maps, as illustrated in Fig. 2, eliminate the effect of the epipole singularities as the contributions of the areas around the singularities tend to zero. For the vertical case, they coincide with the distortion attenuation factor $\sin(\theta)$ but for the horizontal case the corresponding singularity attenuation term $\sin(\phi)$ is added. Hence, the total reconstruction loss is the spherically weighted mean photometric error of all valid pixels:

$$\mathcal{L}_{recon} = \frac{1}{\sum_{\mathbf{p}} M_t(\mathbf{p})} \sum_{\mathbf{p}} A(\mathbf{p}) M_t(\mathbf{p}) \mathcal{L}_{photo}(\mathbf{p}). \tag{6}$$

We also impose a smoothly varying prior on the predicted signal. However, defining smoothness on the sphere is challenging and naive approaches like applying finite element

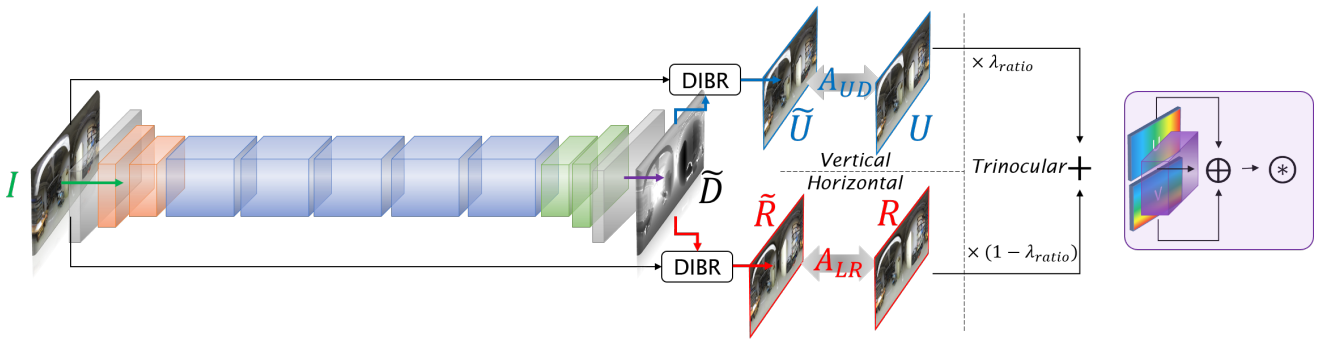

Figure 5. CoordNet and its view synthesis based supervision scheme. An ERP depth map $\tilde{D}$ is predicted using a single monocular ERP color image $\tilde{\mathbf{I}}$. Through the estimated depth, we synthesize stereo viewpoints in vertical (up – $\tilde{\mathbf{U}}$) or horizontal (right – $\tilde{\mathbf{R}}$) baselines. These are supervised using a photometric consistency error using the original viewpoints $\mathbf{U}$ and $\mathbf{R}$ for the up and right reconstructions, as well as placement specific attention maps, A_{UD} and A_{LR} respectively. The trinocular self-supervised scenario is also considered using a blending factor λ_{ratio} to balance loss between the two different viewpoint reconstructions. CoordNet utilizes CoordConvs for all its convolutional layers as shown on the right. Each incoming feature map is concatenated with the horizontal u and vertical v coordinate maps of its resolution before fed into the convolution operation.

gradient operators [49] in the ERP domain will not succeed in enforcing smoothness correctly, as spherical depth inherently varies spatially even for flat surfaces. As an alternative, we enforce a smoothness constraint on the deprojected Cartesian coordinates $\mathbf{v} = (x, y, z)$ for each predicted pixel $(r = \tilde{D}(\mathbf{p}), \phi, \theta)$, by minimizing the following weighted total variation term, using central differences:

$$\mathcal{L}_{smooth} = \bar{A}(\mathbf{p}) e^{\|\nabla \mathbf{I}_s(\mathbf{p})\|_2} \sqrt{(\nabla_u \mathbf{v}(\mathbf{p}))^2 + (\nabla_v \mathbf{v}(\mathbf{p}))^2}. \quad (7)$$

The weighting term $\bar{A}(\mathbf{p}) = \mathbf{1} - A(\mathbf{p})$ more heavily enforces smoothness on the distorted regions. A color guidance weighted factor is also used in order to establish correlated depth and color gradients. Thus, smoothness on the ERP domain is ensured via the Eq. 1 deprojection functions.

4. Results

Dataset: Given the unavailability of stereo 360^o datasets, we take a similar approach to [49] and render panoramas from displaced viewpoints in both vertical and horizontal placements as shown in Fig. 3. We use Blender[1] and set the baseline for both placements to 0.26m, which is a reasonable distance to get high quality results for indoor scenes, which is the context of the rendered 3D datasets used in [49]. However, unlike [49], we use the official train, validation and test splits of Matterport3D [2] and Stanford2D3D [1] (fold#1).In this way, our test set is sufficiently different from our train set, and at least quadruple the size of the test set used in [49]. Further, SunCG [35] is only used during training and validation, but not during testing as our focus is to assess applicability in real world settings.

Implementation Details: We implement our network in PyTorch [26], initialize its weights using [10], and train all our models for 30 epochs using a fixed learning rate of 10^{-4} and a batch size of 16. Across all experiments we use a fixed seed for all the involved random generators to guarantee consistency. We use the AdaBound [22] optimizer with a convergence speed of 2×10^{-3} and a final target SGD learning rate of 10^{-3}. The weights of Eq. 4 are set to λ_{recon}=0.95 and λ_{smooth}=0.05. Inline with prior work, the photometric error is balanced by η=0.85 and a box filter with a kernel size of 5 is used for the SSIM calculations.

Metrics: We use traditional depth evaluation metrics [6], but with a notable difference. While previous works on 360^o depth estimation [49, 41] used these metrics in the ERP domain, they did not take into account its distortion. As a result, distorted areas were given higher precedence in the error calculation. We adapt the absolute relative error, squared relative error, RMSE and RMSLE to use weighted calculations for each pixel using the first case of Eq. 5 in order to alleviate the effect of ERP distortion in our evaluation. However, the percentile threshold metrics require a different approach. Instead of densely sampling the ERP, we sample the sphere using an $\mathcal{S}^2$ generalized spiral set [30] with $N = 0.25 \times w \times h$ points. Consequently, the percentile thresholds are only calculated for these spiral points.

Stereo placement analysis: First we seek to assess which stereo placement is more efficient for view synthesis based depth estimation learning. We train two variants of the network described in Sec. 3.3. For the vertical variant (referred to as UD, *i.e.* up-down) we supervise using the up view (displaced on the y axis) while the network is fed the down/central image. Similarly, for the horizontal vari-

[1]Blender uses different longitudinal and latitudinal ranges ($[\frac{-3\pi}{2}, \frac{\pi}{2}]$ and $[\frac{-\pi}{2}, \frac{\pi}{2}]$ respectively), therefore Eq. 1, Eq. 2 and Eq. 5 get modified accordingly using trigonometric reflections.

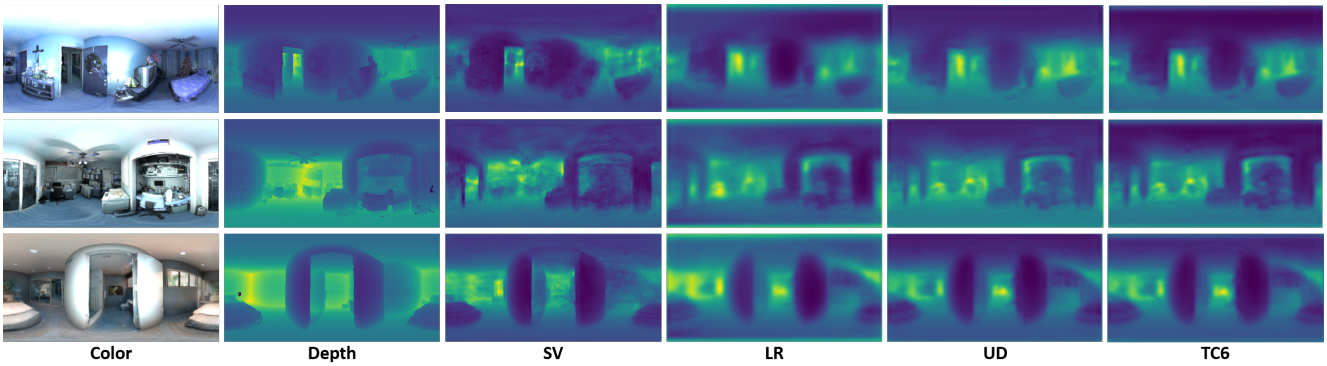

Figure 6. Qualitative results of each category of trained models (TC6 was chosen as it is the best performing trinocular model). From left to right: the input color image, the ground truth depth from [49], the fully supervised prediction (SV), the self-supervised predictions of horizontal (LR), vertical (UD) and trinocular (TC6) placements. Additional examples can be found in our supplementary material.

ant (LR, *i.e.* left-right), we supervise using the right view. Table 1 shows that both converge at about the same epoch, and that UD achieves higher performance. Intuitively this is attributed to the simpler disparity model. Nevertheless, another important factor is that an UD model does not suffer from the prevalent horizontal distortions. Due to this reason, an UD variant can be trained with inverse warping as the view synthesis method, while for the LR variant, convergence with inverse warping was not possible.

Complementarity analysis: Next, we seek to understand whether these two placements are complementary. We train another model using trinocular (referred to as TC) supervision that infers a single depth map from the central view and is jointly supervised by the reconstruction of both the up and right images, as seen in Fig. 5. We explore the effect of blending both view synthesis supervisions by adding a ratio parameter to combine their losses $\mathcal{L}_{recon}{=}\lambda_{ratio}\mathcal{L}^{UD}_{recon} + (1{-}\lambda_{ratio})\mathcal{L}^{LR}_{recon}$. We train 4 variants of the TC network with a 0.2 step for λ_{ratio} and name them by suffixing TC with the ratio's decimal. The results are also presented in Table 1 with the color coded interpolation for each metric illustrating the transition from the best to worst, as we move from LR ($\lambda_{ratio}{=}0$) to UD ($\lambda_{ratio}{=}1$). Interestingly, TC4 indicates that there exist blending factors that will not allow the model to learn a good enough representation as single viewpoint supervisions do. We further observe that performance increases as the ratio increases towards the simpler disparity model. Nonetheless, while UD achieves best performance with respect to outlier predictions (as indicated by the RMS metrics), we find that the slower convergence of TC6 results in a more robust model, offering a compromise for overall performance, attributed to the harder to optimize for, right view reconstruction.

Self-supervision status: Given that we rendered/synthesized our data, we are in the unique position of being able to directly and fairly compare view synthesis self-supervision and direct supervision. Most others self-supervised works resort to view synthesis supervision because no high quality depth ground truth data are available. While datasets with laser scanner depth data exist, they are usually sparse, and/or of limited test samples. On the other hand, synthetic datasets that offer high quality depth renders, do not need to render stereo viewpoints, and consequently, this comparison has not been done before. Further, even if it is possible to perform this comparison with synthetic data, applicability to real world scenes is the ultimate goal, which our dataset supports in assessing.

Table 1. Best performing snapshots (reached at the corresponding epoch on the right) of our trained models. Relative performance for the self-supervised methods is color coded to showcase the gradual transition from LR to UD via the different blending factors of TC. Lower is better for light blue metrics, while for the darker accuracies $\delta_i < 1.25^i$ higher is better.

	Abs Rel	Sq Rel	RMSE	RMSLE	δ_1	δ_2	δ_3	Epoch
SV	0.138	**0.091**	**0.473**	**0.184**	**82.4%**	**95.9%**	**98.5%**	24
LR	0.143	0.129	0.639	0.230	58.1%	88.2%	96.5%	18
TC2	0.132	0.117	0.606	0.216	61.3%	89.3%	96.1%	20
TC4	0.199	0.154	0.651	0.250	65.8%	91.2%	96.7%	17
TC6	0.129	0.112	0.580	0.209	65.1%	91.3%	97.0%	28
TC8	0.133	0.117	0.578	0.209	65.4%	91.0%	96.9%	16
UD	0.134	0.119	0.571	0.208	66.4%	90.8%	96.8%	16

We train our network modifying only the loss function and directly supervising with ground truth depth maps. We use the BerHu loss [18] and refer to the fully supervised train as SV. Table 1 clearly shows the superiority of a fully supervised approach compared to stereo self-supervision, providing food for thought and poses interesting dilemmas.

Convergence analysis: We additionally offer a detailed analysis for the convergence behaviour of all variants as Table 1 only reported the best performing snapshots. Fig. 7 plots the results of four metrics on the whole test set across epochs. It further signifies the importance of direct supervision as it is observed that it consistency improves its pre-

dictions. At the same time, UD plateaus while LR is unable to converge further and instead loses performances across epochs, both after around the middle of the training duration, where they achieve their best performing state. The good TC variants showcase more stable training and consistently higher quality performance, contrary to UD which fluctuates more, albeit achieving a high quality minima.

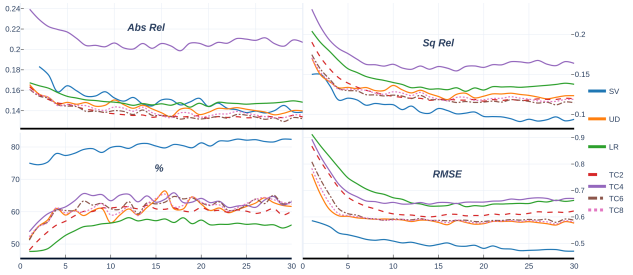

Figure 7. Test set metrics for each epoch for all the conducted experiments. Left to right, top to bottom: Absolute relative error, Squared relative error, $\delta_1 < 1.25$ accuracy, RMSE.

State-of-the-art comparison: We compare our stereo-based learning approach to a recent video-based one [41]. Since [41] similarly renders a sequence dataset using SunCG scenes (PanoSunCG), we train a SunCG only variant of our TC6 model (SCG-TC6). The first two rows of Table 2 compare both methods on the PanoSunCG test set. Since [41] does not provide a publicly available model, and only offers a quantized dataset of significantly smaller variance than ours (our test set alone uses twice as many scenes as the PanoSunCG train and test set combined), the final row of Table 2 presents our model's quantitative performance on our SunCG test set. While the performance of [41] is slightly better on PanoSunCG, our model achieves much higher quality results in our more diverse test set.

Table 2. SunCG & PanoSunCG Comparison Results

	Abs Rel	Sq Rel	RMSE	RMSLE	δ_1	δ_2	δ_3
[41]	**0.337**	**0.196**	**0.337**	0.611	**64.7%**	**82.9%**	**89.9%**
SCG-TC6	0.371	0.440	0.843	**0.421**	56.2%	78.4%	87.8%
SCG-TC6	0.185	0.123	0.491	0.215	72.2%	89.5%	92.0%

CoordConv: Finally, we perform an ablation analysis starting with the effect of CoordConv. We train UD and LR using standard convolutions and report the results in Table 3. We observe that CoordConvs clearly boost the performance in an UD placement but it is harder to determine a similar finding for LR. The discrepancy in RMSE and RMSLE indicate that there is a gain for closer distances (which RMSLE favors) compared to far ones (that RMSE favors), similarly indicated by the discrepancy in the relative metrics (squared against absolute).

Spherical Attention: We conduct two experiments to assess the gains associated to the spherical attention maps by re-training UD and LR without their respective attention masks A_{UD} and A_{LR}. The results are presented in Table 4 where an interesting outcome is apparent. Their effect on LR is significant while for UD it remains questionable as it very slightly hampers performance. ERP distortions are more prevalent in LR and stabilizing the loss during training by reducing their effect, is very important. On the contrary, vertical distortions are gracefully handled by DIBR, therefore rendering the attention insignificant.

Table 3. CoordConv Ablation Results

	Abs Rel	Sq Rel	RMSE	RMSLE	δ_1	δ_2	δ_3
LR	0.143	**0.129**	**0.639**	0.230	58.1%	88.2%	**96.5%**
w/o CC	**0.141**	0.138	0.663	**0.228**	**60.5%**	**88.4%**	96.2%
UD	**0.134**	**0.119**	**0.571**	**0.208**	**66.4%**	**90.8%**	**96.8%**
w/o CC	0.138	0.136	0.650	0.224	61.2%	88.9%	96.3%

Table 4. Spherical Attention Ablation Results

	Abs Rel	Sq Rel	RMSE	RMSLE	δ_1	δ_2	δ_3
LR	**0.143**	**0.129**	**0.639**	**0.230**	**58.1%**	**88.2%**	**96.5%**
w/o A_{LR}	0.269	0.295	0.824	0.324	56.7%	84.4%	93.2%
UD	0.134	0.119	0.571	0.208	**66.4%**	**90.8%**	**96.8%**
w/o A_{UD}	**0.132**	**0.116**	**0.566**	**0.205**	66.2%	90.1%	96.0%

5. Discussion

Spherical view synthesis is a relatively unexplored supervision scheme, mainly due to the lack of data and the challenges that it entails. We have presented a learning scheme under which self-supervised 360^o depth estimation is possible addressing the challenges mainly related to the distortions that ERP introduces. Our work is the first to train a horizontal baseline 360^o self-supervised model and to achieve this, besides introducing the full 360^o disparity model, a more robust 360^o view synthesis was required. The DIBR splatting scheme, in combination with spherical attention, manage to overcome the inconsistent supervision that traditional inverse warping approaches suffer from. Nonetheless, vertical stereo setups are offering higher quality models, further improved by CoordConvs, but as current research focuses on utilizing videos for learning depth estimation, the challenges that horizontal disparity comes with, as well as the full spherical disparity model, are very relevant. Finally, an unsurprising open question is raised with respect to the performance deviation of self-supervised and fully supervised models. Is self-supervision the direction to pursue, or are other approaches like higher quality data acquisition, or synthetic data and domain adaptation, perhaps, better alternatives?

Acknowledgements: We acknowledge HW support by

Nvidia and financial support by the H2020 EC project Hyper360 (GA 761934).

References

[1] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, Feb. 2017. 6

[2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE, 2017. 6

[3] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2018. 2

[4] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 2

[5] B. Coors, A. Paul Condurache, and A. Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018. 2

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 6

[7] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018. 2

[8] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. J. Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *arXiv preprint arXiv:1903.08094*, 2019. 2

[9] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 1, 2

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 6

[11] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 1, 2, 5

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 5

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 5

[14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[16] R. Khasanova and P. Frossard. Graph-based classification of omnidirectional images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 869–878, 2017. 2, 3

[17] H. Kim and A. Hilton. 3d scene reconstruction from multiple spherical stereo pairs. *International journal of computer vision*, 104(1):94–116, 2013. 3

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 7

[19] S. Li. Binocular spherical stereo. *IEEE Transactions on intelligent transportation systems*, 9(4):589–600, 2008. 3

[20] K. Lin and T. P. Breckon. Real-time low-cost omnidirectional stereo vision via bi-polar spherical cameras. In *International Conference Image Analysis and Recognition*, pages 315–325. Springer, 2018. 3

[21] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018. 5

[22] L. Luo, Y. Xiong, Y. Liu, and X. Sun. Adaptive gradient methods with dynamic bound of learning rate. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, May 2019. 6

[23] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. 2

[24] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic. Salnet360: Saliency maps for omni-directional images with cnn. *Signal Processing: Image Communication*, 69:26–34, 2018. 2

[25] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 5

[26] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 6

[27] G. Payen de La Garanderie, A. Atapour Abarghouei, and T. P. Breckon. Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 789–807, 2018. 2

[28] C. Pinard, L. Chevalley, A. Manzanera, and D. Filliat. Learning structure-from-motion from motion. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*,

pages 363–376, Cham, 2019. Springer International Publishing. 3

[29] V. Prasad and B. Bhowmick. Sfmlearner++: Learning monocular depth & ego-motion using meaningful geometric constraints. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2087–2096. IEEE, 2019. 2

[30] E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. *The mathematical intelligencer*, 19(1):5–11, 1997. 6

[31] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008. 1

[32] A. Shah, E. Kadam, H. Shah, S. Shinde, and S. Shingade. Deep residual networks with exponential linear unit. In *Proceedings of the Third International Symposium on Computer Vision and the Internet*, pages 59–65. ACM, 2016. 5

[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 1

[34] R. Skupin, Y. Sanchez, Y.-K. Wang, M. M. Hannuksela, J. Boyce, and M. Wien. Standardization status of 360 degree video coding and delivery. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017. 2

[35] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017. 6

[36] Y.-C. Su and K. Grauman. Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems*, pages 529–539, 2017. 2

[37] Y.-C. Su and K. Grauman. Kernel transformer networks for compact spherical convolution. *arXiv preprint arXiv:1812.03115*, 2018. 2

[38] K. Tateno, N. Navab, and F. Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–722, 2018. 2

[39] S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 4

[40] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018. 2

[41] F. Wang, H. Hu, H. Cheng, J. Lin, S. Yang, M. Shih, H. Chu, and M. Sun. Self-supervised learning of depth and camera motion from 360° videos. *CoRR*, abs/1811.05304, 2018. 2, 3, 6, 8

[42] T.-H. Wang, H.-J. Huang, J.-T. Lin, C.-W. Hu, K.-H. Zeng, and M. Sun. Omnidirectional cnn for visual place recognition and navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2341–2348. IEEE, 2018. 2

[43] M. L. J. P. W. S. H. L. L. L. Yue Luo, Jimmy Ren. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[44] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 3

[45] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5287–5295, 2017. 1

[46] Z. Zhang, Y. Xu, J. Yu, and S. Gao. Saliency detection in 360 videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 488–503, 2018. 2

[47] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 1, 2

[48] Z. Zhu. Omnidirectional stereo vision. In *Proceedings of the Workshop on Omnidirectional Vision, ICAR, Budapest, Hungary*, pages 22–25, 2001. 3

[49] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018. 3, 5, 6, 7