The Midas Touch for Metric Depth
— Supplementary Material —

A. Ethics

In this research, we utilize multiple datasets for depth estimation and completion, including nuScenes [5], DDAD [16], Make3D [33], DIODE [42], ETH3D [34], ScanNet [8], VOID [48], SUN-RGBD [37], HAMMER [19], IBims-1 [22], KITTI [14], and NYU-Depth V2 [35]. These datasets are used in strict accordance with their respective terms of use. While some datasets may contain images with visible faces and other personal data collected without consent, we emphasize that no processing of biometric information has occurred. We utilize the images under the CC-BY license or in a manner compatible with the Data Analysis Permission.

B. Details of Methodology

B.1. Details on the Per-Segment Calibration

As discussed in Sect. 3.1.1 of the main paper, we convert each 3D seed prior to a scalar cue ξ_i^j that is equivalent to inverse depth. Concretely: (i) When metric depth z_i^j is available (e.g., LiDAR or metric depth estimators), we set:

$$\xi_i^j = \frac{\kappa}{z_s(x) + \varepsilon}, \tag{1}$$

where κ is the minimum depth (note that κ can be set to any constant, as long as the same value of κ is used during recovery), and ε is a small constant for stability. (ii) For rectified stereo, disparity is proportional to z^{-1} (up to the baseline–focal-length scale). (iii) For generic multi-view stereo or unrectified stereo (*i.e.*, without epipolar rectification), we perform correspondence matching and recover depth via geometric triangulation from calibrated camera poses, then we follow (i).

For $i \in \mathcal{Q}$, the calibration function g_i models the per-segment transition and, in practice, can be simplified as:

$$g_i(x) = \max\{a_i x + b_i,\ d_{\min}\}, \qquad x \in \mathbb{R}_{\geq 0}, \tag{2}$$

where (a_i, b_i) is parameters, and d_{min} denotes a prescribed minimum depth. g_i can be obtained via least squares, moment matching, or quantile matching. Concretely: (i) For the least-squares situation, we adopt:

$$(a_i, b_i) \ = \ \arg\min_{a,b} \sum_{x \in \mathcal{S}_i \cap \Omega_s} \big(a\, d(x) + b - \xi(x)\big)^2, \tag{3}$$

where Ω_s denotes the set of projected seed pixels; (ii) for the moment matching situation, we set:

$$a = \frac{\sigma_\xi}{\sigma_d}, \qquad b = \mu_\xi - a\,\mu_d, \tag{4}$$

where σ and μ denote the sample standard deviation and the sample mean, respectively. For a special case in mean scaling, we set:

$$a_i = \frac{\mu_\xi}{\mu_d}, \qquad b_i = 0; \tag{5}$$

(iii) For the quantile matching situation, we utilize the empirical quantiles Q_p and the interquartile range $\mathrm{IQR}(x) = Q_{0.75}(x) - Q_{0.25}(x)$:

$$a_i = \frac{\mathrm{IQR}(\xi)}{\mathrm{IQR}(d)}, \qquad b_i = \mathrm{median}(\xi) - a_i\,\mathrm{median}(d), \tag{6}$$

where $\mathrm{median}(\cdot) = Q_{0.5}(\cdot)$ denotes the median operation. For a special case in median scaling, we set:

$$a_i = \frac{\mathrm{median}(\xi)}{\mathrm{median}(d)}, \qquad b_i = 0. \tag{7}$$

B.2. Details on the Segment Propagation and Sparse Graph Optimization

Random Noise Suppression. For $\{\mathcal{S}_i\}_{i \notin \mathcal{Q}}$, which contain no 3D seeds, the above recovery procedure cannot be applied. Moreover, when applying g_i for the calibration process on $\{\mathcal{S}_i\}_{i \in \mathcal{Q}}$, random noise becomes pronounced if the set of 3D seed priors $\mathcal{X}_i$ is extremely small. To simultaneously suppress random noise, we first construct a transfer map $\boldsymbol{T}$ from disparity d_i to depth z_i using data from $\{\mathcal{S}_i\}_{i \in \mathcal{Q}}$. For regions $\{\mathcal{S}_i\}_{i \notin \mathcal{Q}}$, the values in $\boldsymbol{T}$ are initialized as invalid ($\varnothing$).

Assume $\boldsymbol{p}$ is a 2D pixel in transformation map $\boldsymbol{T}$, and $\mathcal{N}_{\boldsymbol{p}}$ denotes the set of neighbors of $\boldsymbol{p}$ (including $\boldsymbol{p}$ itself) restricted to positions where $\boldsymbol{T}$ is defined (*i.e.*, non-$\varnothing$). $\boldsymbol{T}^{(t)}$ denotes the map at iteration t of the bilateral filter. For any point $\boldsymbol{p}$, its probability of occurrence at iteration t, denoted as $\mathbb{P}^{(t)}(\boldsymbol{p})$, can be interpreted as the potential energy of point $\boldsymbol{p}$ under the function $\boldsymbol{T}^{(t)}$, i.e., $\mathbb{P}^{(t)}(\boldsymbol{p}) \propto \exp(-\boldsymbol{T}^{(t)}(\boldsymbol{p}))$. Since the state of $\boldsymbol{p}$ is influenced by its local neighborhood $\mathcal{N}_{\boldsymbol{p}}$, the law of total probability yields:

$$\mathbb{P}^{(t)}(\boldsymbol{p} \mid \mathcal{N}_{\boldsymbol{p}}) = \sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} \mathbb{P}(\boldsymbol{p} \mid \boldsymbol{q}, \mathcal{N}_{\boldsymbol{p}})\, \mathbb{P}^{(t-1)}(\boldsymbol{q} \mid \mathcal{N}_{\boldsymbol{p}}), \tag{8}$$

where $\mathbb{P}(\boldsymbol{p} \mid \boldsymbol{q})$ denotes the conditional probability of transitioning from a neighboring node $\boldsymbol{q}$ to the central node $\boldsymbol{p}$, and $\mathbb{P}^{(t-1)}(\boldsymbol{q})$ represents the prior probability of $\boldsymbol{q}$ from the previous iteration. For each $\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}$, the bilateral kernel weight $k(\boldsymbol{q})$ is defined as:

$$k(\boldsymbol{q}) = \exp\left(-\frac{\|\boldsymbol{p}-\boldsymbol{q}\|^2}{2\sigma_1^2} - \frac{\left(\boldsymbol{I}(\boldsymbol{p})-\boldsymbol{I}(\boldsymbol{q})\right)^2}{2\sigma_2^2}\right), \quad (9)$$

where $\|\cdot\|$ is the Euclidean norm, σ_1 controls the spatial falloff, and σ_2 controls the color similarity. Given that the bilateral kernel $k(\boldsymbol{q})$ defines the influence weight of neighbor $\boldsymbol{q}$ on the center point $\boldsymbol{p}$, we can apply Bayes' theorem to express:

$$\mathbb{P}(\boldsymbol{p} \mid \boldsymbol{q}, \mathcal{N}_{\boldsymbol{p}}) = \frac{\mathbb{P}(\boldsymbol{q} \mid \boldsymbol{p}, \mathcal{N}_{\boldsymbol{p}})\,\mathbb{P}(\boldsymbol{p} \mid \mathcal{N}_{\boldsymbol{p}})}{\mathbb{P}(\boldsymbol{q} \mid \mathcal{N}_{\boldsymbol{p}})} \propto k(\boldsymbol{q}). \quad (10)$$

Substituting this into the (8) yields:

$$\mathbb{P}^{(t)}(\boldsymbol{p} \mid \mathcal{N}_{\boldsymbol{p}}) = \frac{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} k(\boldsymbol{q})\,\mathbb{P}^{(t-1)}(\boldsymbol{q} \mid \mathcal{N}_{\boldsymbol{p}})}{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} k(\boldsymbol{q})}. \quad (11)$$

Taking the logarithm of both sides leads to the filtering function:

$$\boldsymbol{T}^{(t)}(\boldsymbol{p}) = -\log \frac{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} k(\boldsymbol{q})\,\exp\left(-\boldsymbol{T}^{(t-1)}(\boldsymbol{q})\right)}{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} k(\boldsymbol{q})}. \quad (12)$$

From a geometric perspective, this filtering function effectively suppresses outliers of high-value errors through exponential weighting, thereby ensuring the stability and reliability of the transformation process.

Sparse Graph Optimization. In Sect. 3.1.2 of the main paper, we introduce a sparse graph optimization. We first model the $\{\mathcal{S}_i\}$ into a graph $\mathcal{G}$. The vertex of $\mathcal{G}$ stores the calibration parameters, and the edge of $\mathcal{G}$ is the spatial distance of the center in $\{\mathcal{S}_i\}$. We then sparsify the graph by sorting the edge weights of $\mathcal{G}$. We preserve the N-minimum distance of each vertex. Let $\boldsymbol{c}_i$ denote the centroid of $\boldsymbol{S}_i$. The w_{ij} in Sect. 3.1.2 can be computed as follows: $w_{ij} \propto \exp\left(-\|\boldsymbol{c}_i - \boldsymbol{c}_j\|/\tau\right)$, where $\|\cdot\|$ is the Euclidean norm, and τ is the adaptive, median-based scale parameter for numerical stability.

B.3. Proof of Proposition 1 in the Main Paper

Diagonal case. Let $\boldsymbol{p} = (u_0, v_0)^\top$ and $\boldsymbol{q} = (u_1, v_1)^\top$. Consider the two axis-aligned polyline paths between $\boldsymbol{p}$ and $\boldsymbol{q}$. Let $\mathcal{L}_1$ be the axis-aligned path $\boldsymbol{p} \to (u_1, v_0)^\top \to \boldsymbol{q}$ (horizontal then vertical), and $\mathcal{L}_2$ the path $\boldsymbol{p} \to (u_0, v_1)^\top \to \boldsymbol{q}$ (vertical then horizontal). Define the symmetric first-order remainder as follows:

$$R(\boldsymbol{p}, \boldsymbol{q}) := z(\boldsymbol{q}) - z(\boldsymbol{p}) - \tfrac{1}{2}\big(\nabla z(\boldsymbol{p}) + \nabla z(\boldsymbol{q})\big)^\top (\boldsymbol{q} - \boldsymbol{p}). \quad (13)$$

Our goal is to choose one path such that $|R(\boldsymbol{p}, \boldsymbol{q})|$ is bounded by a path integral $\int \phi\, ds$, where $\phi(u, v) = \sqrt{z_{uu}^2(u, v) + z_{vv}^2(u, v)}$ and $ds = \sqrt{du^2 + dv^2}$.

We first rewrite $R(\boldsymbol{p}, \boldsymbol{q})$ as the average of the remainders associated with the two candidate paths. Set $\Delta u = u_1 - u_0$, $\Delta v = v_1 - v_0$. For each one-dimensional axis-aligned path (horizontal paths at fixed v, vertical paths at fixed u), we apply the integral form of the trapezoidal rule error for a single variable and then integrate by parts along that path. This yields the following two equations:

$$\begin{aligned} R_{H\to V} &:= z(\boldsymbol{q}) - z(\boldsymbol{p}) \\ &\quad - \big(\Delta u\, z_u(\boldsymbol{p}) + \Delta v\, z_v(\boldsymbol{q})\big) \\ &= \int_{u_0}^{u_1} (u_1 - u)\, z_{uu}(u, v_0)\, du \\ &\quad - \int_{v_0}^{v_1} (v - v_0)\, z_{vv}(u_1, v)\, dv, \end{aligned} \quad (14)$$

$$\begin{aligned} R_{V\to H} &:= z(\boldsymbol{q}) - z(\boldsymbol{p}) \\ &\quad - \big(\Delta v\, z_v(\boldsymbol{p}) + \Delta u\, z_u(\boldsymbol{q})\big) \\ &= -\int_{u_0}^{u_1} (u - u_0)\, z_{uu}(u, v_1)\, du \\ &\quad + \int_{v_0}^{v_1} (v_1 - v)\, z_{vv}(u_0, v)\, dv. \end{aligned} \quad (15)$$

These two formulas can be proved using a one-dimensional line integral. By definition and a simple rearrangement, we obtain the following equation:

$$R(\boldsymbol{p}, \boldsymbol{q}) = \tfrac{1}{2}\, R_{H\to V} + \tfrac{1}{2}\, R_{V\to H}. \quad (16)$$

From (14) and the triangle inequality,

$$\begin{aligned} |R_{H\to V}| &\le \int_{u_0}^{u_1} (u_1 - u)\, |z_{uu}(u, v_0)|\, du \\ &\quad + \int_{v_0}^{v_1} (v - v_0)\, |z_{vv}(u_1, v)|\, dv. \end{aligned} \quad (17)$$

Since $0 \le (u_1 - u) \le |u_1 - u_0| = |\Delta u|$ and $0 \le (v - v_0) \le |\Delta v|$, we obtain the following equation:

$$\begin{aligned} |R_{H\to V}| &\le |\Delta u| \int_{u_0}^{u_1} |z_{uu}(u, v_0)|\, du \\ &\quad + |\Delta v| \int_{v_0}^{v_1} |z_{vv}(u_1, v)|\, dv. \end{aligned} \quad (18)$$

On the horizontal path $ds = |du|$ and $|z_{uu}| \le \phi$; on the vertical path $ds = |dv|$ and $|z_{vv}| \le \phi$. Therefore, we obtain the following equation:

$$|R_{H\to V}| \le \int_{\mathcal{L}_{H\to V}} \phi(u, v)\, ds. \quad (19)$$

From (15), we can obtain the following expression:

$$|R_{V\to H}| \;\le\; \int_{\mathcal{L}_{V\to H}} \phi(u,v)\, ds. \tag{20}$$

Take absolute values in (16) and apply (19)-(20):

$$\begin{aligned} |R| &\le \tfrac{1}{2}\int_{\mathcal{L}_{H\to V}} \phi\, ds \;+\; \tfrac{1}{2}\int_{\mathcal{L}_{V\to H}} \phi\, ds \\ &\le \max\{\int_{\mathcal{L}_{H\to V}} \phi\, ds,\; \int_{\mathcal{L}_{V\to H}} \phi\, ds\}. \end{aligned} \tag{21}$$

Hence there exists $\mathcal{L} \in \{\mathcal{L}_{H\to V}, \mathcal{L}_{V\to H}\}$ such that

$$\big|R(\boldsymbol{p},\boldsymbol{q})\big| \;\le\; \int_{\mathcal{L}} \phi(u,v)\, ds. \tag{22}$$

This proves the claimed pathwise existence bound (the Cauchy-Schwarz inequality can also be employed to obtain the same result).

Purely horizontal step. We derive the following equation:

$$R^{(\mathrm{hor})} = \int_{u_0}^{u_1} \left(\tfrac{u_0+u_1}{2} - u\right) z_{uu}(u,v_0)\, du. \tag{23}$$

We then derive the following equation:

$$\begin{aligned} \big|R^{(\mathrm{hor})}\big| &\le \int_{u_0}^{u_1} \left|\tfrac{u_0+u_1}{2} - u\right| |z_{uu}(u,v_0)|\, du \\ &\le \int_{u_0}^{u_1} |z_{uu}(u,v_0)|\, du \\ &\le \int_{\mathcal{L}_{\mathrm{hor}}} \phi\, ds, \end{aligned} \tag{24}$$

where $\mathcal{L}_{\mathrm{hor}}$ is the single horizontal axis-aligned path (so $ds = |du|$ and $|z_{uu}| \le \phi$ along it).

Purely vertical step. Similarly, we can derive the following equation:

$$R^{(\mathrm{ver})} = \int_{v_0}^{v_1} \left(\tfrac{v_0+v_1}{2} - v\right) z_{vv}(u_0,v)\, dv, \tag{25}$$

and

$$\big|R^{(\mathrm{ver})}\big| \le \int_{v_0}^{v_1} |z_{vv}(u_0,v)|\, dv \le \int_{\mathcal{L}_{\mathrm{ver}}} \phi\, ds, \tag{26}$$

where $\mathcal{L}_{\mathrm{ver}}$ is the single vertical axis-aligned path (so $ds = |dv|$ and $|z_{vv}| \le \phi$ along it).

B.4. Proof of Geodesic Cost

Define the weighted path length and the induced geodesic cost as follows:

$$L_\phi(\gamma) = \int_\gamma \phi\, ds, \qquad d_\phi(\boldsymbol{p},\boldsymbol{q}) = \inf_{\gamma:\,\boldsymbol{p}\to\boldsymbol{q}} L_\phi(\gamma), \tag{27}$$

where for a piecewise C^1 curve $\gamma : [0,1] \to \Omega$, we have $ds = \|\dot\gamma(t)\|_2\, dt$. Endow $\Omega \subset \mathbb{R}^2$ with the conformal Riemannian metric $g = \phi^2 \boldsymbol{I}_2$, where $\boldsymbol{I}_2$ is the identity matrix. Then the Riemannian length of γ is:

$$\begin{aligned} L_g(\gamma) &= \int_0^1 \sqrt{\dot\gamma(t)^\top g_{\gamma(t)}\, \dot\gamma(t)}\; dt \\ &= \int_0^1 \phi(\gamma(t))\, \|\dot\gamma(t)\|_2\, dt \\ &= \int_\gamma \phi\, ds. \end{aligned} \tag{28}$$

Hence,

$$L_\phi(\gamma) = L_g(\gamma) \quad\Rightarrow\quad d_\phi(\boldsymbol{p},\boldsymbol{q}) = d_g(\boldsymbol{p},\boldsymbol{q})\,. \tag{29}$$

The path length formulation exactly is the geodesic distance under the conformal metric $g = \phi^2 \boldsymbol{I}_2$ (identical values and the same minimizing curves up to reparameterization).

B.5. Details on the Dynamic Programming

To update the value at a point $\boldsymbol{p}$ using information from its neighboring point $\boldsymbol{q}$ in the dynamic programming process, we use the following expansion in Sect. 3.2.2:

$$z^{(k+1)}(\boldsymbol{p}) = \left(1 - \tfrac{1}{k+1}\right) z^{(k)}(\boldsymbol{p}) \;+\; \tfrac{1}{k+1}\, \hat z^{(k)}\,(\boldsymbol{p} \mid \boldsymbol{q}, \Delta\boldsymbol{p})\,. \tag{30}$$

Under a polynomial expansion, the above expression can be reformulated as the following equation:

$$\begin{aligned} z^{(k+1)}(\boldsymbol{p}) = \left(1 - \frac{1}{k+1}\right) z^{(k)}(\boldsymbol{p}) + \frac{1}{k+1}\Big(z^{(k)}(\boldsymbol{q}) \\ + \nabla z^{(k)}(\boldsymbol{q})^\top \Delta\boldsymbol{h} + \frac{1}{2}\Delta\boldsymbol{h}^\top \boldsymbol{H}_z^{(k)}(\boldsymbol{q})\Delta\boldsymbol{h} + \cdots\Big), \end{aligned} \tag{31}$$

where the displacement vector $\Delta\boldsymbol{h} = \boldsymbol{p} - \boldsymbol{q}$. $\nabla z^{(k)}(\boldsymbol{q})$ is the gradient of $z^{(k)}$ at point $\boldsymbol{q}$, and $\boldsymbol{H}_z^{(k)}(\boldsymbol{q})$ is the Hessian matrix of $z^{(k)}$ at $\boldsymbol{q}$, given by:

$$\boldsymbol{H}_z^{(k)}(\boldsymbol{q}) = \begin{bmatrix} z_{xx}^{(k)}(\boldsymbol{q}) & z_{xy}^{(k)}(\boldsymbol{q}) \\ z_{xy}^{(k)}(\boldsymbol{q}) & z_{yy}^{(k)}(\boldsymbol{q}) \end{bmatrix}, \tag{32}$$

where z_{xx}, z_{yy}, and z_{xy} denote the second partial derivatives of z with respect to x, y, and the mixed partial derivative with respect to x and y, respectively.

For (31), we use the step sizes $\frac{1}{k+1}$, which satisfy the following conditions:

$$\sum_{k=1}^{\infty} \frac{1}{k} = \infty, \quad \sum_{k=1}^{\infty} \left(\frac{1}{k}\right)^2 < \infty. \tag{33}$$

Intuitively, the first condition prevents the updates from vanishing too quickly, so new information continues to influence the estimate; the second mitigates the cumulative

effect of noise. We now make the convergence analysis fully two-dimensional by treating the Taylor remainder $R_2^{(k)}(\boldsymbol{p}, \boldsymbol{q})$ rigorously. Assume that each iterate $z^{(k)} : \mathbb{R}^2 \to \mathbb{R}$ is three-times continuously differentiable on an open set containing the line path between $\boldsymbol{p}$ and $\boldsymbol{q}$, and that the third-derivative tensors are uniformly bounded along these paths. Specifically, there exists $M > 0$ such that

$$\sup_{\boldsymbol{\xi}} \left\| \nabla^3 z^{(k)}(\boldsymbol{\xi}) \right\|_{\text{op}} \leq M, \quad \forall\, k. \tag{34}$$

Here, $\| \cdot \|_{\text{op}}$ denotes the operator norm. For each iteration, the second-order Taylor remainder can be written in the form:

$$\epsilon_{k+1} := R_2^{(k)}(\boldsymbol{p}, \boldsymbol{q}) = \tfrac{1}{6} \nabla^3 z^{(k)}(\boldsymbol{\xi}_{k+1}) \big[\Delta \boldsymbol{h}, \Delta \boldsymbol{h}, \Delta \boldsymbol{h}\big], \tag{35}$$

where $\boldsymbol{\xi}_{k+1}$ is in the line path between $\boldsymbol{p}$ and $\boldsymbol{q}$. This implies the bound:

$$|\epsilon_{k+1}| \leq \tfrac{M}{6} \|\Delta \boldsymbol{h}\|^3. \tag{36}$$

Therefore, the step size $\Delta \boldsymbol{h}$ should be controlled.

B.6. Details on the Knowledge Distillation

We leverage high-quality raw data collected from both real-world and virtual scenarios. Our objective is to develop a lightweight encoder capable of robust generalization for applications. To achieve this, we employ DepthAnythingV2 [51] as a teacher network to distill knowledge into our depth foundation models. As shown in previous work [13], virtual datasets offer advantages such as high quality and precise synchronization. Moreover, high-quality datasets tend to provide more reliable supervision for pretrained models [13]. Therefore, we utilize both high-quality virtual and real-world datasets, including VKITTI2 [4], Hypersim [32], TartanAir [45], and SA-1B [21]. During training, both the depth foundation models and the DPT decoder [30] are trained using a combination of real-world and virtual datasets. To facilitate training, the decoder is constrained by pseudo labels generated by DepthAnythingV2. Assume the input image is denoted as $\boldsymbol{I}$. Our teacher features from the DepthAnythingV2 image encoder can be written as $f^{\text{da}}(\boldsymbol{I})$, the output from DepthAnythingV2 DPT head is $\boldsymbol{D}^{\text{da}} = h^{\text{da}}(f^{\text{da}}(\boldsymbol{I}))$. The output from the lightweight depth foundation model's encoder is $f^{\text{tiny}}(\boldsymbol{I})$, and the lightweight DPT decoder output is $\boldsymbol{D}^{\text{tiny}} = h^{\text{tiny}}(f^{\text{tiny}}(\boldsymbol{I}))$. The overall loss is defined as the sum of the logit distillation loss $\mathcal{L}_l$ and the feature distillation $\mathcal{L}_f$. The logit distillation loss between the DepthAnythingV2 [51] and our lightweight DPT decoder [30] can be defined as follows:

$$\begin{aligned} \mathcal{L}_l = & \frac{1}{N} \sum_{i=1}^{N} \left| \rho(\boldsymbol{D}^{\text{da}}) - \rho(\boldsymbol{D}^{\text{tiny}}) \right| \\ & + \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} \Big(\left| \rho(\nabla_x \boldsymbol{D}^{\text{da}}) - \rho(\nabla_x \boldsymbol{D}^{\text{tiny}}) \right| \Big) \\ & + \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} \Big(\left| \rho(\nabla_y \boldsymbol{D}^{\text{da}}) - \rho(\nabla_y \boldsymbol{D}^{\text{tiny}}) \right| \Big), \end{aligned} \tag{37}$$

where N is the number of pixels, $|\cdot|$ denotes a ℓ_1 norm, α is set as in [51], and the function ρ is the scaled and shifted operation. The feature distillation loss can be defined as follows:

$$\mathcal{L}_f = \frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{j=1}^{N} \left\| f^{\text{da}}(\boldsymbol{I}) - f^{\text{tiny}}(\boldsymbol{I}) \right\|^2, \tag{38}$$

where M represents the number of multi-scale features.

C. Experiments

C.1. Datasets

For depth estimation, we use the KITTI [14], NYU-Depth V2 (NYUv2) [35], ScanNet [8], ETH3D [34], and DIODE [42] datasets. We follow the data splits adopted in the studies [20, 38]. KITTI is a street-scene dataset with sparse metric depth captured by a LiDAR sensor, and we employ the Eigen test split [10]. The indoor dataset NYUv2 provides 654 images, while ScanNet contains 800 images. DIODE [42] and ETH3D [34] are high-resolution mixed indoor-outdoor datasets; in the splits, ETH3D contributes 454 images, and DIODE provides 325 indoor samples and 446 outdoor samples.

For depth completion, we use nuScenes [5], DDAD [16], Make3D [33], DIODE [42], ETH3D [34], ScanNet [8], VOID [48], SUN-RGBD [37], IBims-1 [22], and HAMMER [19]. For nuScenes, we follow the preprocessing protocol and test split of prior work [36]. We discard samples with obvious projection errors. For DDAD, we follow the split and preprocessing of prior work [43] and use images at a resolution of 1216×1936. For Make3D [33], we adopt the split used in previous work [56]; due to the inherently low depth resolution of this dataset, the resulting MAE and RMSE are relatively large. For DIODE [42], ETH3D [34], and ScanNet [8], we use the same splits as in our depth prediction experiments. For VOID [48], we again follow the split proposed in prior work [29]. For SUN-RGBD [37], we use the split that excludes images overlapping with NYUv2 [35], resulting in approximately 4.4k images. For HAMMER [19], we randomly sample 400 images for testing. For IBims-1 [22], we use the official evaluation split with 100 images. nuScenes [5], DDAD [16], and

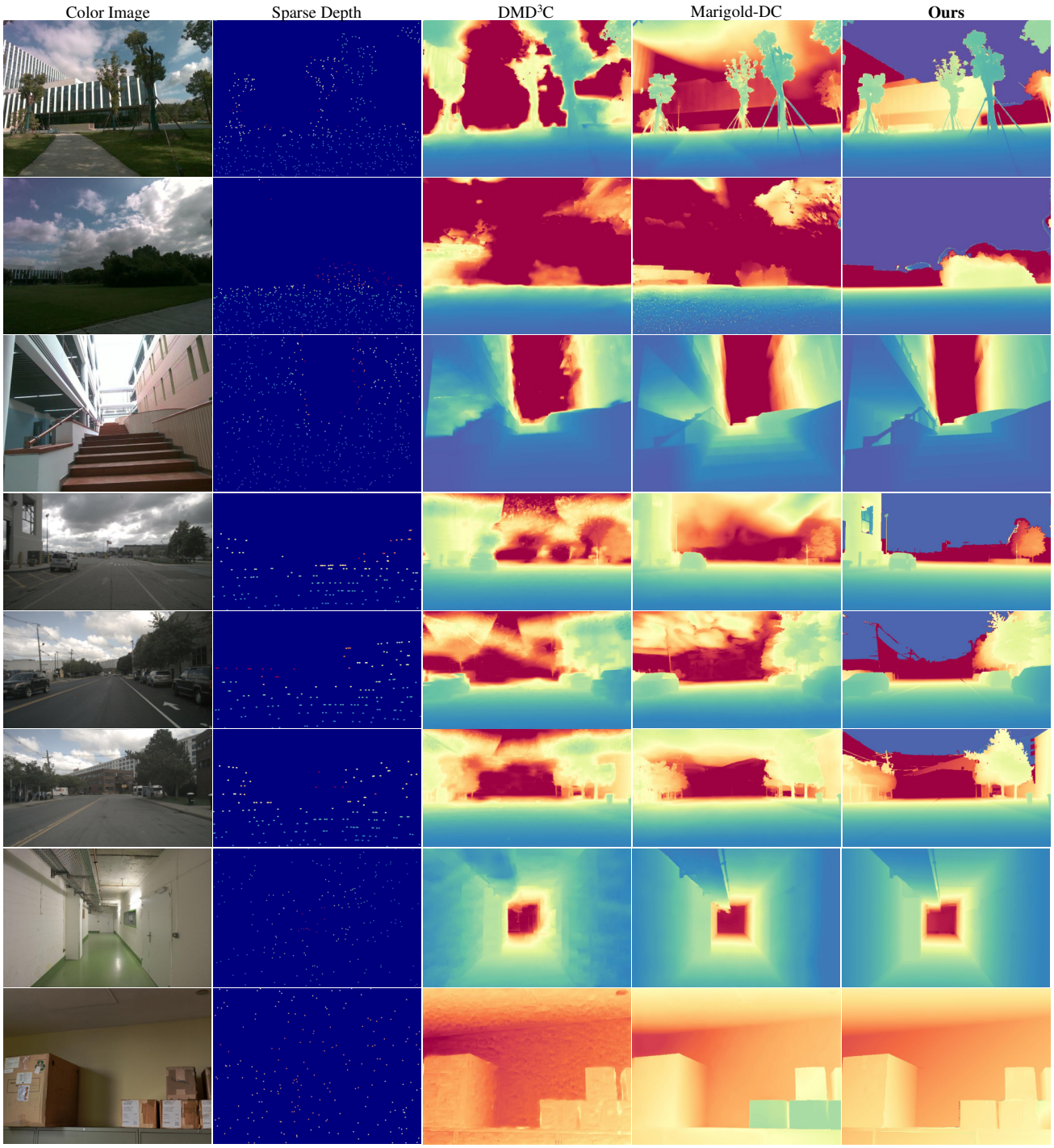

Figure 1. Qualitative comparisons with SoTA depth completion methods.

VOID [48] provide sparse depth maps directly. For the remaining datasets, we simulate sparse depth by randomly retaining 0.01%–0.1% depth points. We apply noisy random sampling, where points are uniformly sampled at varying densities and 10%–20% of them are perturbed with noise. For datasets acquired with line-scan LiDAR sensors, we use LiDAR-simulated sampling.

Table 1. Quantitative comparison with SoTA depth completion methods. All methods are evaluated in a zero-shot setting.

Method	nuScenes		VOID1500		VOID500		VOID150		IBims-1	
	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓
DepthAnythingV2 [51]	5.303	3.163	0.605	0.209	0.582	0.209	0.644	0.230	0.305	0.132
Depth Pro [3]	6.232	3.657	0.734	0.385	0.697	0.373	0.758	0.392	0.332	0.148
Marigold [20]	5.459	3.271	0.630	0.240	0.607	0.241	0.673	0.263	0.306	0.138
DPromting [28]	13.981	9.182	0.779	0.373	0.754	0.373	0.820	0.398	0.297	0.102
BP-Net [40]	15.092	10.592	0.738	0.268	0.790	0.369	0.934	0.470	0.302	0.119
DMD^{3}C [24]	5.556	3.112	0.676	0.225	0.736	0.275	0.762	0.297	0.286	0.083
G2-Monodepth [44]	8.921	4.587	0.568	0.159	0.574	0.182	0.691	0.247	0.267	0.078
Marigold-DC [43]	4.924	2.595	0.505	0.151	0.535	0.158	0.622	**0.194**	0.176	0.038
MTD (Ours)	**4.387**	**2.177**	**0.366**	**0.138**	**0.522**	**0.157**	**0.615**	0.217	0.190	0.072

Table 2. Quantitative comparison with SoTA zero-shot depth estimation methods. We utilize the metric depth estimator to generate the 3D seeds for depth estimation.

Method	KITTI		NYUv2		ETH3D		ScanNet		DIODE	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
DiverseDepth [52]	0.190	0.704	0.117	0.875	0.228	0.694	0.109	0.882	0.376	0.631
MiDaS [2]	0.183	0.711	0.095	0.915	0.190	0.884	0.099	0.907	0.266	0.713
LeReS [53]	0.149	0.784	0.090	0.916	0.171	0.777	0.091	0.917	0.271	0.766
Omnidata [9]	0.149	0.835	0.074	0.945	0.166	0.778	0.075	0.936	0.339	0.742
HDN [54]	0.115	0.867	0.069	0.948	0.121	0.833	0.080	0.939	0.246	0.780
DPT [30]	0.111	0.881	0.091	0.919	0.115	0.929	0.084	0.932	0.269	0.730
Depth Pro [3]	0.077	0.949	0.044	0.975	0.060	0.965	**0.042**	0.980	0.321	0.752
DepthAnythingV2 [51]	0.080	0.946	0.043	0.980	0.062	0.980	0.043	**0.981**	0.260	0.759
Marigold [20]	0.099	0.916	0.055	0.964	0.065	0.960	0.064	0.951	0.308	0.773
GeoWizard [12]	0.097	0.921	0.052	0.966	0.064	0.961	0.061	0.953	0.297	0.792
Lotus [17]	0.093	0.928	0.053	0.967	0.068	0.953	0.060	0.963	0.228	0.738
DepthMaster [38]	0.082	0.937	0.050	0.972	0.053	0.974	0.055	0.967	0.215	0.776
Ours	**0.075**	**0.953**	**0.038**	**0.980**	**0.051**	**0.981**	0.044	0.978	**0.207**	**0.816**

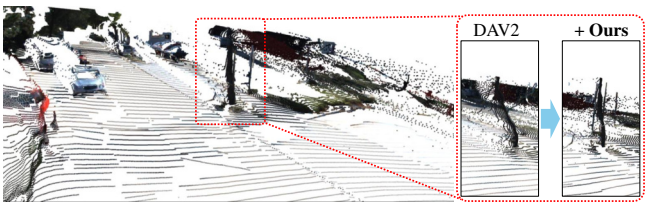

Figure 2. Illustration of reducing local scale inconsistencies in our method. "DAV2" denotes the DepthAnythingV2, which exhibits noticeable local scale inconsistencies around the tree trunks.

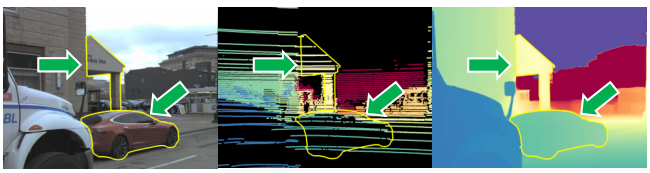

Figure 3. RGB-LiDAR misalignment on Argoverse 2.

C.2. The Details of Comparison with SoTA Methods

In our main paper, for the depth estimation baselines, the quantitative results reported in our tables (the section without our method) are taken from the original studies [20, 38]. In our main paper, for depth completion baselines, we follow the protocol of prior work [24, 59] and conduct zero-shot evaluation for CFormer [57], BP-Net [40], and LRRU [47] by using weights trained on KITTI for outdoor datasets and weights trained on NYUv2 for indoor datasets. For DMD^{3}C [24], we use the official weights trained on a large-scale dataset for zero-shot evaluation. It is worth noting that the pretraining of

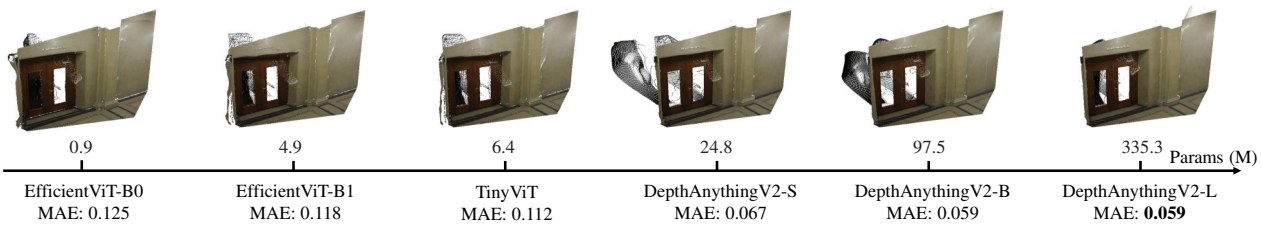

Figure 4. Qualitative results across different depth foundation models on the ETH3D Indoor dataset.

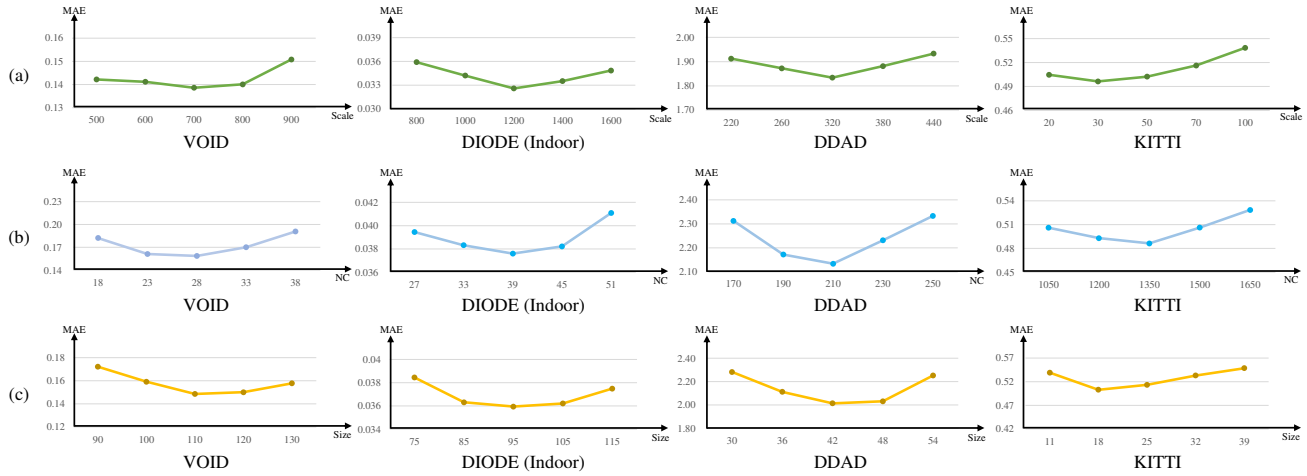

Figure 5. MAE comparison under different superpixel segmentation algorithms. The subfigures correspond to (a) the Felzenszwalb algorithm [11], (b) the SLIC algorithm [1], and (c) the LSC algorithm [23]. In (b), "NC" denotes the number of components in SLIC [1]. Under a fixed number of 3D seed points, we search over the hyperparameters of each algorithm to minimize MAE. VOID and DIODE are indoor datasets, while DDAD and KITTI are outdoor datasets.

Table 3. Zero-shot evaluation. * NLSPN's KITTI-DC results from VPP4DC (as in Marigold-DC). Prop-Time is benchmarked on an RTX 3090 with 480×640 inputs. For SPN-based methods, we report "guidance generation + propagation" time; for ours, we report the full pixel-wise refinement time, as guidance and propagation are performed jointly. For completeness, the total back-end runtime (segment-wise recovery + pixel-wise refinement) is 1.9 ms.

Method	nuScenes		VOID		KITTI-DC		Prop-Time (ms)
	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	
CSPN [7]	15.871	10.792	1.571	0.548	-	-	46.5+22.1
NLSPN* [27]	13.630	8.809	1.353	0.427	2.076	1.335	13.6+6.0
Marigold-DC	4.924	2.595	0.505	0.151	**1.465**	0.434	-
Ours	**4.387**	**2.177**	**0.366**	**0.138**	1.471	**0.422**	**0.9**

Table 4. Additional ablation studies on the KITTI and VOID datasets.

Factor	KITTI		VOID	
	RMSE↓	MAE↓	RMSE↓	MAE↓
Fitting: median	10.891	2.169	0.898	0.358
Fitting: quantile matching	9.178	2.114	0.821	0.346
Fitting: moment matching	8.756	1.983	0.815	0.329
Fitting: least squares	7.013	1.802	0.791	0.307
Domain: z^{-1}	6.782	1.794	0.614	0.238
Graph: global	2.521	0.687	0.554	0.169
Graph: graph-based (2D)	2.232	0.608	0.468	0.157
Graph: graph-based (3D)	2.318	0.634	0.459	0.150
With bilateral filtering	**2.201**	**0.597**	**0.452**	**0.148**

DMD^{3}C [24] includes nuScenes, which partly explains its superior performance on the nuScenes benchmark. Although DMD^{3}C [24] and BP-Net [40] share similar network architectures, we observe that this large-scale pretraining substantially improves the generalization ability of DMD^{3}C. For PromptDA [25], since the official weights are trained on a specific dataset and exhibit poor generalization, we reimplement the method. We configure the denoising process of Marigold-DC [43] within a reasonable range.

Table 5. Quantitative comparison with zero-shot stereo matching methods.

Method	Middlebury BP-2	ETH3D BP-1	KITTI-12 D1	KITTI-15 D1
CREStereo++ [18]	14.8	4.4	4.7	5.2
DSMNet [55]	13.8	6.2	6.2	6.5
Mask-CFNet [31]	13.7	5.7	4.8	5.8
HVT-RAFT [6]	10.4	3.0	3.7	5.2
RAFT-Stereo [26]	12.6	3.3	4.7	5.5
Selective-IGEV [46]	9.2	5.7	4.5	5.6
IGEV [49]	8.8	4.0	5.2	5.7
Former-RAFT-DAM [58]	8.1	3.3	3.9	5.1
IGEV++ [50]	7.8	4.1	5.1	5.9
NMRF [15]	7.5	3.8	4.2	5.1
Ours	**6.1**	**2.8**	**3.6**	**5.0**

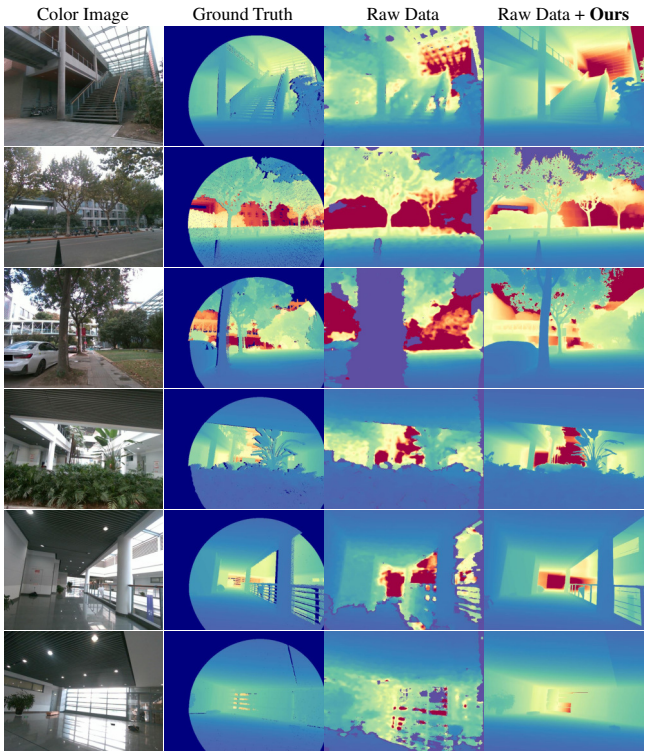

Figure 6. Qualitative results for the rectification of commonly used range cameras.

Furthermore, Table 1 provides additional comparisons on multiple depth completion datasets, complementing comparisons for the main paper. In Table 2, we fine-tune UniDepth [29] on the downstream datasets to obtain 3D seeds. This setup allows our method to use only monocular color images as input. Fig. 1 shows qualitative comparisons among our method and SoTA depth completion methods on multiple datasets. Fig. 2 illustrates the process of reducing local scale inconsistencies. As shown in the figure, the original predictions of DepthAnythingV2 [51] exhibit noticeable local inconsistencies around the tree trunks, while our segment-wise recovery strategy reduces these inconsistencies. This qualitative evidence further validates the motivation underlying our method. In addition, Fig. 3 shows that when the color image and the LiDAR projected sparse depth are misaligned, a condition commonly encountered in outdoor datasets, our method can still robustly recover a well-aligned dense depth map.

We also provide a quantitative comparison with SPN-based methods in Table 3. Unlike SPN-based methods, we derive a discontinuity-aware propagation metric from second-order residuals (Proposition 1), leading to a geodesic shortest path formulation that can be solved efficiently via dynamic programming. Beyond not requiring training, our method has three advantages. (1) Better interpretability: d_ϕ has a clear geometric meaning as a geodesic cost that penalizes discontinuity crossings, unlike implicitly learned affinities. (2) Better cross-domain generalization: our geometry-driven cost avoids affinity learning and is therefore less sensitive to domain shifts. (3) Higher efficiency and lower computational overhead: our approach is lightweight and plug-and-play, requiring no additional neural network to produce pixel-wise propagation.

C.3. Ablation Study

In this section, we first describe the ablation setting used in the main paper. In Fig. 4 of the main paper, the ETH3D and DIODE results are computed only on their indoor subsets. The purpose of this ablation is to analyze how the number of 3D seed points (NP) influences performance under different scene types (indoor versus outdoor). In contrast, Table 1 of the main paper reports results on the full splits, where both indoor and outdoor scenes of ETH3D and DIODE are included. Consequently, the results are different between these two settings.

The main paper shows the effectiveness of our segment-wise recovery and pixel-wise refinement. Since the supplementary material describes the detailed methods for computing the per-segment calibration function and explains the iterative bilateral filtering used in segment propagation and graph optimization, we present more detailed ablation studies for segment-wise recovery. In Table 4, the upper part of the table compares the results under different calibration functions. The lower part of the table compares whether the distance term $\|\boldsymbol{c}_i - \boldsymbol{c}_j\|$ in the sparse graph optimization uses 2D or 3D coordinates. We also demonstrate the benefit of adding our bilateral filtering.

Fig. 4 compares the results of our method across depth foundation models with different numbers of parameters for the same input. The figure shows point cloud visualizations on the ETH3D Indoor dataset. EfficientViT-

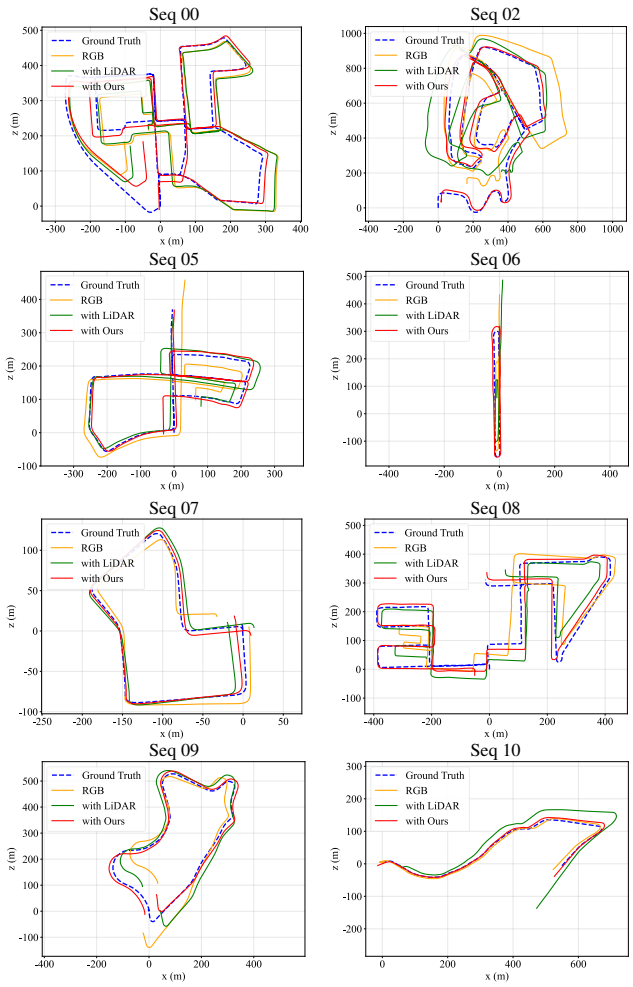

Figure 7. Additional qualitative results on the KITTI Odometry benchmark.

B0, EfficientViT-B1, and TinyViT are models that we obtain by distillation. These lightweight depth foundation models achieve MAE within an acceptable range and provide a good balance between accuracy and efficiency, which makes them suitable for other downstream tasks. The MAE difference between DepthAnythingV2-S and DepthAnythingV2-B is relatively minor, and the difference between DepthAnythingV2-B and DepthAnythingV2-L is negligible. This result also indicates that our method does not depend on high-capacity relative depth estimators to achieve accurate metric depth estimation.

Moreover, in Fig. 5, we search for the optimal hyperparameters of the superpixel segmentation algorithms to minimize MAE, and we also compare different superpixel algorithms under a fixed number of 3D seed points. The optimal parameters of the three superpixel methods are related. For example, for (a) the Felzenszwalb algorithm [11], if the optimal scale produces N distinct segments, then for (b) the SLIC algorithm [1], the optimal number of components is also close to N. If $A = H \times W/N$ denotes the approximate area of each segment, then for (c) the LSC algorithm [23], the optimal value of the size parameter lies near $\sqrt{A}$. In addition, the MAE does not change significantly across different hyperparameter settings, and the optimal MAE values of the three algorithms are relatively close. These observations demonstrate the robustness of our method. Therefore, in the ablation studies in the main paper, we use the Felzenszwalb algorithm [11], as it achieves the best overall performance on both indoor and outdoor datasets.

Figure 8. Qualitative results on our method for the Radar depth completion task.

C.4. Applications

Fig. 6 presents qualitative results obtained by applying our method to depth rectification for commonly used range cameras. Our method substantially improves the quality of raw depth data by filling missing regions, sharpening depth textures, and preserving object boundaries. Comparing our results with the ground-truth visualizations, we observe that many objects that appear blurry in the raw data become clearer and exhibit more accurate depth values. Fig. 7 shows additional SLAM results from the KITTI Odometry benchmark. Compared with using DROID-SLAM [41] directly on raw point clouds, incorporating our method clearly improves the results.

In the main paper, we show that our method can be integrated with vision foundation models and applied to multi-view stereo. Table 7 further demonstrates that our method can also be applied to stereo matching, where it achieves competitive performance. We use the high-confidence matches from [39] as disparity inputs and reconstruct dense disparity maps. Fig. 8 presents qualitative radar depth completion results on the nuScenes dataset. These experiments demonstrate that our method generalizes to different input modalities, thereby confirming its effectiveness on downstream tasks.

References

[1] Radhakrishna Achanta et al. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 7, 9

[2] Reiner Birkl et al. MiDaS v3.1 – A Model Zoo for Robust Monocular Relative Depth Estimation. *arXiv preprint arXiv:2307.14460*, 2023. 6

[3] Aleksei Bochkovskii et al. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 6

[4] Yohann Cabon et al. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 4

[5] Holger Caesar et al. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 4

[6] Tianyu Chang et al. Domain Generalized Stereo Matching via Hierarchical Visual Transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9559–9568, 2023. 8

[7] Xinjing Cheng et al. Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 7

[8] Angela Dai et al. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4

[9] Ainaz Eftekhar et al. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10786–10796, 2021. 6

[10] David Eigen et al. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems (NeurIPS)*, 27, 2014. 4

[11] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 7, 9

[12] Xiao Fu et al. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 241–258. Springer, 2024. 6

[13] Yongtao Ge et al. GeoBench: Benchmarking and Analyzing Monocular Geometry Estimation Models. *arXiv preprint arXiv:2406.12671*, 2024. 4

[14] Andreas Geiger et al. 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)*, pages 3354–3361. IEEE, 2012. 1, 4

[15] Tongfan Guan et al. Neural Markov Random Field for Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5459–5469, 2024. 8

[16] Vitor Guizilini et al. 3D Packing for Self-Supervised Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 4

[17] Jing He et al. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 6

[18] Junpeng Jing et al. Uncertainty Guided Adaptive Warping for Robust and Efficient Stereo Matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3318–3327, 2023. 8

[19] HyunJun Jung et al. Is my Depth Ground-Truth Good Enough? HAMMER–Highly Accurate Multi-Modal Dataset for DEnse 3D Scene Regression. *arXiv preprint arXiv:2205.04565*, 2022. 1, 4

[20] Bingxin Ke et al. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9492–9502, 2024. 4, 6

[21] Alexander Kirillov et al. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 4

[22] Tobias Koch et al. Comparison of monocular depth estimation methods using geometrically relevant metrics on the IBims-1 dataset. *Computer Vision and Image Understanding*, 191:102877, 2020. 1, 4

[23] Zhengqin Li and Jiansheng Chen. Superpixel Segmentation Using Linear Spectral Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7, 9

[24] Yingping Liang et al. Distilling Monocular Foundation Model for Fine-grained Depth Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22254–22265, 2025. 6, 7

[25] Haotong Lin et al. Prompting Depth Anything for 4K Resolution Accurate Metric Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17070–17080, 2025. 7

[26] Lahav Lipson et al. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227, 2021. 8

[27] Jinsun Park et al. Non-local spatial propagation network for depth completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 120–136. Springer, 2020. 7

[28] Jin-Hwi Park et al. Depth prompting for sensor-agnostic depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9859–9869, 2024. 6

[29] Luigi Piccinelli et al. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10106–10116, 2024. 4, 8

[30] René Ranftl et al. Vision Transformers for Dense Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 4, 6

[31] Zhibo Rao et al. Masked Representation Learning for Domain Generalized Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5435–5444, 2023. 8

[32] Mike Roberts et al. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10912–10922, 2021. 4

[33] Ashutosh Saxena et al. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 1, 4

[34] Thomas Schops et al. A Multi-View Stereo Benchmark With High-Resolution Images and Multi-Camera Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4

[35] Nathan Silberman et al. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 1, 4

[36] Akash Deep Singh et al. Depth Estimation From Camera Image and mmWave Radar Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9285, 2023. 4

[37] Shuran Song et al. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 4

[38] Ziyang Song, Zerong Wang, Bo Li, Hao Zhang, Ruijie Zhu, Li Liu, Peng-Tao Jiang, and Tianzhu Zhang. Depthmaster: Taming diffusion models for monocular depth estimation. *arXiv preprint arXiv:2501.02576*, 2025. 4, 6

[39] Jiaming Sun et al. LoFTR: Detector-Free Local Feature Matching With Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8922–8931, 2021. 9

[40] Jie Tang et al. Bilateral propagation network for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9763–9772, 2024. 6, 7

[41] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems (NeurIPS)*, 34:16558–16569, 2021. 9

[42] Igor Vasiljevic et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 1, 4

[43] Massimiliano Viola et al. Marigold-DC: Zero-Shot Monocular Depth Completion with Guided Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5359–5370, 2025. 4, 6, 7

[44] Haotian Wang et al. G2-MonoDepth: A General Framework of Generalized Depth Inference From Monocular RGB+X Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3753–3771, 2024. 6

[45] Wenshan Wang et al. TartanAir: A Dataset to Push the Limits of Visual SLAM. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916, 2020. 4

[46] Xianqi Wang et al. Selective-Stereo: Adaptive Frequency Information Selection for Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19701–19710, 2024. 8

[47] Yufei Wang et al. LRRU: Long-short Range Recurrent Updating Networks for Depth Completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9422–9432, 2023. 6

[48] Alex Wong et al. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5 (2):1899–1906, 2020. 1, 4, 5

[49] Gangwei Xu et al. Iterative Geometry Encoding Volume for Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21919–21928, 2023. 8

[50] Gangwei Xu et al. IGEV++: Iterative Multi-Range Geometry Encoding Volumes for Stereo Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8): 7108–7122, 2025. 8

[51] Lihe Yang et al. Depth anything v2. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:21875–21911, 2024. 4, 6, 8

[52] Wei Yin et al. Diversedepth: Affine-invariant depth prediction using diverse data. *arXiv preprint arXiv:2002.00569*, 2020. 6

[53] Wei Yin et al. Learning To Recover 3D Scene Shape From a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 204–213, 2021. 6

[54] Chi Zhang et al. Hierarchical normalization for robust monocular depth estimation. *Advances in neural information processing systems (NeurIPS)*, 35:14128–14139, 2022. 6

[55] Feihu Zhang et al. Domain-invariant stereo matching networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–439. Springer, 2020. 8

[56] Ning Zhang et al. Lite-Mono: A Lightweight CNN and Transformer Architecture for Self-Supervised Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18537–18546, 2023. 4

[57] Youmin Zhang et al. CompletionFormer: Depth Completion With Convolutions and Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18527–18536, 2023. 6

[58] Yongjian Zhang et al. Learning representations from foundation models for domain generalized stereo matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 146–162. Springer, 2024. 8

[59] Yiming Zuo and Jia Deng. OGNI-DC: Robust depth completion with optimization-guided neural iterations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 78–95. Springer, 2024. 6