



*Citation for published version:*

Yuan, M & Richardt, C 2021, 360° Optical Flow using Tangent Images. in *British Machine Vision Conference : (BMVC)*. vol. 2021. <<https://richardt.name/360-flow/>>

*Publication date:*  
2021

*Document Version*  
Peer reviewed version

[Link to publication](#)

**University of Bath**

**Alternative formats**

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# 360° Optical Flow using Tangent Images

Mingze Yuan

<https://yuanmingze.github.io/>

Christian Richardt

<https://richardt.name>

University of Bath

Bath, UK

---

## Abstract

Omnidirectional 360° images have found many promising and exciting applications in computer vision, robotics and other fields, thanks to their increasing affordability, portability and their 360° field of view. The most common format for storing, processing and visualising 360° images is equirectangular projection (ERP). However, the distortion introduced by the nonlinear mapping from 360° images to ERP images is still a barrier that holds back ERP images from being used as easily as conventional perspective images. This is especially relevant when estimating 360° optical flow, as the distortions need to be mitigated appropriately. In this paper, we propose a 360° optical flow method based on tangent images. Our method leverages gnomonic projection to locally convert ERP images to perspective images, and uniformly samples the ERP image by projection to a cubemap and regular icosahedron faces, to incrementally refine the estimated 360° flow fields even in the presence of large rotations. Our experiments demonstrate the benefits of our proposed method both quantitatively and qualitatively.

## 1 Introduction

Commercial 360° video cameras have recently grown in popularity. Companies such as GoPro, Insta360, Ricoh, and many others, are now offering affordable consumer 360° video cameras. At the same time, support for 360° videos has been added across the content creation and consumption pipeline, including video editing software, such as DaVinci Resolve and Adobe Premiere Pro, video distribution channels such as YouTube, Facebook and Vimeo, video players like VLC, and virtually all current VR headsets.

Optical flow is a crucial component for processing and editing 360° images and videos as it provides correspondences over time. Among other things, this enables 360° video stabilisation [22], depth estimation [51], and novel-view synthesis [6]. However, the most common format of 360° images, equirectangular projection (ERP), suffers from severe distortions when mapping a spherical 360° image to the 2D image plane. This distortion is not considered by most existing optical flow algorithms, which may lead to flow estimation errors.

Previous work has adapted existing CNN-based techniques designed for perspective images to ERP images by transforming convolution kernels to match the projection [10, 53, 58]. To overcome the limited spatial resolution supported by these methods, other work proposed extracting, processing and recombining tangent images [3, 25, 27, 41, 42, 48], which supports both CNN-based and traditional methods. Moreover, these tangent images are based

on *gnomonic* projection, which is a common tool used in geography to map a sphere to a plane, as the great circle on the sphere maps to a straight line in the tangent image [13]. This principle has been applied to tasks such as monocular depth estimation [41, 42], semantic segmentation [13, 43], and classification [13, 25, 27], but not yet to optical flow. Our approach builds on this insight to estimate accurate 360° optical flow for high-resolution ERP images by sampling tangent images uniformly based on regular polyhedra (cube and icosahedron).

This paper makes the following technical contributions:

1. We propose a 360° optical flow method based on gnomonic projection to overcome the distortions induced by the equirectangular projection of 360° images.
2. We integrate a global warping method to align large rotations between 360° images.
3. We introduce a tangent image optical flow blending method that can dramatically reduce discontinuities at face boundaries.
4. We refine our flow estimates by increasing the sampling density of tangent images from an initial cubemap to the final icosahedron-based uniform sampling of tangent images.

## 2 Related Work

**Optical Flow** has been a fundamental computer vision technique for decades, as it estimates dense correspondences between two input images [1, 29]. These correspondences can be used to identify object or camera ego-motion, and optical flow thus finds many applications in robotics, scene understanding, geometry reconstruction etc. Traditional optical flow methods are formulated using energy minimization based on photoconsistency and regularisation with a smoothness term [1, 15, 23, 26]. Brox et al. [8] proposed a warping-based method that refines optical flow estimations in a coarse-to-fine fashion. Sun et al. [35] uses a median filter post-processing step to produce sharper object boundaries in the optical flow fields.

In recent years, learning-based methods have defined a new state of the art in optical flow estimation. The first methods were based on supervised learning, generally on synthetic training data. FlowNet [10] lifts the correlation operation into feature space, and uses a multi-scale architecture to effectively predict optical flow from two input images. FlowNet2 [18] introduces image warping between multiple cascaded FlowNets to increase the accuracy of large pixel motion. PWC-Net [57] incorporates the best practices of traditional optical flow methods into a neural network: pyramid processing, image warping, and cost volume processing. RAFT [69] employs a recurrent network to iteratively estimate optical flow from the 4D correlation volume between all pairs of pixels. Aleotti et al. [1] propose a data generation method using monocular depth estimation to synthesise a second view from a single input image. This enables self-supervised training of existing optical flow methods.

Virtually all optical flow methods focus on perspective input images, although spherical images have recently received more attention. To this end, recent approaches by Artizzu et al. [2] and Bhandari et al. [6] transfer the architecture and weights of pre-trained networks [10, 1] to spherical images in the equirectangular projection (ERP) format.

**Panoramic Image Processing.** Panoramic (aka 360° or *spherical*) images have been used in a wide variety of application areas, including depth estimation [19, 20, 22, 54, 41, 47, 44, 50, 51], room layout estimation [12, 14, 21, 52, 40, 43, 47], semantic segmentation [25, 52, 46, 48], novel-view synthesis [5, 16, 28, 45] and so on. The key problem when working with spherical images is to project them onto a regular 2D pixel grid for easy processing, as any projection introduces some kind of distortion – similar to maps of the world.

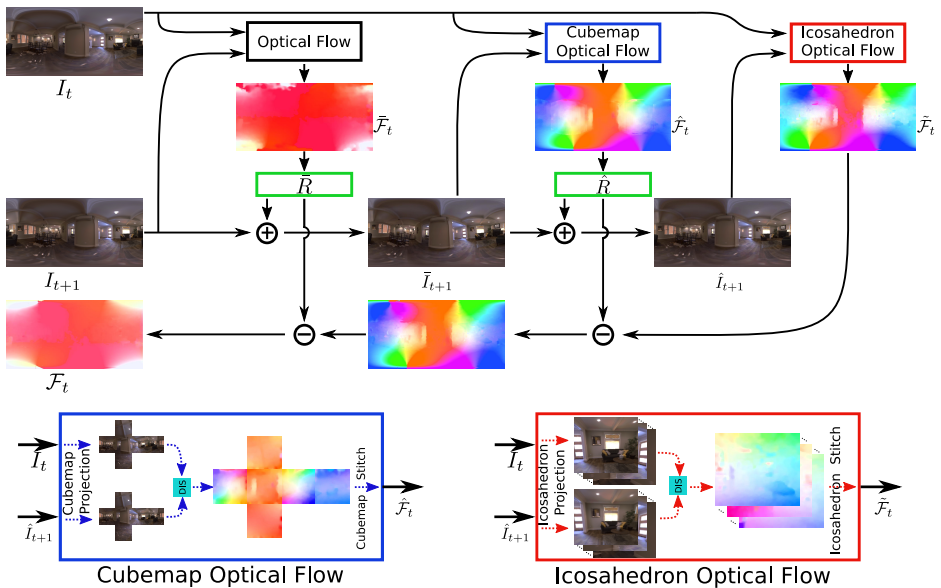


Figure 1: Method overview. The input is a pair of equirectangular images,  $I_t$  and  $I_{t+1}$ , and the output is the optical flow  $\mathcal{F}_t$ . The operators ‘ $\oplus$ ’ and ‘ $\ominus$ ’ rotate images forward and optical flow fields backward, respectively, using image warping (see Section 3.3). The green box estimates the optimal rotation from a 360° optical flow field.

Spherical images are sometimes projected to a cubemap, i.e. the six sides of a cube, potentially with overlap between sides [4, 41, 42]. However, the resulting perspective images have large angles of view of  $\geq 90^\circ$ , which introduces significant perspective distortion. A smaller field of view can be achieved using icosahedron projection, producing 20 triangular faces, or in fact any subdivision of the icosahedron [25, 27, 48]. In the limit, perspective tangent images can be obtained anywhere on the sphere using gnomonic projection [11, 13]. To overcome the significant distortions near the poles at the top/bottom image edges in ERP images, distortion-aware convolutions adapt kernels using gnomonic projection [11, 63, 68], which enable training on perspective images and testing on ERP images.

Our method combines ERP, cubemap and icosahedron projections to incrementally estimate global rotation and to refine spherical optical flow estimates to achieve higher accuracy.

### 3 Approach

Our method comprises a global rotation warping and tangent-image-based panoramic optical flow method to mitigate the distortions in equirectangular image and the large displacements of pixels. At the same time, our high-level approach generalises to any off-the-shelf perspective optical flow method, and thus will benefit from future improvements in optical flow methods.

As shown in Figure 1, our method comprises three steps: (1) We estimate the optical flow  $\hat{\mathcal{F}}_t$  from ERP image  $I_t$  to  $I_{t+1}$ , followed by rotating  $I_{t+1}$  towards to  $I_t$  based on the rotation estimate  $\hat{R}$  from the flow  $\hat{\mathcal{F}}_t$ , which generates the warped ERP image  $\tilde{I}_{t+1}$  (see Section 3.3). (2) We estimate the panoramic optical flow  $\hat{\mathcal{F}}_t$  from  $I_t$  to  $\tilde{I}_{t+1}$  based on cubemap projection

optical flow (see Section 3.2), and then rotate the image  $\bar{I}_{t+1}$  according to the rotation  $\hat{R}$  estimated from the cubemap flow  $\hat{\mathcal{F}}_t$  to generate  $\hat{I}_{t+1}$ . (3) We estimate the fine-scale flow  $\hat{\mathcal{F}}_t$  from  $I_t$  to  $\hat{I}_{t+1}$  using icosahedron projection optical flow (see Section 3.3), then backward-rotate the fine-scale flow  $\hat{\mathcal{F}}_t$  by rotations  $\bar{R}$  and  $\hat{R}$  to generate the final 360° optical flow  $\mathcal{F}_t$ .

Compared to perspective images, equirectangular images are continuous in all direction. In Section 3.1, we analyse and define the panoramic optical flow. To solve the equirectangular image distortion, especially at the top and bottom, our method employs gnomonic projection to project the equirectangular image to tangent image [13]. We use cubemap and regular icosahedron faces to uniformly sample the equirectangular image to solve this problem (see Section 3.2). However, compared to a panoramic image, a tangent image has a smaller field of view, not larger than 180°, and just 90° for cubemap projection (without padding). The tangent image optical flow operates on a pair of corresponding tangent images from the same ERP image area, which can fail when an object is only visible in one of the tangent images. To compensate for the tangent images’ narrow field of view, the global rotation operation is used on the target image  $I_{t+1}$  to pre-align it with the source image  $I_t$  (Section 3.3).

### 3.1 Definition of 360° Optical Flow

Spherical image coordinates are continuous in any direction on the image, e.g. pixel locations overflowing the image width will wrap around to the other side of the image. If panoramic optical flow follows an object, a pixel’s motion vector can fall outside the image boundary. This is one reason that perspective optical flow methods cannot track pixels moving outside the equirectangular image boundary, because they do not support the horizontal coordinate wrap-around, as illustrated in Figure 2. However, this introduces an ambiguity into 360° optical flow estimation, as there is more than one path from the source point to the target point along a great-circle on the sphere: usually there is one shorter and one longer path.<sup>1</sup> To uniquely define 360° optical flow in the equirectangular image format, we define the optical flow to follow the shortest path from source to target along the great circle between them. This naturally limits the maximum flow magnitude to  $\leq 180^\circ$ .

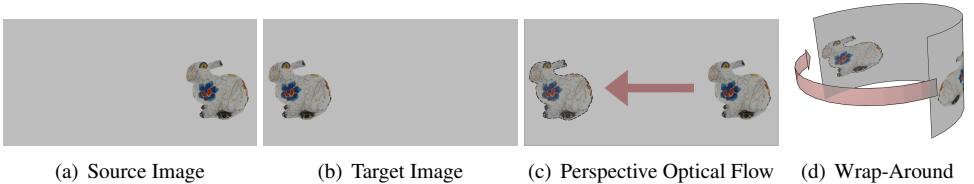


Figure 2: 360° optical flow illustrated: A bunny moves from (a) to (b) within an ERP image. (c) Perspective optical flow methods estimate the bunny’s motion from the right to the left via the front. (d) However, the shortest path for the bunny to move is along the back wrap-around.

### 3.2 Projection & Stitching

To mitigate the distortions in ERP images, we locally undistort the ERP image to a perspective tangent image using gnomonic projection [13]. This way, we can apply any off-the-shelf perspective optical flow method directly on pairs of tangent images, first using a cubemap

<sup>1</sup>One could also travel along the great circle a few more times, but these paths are getting longer and longer.

projection and later using an icosahedron projection to refine the 360° flow estimates. In both the cubemap and icosahedron cases, we proceed as follows: we first project the ERP image with gnomonic projection to generate 6 (cubemap) or 20 (icosahedron) perspective images that are tangent to the unit sphere; we then use an optical flow method for perspective images to estimate optical flow for pairs of corresponding tangent images; and finally, we stitch the optical flow fields for all tangent images back together in the ERP format. Our approach in principle supports any perspective optical flow method, and results are therefore expected to improve as improved optical flow methods become available in future.

**Gnomonic Projection.** Like Zhao et al. [49], we use the gnomonic projection to produce tangent images. These images are perspective images with the same camera centre as the ERP image, but their principal axis intersects the unit sphere at the tangent point with a circumscribed cube or icosahedron. Depending on the choice and density of tangent points, tangent images cover different spherical surface areas, which determines the field of view (FoV) of the tangent images. For estimating multi-scale motion and to uniformly sample the sphere surface, we select tangent points according to a regular cubemap and a regular icosahedron. These are the most common convex regular polyhedra with 6 and 20 faces, respectively, and this results in different FoV with cubemap tangent images having a considerably wider FoV than icosahedron tangent images. Therefore, cubemaps can estimate a larger motion vectors from the tangent images, although their angular resolution is coarser than the icosahedron’s tangent images for the same spatial tangent image resolution. To increase the FoV of tangent images and improve the continuity of optical flow at the boundary between tangent images, we add image padding to extend the area of tangent images, as illustrated in Figure 3.

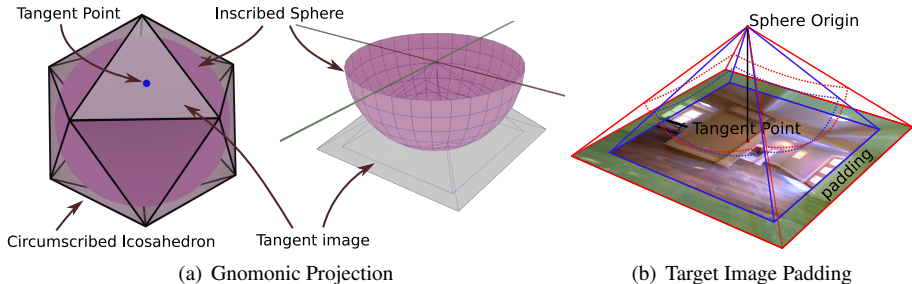


Figure 3: Tangent image projection and padding. (a) Gnomonic projection maps points on the surface of a sphere from the sphere’s centre to a point on the tangent plane. (b) We add padding to the tangent images to extend their area and ensure overlap between tangent images.

**Tangent Optical Flow Blending.** As each tangent image’s optical flow is estimated independently, the estimates might not be consistent in the overlap areas of adjacent tangent images. When stitching the tangent flow fields in the equirectangular format, we thus compute a per-pixel blending weight  $\omega_i$  for each tangent image  $I_{t,i}$  to smoothly blend its optical flows  $\mathcal{F}_{t,i}$  in the overlap areas. The  $\omega_i$  computes the colour difference between the source tangent image  $I_{t,i}$  and the backwards-warped target image, to reduce the weight of badly estimated tangent images’ optical flow  $\mathcal{F}_{t,i}$ . The  $\omega_i$  is estimated by  $e^{-|I_{t,i} - \text{Warp}(I_{t+1,i}, \mathcal{F}_{t,i})|}$ , the  $\text{Warp}(\cdot, \cdot)$  is the backwards warping operation, and  $I_{t+1,i}$  is the corresponding tangent image of  $I_{t,i}$  at time  $t + 1$ . The  $\mathcal{F}_t$  blended with the weighted tangent image optical flows,  $\frac{\sum_i \mathcal{F}_{t,i} \omega_i}{\sum_i \omega_i}$ .

### 3.3 Global Rotation Warping

Inspired by warping-based optical flow [8], we estimate the global rotation to pre-align the ERP image  $I_{t+1}$  to  $I_t$ . This helps reduce the range of pixel motion to a level that is more suitable for the smaller FoV of tangent images. We compute the global rotation  $\bar{R}$  from the computed optical flow field as follows. First, we convert the coordinates of the start and end points of all optical flow vectors to spherical coordinates with unit length. Then, we solve for the optimal rotation between the start points and the corresponding end points using least-squares fitting, which can be solved in closed form by singular value decomposition [9, 50].

**Global Rotation Image Warping.** After estimating the global rotation  $\bar{R}$  from the 360° optical flow, we warp the target image  $I_{t+1}$  by the rotation  $\bar{R}$  to align it to the source image  $I_t$ . We apply global rotation warping based on the coarse to fine principle. First, we estimate the rotation  $\bar{R}$  directly from ERP optical flow to roughly align the image  $I_{t+1}$  and compensate for camera ego-motion. We then use the cubemap optical flow to estimate the residual rotation  $\hat{R}$  to fine-tune the global rotation alignment.

**Global Rotation Optical Flow Warping.** To recover the final 360° optical flow field  $\mathcal{F}$  from the icosahedron optical flow  $\tilde{\mathcal{F}}$ , the end points of  $\tilde{\mathcal{F}}$  need to align with the pixel position in the input image  $I_{t+1}$ . When estimating the optical flow, only the target image  $I_{t+1}$  is rotated, so we now need to rotate the end points of  $\tilde{\mathcal{F}}$ , i.e.  $\tilde{\mathcal{F}}^{\text{EP}}$ , in the opposite direction. To ‘rotate’ an ERP pixel, we first convert its ERP image coordinates  $\mathbf{x}$  to Cartesian 3D coordinates using the inverse projection operator  $\mathcal{P}^{-1}$ , rotate it about the centre of the sphere, and then convert it back to ERP image coordinates using  $\mathcal{P}$ . Applied to optical flow, this yields:

$$\mathcal{F}(\mathbf{x}) = \mathcal{F}^{\text{EP}}(\mathbf{x}) - \mathbf{x} = \mathcal{P} \left( \hat{R}^\top \cdot \bar{R}^\top \cdot \mathcal{P}^{-1}(\tilde{\mathcal{F}}^{\text{EP}}(\mathbf{x})) \right) - \mathbf{x}. \quad (1)$$

## 4 Experiments

We evaluate our method on the Replica [52] and OmniPhotos [5] datasets and achieve state-of-the-art performance on both datasets. To evaluate the accuracy of 360° optical flow regardless projection, and to account for the wrap-around (Section 3.1), we measure the *spherical* average endpoint error (SEPE), *spherical* average angle error (SAAE), and the *spherical* root mean square error (SRMS). Our single-threaded Python implementation computes 360° optical flow at 1280×640 resolution in about 20 seconds. We use DIS flow [23] for the tangent flow fields.

### 4.1 Datasets & Error Metrics

We evaluate the performance of our method on synthetic and real-world datasets using multiple spherical error metrics.

**Datasets.** Unlike depth sensors, there is no feasible hardware for capturing high-quality optical flow from a real-world scene, especially for 360° optical flow. For quantitative evaluation of optical flow, ground truth is necessary. Inspired by Shugrina et al. [30], we render ground-truth 360° images and optical flow from the reconstructed 3D mesh of a real-world scene. The ground-truth data is rendered using the Replica dataset [52], which not only contains high-quality HDR textures and 3D meshes, but also public code for a versatile renderer. However, the official Replica rendering pipeline does not support any 360° camera model or optical

flow generation. We therefore implemented an ERP camera model using OpenGL geometry shaders that transform the 3D mesh from Cartesian coordinates to spherical coordinates, finally render the 3D mesh and ground-truth optical flow in the equirectangular format.

We render three types of camera paths for ground-truth evaluation at a resolution of  $1280 \times 640$  pixels. A *Circle*, in which the camera moves along a 50 cm radius circle and faces outwards, with images rendered every  $10^\circ$ . A *Line*, where the camera faces the same direction and moves in a straight line, with images rendered every 20 cm. And *Random*, with camera centres uniformly randomly sampled within a box of  $1 \times 1 \times 1$  metres and rotations jittered by  $\pm 10^\circ$  degrees along each of the Euler angle axes.

Although the synthetic dataset is reasonably realistic, real-world data tends to be more challenging, with complex lighting and motions. To further explore our method’s performance qualitatively, we also test our method on the OmniPhotos dataset [9], which contains diverse  $360^\circ$  videos of real scenes captured with a high-quality  $360^\circ$  video camera.

**Error Metrics.** Optical flow evaluation commonly uses the average angle error (AAE), end-point error (EPE), and root-mean-square error (RMS) metrics [4]. To evaluate spherical  $360^\circ$  optical flow, we extend these metrics to the spherical domain by mapping ERP pixel coordinates to spherical coordinates on the unit sphere, and measuring geodesic distances instead of Euclidean distances in image space as used for perspective flow metrics. This overcomes both distortions due to the equirectangular projection and also the wrap-around on the sphere. Specifically, we propose the spherical AAE (SAAE), spherical EPE (SEPE, see Equation 2) and spherical RMS (SRMS) metrics to evaluate  $360^\circ$  optical flow. The spherical AAE (SAAE) measures the angle between the estimated and ground-truth optical flow vectors on the surface of a unit sphere using spherical trigonometry. For SEPE and SRMS,  $d(\cdot, \cdot)$  is the geodesic distance between endpoints on the unit sphere:

$$\text{SEPE} = \frac{1}{|\Omega|} \sum_{i \in \Omega} d((\theta_i^{\text{est}}, \phi_i^{\text{est}}), (\theta_i^{\text{GT}}, \phi_i^{\text{GT}})). \quad (2)$$

## 4.2 Comparison

We compare our method with OmniFlowNet [4], a state-of-the-art  $360^\circ$  optical flow method, as well as the following state-of-the-art perspective methods: RAFT [39], PWC-Net [36] and DIS [23]. For OmniFlowNet<sup>2</sup>, RAFT<sup>3</sup> and PWC-Net<sup>4</sup>, we use the official released code, and for DIS, we use OpenCV’s implementation. OmniFlowNet ran out of GPU memory on our NVIDIA RTX 2060 (6 GB RAM), so we downscale the input images to a resolution of  $1024 \times 512$  pixels, and upsample the estimated flow fields back to the original image resolution afterwards. The performance evaluation comprises quantitative and qualitative evaluation on the Replica  $360^\circ$  and OmniPhotos datasets.

**Quantitative Evaluation.** We show a quantitative comparison on the Replica  $360^\circ$  dataset using both perspective and spherical optical flow evaluation metrics in Table 1. The table shows that our method achieves the best performance across the board. All methods tend to be better on the *Circle* paths than *Line* paths, as the latter contains larger displacements that are more difficult to estimate accurately. Meanwhile, our global rotation warping pre-aligns the input images, so that our method more easily handles camera ego-motion.

<sup>2</sup><https://github.com/COATZ/OmniFlowNet>

<sup>3</sup><https://github.com/princeton-vl/RAFT>

<sup>4</sup><https://github.com/NVLabs/PWC-Net>



Table 1: Optical flow errors on the synthetic Replica 360° dataset. See Section 4.1 for metrics.

	Method	<i>EPE</i> ↓	<i>AAE</i> ↓	<i>RMS</i> ↓	<i>SEPE</i> ↓	<i>SAAE</i> ↓	<i>SRMS</i> ↓
<i>Circle</i>	OmniFlowNet	15.12	0.2618	60.79	0.06025	0.1316	0.3963
	PWC-Net	15.73	0.2701	61.50	0.05403	0.1406	0.3693
	RAFT	15.75	0.2769	63.34	0.04619	0.1461	0.3349
	DIS	16.31	0.2818	62.29	0.05430	0.1503	0.3578
	Ours	<b>3.507</b>	<b>0.1694</b>	<b>34.21</b>	<b>0.005370</b>	<b>0.03480</b>	<b>0.01021</b>
<i>Line</i>	OmniFlowNet	30.64	0.2515	92.36	0.1229	0.1276	0.5390
	PWC-Net	32.23	0.2594	93.27	0.1259	0.1358	0.5414
	RAFT	32.38	0.2935	97.46	0.08787	0.1642	0.4180
	DIS	36.06	0.3357	99.07	0.1127	0.2093	0.4668
	Ours	<b>5.839</b>	<b>0.1971</b>	<b>40.74</b>	<b>0.01063</b>	<b>0.05951</b>	<b>0.02098</b>
<i>Random</i>	OmniFlowNet	40.50	0.3229	106.2	0.1321	0.1940	0.5034
	PWC-Net	41.98	0.3294	106.9	0.1438	0.2118	0.5203
	RAFT	38.31	0.2978	104.2	0.1143	0.1766	0.4557
	DIS	47.29	0.4203	113.2	0.1728	0.3000	0.5545
	Ours	<b>14.10</b>	<b>0.2192</b>	<b>59.78</b>	<b>0.02717</b>	<b>0.08849</b>	<b>0.05753</b>
<i>All</i>	OmniFlowNet	28.76	0.2788	86.48	0.1051	0.1511	0.4797
	PWC-Net	29.98	0.2863	87.23	0.1079	0.1627	0.4770
	RAFT	28.81	0.2893	88.34	0.08278	0.1623	0.4029
	DIS	33.22	0.3459	91.54	0.1133	0.2199	0.4597
	Ours	<b>7.701</b>	<b>0.1946</b>	<b>44.62</b>	<b>0.01411</b>	<b>0.06027</b>	<b>0.02905</b>

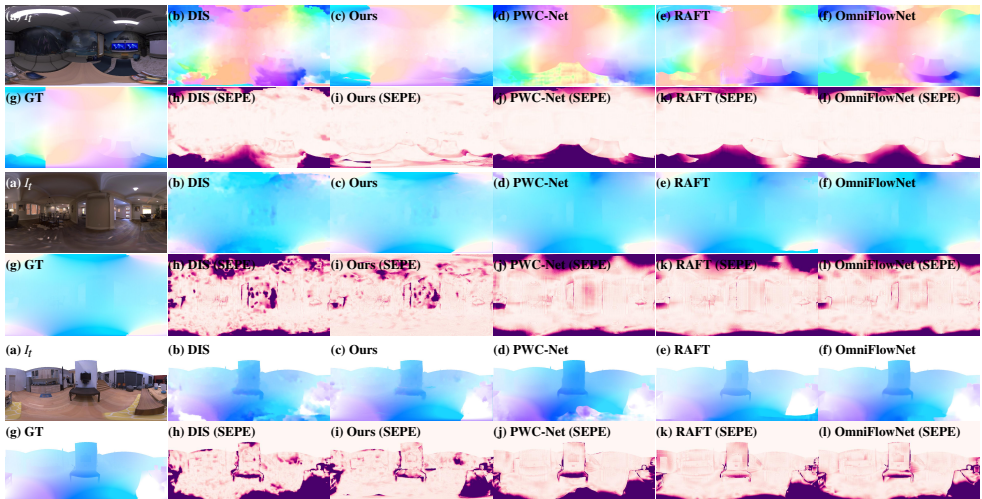


Figure 4: Estimated 360° optical flow and error heatmaps on the Replica 360° dataset: (a) Source image. (b–f) Estimated flow fields. (g) Ground-truth flow. (h–l) SEPE (spherical end-point error) heatmaps (lighter is better). Top: *office\_0* scene, line camera motion. Middle: *apartment\_0* scene, circle camera motion. Bottom: *frl\_apartment\_0* scene, random camera motion.

Figure 4 shows some example estimated 360° optical flow fields and their SEPE visualised as heatmaps (lighter is better). Other methods mainly introduce errors on the top and bottom edges of the optical flow. Our method has the lowest errors at the top and bottom, while the

middle region is quite similar to the DIS result.

Based on the [Table 1](#) and [Figure 4](#), we can conclude that perspective optical flow cannot correctly handle the top and bottom regions of ERP images, but our method can help them improve dramatically in these areas. Our method succeeds where the underlying optical flow method (DIS) fails, while maintaining the same performance in the equatorial image region.

**Qualitative Evaluation.** To analyse the performance of 360° optical flow methods on a real-world dataset without ground-truth optical flow, we measure the interpolation error  $[4]$ , i.e. the RGB colour difference between the source image and backward-warped target image. [Figure 5](#) shows an example result from the OmniPhotos dataset. The error maps in [Figure 5](#)(m–q) show less error at the top and bottom of images for our method’s result. The differences between methods is less pronounced than in the Replica 360° dataset, as the baseline between frames is smaller in OmniPhotos due to a higher video frame rate.

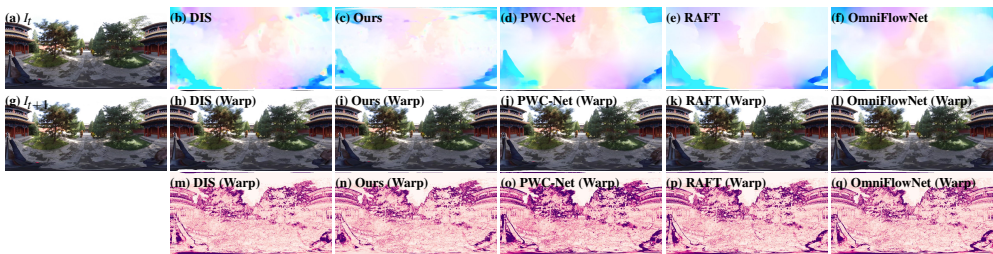


Figure 5: Backward warping results on OmniPhotos’ *BeihaiPark* scene: (a, g) Source and target images. (b–f) Estimated flow fields. (h–l) Target image (g) warped to the source image (a) using the estimated optical flow (b–f). (m–q) Interpolation error heatmaps (lighter is better).

Table 2: The results of our ablation study. See [Section 4.3](#) for details.

	Method	<i>EPE</i> ↓	<i>AAE</i> ↓	<i>RMS</i> ↓	<i>SEPE</i> ↓	<i>SAAE</i> ↓	<i>SRMS</i> ↓
<i>Circle</i>	Full method	3.507	0.1694	34.21	<b>0.005370</b>	0.03480	0.01021
	w/o weight	3.616	0.1692	34.86	0.005405	0.03478	0.01235
	w/o ERP	<b>3.434</b>	<b>0.1688</b>	<b>33.81</b>	0.005376	<b>0.03437</b>	0.009951
	w/o cubemap	3.533	0.1698	34.04	0.005399	0.03519	<b>0.009730</b>
	w/o ico	3.481	0.1711	30.45	0.006040	0.03724	0.01412
<i>Line</i>	Full method	5.839	0.1971	40.74	0.01063	0.05951	0.02098
	w/o weight	5.475	0.1960	32.36	0.01100	<b>0.05815</b>	0.02214
	w/o ERP	5.749	<b>0.1945</b>	40.11	0.01069	0.05881	0.02107
	w/o cubemap	5.816	0.1953	40.09	0.01075	0.05825	0.02136
	w/o ico	<b>4.262</b>	0.2042	<b>26.28</b>	<b>0.01015</b>	0.07242	<b>0.01818</b>
<i>Random</i>	Full method	14.10	0.2192	59.78	0.02717	0.08849	0.05753
	w/o weight	<b>13.66</b>	0.2176	57.87	0.02723	0.08679	0.05684
	w/o ERP	15.71	0.2211	60.81	0.03212	0.09007	0.06161
	w/o cubemap	13.96	<b>0.2145</b>	58.92	<b>0.02652</b>	<b>0.08428</b>	<b>0.05619</b>
	w/o ico	14.81	0.2490	<b>57.15</b>	0.03389	0.1167	0.06173
<i>All</i>	Full method	7.701	0.1946	44.62	<b>0.01411</b>	0.06027	<b>0.02905</b>
	w/o weight	7.585	0.1942	41.70	0.01455	0.05991	0.03045
	w/o ERP	8.298	0.1948	44.91	0.01606	0.06109	0.03088
	w/o cubemap	7.771	<b>0.1932</b>	44.35	0.01422	<b>0.05924</b>	0.02909
	w/o ico	<b>7.519</b>	0.2081	<b>37.96</b>	0.01670	0.07547	0.03134

### 4.3 Ablation Studies

**Global Rotation Warping.** The multi-step rotation warping aligns image  $I_{t+1}$  to  $I_t$  by estimating rotation using the ERP and cubemap optical flow. The results without ERP optical flow alignment (‘w/o ERP’) and without cubemap optical flow alignment (‘w/o cubemap’) are shown in [Table 2](#). In different type camera motion data different rotation gets best performance. But ‘Full method’ make most of and balance each term’s advantage, to make it get the best performance on ‘All’ data.

**Stitch Blending Weight.** The blending weight aims to smoothly and consistently stitch the overlap area of different faces’ optical flow. The result without blending weight is shown as ‘w/o weight’ in [Table 2](#), which replaces all weights with the unit weight. The ‘Full Method’ outperforms the ‘w/o weight’ because the blending weight brings down the unreliable optical flow, such as the pixels move out or in the tangent image boundary but their reliable optical flow can get from corresponding pixels on other tangent images.

## 5 Conclusion

We proposed a flexible method for estimating 360° optical flow using tangent images and global rotation warping. Tangent images use gnomonic projection to uniformly sample a 360° ERP image from an initial cubemap to the final icosahedron-based tangent images to overcome the equirectangular projection distortions. Any existing off-the-shelf optical flow method can be used on the tangent images. Our method uses global rotation warping to pre-align the input images and overcome large motions between frames. Finally, we blend optical flow to reduce discontinuities at face boundaries. Our evaluation shows that our method achieves state-of-the-art performance for 360° optical flow estimation.

Our proposed approach overcomes the limited resolutions supported by CNN-based methods due to model size constraints (e.g. 1024×512 [10]) by processing tangent images separately. This enables the estimation of high-resolution 360° optical flow, for example at 2K×1K resolution, which is not supported by existing methods.

**Limitations.** Large pose changes are a common limitation of most optical flow methods, which are primarily designed for handling small motions. In our method, we use global rotation alignment to effectively minimise the net disparity across the unit sphere. This helps in the case of moderate pose changes, but large pose changes remain challenging.

**Future Work.** Our method struggles with large camera translations; one could replace the global rotation warping with a meshgrid-based warping method that can pre-align both rotational and translational camera motions.

## Acknowledgements

We thank the reviewers for their valuable feedback that has helped us improve our paper. This work was supported by an EPSRC-UKRI Innovation Fellowship (EP/S001050/1) and RCUK grant CAMERA (EP/M023281/1, EP/T022523/1).

## References

- [1] Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning optical flow from still images. In *CVPR*, 2021.
- [2] Charles-Olivier Artizzu, Haozhou Zhang, Guillaume Allibert, and Cédric Demonceaux. OmniFlowNet: a perspective neural network adaptation for optical flow estimation in omnidirectional images. In *ICPR*, 2020. doi:[10.1109/ICPR48806.2021.9412745](https://doi.org/10.1109/ICPR48806.2021.9412745).
- [3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *TPAMI*, 9(5):698–700, 1987. doi:[10.1109/TPAMI.1987.4767965](https://doi.org/10.1109/TPAMI.1987.4767965).
- [4] Simon Baker, Daniel Scharstein, J. Lewis, Stefan Roth, Michael Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011. doi:[10.1007/s11263-010-0390-2](https://doi.org/10.1007/s11263-010-0390-2).
- [5] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. OmniPhotos: Casual 360° VR photography. *ACM Trans. Graph.*, 39(6):267:1–12, 2020. doi:[10.1145/3414685.3417770](https://doi.org/10.1145/3414685.3417770).
- [6] Keshav Bhandari, Ziliang Zong, and Yan Yan. Revisiting optical flow estimation in 360 videos. In *ICPR*, 2020. doi:[10.1109/ICPR48806.2021.9412035](https://doi.org/10.1109/ICPR48806.2021.9412035).
- [7] Michael J. Black and P. Anandan. Robust dynamic motion estimation over time. In *CVPR*, pages 296–302, 1991. doi:[10.1109/CVPR.1991.139705](https://doi.org/10.1109/CVPR.1991.139705).
- [8] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, volume 3024, pages 25–36, 2004. doi:[10.1007/978-3-540-24673-2\\_3](https://doi.org/10.1007/978-3-540-24673-2_3).
- [9] 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 *CVPR*, pages 1420–1429, 2018. doi:[10.1109/CVPR.2018.00154](https://doi.org/10.1109/CVPR.2018.00154).
- [10] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *ECCV*, pages 518–533, 2018. doi:[10.1007/978-3-030-01240-3\\_32](https://doi.org/10.1007/978-3-030-01240-3_32).
- [11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015. doi:[10.1109/ICCV.2015.316](https://doi.org/10.1109/ICCV.2015.316).
- [12] Marc Eder, Pierre Moulon, and Li Guan. Pano popups: Indoor 3D reconstruction with a plane-aware network. In *3DV*, pages 76–84, 2019. doi:[10.1109/3DV.2019.00018](https://doi.org/10.1109/3DV.2019.00018).
- [13] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent images for mitigating spherical distortion. In *CVPR*, 2020. doi:[10.1109/CVPR42600.2020.01244](https://doi.org/10.1109/CVPR42600.2020.01244).
- [14] Clara Fernandez-Labrador, Jose M. Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and Jose J. Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 5(2):1255–1262, 2020. doi:[10.1109/LRA.2020.2967274](https://doi.org/10.1109/LRA.2020.2967274).

- [15] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. doi:[10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2).
- [16] Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. 6-DOF VR videos with a single 360-camera. In *IEEE VR*, pages 37–44, 2017. doi:[10.1109/VR.2017.7892229](https://doi.org/10.1109/VR.2017.7892229).
- [17] Tak-Wai Hui, Xiaou Tang, and Chen Change Loy. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, pages 8981–8989, 2018. doi:[10.1109/CVPR.2018.00936](https://doi.org/10.1109/CVPR.2018.00936).
- [18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. doi:[10.1109/CVPR.2017.179](https://doi.org/10.1109/CVPR.2017.179).
- [19] Sunghoon Im, Hyowon Ha, François Rameau, Hae-Gon Jeon, Gyeongmin Choe, and In So Kweon. All-around depth from small motion with a spherical panoramic camera. In *ECCV*, 2016. doi:[10.1007/978-3-319-46487-9\\_10](https://doi.org/10.1007/978-3-319-46487-9_10).
- [20] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. UniFuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics and Automation Letters*, 6(2):1519–1526, 2021. doi:[10.1109/LRA.2021.3058957](https://doi.org/10.1109/LRA.2021.3058957).
- [21] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *CVPR*, pages 886–895, 2020. doi:[10.1109/CVPR42600.2020.00097](https://doi.org/10.1109/CVPR42600.2020.00097).
- [22] Johannes Kopf. 360° video stabilization. *ACM Trans. Graph.*, 35(6):195:1–9, 2016. doi:[10.1145/2980179.2982405](https://doi.org/10.1145/2980179.2982405).
- [23] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *ECCV*, pages 471–488, 2016. doi:[10.1007/978-3-319-46493-0\\_29](https://doi.org/10.1007/978-3-319-46493-0_29).
- [24] Po Kong Lai, Shuang Xie, Jochen Lang, and Robert Laganière. Real-time panoramic depth maps from omni-directional stereo images for 6 DoF videos in virtual reality. In *IEEE VR*, pages 405–412, 2019. doi:[10.1109/VR.2019.8798016](https://doi.org/10.1109/VR.2019.8798016).
- [25] Yeon Kun Lee, Jaeseok Jeong, Jong Seob Yun, Cho Won June, and Kuk-Jin Yoon. SpherePHD: Applying CNNs on a spherical PolyHeDron representation of 360° images. In *CVPR*, pages 9173–9181, 2019. doi:[10.1109/CVPR.2019.00940](https://doi.org/10.1109/CVPR.2019.00940).
- [26] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981.
- [27] Junren Luo, Wanpeng Zhang, Jiongming Su, and Fengtao Xiang. Hexagonal convolutional neural networks for hexagonal grids. *IEEE Access*, 7:142738–142749, 2019. doi:[10.1109/ACCESS.2019.2944766](https://doi.org/10.1109/ACCESS.2019.2944766).
- [28] Kevin Matzen, Michael F. Cohen, Bryce Evans, Johannes Kopf, and Richard Szeliski. Low-cost 360 stereo photography and video capture. *ACM Trans. Graph.*, 36(4):148:1–12, 2017. doi:[10.1145/3072959.3073645](https://doi.org/10.1145/3072959.3073645).

- [29] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. doi:[10.1016/j.isprsjprs.2017.09.013](https://doi.org/10.1016/j.isprsjprs.2017.09.013).
- [30] Maria Shugrina, Ziheng Liang, Amlan Kar, Jiaman Li, Angad Singh, Karan Singh, and Sanja Fidler. Creative flow+ dataset. In *CVPR*, pages 5384–5393, 2019. doi:[10.1109/CVPR.2019.00553](https://doi.org/10.1109/CVPR.2019.00553).
- [31] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using SVD. Note, 2017.
- [32] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. arXiv:[1906.05797](https://arxiv.org/abs/1906.05797), 2019.
- [33] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *CVPR*, pages 9442–9451, 2019. doi:[10.1109/CVPR.2019.00967](https://doi.org/10.1109/CVPR.2019.00967).
- [34] Cheng Sun, Min Sun, and Hwann-Tzong Chen. HoHoNet: 360 indoor holistic understanding with latent horizontal features. In *CVPR*, 2021.
- [35] Deqing Sun, Stefan Roth, and Michael J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 106(2): 115–137, 2014. doi:[10.1007/s11263-013-0644-x](https://doi.org/10.1007/s11263-013-0644-x).
- [36] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. doi:[10.1109/CVPR.2018.00931](https://doi.org/10.1109/CVPR.2018.00931).
- [37] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of CNNs for optical flow estimation. *TPAMI*, 42(6): 1408–1423, 2020. doi:[10.1109/TPAMI.2019.2894353](https://doi.org/10.1109/TPAMI.2019.2894353).
- [38] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *ECCV*, pages 732–750, 2018. doi:[10.1007/978-3-030-01270-0\\_43](https://doi.org/10.1007/978-3-030-01270-0_43).
- [39] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. doi:[10.1007/978-3-030-58536-5\\_24](https://doi.org/10.1007/978-3-030-58536-5_24).
- [40] Phi Vu Tran. SSLLayout360: Semi-supervised indoor layout estimation from 360-degree panorama. In *CVPR*, 2021.
- [41] Fu-En Wang, Hou-Ning Hu, Hsien-Tzu Cheng, Juan-Ting Lin, Shang-Ta Yang, Meng-Li Shih, Hung-Kuo Chu, and Min Sun. Self-supervised learning of depth and camera motion from 360° videos. In *ACCV*, pages 53–68, 2018. doi:[10.1007/978-3-030-20873-8\\_4](https://doi.org/10.1007/978-3-030-20873-8_4).

- [42] 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 *CVPR*, pages 462–471, 2020. doi:[10.1109/CVPR42600.2020.00054](https://doi.org/10.1109/CVPR42600.2020.00054).
- [43] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. LED<sup>2</sup>-net: Monocular 360 layout estimation via differentiable depth rendering. In *CVPR*, 2021.
- [44] Ning-Hsu Wang, Bolivar Solarte, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 360SD-net: 360° stereo depth estimation with learnable cost volume. In *ICRA*, pages 582–588, 2020. doi:[10.1109/ICRA40945.2020.9196975](https://doi.org/10.1109/ICRA40945.2020.9196975).
- [45] Jiale Xu, Jia Zheng, Yanyu Xu, Rui Tang, and Shenghua Gao. Layout-guided novel view synthesis from a single indoor panorama. In *CVPR*, 2021.
- [46] Kailun Yang, Jiaming Zhang, Simon Reiß, Xinxin Hu, and Rainer Stiefelhagen. Capturing omni-range context for omnidirectional segmentation. In *CVPR*, 2021.
- [47] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Joint 3D layout and depth prediction from a single indoor panorama image. In *ECCV*, 2020. doi:[10.1007/978-3-030-58517-4\\_39](https://doi.org/10.1007/978-3-030-58517-4_39).
- [48] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In *ICCV*, pages 3533–3541, 2019. doi:[10.1109/ICCV.2019.00363](https://doi.org/10.1109/ICCV.2019.00363).
- [49] Pengyu Zhao, Ansheng You, Yuanxing Zhang, Jiaying Liu, Kaigui Bian, and Yunhai Tong. Spherical criteria for fast and accurate 360° object detection. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, volume 34, pages 12959–12966, 2020. doi:[10.1609/aaai.v34i07.6995](https://doi.org/10.1609/aaai.v34i07.6995).
- [50] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. OmniDepth: Dense depth estimation for indoors spherical panoramas. In *ECCV*, pages 448–465, 2018. doi:[10.1007/978-3-030-01231-1\\_28](https://doi.org/10.1007/978-3-030-01231-1_28).
- [51] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. Spherical view synthesis for self-supervised 360° depth estimation. In *3DV*, pages 690–699, 2019. doi:[10.1109/3DV.2019.00081](https://doi.org/10.1109/3DV.2019.00081).