

# SCE-Depth: A Spherical Compound Eye Framework for Wide FOV Depth Estimation –Supplementary File–

Yi Zhu<sup>1,2,†</sup> Hao Xiong<sup>1,†</sup> Lin Xiao<sup>1</sup> Ranfeng Shi<sup>1</sup> Qinying Gu<sup>2</sup> Leilei Gu<sup>1,2,\*</sup>  
<sup>1</sup>Shanghai Jiao Tong University <sup>2</sup>Shanghai Artificial Intelligence Laboratory  
 {zhuyi2000, biaji233, xlin12, hh-fy424, leilei.gu}@sjtu.edu.cn, guqinying@pjlab.org.cn

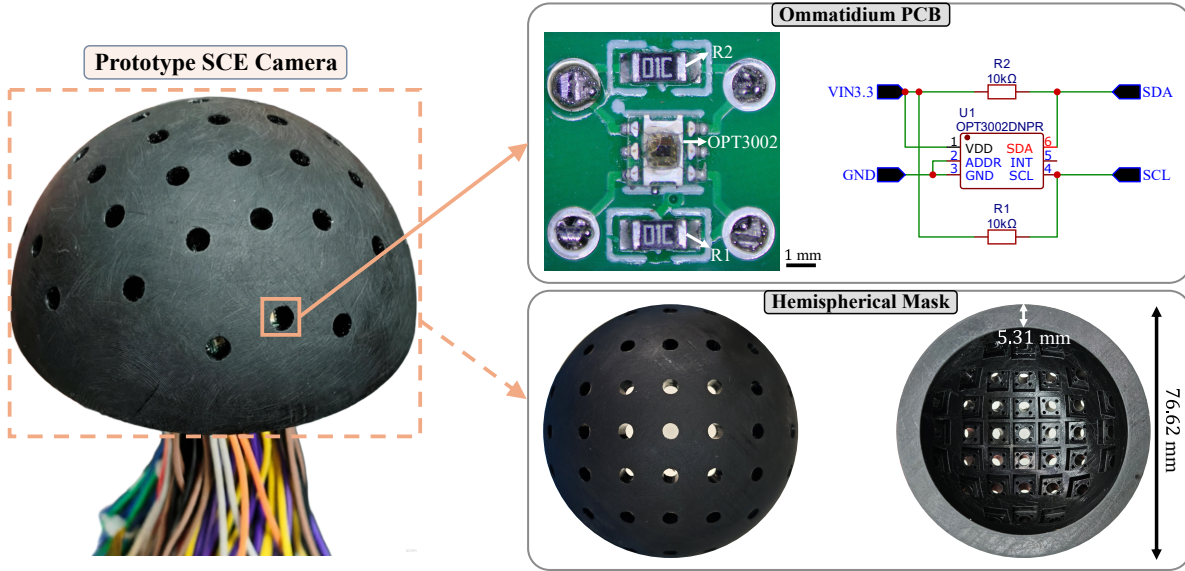


Figure 1. Prototype SCE camera and its hardware components. The left panel shows the assembled 37-ommatidium SCE camera, where each aperture houses an OPT3002-based ommatidium PCB; the right panels depict the corresponding ommatidium PCB and circuit schematic (top) and the 3D-printed hemispherical mask (diameter 76.62 mm, shell thickness 5.31 mm) with 37 apertures, whose inner–outer shell thickness defines an effective per-ommatidium FOV of approximately 45°.

*Due to space limitations in the main paper, we provide additional technical details in this supplementary document. In Sec. 1, we describe the fabrication of the prototype SCE camera, including sensor arrangement and optical isolation. Section 2 presents a comprehensive description of the CompoundDepth dataset, covering ray-tracing configurations, ommatidial placement on the spherical surface, and data preprocessing before training. In Sec. 3, we further analyze design choices of SCE-Depth, including the effects of the ommatidial FOV parameter  $\alpha$ , different gradient operators, and model scaling. In Sec. 4, we evaluate omnidirectional consistency by comparing robustness against fisheye-based methods under large viewpoint rotations. Finally, in Sec. 5, we present a depth-gradient for-*

*mal model that mathematically characterizes the compound eye’s depth-sensitive property.*

## 1. Prototype SCE Camera

We build a prototype SCE camera using OPT3002 light sensors. Each sensor is mounted on a custom 8 mm × 8 mm PCB that integrates the pull-up resistors and I<sup>2</sup>C interface, forming a compact ommatidium module; Fig. 1 shows the circuit schematic and the fabricated PCB. The ommatidium modules are grouped and read out via TCA9548 I<sup>2</sup>C multiplexers, with each TCA9548 handling up to eight modules, and all multiplexers are connected to a Raspberry Pi 4B for synchronized data acquisition.

To assemble the compound eye, we design a 3D-printed hemispherical mask with an outer diameter of 76.62 mm and a shell thickness of 5.31 mm, as shown in Fig. 1. The

<sup>†</sup>Equal contribution.

<sup>\*</sup>Leilei Gu is the corresponding author.

mask contains 37 apertures distributed on the hemisphere, and one OPT3002-based ommatidium module is inserted behind each aperture. For each ommatidium, the effective field of view (FOV)  $\alpha$  is determined purely by the geometry of the aperture and shell thickness, and is approximately  $45^\circ$  in the current prototype. This value is intentionally chosen to be larger than the inter-ommatidial angular spacing  $\beta$  (which varies slightly due to the near-square layout), ensuring substantial overlap between neighboring FOVs and matching the overlap behavior assumed in our simulation.

In the simulated SCE camera, the number of ommatidia is on the order of  $2 \times 10^4$ , which requires a much smaller per-ommatidium FOV ( $\alpha \in [1^\circ, 7^\circ]$ ) to avoid excessive blur. In contrast, the physical prototype has only 37 ommatidia, so a larger  $\alpha$  is necessary to maintain FOV overlap. As long as  $\alpha > \beta$ , the prototype preserves the key property observed in simulation, the intensity difference  $\Delta \text{Intensity}$  between overlapping ommatidia decreases as the object distance increases, enabling a consistent validation of the depth-sensitive behavior of the SCE design.

## 2. CompoundDepth Datasets

### 2.1. Simulator Setting

Simulation data for the CompoundDepth dataset are generated using NVIDIA Isaac Sim, which provides a physically based rendering engine and supports scene authoring through USD assets. Custom Python scripts are used to instantiate scenes, configure cameras, and automate data capture, with the full implementation provided in the supplementary file `SCE_simulator.py`.

To construct spherical SCE data, virtual cameras are placed at predefined ommatidial locations on a hemispherical icosahedron mesh. Due to implementation constraints in Isaac Sim, cameras are instantiated in batches of ten for rendering, but each ommatidium is always modeled as an individual camera. Every ommatidium renders a  $100 \times 100$  RGB patch together with a corresponding depth map. The 3D positions and viewing directions of all ommatidia are provided in the supplementary files `eye_position.npy` and `eye_direction.npy`. As illustrated in Fig. 2, these positions are obtained by first constructing a spherical mesh through iterative subdivision of a regular icosahedron and then selecting the hemispherical subset with  $x < 0$ , yielding a well-distributed hemispherical sampling with 20,609 ommatidia. Each ommatidium’s viewing direction is aligned with the radial vector from the sphere center to its corresponding position on the tessellated mesh.

Fig. 3(a) illustrates the office environment used for data acquisition, where each red marker denotes a camera placement on a 1.4 m sampling grid. These locations provide dense coverage across the scene and ensure diverse viewpoints. Fig. 3(b) presents a second indoor environment, a

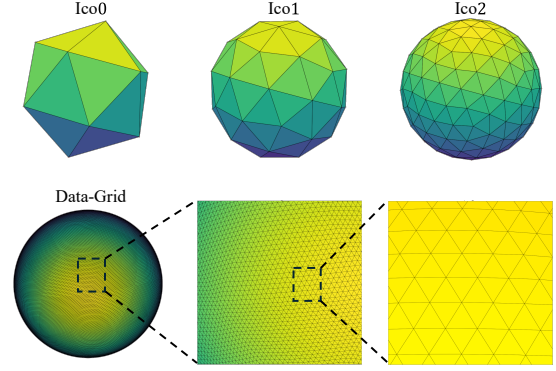


Figure 2. Icosahedron-based spherical sampling used for ommatidial placement. *Top*: A regular icosahedron and its subdivisions (0–2 iterations), illustrating the refinement process toward a uniform spherical mesh. *Bottom*: The resulting high-resolution hemispherical grid used for SCE data acquisition, where each vertex corresponds to an ommatidium location.

hospital layout, captured using the same procedure. Incorporating multiple indoor scenes increases variation in geometry and appearance, thereby improving the generalization ability of the trained model.

At each sampling point, the camera performs a horizontal sweep by capturing an image and then rotating  $90$  degrees clockwise about its vertical axis. Four images are collected per point, completing a full  $360$  degree rotation with  $45$  degree overlap between adjacent views. This configuration eliminates blind spots and ensures that all objects in the environment are observed from multiple directions.

For fisheye images, a single virtual camera is created using the polynomial fisheye model with a  $180^\circ$  FOV. Its effective resolution is matched to the SCE hemisphere to ensure consistent supervision across modalities. Figure 4 illustrates the rendering interface, including the fisheye visualization and corresponding depth outputs.

### 2.2. Data Pre-processing

Each virtual ommatidium renders a  $100 \times 100$  RGB patch together with a corresponding depth map. To obtain a single measurement per ommatidium, we apply a circular mask of radius 50 px centered on the patch, retaining only rays that fall within the effective aperture. Grayscale intensity is computed as the mean pixel value inside this mask. For depth, values larger than 10 m are clipped before averaging over the masked region to produce a single depth value per ommatidium. In Isaac Sim, the hemispherical SCE is rendered using ten virtual cameras, each responsible for a disjoint subset of ommatidia. We iterate over these camera folders, apply the above masking and aggregation to every patch and depth map, and concatenate the per-camera results into two one-dimensional arrays (inten-

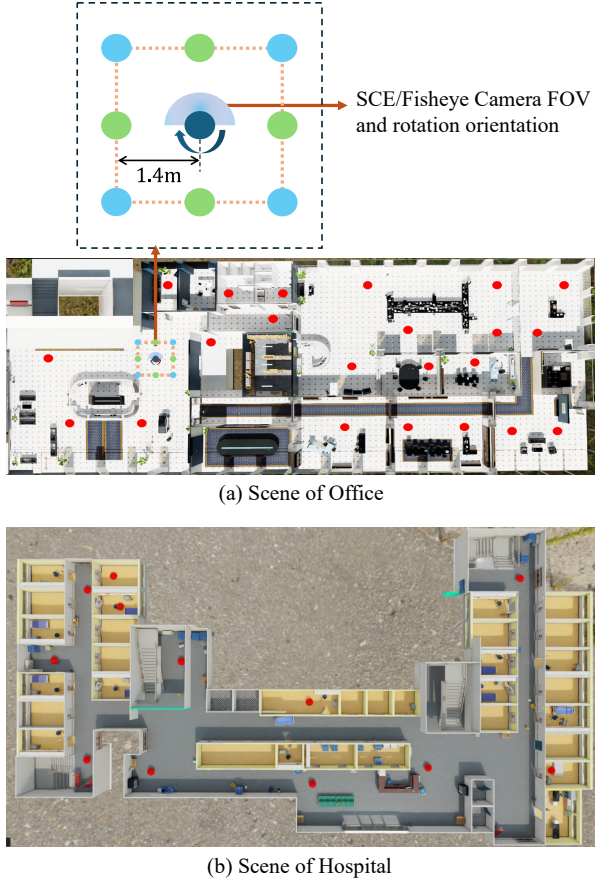


Figure 3. (a) Top view of the office scene used for data collection. (b) Top view of the hospital scene. Fisheye and SCE images are captured at representative sampling locations in both environments.

sity and depth) that together describe a complete hemispherical measurement with 20,609 ommatidia. The corresponding implementation is provided in the supplementary file `post_process.py`.

For the FisheyeDepth dataset, a similar pre-processing strategy is applied to minimize modality-induced discrepancies between fisheye and SCE inputs. Isaac Sim provides depth only at pixel centers for the fisheye camera, which differs fundamentally from the SCE’s per-ommatidium averaging model. To approximate the SCE sampling behavior, fisheye images are first rendered at  $1620 \times 1620$  resolution and converted to grayscale. A  $10 \times 10$  average pooling kernel is then applied to both the grayscale image and the depth map, producing a  $162 \times 162$  representation with 20,612 effective pixels. Each pooled value corresponds to the local mean intensity or depth within the receptive region of the kernel, mirroring the SCE pipeline in which each ommatidium integrates all rays within its FOV into a single measurement. This ensures that differences between FisheyeDepth and CompoundDepth arise from imaging geometry rather

than from inconsistent sampling strategies.

To interface with SCE-Depth, both datasets are converted into a unified spherical representation using the HEALPix coordinates provided in `eye_healpix_131072.npy`. CompoundDepth is directly resampled onto this 131,072-point mesh, while FisheyeDepth is first lifted to the sphere via the equidistant model and then resampled in the same manner. This normalization introduces only minor spherical resampling errors ( $4.99 \times 10^{-3}$  and  $5.16 \times 10^{-3}$ ) and serves purely to standardize the input format.

### 3. Design Choices of SCE-Depth

#### 3.1. Gradient Operator

To analyze our Spherical Gradient Feature Extractor (SGFE), we compute gradient maps on SCE grayscale inputs with  $\alpha = 3^\circ$ . Figure 5 compares Laplacian, Sobel, and Scharr. Sobel yields clean, stable gradients that preserve overlap-induced SCE structures, whereas Laplacian is too weak and Scharr amplifies high-frequency noise.

Since SGFE features are concatenated with the grayscale input, we prefer an operator with a compatible magnitude scale for stable joint learning. Sobel best matches this requirement, so we use it as the default. Implementation details are provided in `sgfe.py`.

#### 3.2. Ommatidial FOV

Figure 6 studies how the ommatidial FOV  $\alpha$  affects SCE imaging and depth estimation. Increasing  $\alpha$  from  $1^\circ$  to  $7^\circ$  increases inter-ommatidial overlap, yielding smoother but blurrier images. With  $\alpha = 1^\circ$ , images are sharp but lack overlap, leading to discontinuities that harm geometric consistency. Larger  $\alpha$  values suppress fine structures, especially near depth boundaries. Moderate overlap at  $\alpha = 3^\circ$  provides the best trade-off and yields accurate predictions, whereas  $\alpha \in \{5^\circ, 7^\circ\}$  introduces excessive blur and degrades depth estimation.

#### 3.3. Model Size

To assess the feasibility of edge deployment, we further reduce the model size to improve inference speed. SCE-Depth-Small (10.4M) is obtained by decreasing the SwinHP embedding dimension from 96 to 48 while keeping all other settings unchanged. As shown in Tab. 1, this lightweight variant incurs a modest accuracy drop compared to the full model, yet remains significantly stronger than the HealSwin baseline. In return, it achieves 117.2 FPS on an RTX 4090, nearly  $2\times$  faster than the original 60.9 FPS. Using a first-order linear TOPS scaling, this corresponds to a coarse estimate of  $\approx 13.9$  FPS on Jetson Orin NX (1321 vs. 157 TOPS), suggesting promising potential for real-time edge deployment.



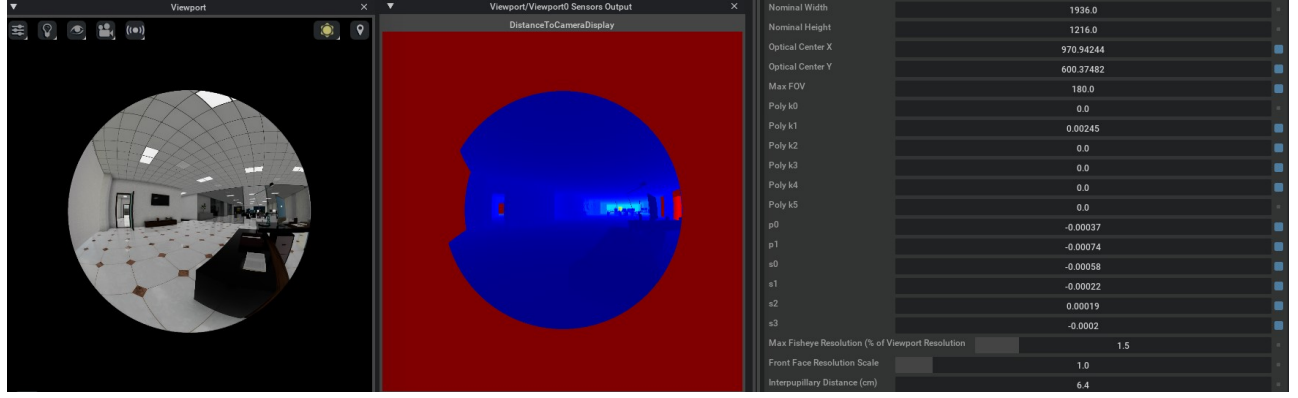


Figure 4. Isaac Sim interface used for generating datasets. *Left*: Rendered fisheye RGB view with a  $180^\circ$  FOV. *Center*: Depth map visualized using Isaac Sim’s built-in colormap scale (note that this scale differs from that used in the main paper). *Right*: Fisheye camera configuration parameters.

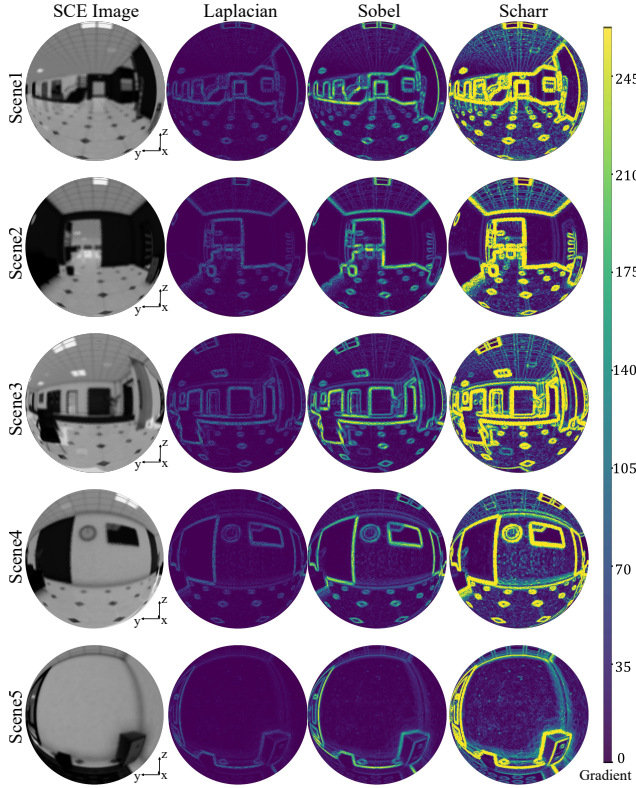


Figure 5. Comparison of SGFE gradient maps computed with Laplacian, Sobel, and Scharr operators on three representative scenes, using SCE grayscale inputs with  $\alpha = 3^\circ$ . Sobel-based SGFE produces clean and informative gradients that preserve overlap-induced structure, whereas Laplacian-based SGFE responses are too weak and Scharr-based responses excessively amplify noise.

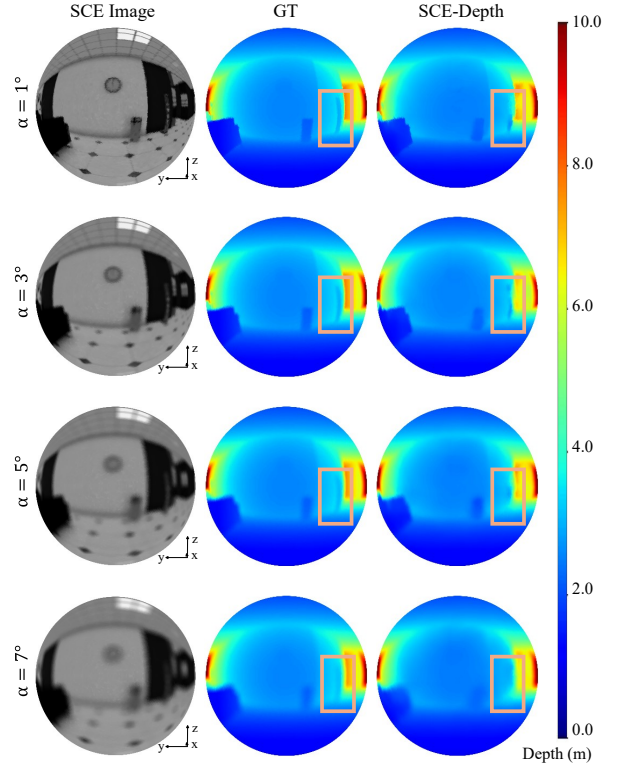


Figure 6. Depth estimation results across different ommatidial FOVs, showing the SCE grayscale input, ground-truth depth, and SCE-Depth predictions for  $\alpha \in \{1^\circ, 3^\circ, 5^\circ, 7^\circ\}$ . The  $\alpha = 3^\circ$  configuration achieves the best balance between sharpness and overlap, whereas  $\alpha = 1^\circ$  lacks sufficient inter-ommatidial overlap and  $\alpha \in 5^\circ, 7^\circ$  produces overly blurred measurements.



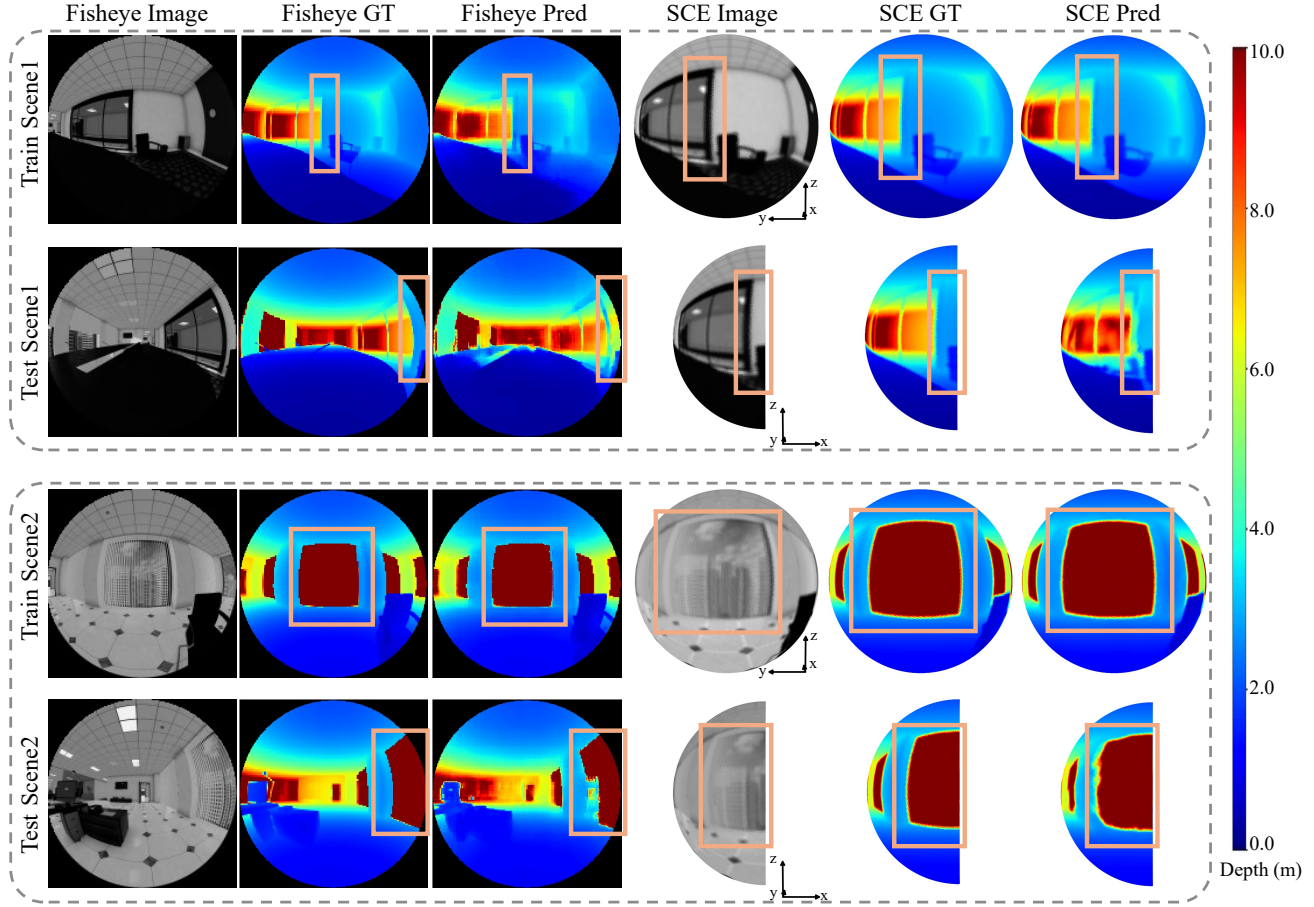


Figure 7. Comparison between the fisheye-based baseline (Fisheye-Swin) and our SCE-Depth on two representative train-test scene pairs. For each pair, the object of interest (wall in Scene 1, window in Scene 2) appears near the image center in the training view and near the periphery in the test view. While the fisheye model exhibits distortion-induced shape deformation and depth errors at the boundaries, SCE-Depth preserves both object geometry and depth consistency across viewpoints.

Model	#Paras(M)	Inference FPS $\uparrow$	RMSE $\downarrow$	Abs Rel $\downarrow$	$\delta_1 \uparrow$	$\delta_2 \uparrow$
HealSwin	41.3	61.1	0.350	0.082	92.76	96.99
SCE-Depth-Small	<b>10.4</b>	<b>117.2</b>	0.282	0.069	94.38	97.18
SCE-Depth	41.3	60.9	<b>0.256</b>	<b>0.050</b>	<b>95.74</b>	<b>98.48</b>

Table 1. Accuracy-efficiency comparison across HealSwin, SCE-Depth, and its lightweight variant (green: newly added).

#### 4. Omnidirectional Consistency

To further analyze the omnidirectional consistency of SCE-Depth, we examine objects that appear in both the training and test splits but at different azimuth angles. We select cases where the object is located near the center of the FOV in the training scene, while the same object moves to the periphery in the test scene due to the acquisition protocol, which rotates the camera around the scene by at least  $90^\circ$ . Figure 7 compares the fisheye based baseline (Fisheye-Swin) with our SCE-Depth in two representative scenarios.

In Scene 1, the reference object is a wall that lies in front of the camera. When the wall is centered in the training

view, both methods produce accurate depth estimates. In the test view, the same wall appears near the right boundary. The fisheye image suffers from strong peripheral distortion, and the baseline predicts an incorrect depth range where the wall region becomes green instead of the expected blue, indicating a depth underestimation. In contrast, our spherical imaging is distortion free on the unit sphere, so the appearance of the wall remains consistent, and SCE-Depth preserves the correct depth across both central and peripheral views.

Scene 2 focuses on a distant window. All objects farther than 10 m are encoded with the maximum depth, which appears in red in the ground truth. When the window is centered in the training scene, both methods recover the correct red depth. In the test scene, only part of the window moves to the right edge of the FOV. The fisheye baseline again degrades near the boundary: the window contour is distorted and large regions that should remain red are incorrectly pre-

dicted as blue. Our SCE-Depth model maintains both the rectangular shape of the window and the correct maximum depth, illustrating that the proposed spherical representation and network achieve more stable predictions under large viewpoint rotations.

## 5. Depth-Gradient Formal Model

While the main paper has already verified this depth-dependent property of the compound eye experimentally in both simulation and real hardware, a simple mathematical model can further substantiate and explain our finding. Any pair of SCE ommatidia with overlapping FOVs can be viewed as a single-pixel stereo pair: each ommatidium outputs a scalar intensity, so stereo disparity is replaced by an inter-ommatidial intensity difference (gradient). With many overlapping pairs across the SCE, this forms a multi-view system with adjustable baselines.

As illustrated in Fig. 8, we consider a minimal 2D step-edge scene observed by two neighboring ommatidia on the spherical compound eye. Each ommatidium has FOV  $\alpha$ , the angular spacing between their optical axes is  $\beta$ , the SCE radius is  $r$ , and the corresponding center-to-center baseline is approximated by  $b \approx r\beta$  for small  $\beta$  (in radians). The scene contains a single bright–dark edge, with constant radiance  $L_1$  on the left and  $L_2$  on the right. For ommatidium  $i \in \{L, R\}$ ,  $\theta_i(d)$  denotes the signed angular offset of the edge relative to its optical axis at depth  $d$ . Under a box angular kernel, the response of ommatidium  $i$  is the average radiance over its FOV:

$$I_i(d) = \frac{1}{\alpha} \left[ \int_{-\alpha/2}^{\theta_i(d)} L_1 d\theta + \int_{\theta_i(d)}^{\alpha/2} L_2 d\theta \right] \quad (1)$$

$$= \frac{L_1 + L_2}{2} + \frac{L_1 - L_2}{\alpha} \theta_i(d), \quad i \in \{L, R\},$$

We define the inter-ommatidial gradient  $\Delta I(d)$  as the magnitude of the intensity difference between the left and right ommatidia (i.e.,  $\Delta Intensity$  in the main paper experiments):

$$\Delta I(d) = |I_L(d) - I_R(d)|$$

$$= \left| \frac{L_1 - L_2}{\alpha} (\theta_L(d) - \theta_R(d)) \right|, \quad (2)$$

The relative edge offsets differ by two parts: a constant term  $\beta$  from the axis-orientation difference, and a depth-dependent disparity term  $\delta(d)$ :

$$\theta_L(d) - \theta_R(d) = \beta + \delta(d), \quad (3)$$

Under the far-field small-angle assumption  $d \gg b$ , standard stereo geometry gives

$$\delta(d) \approx \frac{b}{d} \approx \frac{r\beta}{d} \quad (d \gg b), \quad (4)$$

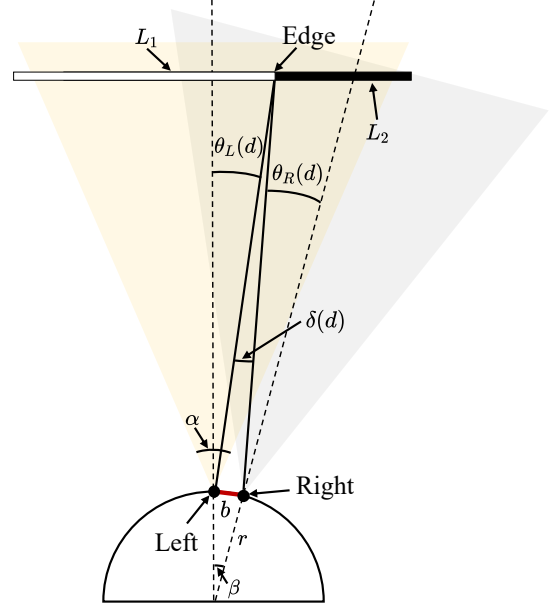


Figure 8. Depth-Gradient formal model.

Substituting Eqs. (3) and (4) into Eq. (2) yields the closed-form gradient:

$$\Delta I(d) = \left| \frac{L_1 - L_2}{\alpha} \left( \beta + \frac{r\beta}{d} \right) \right|, \quad (5)$$

Differentiating Eq. (5) with respect to  $d$  gives

$$\frac{\partial \Delta I(d)}{\partial d} = - \frac{|L_1 - L_2| r \beta}{\alpha} \cdot \frac{1}{d^2}. \quad (6)$$

Therefore,  $\Delta I(d)$  decreases as  $d$  increases, and its depth-varying component follows the expected inverse-depth trend, consistent with our empirical observations.