2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

FastViDAR: Real-Time Omnidirectional Depth Estimation via Alternative Hierarchical Attention

Hangtian Zhao[1,2], Xiang Chen[3], Yizhe Li[4], Qianhao Wang[5], Haibo Lu[1,*], and Fei Gao[5,*]

***Abstract*— In this paper, we propose FastViDAR, a novel framework that takes four fisheye camera inputs and produces a full 360° depth map along with per-camera depth, fusion depth, and confidence estimates. Our main contributions are: (1) We introduce an Alternative Hierarchical Attention (AHA) mechanism that efficiently fuses features across views through separate intra-frame and inter-frame windowed self-attention, achieving cross-view feature mixing with reduced overhead. (2) We propose a novel equirectangular projection (ERP) fusion approach that projects multi-view depth estimates to a shared equirectangular coordinate system to obtain the final fusion depth. (3) We generate ERP image-depth pairs using HM3D and 2D-3D-S datasets for comprehensive evaluation, demonstrating competitive zero-shot performance on real datasets while achieving up to 20 FPS on NVIDIA Orin NX embedded hardware. Project page: https://zhao-hangtian.github.io/FastViDAR/**

I. Introduction

Fast and reliable omnidirectional depth is crucial for robotics and autonomous driving. Active sensors (e.g., LiDAR) provide accurate 360° depth but are costly and power-hungry, whereas multi-camera rigs with fisheye lenses offer a practical alternative. A four-camera rig with ultra-wide field of view (FOV) ($> 180^\circ$) covers the full sphere, but inferring a consistent, accurate, and efficient depth map from these views remains challenging. Classic extensions of stereo to fisheye rely on spherical epipolar geometry and plane sweeping with volumetric cost aggregation, which often hampers real-time deployment and assumes perfect inter-camera extrinsics [1], [2]. Recent monocular approaches generalize well across cameras by factoring out intrinsics; notably Depth Any Camera [3] converts inputs (perspective/fisheye/panorama) to a common ERP representation to achieve zero-shot metric depth. However, single-image methods cannot leverage multi-view geometry nor estimate inter-camera parameters. In parallel, transformer-based multi-view models such as VGGT [4] aggregate cross-view information with alternating local/global attention and predict camera parameters and dense depth in a feed-forward manner, suggesting self-attention [5] can replace heavy cost volumes—though achieving real-time inference on autonomous mobile robot platforms remains challenging.

Our approach. To address the real-time inference challenge of transformer-based multi-view models like VGGT [4], we propose FastViDAR, an efficient omnidirectional depth system that processes multiple fisheye images and produces a full 360° depth map along with per-camera depth, fusion depth, and confidence estimates in real time, as demonstrated in Figure 1. While our experiments demonstrate the method using four-camera setups, FastViDAR supports arbitrary numbers of cameras and arbitrary FOV cameras including ultra-wide FOV ($> 180^\circ$). We project each fisheye image to a unified ERP representation, avoiding the need for the model to learn various fisheye lens distortion parameters and allowing it to focus on the ERP representation. Building on hierarchical/windowed attention [6], we introduce an AHA mechanism that alternates frame-local windowed self-attention with cross-frame attention over corresponding windows. Cross-view tokens are concatenated and blended via MLPs, acting as global tokens that propagate depth cues without explicit 3D or 4D cost volumes. Compared to pure window attention methods, AHA achieves good cross-frame self-attention capability with less than 10% additional computational and memory overhead, enabling accurate depth prediction for regions outside each frame's FOV and improving multi-frame depth consistency. Compared to

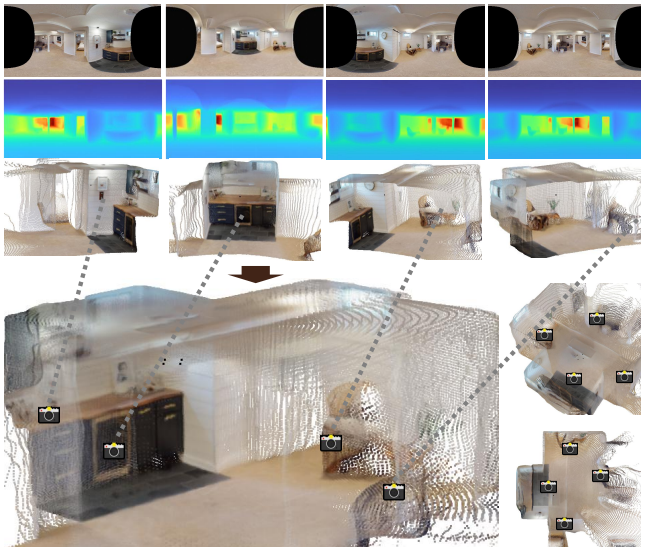

Fig. 1: Real-world performance demonstration of FastViDAR. First row: ERP images converted from 220° FOV fisheye images. Second row: ERP depth maps output by FastViDAR for each viewpoint. Third row: Predicted point clouds corresponding to each viewpoint. Fourth row: Predicted fused point clouds and omnidirectional fisheye camera orientations.

[1]Peng Cheng Laboratory, Shenzhen, China.
[2]Differential Robotics Technology Co., Ltd., Hangzhou, China.
[3]East China Normal University, Shanghai, China.
[4]Xidian University, Xi'an, China.
[5]FAST Lab, Zhejiang University, Hangzhou, China.
*Corresponding authors: Haibo Lu (luhb@pcl.ac.cn) and Fei Gao (feigao@zju.edu.cn).

full attention, AHA achieves around 16× inference speed improvement in theory. In practice, FastViDAR achieves 3.3× speedup over VGGT at 640×320 resolution with 4 frames, with this advantage expanding as input resolution or frame count increases.

Evaluation. On real-world benchmarks including HM3D [7] and Stanford 2D-3D-S [8], our AHA and ERP fusion methods contribute to accuracy and robustness. FastViDAR shows competitive performance compared to recent omnidirectional stereo and transformer baselines, and generalizes zero-shot to real panoramic data, delivering dense, accurate 360° depth in real time. On embedded hardware platforms such as NVIDIA Orin NX, our method achieves up to 20 FPS inference speed with TensorRT fp16 optimization while maintaining high accuracy, demonstrating its practicality for real-world robotic applications.

Contributions. Our main contributions are:

1) **AHA Mechanism.** We introduce a novel attention mechanism that efficiently fuses features across views through separate intra-frame and inter-frame windowed self-attention, achieving cross-view feature mixing with reduced overhead while enabling real-time processing on embedded hardware.
2) **ERP Fusion Method.** We propose a novel ERP fusion approach that projects multi-view depth estimates to a shared equirectangular coordinate system, enabling seamless 360° depth fusion without expensive point cloud alignment.
3) **ERP-aware Dataset Generation and Evaluation.** We generate ERP-aware image-depth pairs using HM3D and 2D-3D-S datasets for comprehensive evaluation, demonstrating competitive zero-shot performance on real 360° datasets while maintaining real-time processing capabilities on embedded platforms.

II. Related Work

Omnidirectional multi-view depth. Early deep models extend stereo to fisheye by projecting to the sphere and building cost volumes; later works improve efficiency and accuracy via spherical plane sweeping and multi-stage aggregation. Recent real-time systems rectify multi-fisheye inputs to stereo panoramas [9] or adopt Cassini-like projections with lightweight stereo backbones and fusion [10]. These achieve strong accuracy/speed but usually assume fixed calibration and rely on explicit cost volumes. In contrast, FastViDAR fuses arbitrary views with the proposed AHA without constructing stereo volumes and employs ERP fusion for seamless 360° depth estimation, while flexibly outputting metric depth for each individual view.

Panoramic/fisheye monocular depth. Standard pinhole networks face challenges with limited field of view and multi-view depth consistency. Specialized spherical/cubemap representations mitigate distortion, while DAC [3] attains zero-shot metric depth by mapping any input to a standard ERP representation. In robotics, FreDSNet [11] jointly models monocular panoramic depth and semantics using Fast Fourier Convolutions for efficient single-panorama perception. However, monocular methods still face scale ambiguity and cannot exploit cross-view constraints. FastViDAR also uses ERP representation but employs global attention across frames to achieve implicit multi-view scale constraints and better scale consistency.

Efficient stereo and MVS. Efficiency-oriented designs reduce the burden of 3D cost volumes via 2D aggregation and lightweight modules [12]. Coarse-to-fine multi-view stereo (e.g., MVSNet/CasMVSNet [13], [14]) limits depth hypotheses. CasOmniMVS [1] adapts spherical sweeping density for omnidirectional scenes. Our method avoids explicit volumes altogether and relies on learned implicit correlations via AHA.

Transformers for 3D perception. Alternating local/global attention has proven effective for multi-view geometry and camera prediction [4]. However, in omnidirectional multi-fisheye settings, such generic multi-view transformers do not fully exploit spherical coverage and overlap, and in our setting they are both less accurate for depth and substantially slower than real-time requirements. Hierarchical/windowed attention such as FasterViT [6] scales attention with carrier tokens. FastViDAR tailors this paradigm to multi-camera omnidirectional depth, and we explicitly evaluate the contribution of global summary interaction and fusion strategy. This clarifies where the gains come from beyond straightforward backbone reuse.

III. Method

A. Fisheye Camera and ERP

We adopt ERP images as the network input to decouple lens-specific intrinsics from learning. Any *central* fisheye model (e.g., KB/equidistant/equisolid [15], OCamCalib polynomial [16], Double Sphere (DSCamera) [17], unified central [18]) maps a pixel $\mathbf{u} = (u, v)$ to a unit viewing ray $\mathbf{d} = (d_x, d_y, d_z) \in \mathbb{S}^2$ in the camera frame via π_θ^{-1} (intrinsics θ). After this step, ERP coordinates are a *pure spherical reparameterization* independent of θ; we take $+z$ forward, $+x$ right, $+y$ up, ERP size $W \times H$ with origin at top-left, and longitude/latitude (λ, ϕ) in radians:

$$\lambda = \operatorname{atan2}(d_x, d_z), \qquad \phi = \arcsin(d_y), \tag{1}$$

$$x = \left(\tfrac{\lambda}{2\pi} + \tfrac{1}{2}\right)W, \qquad y = \left(\tfrac{1}{2} - \tfrac{\phi}{\pi}\right)H. \tag{2}$$

Thus images from heterogeneous fisheye lenses land on the same ERP lattice, as shown in Figure 2, allowing the network to learn on a stable, camera-agnostic domain while preserving the native wide FOV that perspective pinhole would crop away. Although ERP introduces polar area distortion (local scale $\propto 1/\cos\phi$), our experiments show the proposed method handles it well. For visualization or synthesis, the reverse path is direct: from ERP (x, y) recover (λ, ϕ), form $\mathbf{d} = [\sin\lambda\cos\phi,\ \sin\phi,\ \cos\lambda\cos\phi]^\top$, and project via $\mathbf{u} = \pi_\theta(\mathbf{d})$ (e.g., DSCamera) to render fisheye views within the lens FOV.

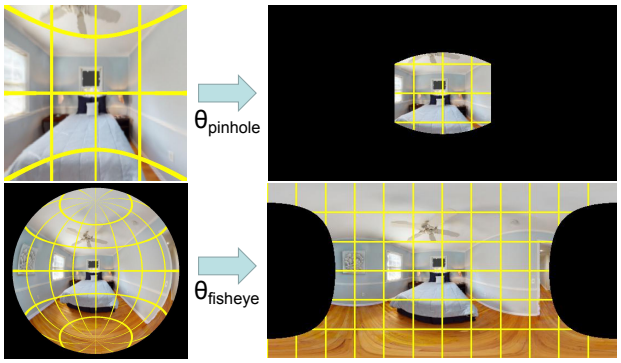

Fig. 2: Unified ERP representation for different camera types. Top row: pinhole camera (FOV=100°) with intrinsics θ_{pinhole} projects to ERP. Bottom row: fisheye camera (FOV=220°) with intrinsics θ_{fisheye} projects to the same ERP lattice. Yellow lines show the correspondence between ERP lattice positions and the original camera-specific lattice positions.

B. Alternative Hierarchical Attention

Monocular depth is fundamentally *scale ambiguous*. Under a calibrated pinhole model, if 3D points $\mathbf{X} \in \mathbb{R}^3$ and camera pose $(\mathbf{R}, \mathbf{t})$ explain the observations, then for any $\alpha > 0$, the projected image coordinates satisfy

$$\mathbf{x} \propto \mathbf{K}(\mathbf{R}(\alpha\mathbf{X}) + \alpha\mathbf{t}) = \alpha\,\mathbf{K}(\mathbf{R}\mathbf{X} + \mathbf{t}) \propto \mathbf{K}(\mathbf{R}\mathbf{X} + \mathbf{t}) \tag{3}$$

where $\mathbf{x} = (u, v, 1)^\top$ represents homogeneous image coordinates and $\mathbf{K}$ is the camera intrinsic matrix. Thus, depths are identifiable only up to a global scale in the absence of absolute metric cues (e.g., known baseline, object size priors, or other metric information). Multi-frame modeling constrains *relative* structure and motion; however, naïvely applying global self-attention over all tokens from S frames with N tokens each incurs quadratic cost $\mathcal{O}\big((SN)^2\big)$ in time and memory. VGGT [4] leverages multi-frame attention effectively yet remains computationally heavy for embedded deployment. Inspired by FasterViT [6] and convolutional tokenization as in CvT [19], we propose AHA, which applies window self-attention on *local tokens* and alternates frame- and global-level self-attention on compact *summary tokens*.

a) Overview and backbone: As shown in Figure 3, the model comprises *four stacked stages* and supports variable frame counts S (e.g., 4-camera rigs or temporal clips) and arbitrary input resolutions, enabled by adaptive padding and average pooling. In this paper, we primarily evaluate $S = 4$ camera settings. We therefore describe pose/count flexibility as an architectural property, while keeping empirical claims to the tested configurations. Stages 1-2 form a convolutional stem composed of repeated `conv2d-bn-gelu` blocks with downsampling at stage boundaries, producing feature maps $F \in \mathbb{R}^{B\times S\times C\times H\times W}$, where B is the batch size, S the number of frames/cameras, C channels, and $H\times W$ the input spatial size. Stage i outputs resolution $H/2^{i+2} \times W/2^{i+2}$ with $\approx 2^i C$ channels. Stage 3 stacks L *AHA blocks* that apply window self-attention on *local tokens* and alternate frame- and global-level self-attention on compact *summary tokens*, as detailed in Figure 4. Stage 4 performs *local refinement* via a stack of self-attention layers applied to the windowed local tokens.

b) Tokenizers and notation: We form two token sets from the stage-2 feature map $F \in \mathbb{R}^{B\times S\times C\times H\times W}$ by applying learnable positional bias to the feature maps. Windows are *non-overlapping* with size $P_h \times P_w$ and stride (P_h, P_w). To ensure complete window coverage, we use adaptive padding to $H' = \lceil H/P_h \rceil P_h$ and $W' = \lceil W/P_w \rceil P_w$, then partition each frame into $N_h = H'/P_h$ by $N_w = W'/P_w$ windows; the number of windows per frame is $M = N_h N_w$. We index windows within a frame by $m \in \{1, \dots, M\}$ (row-major over the $N_h \times N_w$ grid), and positions inside a window by $p \in \{1, \dots, P_h P_w\}$. For clarity, we denote the per-frame *local tokens* by $L_{b,s} \in \mathbb{R}^{M\times(P_h P_w)\times C}$ and the *local summary tokens* by $S^{\text{loc}}_{b,s} \in \mathbb{R}^{M\times C}$.

Tokenizers as operators (input → output shape):

$$\text{WinTok}_{(P_h,P_w)} : \mathbb{R}^{C\times H'\times W'} \to \mathbb{R}^{M\times(P_h P_w)\times C}, \tag{4}$$

$$\text{FrameTok} : \mathbb{R}^{M\times(P_h P_w)\times C} \to \mathbb{R}^{M\times C}. \tag{5}$$

Local tokens. For a padded frame feature map $F_{b,s} \in \mathbb{R}^{C\times H'\times W'}$, we define

$$L_{b,s} \coloneqq \text{WinTok}_{(P_h,P_w)}\big(F_{b,s}\big) \in \mathbb{R}^{M\times(P_h P_w)\times C}.$$

Local summary tokens. Per-window pooled descriptors are given by

$$S^{\text{loc}}_{b,s} \coloneqq \text{FrameTok}\big(L_{b,s}\big) \in \mathbb{R}^{M\times C}.$$

Elementwise, for window index m, we perform average pooling over the window dimension:

$$S^{\text{loc}}_{b,s}[m] = \frac{1}{P_h P_w} \sum_{p=1}^{P_h P_w} L_{b,s}[m, p, :]. \tag{6}$$

Stacking batch and frames yields $L \in \mathbb{R}^{(BS)M\times(P_h P_w)\times C}$ and $S^{\text{loc}} \in \mathbb{R}^{(BS)\times M\times C}$.

c) Three-level attention in AHA: Each AHA block alternates three attentions, as shown in Figure 4. Here, self-attention refers to Multi-Head Self-Attention (MHSA) [5]:

1) **Window attention (local).** Self-attention is applied *within* each window independently. For $m \in \{1, \dots, M\}$,

$$\text{Attn}_{\text{win}} : L_{b,s}[m] \in \mathbb{R}^{(P_h P_w)\times C} \mapsto \tilde{L}_{b,s}[m] \in \mathbb{R}^{(P_h P_w)\times C}, \tag{7}$$

with relative positional bias inside the window.

2) **Frame attention (per-frame summaries).** Self-attention over summary tokens *within* a single frame:

$$\text{Attn}_{\text{frame}} : S^{\text{loc}}_{b,s} \in \mathbb{R}^{M\times C} \mapsto \hat{S}_{b,s} \in \mathbb{R}^{M\times C}. \tag{8}$$

A learnable frame/camera embedding $\mathbf{e}_s$ is added to encode view/time identity.

3) **Global attention (multi-frame summaries).** Self-attention over all summary tokens across S frames:

$$\text{Attn}_{\text{global}} : \text{concat}_s\big(\hat{S}_{b,s}\big) \in \mathbb{R}^{(SM)\times C} \mapsto \bar{S}_b \in \mathbb{R}^{(SM)\times C}. \tag{9}$$

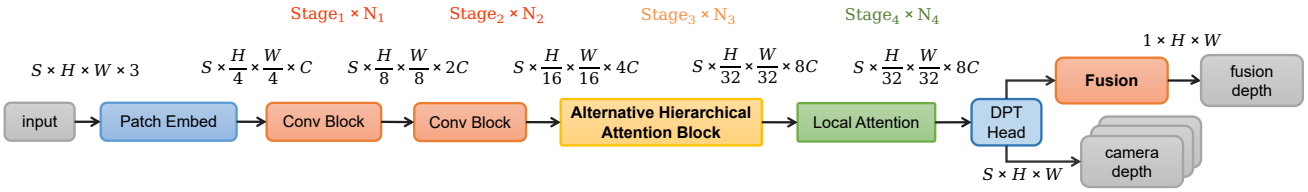

Fig. 3: FastViDAR architecture overview. Stages 1-2: convolutional stem for pyramidal feature extraction. Stage 3: AHA blocks that alternate window self-attention on *local tokens* with frame- and global-level self-attention on pooled *summary tokens*. Stage 4: local refinement via stacked window self-attention.

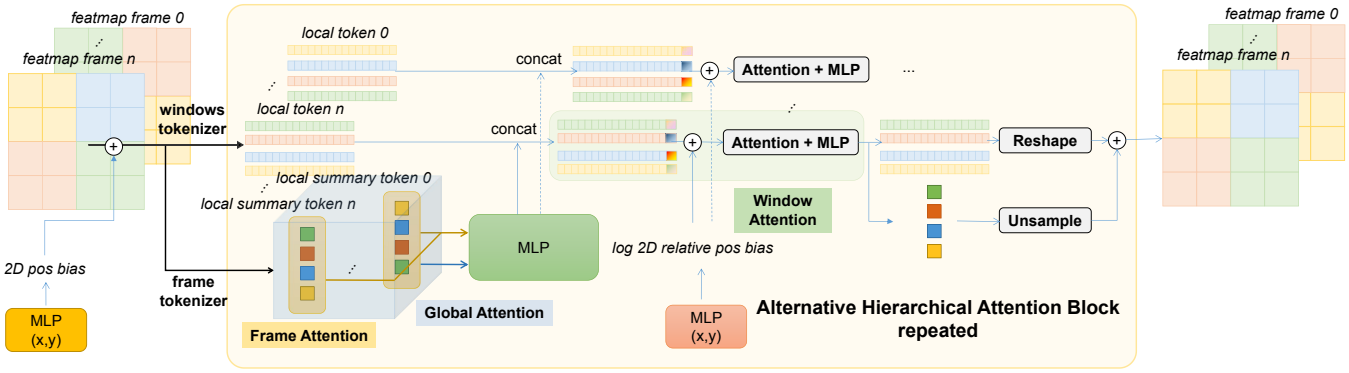

Fig. 4: AHA block. Window self-attention over local tokens, frame-level self-attention over summary tokens and global self-attention over all summaries.

This fuses cross-view or temporal context that helps resolve scale ambiguity and improves geometric consistency.

d) Local refinement via stacked window self-attention: Stage 4 is a *purely local* refinement: we apply N_4 layers of window MHSA on the same (P_h, P_w) partition. For each (b, s, m), let $X^{(0)} = L_{b,s}[m] \in \mathbb{R}^{(P_h P_w) \times C}$. Each layer uses pre-norm MHSA with a 1-layer MLP:

$$X^{(r)} = X^{(r-1)} + \mathrm{MHSA}_{\mathrm{win}}\big(\mathrm{LN}(X^{(r-1)});\, B_{\mathrm{rel}}\big),$$
$$X^{(r)} \leftarrow X^{(r)} + \mathrm{MLP}\big(\mathrm{LN}(X^{(r)})\big), \quad r = 1, \ldots, N_4, \quad (10)$$

where B_{rel} is the within-window relative positional bias. Stacking all windows yields $L^{\star}_{b,s} \in \mathbb{R}^{M \times (P_h P_w) \times C}$.

e) Complexity: Let $N = H'W'$ be the number of dense tokens per frame (after padding), S the number of frames, and $P = P_h P_w$ the window size in tokens (so $M = N/P$ windows per frame). We report leading-order costs *per head* and the size of the attention matrix (which dominates activation memory), ignoring linear projections/MLPs.

VGGT (full attention over all dense tokens). Sequence length $L_{\mathrm{full}} = SN$:

$$\text{Compute} = \mathcal{O}\big((SN)^2\big), \qquad \text{Memory} = \mathcal{O}\big((SN)^2\big). \quad (11)$$

AHA (ours). Sequence lengths: window P, per-frame summaries $M = N/P$, global summaries $SM = SN/P$.

$$\underbrace{\mathcal{O}\big(S\,M\,P^2\big)}_{\text{window attention}} + \underbrace{\mathcal{O}\big(S\,M^2\big)}_{\text{frame attention}} + \underbrace{\mathcal{O}\big((SM)^2\big)}_{\text{global attention}}$$
$$= \mathcal{O}\big(SNP + (SN/P)^2\big) \quad (12)$$

with the same orders for activation memory (attention matrices of sizes SNP, $S\,M^2$, and $(SM)^2$, respectively).

Comparison (per block, per head).

$$\frac{\mathrm{AHA}}{\mathrm{Full}} = \frac{SNP + (SN/P)^2}{(SN)^2} = \frac{P}{SN} + \frac{1}{P^2}, \quad (13)$$

For 7×7 windows (P=49), 640×320 input ($N \approx 200$), and S=4, the ratio becomes $\frac{\mathrm{AHA}}{\mathrm{Full}} \approx \frac{49}{4 \cdot 200} + \frac{1}{49^2} \approx 0.0604$, i.e., a $\sim$16× reduction.

As SN grows, the linear term $\frac{P}{SN}$ vanishes and the *ratio* saturates at $\lim_{SN \to \infty} \frac{\mathrm{AHA}}{\mathrm{Full}} = \frac{1}{P^2}$, so the maximal theoretical speedup is P^2 (for P=49, $\sim$2401×). Equivalently, when $SN \gg P^3$, AHA's *absolute* cost is dominated by its grouped quadratic term $(SN/P)^2$, but the *relative* complexity no longer decreases and remains near $1/P^2$.

C. ERP fusion

While our AHA mechanism leverages efficient global attention to achieve good cross-frame/view consistency, we can further enhance depth accuracy through multi-view depth fusion. Traditional fusion methods often struggle with scale drift and misalignment issues common in monocular and multi-view depth estimation [20], [21]. We propose an ERP fusion approach that projects multi-view depth to a shared ERP coordinate system.

a) ERP→3D and pose merging (reference): **Notation.** Let the ERP grid be $W \times H$ with pixel indices $(x, y) \in \{0, \ldots, W-1\} \times \{0, \ldots, H-1\}$. For frame $s \in \{1, \ldots, S\}$, $D_s(x, y)$ is the per-pixel depth, $\mathbf{d}(x, y) \in \mathbb{S}^2$ the unit ray recovered from Sec. III-A, and $(\mathbf{R}_s, \mathbf{t}_s)$ the extrinsics to a common rig/world frame. We form 3D points

$$\mathbf{p}_s(x, y) \;=\; \mathbf{R}_s\big(D_s(x, y)\,\mathbf{d}(x, y)\big) + \mathbf{t}_s,$$

yielding per-frame point sets in a shared coordinate system. This step assumes accurate calibrated extrinsics; in our setup, the per-camera average reprojection error is below 0.5 pixels.

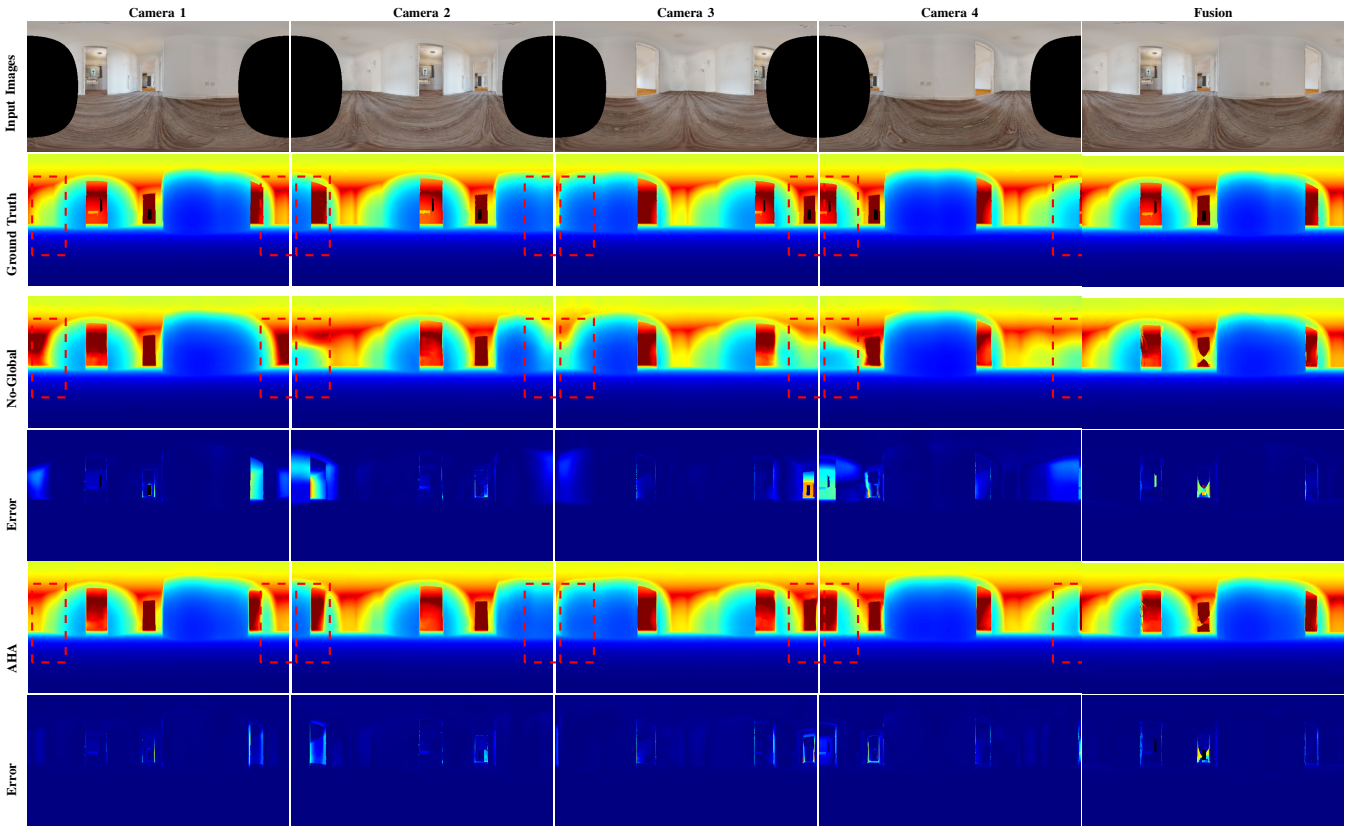

Fig. 5: Qualitative visualization of AHA vs No-Global attention comparison. The figure demonstrates the performance difference between AHA (with global attention) and No-Global attention across different camera views and their fusion results. Main differences are highlighted with red dashed rectangles.

b) Distance-aware splatting on ERP: We reproject points $\{\mathbf{p}_s\}$ to the ERP lattice (u, v) and splat each sample with a distance-adaptive kernel (near points use larger kernels):

$$r(d) = \mathrm{clip}\Big(\lfloor r_0 \tfrac{d_0}{d} + 0.5 \rfloor, 0, 3\Big), \qquad k(d) = 2r(d) + 1, \tag{14}$$

with $d_0 = 2.0$ and $r_0 = 0.5$. We use `amin` for depth (z-buffer), sum/count for color averaging, and then apply light hole filling. This distance-dependent footprint is inspired by the footprint control in 3D Gaussian Splatting [22], adapted to the spherical (ERP) domain. Let $M_s(u, v) \in \{0, 1\}$ denote the per-frame FOV mask on ERP.

c) Multi-frame ERP fusion (mean): On the same ERP grid,

$$D_{\text{fuse}}(u, v) = \frac{\sum_{s=1}^{S} M_s(u, v)\, D_s(u, v)}{\max\big(1, \sum_s M_s(u, v)\big)}, \tag{15}$$

$$C_{\text{fuse}}(u, v) = \frac{\sum_{s=1}^{S} M_s(u, v)\, C_s(u, v)}{\max\big(1, \sum_s M_s(u, v)\big)}. \tag{16}$$

Empirically, predictions extend slightly beyond nominal FOVs, so masked means provide complementary coverage and reduce inter-view variance. Nearest/confidence-weighted variants are reported in Table II.

D. ERP-weighted loss function

We supervise depth with a *masked, ERP-area-weighted* data term and a *multi-scale gradient* term; both operate per frame and are averaged across S frames. Let $M_s \in \{0, 1\}^{H \times W}$ be the validity mask, $\hat{D}_s$ the prediction, D_s the target, and (optionally) $C_s \geq 0$ a per-pixel confidence (from the network or set to $\mathbf{1}$). To compensate ERP latitude distortion, rows are weighted by

$$w(v) = \cos \phi(v), \qquad \phi(v) = \pi\big((v + 0.5)/H - 1/2\big).$$

a) Data term: We use a robust residual with (optional) confidence:

$$\mathcal{L}_{\text{data}}^{(s)} = \sum_{u,v} w(v)\; M_s(u, v)\, C_s(u, v) \times \rho\big(\hat{D}_s(u, v) - D_s(u, v)\big), \tag{17}$$

TABLE I: Ablation A - Hierarchy. Only global attention on summary tokens is toggled. (w: window, f: frame-level, g: global-level) Best results are highlighted in **bold**.

Variant	AbsRel↓	RMSE↓	Log10↓	$\delta < 1.25 \uparrow$	Time (ms)
No-Global (w+f)	0.135	0.454	0.181	0.892	34
AHA (w+f+g)	**0.111**	**0.384**	**0.163**	**0.904**	**36**

TABLE II: Fusion strategies comparison. Best results are highlighted in **bold**.

Method	AbsRel↓	RMSE↓	Log10↓	$\delta < 1.25 \uparrow$
No-fusion	0.109	0.369	0.149	0.897
Nearest	0.113	0.384	0.153	0.898
Weighted	**0.108**	0.365	**0.146**	**0.901**
Mean	**0.108**	**0.364**	**0.146**	**0.901**

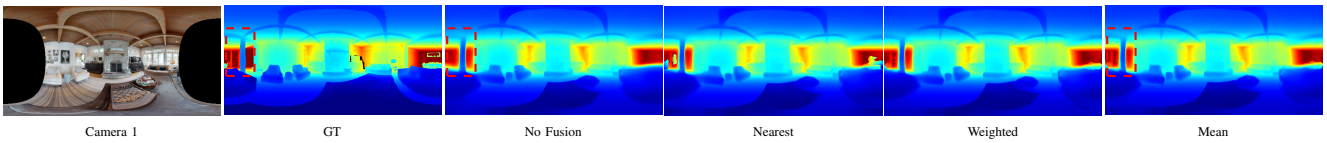

Fig. 6: Fusion strategy comparison showing different fusion methods. Main differences are highlighted with red dashed rectangles.

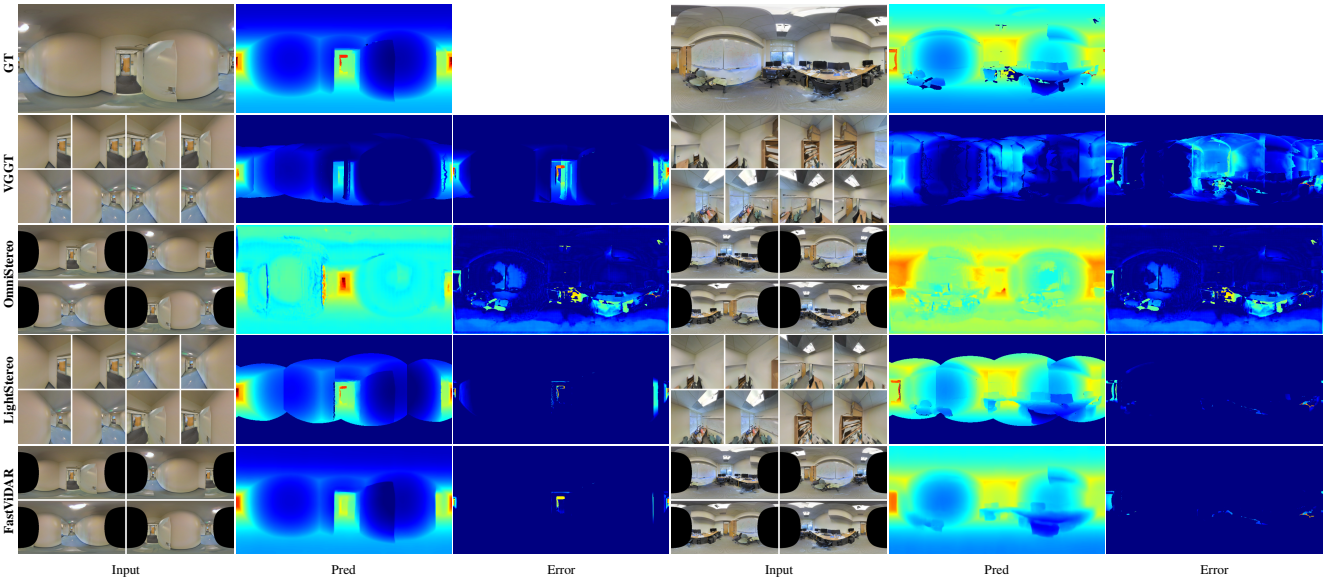

Fig. 7: Comprehensive comparison of FastViDAR with baselines on 2D-3D-S datasets. First row shows the ground truth depth map. Subsequent rows present predictions and error maps for each method. In error maps, regions with $\delta < 1.25$ are shown in pure dark blue.

where $\rho(\cdot)$ is the Huber loss (we adopt $\delta=1$ unless noted).

b) Multi-scale gradient term: At scales $r = 0, \ldots, R-1$ (subsample by 2^r in u and v), we match ERP gradients with the same area weights:

$$\mathcal{L}_{\nabla}^{(s)} = \frac{1}{R}\sum_{r=0}^{R-1}\sum_{u,v} w_r(v)\, M_s^{(r)}(u,v)\, C_s^{(r)}(u,v) \times \rho\big(\nabla \hat{D}_s^{(r)}(u,v) - \nabla D_s^{(r)}(u,v)\big), \tag{18}$$

with $w_r(v) = \cos\phi_r(v)$ defined analogously at height $H/2^r$, and ∇ the finite-difference operator.

c) Depth objective and optional regularizer: Our per-frame depth loss is

$$\mathcal{L}_{\text{depth}} = \frac{1}{S}\sum_{s=1}^{S}\Big(\mathcal{L}_{\text{data}}^{(s)} + \lambda_{\nabla}\,\mathcal{L}_{\nabla}^{(s)}\Big). \tag{19}$$

In implementation, we set $\lambda_{\nabla} = 1$, $R = 4$, $\delta = 1$, and use confidence regularization parameters $\gamma = 1, \alpha = 0.2$. The ERP weighting $w(v)$ mirrors our implementation (row-wise $\cos\phi$) and ensures equal-solid-angle supervision; it consistently improves stability by preventing the poles—small area but dense pixels—from dominating the loss.

IV. Experiments

We evaluate FastViDAR with the AHA backbone (Sec. III-B), ERP fusion (Sec. III-C), and the ERP-weighted loss (Sec. III-D). Unless noted, inputs are ERP 640×320, frame count $S=4$, window $(P_h, P_w)=(7,7)$, and Stage-4 uses $N_4=2$ window-MHSA layers. Our experiments use four cameras where each captures one frame, and the network processes all 4 frames simultaneously. We train the model from scratch using the AdamW optimizer [23] with learning rate 1×10^{-4}, batch size 20, for 20 epochs. We employ OneCycleLR scheduler [24] with 10% warm-up and cosine annealing strategy. For inference time comparison, we test on RTX 4090 with the following input resolutions: LightStereo and VGGT at 512×512, OmniStereo and FastViDAR at 640×320.

TABLE III: Performance and efficiency comparison on 2D-3D-S (zero-shot) dataset. Best results are shown in **bold**.

Method	AbsRel↓	RMSE↓	Log10↓	$\delta < 1.25$ ↑	Time (ms)
VGGT	0.557	1.934	0.396	0.043	120
OmniStereo	0.619	1.450	0.154	0.554	66
LightStereo	0.125	0.667	0.050	0.851	**33**
FastViDAR	**0.119**	**0.433**	**0.046**	**0.929**	36

A. Datasets and Evaluation Protocol

HM3D (train/ablate). We render multi-view ERP from 800/200 train/test scenes of HM3D [7]. Each sample uses a 4-camera rigid rig with *randomized* relative poses and FOV in $[160°, 360°]$, yielding diverse baselines and overlaps. The dataset contains 421,127 training groups and 52,484 test groups, where each group consists of 4 ERP views.

2D-3D-S (zero-shot). We render a *fixed* rigid ring of 4 fisheye cameras (FOV 220°) with baseline $20\sqrt{2}$ mm and

90° angular separation between adjacent cameras from 6 large scenes, totaling 6,000 groups. Each camera captures one frame, and the network processes all 4 frames simultaneously. No fine-tuning is performed on 2D-3D-S [8].

Training data. For ablation experiments, our method is trained only on images collected from the 800 training scenes of HM3D. For zero-shot evaluation on 2D-3D-S, our method is trained on the complete HM3D dataset with 1,000 scenes plus 200 publicly available Blend scenes (excluding 3D models from 2D-3D-S), with no additional data used.

Preprocessing & evaluation domain. All fisheye views are mapped to a common ERP lattice (Sec. III-A). All metrics are computed on the ERP grid using the *intersection* validity mask $M(u,v) = M_{\text{gt}}(u,v) \wedge M_{\text{meth}}(u,v)$ (i.e., GT-valid $\cap$ method-available FOV). For fairness, all depth predictions are considered regardless of confidence values. All multi-view depth predictions are transformed to the first view's coordinate system before evaluation.

B. Metrics

All scores are computed on the ERP grid under the intersection mask M defined above. Let $Z = \sum_{u,v} M(u,v)$ and define the masked mean

$$\mathbb{E}_M[f] \triangleq Z^{-1} \sum_{u,v} M(u,v)\, f(u,v). \tag{20}$$

Given ground-truth depth D and prediction $\hat{D}$ (in meters), set $\Delta = \hat{D} - D$ and fix $\varepsilon > 0$.

a) Unified error functionals (lower is better): For a transform $g : \mathbb{R}_+ \to \mathbb{R}$ and $p \geq 1$,

$$\mathcal{L}_p(g) \triangleq \Big(\mathbb{E}_M\big[\, |g(\hat{D}) - g(D)|^p \,\big] \Big)^{1/p}, \tag{21}$$

$$\mathcal{R}_q \triangleq \mathbb{E}_M\Big[\big(\tfrac{|\Delta|}{D+\varepsilon} \big)^q \Big], \quad q \geq 1. \tag{22}$$

For threshold accuracy (higher is better), let $\tau = \max(\hat{D}/D,\ D/\hat{D})$ and define

$$\mathrm{A}(\alpha) = \mathbb{E}_M\big[\mathbf{1}(\tau < \alpha)\big]. \tag{23}$$

Reported metrics. We report: **AbsRel**$= \mathcal{R}_1$, **RMSE**$= \mathcal{L}_2$ with $g(x) = x$, **Log10**$= \mathcal{L}_1(\log_{10})$, and $\boldsymbol{\delta < 1.25} = \mathrm{A}(1.25)$.

b) Protocol: Scores are computed per scene (the same M for all methods within a scene) and averaged over the test split. Unless stated, $\varepsilon = 10^{-6}$. RMSE is reported in meters; other metrics are dimensionless.

C. Ablation Studies and Analysis

We conduct ablation studies focusing on our two key contributions—*AHA* and *ERP fusion*—under identical training and evaluation settings.

a) AHA vs No-Global: We disable global attention on summary tokens (keeping window and per-frame attention). Capacity, loss, and training schedule remain unchanged. Table I shows that AHA improves accuracy metrics while maintaining efficiency, confirming that summary-level global reasoning improves cross-frame consistency. Qualitative results are shown in Figure 5. AHA demonstrates improved performance with better scale accuracy and consistency, particularly in regions outside each camera's FOV, providing more accurate depth predictions and richer details compared to the No-Global attention baseline.

b) Fusion strategy: We compare no-fusion (per-frame), mean (ours), nearest, and confidence-weighted fusion strategies. Table II demonstrates that mean fusion consistently achieves the best performance, providing stable and accurate depth predictions across different scenarios. Qualitative comparison results are shown in Figure 6. When two views strongly disagree due to occlusion or specular surfaces, mean fusion may produce intermediate artifacts; however, AHA's cross-view consistency mitigates such cases, making mean fusion more robust than alternatives across our benchmarks.

D. Zero-Shot Comparison with State-of-the-Art Methods

We evaluate FastViDAR against state-of-the-art baselines: VGGT (MVS), OmniStereo (omnidirectional depth), and LightStereo-M (real-time stereo). VGGT and OmniStereo use pre-trained weights without adaptation, while LightStereo-M is fine-tuned on our training data. LightStereo-M has been extensively trained on over 1.5 million samples from FoundationStereo [25] and other datasets [26]–[40]. For LightStereo, we split ERP into stereo pairs and convert disparity to depth. For VGGT, we split 4 ERP images into 8 directional pinhole views (100° FOV) and project output point clouds to ERP depth maps. OmniStereo and FastViDAR use the original 4 ERP images (220° FOV).

Table III presents zero-shot results on the 2D-3D-S dataset. FastViDAR achieves competitive performance across all metrics without fine-tuning, demonstrating strong cross-domain generalization. Qualitative results are shown in Figure 7. Although slightly slower than LightStereo, our method provides comprehensive 360° depth coverage with better multi-view consistency and accuracy. We also note several practical limitations: (1) in extreme near-field scenes (object distance < 0.1 m occupying more than half of a view), co-visible regions become too small, potentially causing scale inaccuracies; (2) although our method implicitly learns cross-camera correlations and can predict depth even beyond each camera's FOV, the fusion pipeline assumes accurate fixed extrinsics, and performance may degrade when extrinsic errors are large (e.g., average reprojection error > 1.0 pixels); (3) the central-camera ray assumption may introduce geometric bias for non-central cameras, especially at close range. In future work, we plan to address these limitations and incorporate online pose/extrinsic refinement.

V. Conclusion

We propose FastViDAR, a real-time omnidirectional multi-view depth estimation method featuring AHA and ERP fusion. Despite being trained on only 1,200 scenes, our method shows competitive results compared to OmniStereo, VGGT, and LightStereo (trained on 1.5M+ samples). AHA accommodates arbitrary camera configurations with improved cross-view consistency, while ERP fusion further refines depth in overlap regions with calibrated extrinsics.

REFERENCES

[1] X. Li *et al.*, "Cascade omnidirectional depth estimation with dynamic spherical sweeping," *Applied Sciences*, vol. 14, no. 2, p. 5173, 2024.

[2] C. Won, J. Ryu, and J. Lim, "Sweepnet: Wide-baseline omnidirectional depth estimation," in *Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019, p. 6073–6079.

[3] Y. Guo *et al.*, "Depth any camera: Zero-shot metric depth estimation from any camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[4] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[6] A. Hatamizadeh, G. Heinrich, H. Yin, A. Tao, J. M. Alvarez, J. Kautz, and P. Molchanov, "Fastervit: Fast vision transformers with hierarchical attention," 2024.

[7] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," in *Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[8] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," 2017.

[9] Y. Xie *et al.*, "Omnividar: Omnidirectional depth estimation from multi-fisheye images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[10] J. Deng *et al.*, "Omnistereo: Real-time omnidirectional depth estimation with multiview fisheye cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[11] B. Berenguel-Baeta *et al.*, "Fredsnet: Joint monocular depth and semantic segmentation with fast fourier convolutions from single panoramas," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[12] J. Zhang *et al.*, "Lightstereo: Channel boost is all you need for efficient 2d cost aggregation," 2024.

[13] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *ECCV*, 2018.

[14] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent multi-view stereo with sequential consistency," in *ICCV*, 2019, commonly referred to as CasMVSNet.

[15] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[16] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 5695–5701.

[17] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2018, pp. 552–560.

[18] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *Computer Vision – ECCV 2000, Lecture Notes in Computer Science*, vol. 1843. Springer, 2000, pp. 445–461.

[19] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," Mar. 2021.

[20] C. Wang, J. M. Buenaposada, Z. Rui, and S. Lucey, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 35–45.

[21] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1281–1290.

[22] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023.

[23] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[24] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," 2018.

[25] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," 2025.

[26] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, and S. Liu, "Practical stereo matching via cascaded recurrent network with adaptive correlation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[29] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[30] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," 2020.

[31] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2017.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[33] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] H. Hirschmüller and D. Scharstein, "Middle-resolution stereo datasets (2005 - 2006)," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[35] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2014.

[36] G. Pan, T. Sun, T. Weed, and D. Scharstein, "2021 stereo datasets," 2021.

[37] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision (IJCV)*, vol. 47, no. 1-3, pp. 7–42, 2002.

[38] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] W. Bao, W. Wang, Y. Xu, Y. Guo, S. Hong, and X. Zhang, "Instereo2k: A large real dataset for stereo matching in indoor scenes," *Science China Information Sciences*, vol. 63, no. 11, pp. 1–11, 2020.

[40] P. Z. Ramirez, F. Tosi, M. Poggi, S. Salti, S. Mattoccia, and L. Di Stefano, "Open challenges in deep stereo: the booster dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.