Self-Distilled Depth Refinement with Noisy Poisson Fusion

Jiaqi Li[1,*] **Yiran Wang**[1,*] **Jinghong Zheng**[1]
Zihao Huang[1] **Ke Xian**[2] **Zhiguo Cao**[1,†] **Jianming Zhang**[3]

[1]School of AIA, Huazhong University of Science and Technology
[2]School of EIC, Huazhong University of Science and Technology
[3]Adobe Research
[*]Equal contribution [†]Corresponding author

{lijiaqi_mail,wangyiran,deepzheng,zihaohuang,kxian,zgcao}@hust.edu.cn
jianmzha@adobe.com
https://github.com/lijia7/SDDR

Abstract

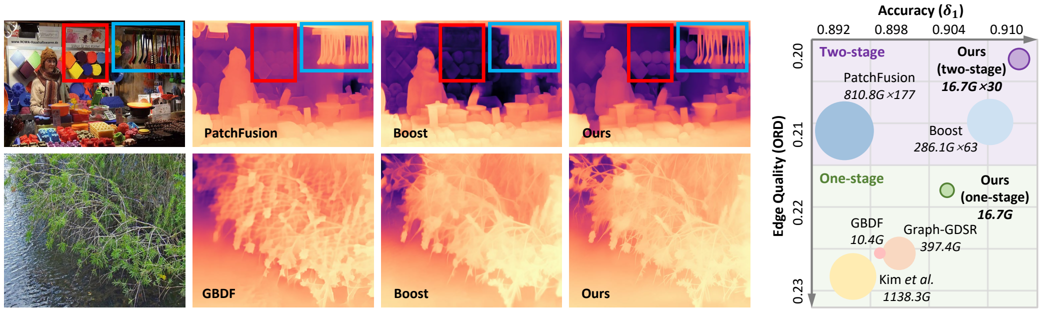
Depth refinement aims to infer high-resolution depth with fine-grained edges and details, refining low-resolution results of depth estimation models. The prevailing methods adopt tile-based manners by merging numerous patches, which lacks efficiency and produces inconsistency. Besides, prior arts suffer from fuzzy depth boundaries and limited generalizability. Analyzing the fundamental reasons for these limitations, we model depth refinement as a noisy Poisson fusion problem with local inconsistency and edge deformation noises. We propose the Self-distilled Depth Refinement (SDDR) framework to enforce robustness against the noises, which mainly consists of depth edge representation and edge-based guidance. With noisy depth predictions as input, SDDR generates low-noise depth edge representations as pseudo-labels by coarse-to-fine self-distillation. Edge-based guidance with edge-guided gradient loss and edge-based fusion loss serves as the optimization objective equivalent to Poisson fusion. When depth maps are better refined, the labels also become more noise-free. Our model can acquire strong robustness to the noises, achieving significant improvements in accuracy, edge quality, efficiency, and generalizability on five different benchmarks. Moreover, directly training another model with edge labels produced by SDDR brings improvements, suggesting that our method could help with training robust refinement models in future works.

1 Introduction

Depth refinement infers high-resolution depth with accurate edges and details, refining the low-resolution counterparts from depth estimation models [30, 51, 1]. With increasing demands for high resolutions in modern applications, depth refinement becomes a prerequisite for virtual reality [24, 13], bokeh rendering [27, 28], and image generation [33, 54]. The prevailing methods [25, 21] adopt two-stage tile-based frameworks. Based on the one-stage refined depth of the whole image, they merge high-frequency details by fusing extensive patches with complex patch selection strategies. However, numerous patches lead to heavy computational costs. Besides, as in Fig. 1 (a), excessive integration of local information leads to inconsistent depth structures, *e.g.*, the disrupted billboard.

Apart from efficiency and consistency, depth refinement [25, 14, 4, 3, 37, 21] is restricted by noisy and blurred depth edges. Highly accurate depth annotations with meticulous boundaries are necessary to enforce fine-grained details. For this reason, prior arts [14, 37, 21] only use synthetic datasets [32,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Visualization of Depth Refinement Approaches

(b) Performance and Efficiency

Figure 1: **(a) Visual comparisons.** We model depth refinement by noisy Poisson fusion with the local inconsistency noise (representing the inconsistent billboard and wall in red box) and the edge deformation noise (indicating blurred depth edges in the blue box and second row). Better viewed when zoomed in. **(b) Performance and efficiency.** Circle area represents FLOPs. The two-stage methods [25, 21] are reported by multiplying FLOPs per patch with patch numbers. SDDR outperforms prior arts in depth accuracy (δ_1), edge quality (ORD), and model efficiency (FLOPs).

39, 11, 45, 44] for the highly accurate depth values and edges. However, synthetic data falls short of the real world in realism and diversity, causing limited generalizability with blurred depth and degraded performance on in-the-wild scenarios. Some attempts [25, 3] simply adopt natural-scene datasets [35, 47, 49, 5, 20] for the problem. The varying characteristics of real-world depth annotations, *e.g.*, sparsity [2, 5, 7], inaccuracy [35, 36, 55], or blurred edges [48, 47, 43, 10], make them infeasible for supervising refinement models. Thus, GBDF [3] uses depth predictions [51] as pseudo-labels, while Boost [25] leverages adversarial training [6] as guidance. Those inaccurate pseudo-labels and guidance still lead to blurred edges as shown in Fig. 1 (a). The key problem is to alleviate the noise of depth boundaries by constructing accurate edge representations and guidance.

To tackle these challenges, we dig into the underlying reasons for the limitations, instead of the straightforward merging of local details. We model depth refinement as a noisy Poisson fusion problem, decoupling depth prediction errors into two degradation components: local inconsistency noise and edge deformation noise. We use regional linear transformation perturbation as the local inconsistency noise to measure inconsistent depth structures. The edge deformation noise represents fuzzy boundaries with Gaussian blur. Experiments in Sec. 3.1 showcase that the noises can effectively depict general depth errors, serving as our basic principle to improve refinement results.

In pursuit of the robustness against the local inconsistency noise and edge deformation noise, we propose the Self-distilled Depth Refinement (SDDR) framework, which mainly consists of depth edge representation and edge-based guidance. A refinement network is considered as the Poisson fusion operator, recovering high-resolution depth from noisy predictions of depth models [51, 30, 1]. Given the noisy input, SDDR can generate low-noise and accurate depth edge representation as pseudo-labels through coarse-to-fine self-distillation. The edge-based guidance including edge-guided gradient loss and edge-based fusion loss is designed as the optimization objective of Poisson fusion. When depth maps are better refined, the pseudo-labels also become more noise-free. Our approach establishes accurate depth edge representations and guidance, endowing SDDR with strong robustness to the two types of noises. Consequently, as shown in Fig. 1 (b), SDDR significantly outperforms prior arts [25, 21, 3] in depth accuracy and edge quality. Besides, without merging numerous patches as the two-stage tile-based methods [21, 25], SDDR achieves much higher efficiency.

We conduct extensive experiments on five benchmarks. SDDR achieves state-of-the-art performance on the commonly-used Middlebury2021 [34], Multiscopic [52], and Hypersim [32]. Meanwhile, since SDDR can establish self-distillation with accurate depth edge representation and guidance on natural scenes, the evaluations on in-the-wild DIML [15] and DIODE [40] datasets showcase our superior generalizability. Analytical experiments demonstrate that these noticeable improvements essentially arise from the strong robustness to the noises. Furthermore, the precise depth edge labels produced by SDDR can be directly used to train another model [3] and yield improvements, which indicates that our method could help with training robust refinement models in future works.

In summary, our main contributions can be summarized as follows:

- We model the depth refinement task through the noisy Poisson fusion problem with local inconsistency noise and edge deformation noise as two types of depth degradation.
- We present the robust and efficient Self-distilled Depth Refinement (SDDR) framework, which can generate accurate depth edge representation by the coarse-to-fine self-distillation paradigm.
- We design the edge-guided gradient loss and edge-based fusion loss, as the edge-based guidance to enforce the model with both consistent depth structures and meticulous depth edges.

2 Related Work

Depth Refinement Models. Depth refinement refines low-resolution depth from depth estimation models [30, 51, 1], predicting high-resolution depth with fine-grained edges and details. Existing methods [3, 14, 21, 25] can be categorized into one-stage [3, 14] and two-stage [25, 21] frameworks. One-stage methods [3, 14] conduct global refinement of the whole image, which could produce blurred depth edges and details. To further enhance local details, based on the globally refined results, the prevailing refinement approaches [25, 21] adopt the two-stage tile-based manner by selecting and merging numerous patches. For example, Boost [25] proposes a complex patch-sampling strategy based on the gradients of input images. PatchFusion [21] improves the sampling by shifted and tidily arranged tile placement. However, the massive patches lead to low efficiency. The excessive local information produces inconsistent depth structures or even artifacts. In this paper, we propose the Self-distilled Depth Refinement (SDDR) framework, which can predict both consistent structures and accurate details with much higher efficiency by tackling the noisy Poisson fusion problem.

Depth Refinement Datasets. Depth datasets with highly accurate annotations and edges are necessary for refinement models. Prior arts [21, 14] utilize CG-rendered datasets [45, 44, 39, 11, 32] for accurate depth, but the realism and diversity fail to match the real world. For instance, neither the UnrealStereo4K [39] nor the MVS-Synth [11] contain people, restricting the generalizability of refinement models. A simple idea for the problem is to leverage natural-scene data [35, 47, 49, 5, 20]. However, different annotation methods lead to varying characteristics, *e.g.*, sparsity of LiDAR [2, 5, 7], inaccurate depth of structured light [55, 35, 36], and blurred edges of stereo matching [49, 47, 43]. To address the challenge, Boost [25] adopts adversarial training as guidance only with a small amount of accurately annotated real-world images. GBDF [3] employs depth predictions [51] with guided filtering [9] as pseudo-labels. Due to the inaccurate pseudo-labels and guidance, they [3, 25] produce blurred edges and details. By contrast, SDDR constructs accurate depth edge representation and edge-based guidance for self-distillation, leading to fine-grained details and strong generalizability.

3 SDDR: Self-Distilled Depth Refinement

We present a detailed illustration of our Self-distilled Depth Refinement (SDDR) framework. In Sec. 3.1, we introduce the noisy Poisson fusion to model the depth refinement task and provide an overview to outline our approach. SDDR mainly consists of depth edge representation and edge-based guidance, which will be described in Sec. 3.2 and Sec. 3.3 respectively.

3.1 Noisy Poisson Fusion

Problem Statement. Based on depth maps of depth prediction models, *i.e.*, depth predictor $\mathcal{N}_d$, depth refinement recovers high-resolution depth with accurate edges and details by refinement network $\mathcal{N}_r$. Some attempts in image super-resolution [31, 56, 26] and multi-modal integration [19, 17, 18, 53] utilize Poisson fusion to merge features and restore details. Motivated by this, we propose to model depth refinement as a noisy Poisson fusion problem. The ideal depth D^* with completely accurate depth values and precise depth edges are unobtainable in real world. A general depth prediction D, whether produced by $\mathcal{N}_d$ or $\mathcal{N}_r$ for an input image I, can be expressed as a noisy approximation of D^*:

$$D \approx D^* + \epsilon_{\text{cons}} + \epsilon_{\text{edge}} \,. \tag{1}$$

ϵ_{cons} and ϵ_{edge} denote local inconsistency and edge deformation noise to decouple depth prediction errors. Local inconsistency noise ϵ_{cons} represents inconsistent depth structures through regional linear transformation perturbation. Based on masked Gaussian blur, edge deformation noise ϵ_{edge} showcases degradation and blurring of depth edges. Refer to Appendix A.4 for details of the noises. As in Fig. 2,

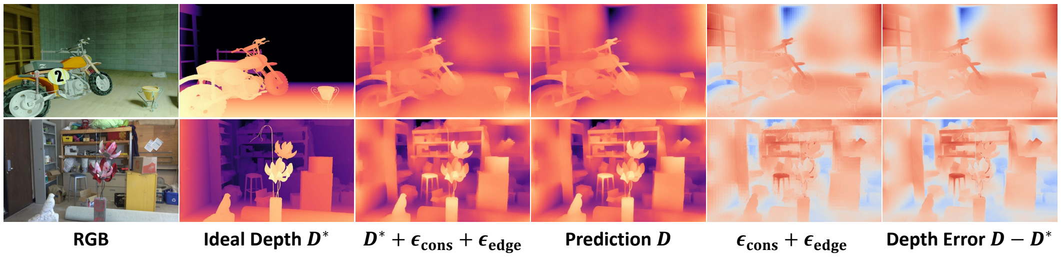

RGB
Ideal Depth D^*
$D^* + \epsilon_{\text{cons}} + \epsilon_{\text{edge}}$
Prediction D
$\epsilon_{\text{cons}} + \epsilon_{\text{edge}}$
Depth Error $D - D^*$

Figure 2: **Depiction of depth errors.** We utilize two samples of high-quality depth maps as ideal depth D^*. For the predicted depth D, the combination of local inconsistency noise ϵ_{cons} and edge deformation noise ϵ_{edge} can approximate real depth error $D - D^*$ (the last two columns). Thus, as in the third and fourth columns, prediction D can be depicted by the summation of D^*, ϵ_{cons}, and ϵ_{edge}.

depth errors can be depicted by combinations of ϵ_{cons} and ϵ_{edge}. Thus, considering refinement network $\mathcal{N}_r$ as a Poisson fusion operator, depth refinement can be defined as a noisy Poisson fusion problem:

$$\begin{gathered} D_0 = \mathcal{N}_r(\mathcal{N}_d(L), \mathcal{N}_d(H)) \,, \\ \text{s.t.} \min_{D_0, \Omega} \iint_{\Omega} |\nabla D_0 - \nabla D^*| \, \partial\Omega + \iint_{I-\Omega} |D_0 - D^*| \, \partial\Omega \,. \end{gathered} \tag{2}$$

The refined depth of $\mathcal{N}_r$ is denoted as D_0. ∇ refers to the gradient operator. Typically for depth refinement [3, 25, 21] task, input image I is resized to low-resolution L and high-resolution H for $\mathcal{N}_d$. Ω represents high-frequency areas, while $I - \Omega$ showcases low-frequency regions.

Motivation Elaboration. In practice, due to the inaccessibility of truly ideal depth, approximation of D^* is required for training $\mathcal{N}_r$. For this reason, the optimization objective in Eq. 2 is divided into Ω and $I - \Omega$. For the low-frequency $I - \Omega$, D^* can be simply represented by the ground truth D^*_{gt} of training data. However, as illustrated in Sec. 2, depth annotations inevitably suffer from imperfect edge quality for the high-frequency Ω. It is essential to generate accurate approximations of ideal depth boundaries as training labels, which are robust to ϵ_{cons} and ϵ_{edge}. Some prior arts adopts synthetic depth [39, 11, 32] for higher edge quality, while leading to limited generalization capability with blurred predictions in real-world scenes. To leverage real depth data [35, 47, 46, 49, 5, 20], GBDF [3] employs depth predictions [51] with guided filter as pseudo-labels, which still contain significant noises and result in blurred depth. Besides, optimization of Ω is also ignored. Kim *et al.* [14] relies on manually annotated Ω regions as input. GBDF [3, 30, 29] omits the selection of Ω and supervises depth gradients on the whole image. Inaccurate approximations of ∇D^* and inappropriate division of Ω lead to limited robustness to local inconsistency noise and edge deformation noise.

Method Overview. To address the challenges, as shown in Fig. 3, we propose our SDDR framework with two main components: depth edge representation and edge-based guidance. To achieve low-noise approximations of ∇D^*, we construct the depth edge representation G_s through coarse-to-fine self-distillation, where $s \in \{1, 2, \cdots, S\}$ refers to iteration numbers. The input image is divided into several windows with overlaps from coarse to fine. For instance, we denote the high-frequency area of a certain window w in iteration s as Ω^w_s, and the refined depth of $\mathcal{N}_r$ as D^w_s. In this way, the self-distilled optimization of depth edge representation G_s can be expressed as follows:

$$\begin{gathered} D^w_s \approx D^* + \epsilon_{\text{cons}} + \epsilon_{\text{edge}} \,, \\ \min_{G_s} \sum_{w} \iint_{\Omega^w_s} |G^w_s - \nabla D^w_s| \, \partial\Omega^w_s \,. \end{gathered} \tag{3}$$

During training, depth edge representation G^w_s is further optimized based on the gradient of current refined depth D^w_s. The final edge representation G_S of the whole image will be utilized as the pseudo-label to supervise the refinement network $\mathcal{N}_r$ after S iterations. SDDR can generate low-noise and robust edge representation, mitigating the impact of ϵ_{cons} and ϵ_{edge} (More results in Appendix A.1).

With G_S as the training label, the next is to enforce $\mathcal{N}_r$ with robustness to the noises, achieving consistent structures and meticulous boundaries. To optimize $\mathcal{N}_r$, we propose edge-based guidance as an equivalent optimization objective to noisy Poisson fusion problem, which is presented by:

$$\min_{D0, \Omega} \iint_{\Omega} |\nabla D_0 - G_S| \, \partial\Omega + \iint_{I-\Omega} |D_0 - D^*_{gt}| \, \partial\Omega \,. \tag{4}$$

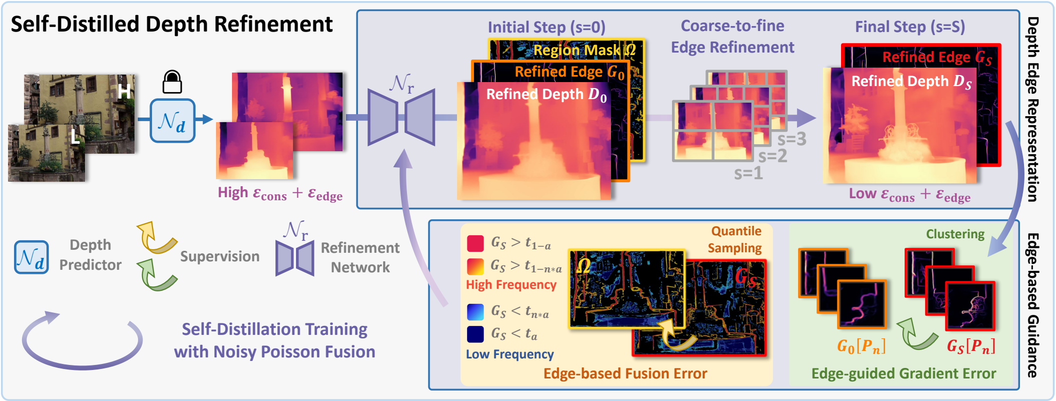


Figure 3: **Overview of self-distilled depth refinement.** SDDR consists of depth edge representation and edge-based guidance. Refinement network $\mathcal{N}_r$ produces initial refined depth D_0, edge representation G_0, and learnable soft mask Ω of high-frequency areas. The final depth edge representation G_S is updated from coarse to fine as pseudo-labels. The edge-based guidance with edge-guided gradient loss and edge-based fusion loss supervises $\mathcal{N}_r$ to achieve consistent structures and fine-grained edges.

For the second term of $I - \Omega$, we adopt depth annotations D^*_{gt} as the approximation of D^*. For the first term, with the generated G_S as pseudo-labels of ∇D^*, we propose edge-guided gradient loss and edge-based fusion loss to optimize D_0 and Ω predicted by $\mathcal{N}_r$. The edge-guided gradient loss supervises the model to consistently refine depth edges with local scale and shift alignment. The edge-based fusion loss guides $\mathcal{N}_r$ to adaptively fuse low- and high-frequency features based on the learned soft region mask Ω, achieving balanced consistency and details by quantile sampling.

Overall, when depth maps are better refined under the edge-based guidance, the edge representation also becomes more accurate and noise-free with the carefully designed coarse-to-fine manner. The self-distillation paradigm can be naturally conducted based on the noisy Poisson fusion, enforcing our model with strong robustness against the local inconsistency noise and edge deformation noise.

3.2 Depth Edge Representation

To build the self-distilled training paradigm, the prerequisite is to construct accurate and low-noise depth edge representations as pseudo-labels. Meticulous steps are designed to generate the representations with both consistent structures and accurate details.

Initial Depth Edge Representation. We generate an initial depth edge representation based on the global refinement results of the whole image. For the input image I, we obtain the refined depth results D_0 from $\mathcal{N}_r$ as in Eq. 2. Depth gradient $G_0 = \nabla D_0$ is calculated as the initial representation. An edge-preserving filter [38] is applied on G_0 to reduce noises in low-frequency area $I - \Omega$. With global information of the whole image, G_0 can preserve spatial structures and depth consistency. It also incorporates certain detailed information from the high-resolution input H. To enhance edges and details in high-frequency region Ω, we conduct coarse-to-fine edge refinement in the next step.

Coarse-to-fine Edge Refinement. The initial D_0 is then refined from course to fine with S iterations to generate final depth edge representation. For a specific iteration $s \in \{1, 2, \cdots, S\}$, we uniformly divide input image I into $(s+1)^2$ windows with overlaps. We denote a certain window w in iteration s of the input image I as I^w_s. The high-resolution H^w_s is then fed to the depth predictor $\mathcal{N}_d$. D^w_{s-1} represents the depth refinement results of the corresponding window w in the previous iteration $s - 1$. The refined depth D^w_s of window w in current iteration s as Eq. 3 can be obtained by $\mathcal{N}_d$ and $\mathcal{N}_r$:

$$D^w_s = \mathcal{N}_r(D^w_{s-1}, \mathcal{N}_d(H^w_s)), s \in \{1, 2, \cdots, S\}, \tag{5}$$

After that, depth gradient ∇D^w_s is used to update the depth edge representation. The coarse-to-fine manner achieves consistent spatial structures and accurate depth details with balanced global and regional information. In the refinement process, only limited iterations and windows are needed. Thus, SDDR achieves much higher efficiency than tile-based methods [25, 21], as shown in Sec. C.1.

Scale and Shift Alignment. The windows are different among varied iterations. Depth results and edge labels on corresponding window w of consecutive iterations could be inconsistent in depth scale

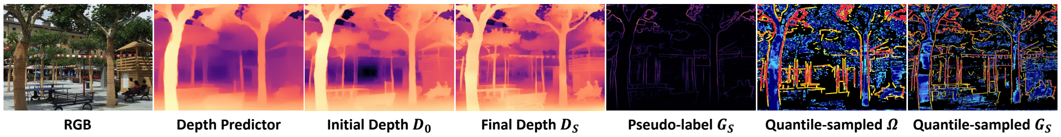


Figure 4: **Visualization of intermediate results.** We visualize the results of several important steps within the SDDR framework. The quantile sampling utilizes the same color map as in Fig. 3.

and shift. Therefore, alignment is required before updating the depth edge representation:

$$\begin{gathered}(\beta_1, \beta_0) = \underset{\beta_1,\beta_0}{\arg\min} \|(\beta_1 \nabla D_s^w + \beta_0) - G_{s-1}^w\|_2^2 \,, \\ G_s^w = \beta_1 \nabla D_s^w + \beta_0 \,,\end{gathered} \tag{6}$$

where β_1 and β_0 are affine transformation coefficients as scale and shift respectively. The aligned G_s^w represents the depth edge pseudo-labels for image patch I_s^w generated from the refined depth D_s^w. At last, after S iterations, we can obtain the pseudo-label G_S as the final depth edge representation for self-distillation. For better understanding, we showcase visualization of D_0, D_S, and G_S in Fig. 4.

Robustness to Noises. In each window, we merge high-resolution $\mathcal{N}_d(H_s^w)$ to enhance details and suppress ϵ_{edge}. Meanwhile, coarse-to-fine window partitioning and scale alignment mitigate ϵ_{cons} and bring consistency. Thus, G_S exhibits strong robustness to the two types of noises by self-distillation.

3.3 Edge-based Guidance

With depth edge representation G_S as pseudo-label for self-distillation, we propose the edge-based guidance including edge-guided gradient loss and edge-based fusion loss to supervise $\mathcal{N}_r$.

Edge-guided Gradient Loss. We aim for fine-grained depth by one-stage refinement, while the two-stage coarse-to-fine manner can further improve the results. Thus, edge-guided gradient loss instructs the initial D_0 with the accurate G_S. Some problems need to be tackled for this purpose.

As $\mathcal{N}_r$ has not converged in the early training phase, G_S is not sufficiently reliable with inconsistent scales and high-level noises between local areas. Therefore, we extract several non-overlapping regions $P_n, n \in \{1, 2, \cdots, N_g\}$ with high gradient density by clustering [8], where N_g represents the number of clustering centroids. The edge-guided gradient loss is only calculated inside P_n with scale and shift alignment. By doing so, the model can focus on improving details in high-frequency regions and preserving depth structures in flat areas. The training process can also be more stable. The edge-guided gradient loss can be calculated by:

$$\mathcal{L}_{grad} = \frac{1}{N_g} \sum_{n=1}^{N_g} ||(\beta_1 G_0 [P_n] + \beta_0) - G_S [P_n]||_1 \,, \tag{7}$$

where β_1 and β_0 are the scale and shift coefficients similar to Eq. 6. We use $[\cdot]$ to depict mask fetching operations, *i.e.*, extracting local area P_n from G_0 and G_S. With the edge-guided gradient loss, SDDR predicts refined depth with meticulous edges and consistent structures.

Edge-based Fusion Loss. High-resolution feature F_H extracted from H brings finer details but could lead to inconsistency, while the low-resolution feature F_L from L can better maintain depth structures. $\mathcal{N}_r$ should primarily rely on F_L for consistent spatial structures within low-frequency $I - \Omega$, while it should preferentially fuse F_H for edges and details in high-frequency areas Ω. The fusion of F_L and F_H noticeably influence the refined depth. However, prior arts [14, 3, 25] adopt manually-annotated Ω regions as fixed masks or even omit Ω as the whole image, leading to inconsistency and blurring. To this end, we implement Ω as a learnable soft mask, with quantile sampling strategy to guide the adaptive fusion of F_L and F_H. The fusion process is expressed by:

$$F = (1 - \Omega) \odot F_L + \Omega \odot F_H \,, \tag{8}$$

where $\odot$ refers to the Hadamard product. Ω is the learnable mask ranging from zero to one. Larger values in Ω showcases higher frequency with denser edges, requiring more detailed information from the high-resolution feature F_H. Thus, Ω can naturally serve as the fusion weight of F_L and F_H.

To be specific, we denote the lower quantile of G_S as t_a, *i.e.*, $P(X < t_a | X \in G_S) = a$. $\{G_S < t_a\}$ indicates flat areas with low gradient magnitude, while $\{G_S > t_{1-a}\}$ represents high-frequency

Predictor	Method	Middlebury2021			Multiscopic			Hypersim		
		δ_1↑	REL↓	ORD↓	δ_1↑	REL↓	ORD↓	δ_1↑	REL↓	ORD↓
MiDaS	MiDaS [30]	0.868	0.117	0.384	0.839	0.130	0.292	0.781	0.169	0.344
	Kim *et al.* [14]	0.864	0.120	0.377	0.839	0.130	0.293	0.778	0.175	0.344
	Graph-GDSR [4]	0.865	0.121	0.380	0.839	0.130	0.292	0.781	0.169	0.345
	GBDF [3]	0.871	0.115	0.305	0.841	0.129	0.289	0.787	0.168	0.338
	Ours	**0.879**	**0.112**	**0.299**	**0.852**	**0.122**	**0.267**	**0.791**	**0.166**	**0.318**
LeReS	LeReS [51]	0.847	0.123	0.326	0.863	0.111	0.272	0.853	0.123	0.279
	Kim *et al.* [14]	0.846	0.124	0.328	0.860	0.113	0.286	0.850	0.125	0.286
	Graph-GDSR [4]	0.847	0.124	0.327	0.862	0.111	0.273	0.852	0.123	0.281
	GBDF [3]	0.852	0.122	0.316	0.865	0.110	0.270	0.857	0.121	0.273
	Ours	**0.862**	**0.120**	**0.305**	**0.870**	**0.108**	**0.259**	**0.862**	**0.120**	**0.273**
ZoeDepth	ZoeDepth [1]	0.900	0.104	0.225	0.896	0.097	0.205	0.927	0.088	0.198
	Kim *et al.* [14]	0.896	0.107	0.228	0.890	0.099	0.204	0.923	0.091	0.204
	Graph-GDSR [4]	0.901	0.103	0.226	0.895	0.096	0.208	0.926	0.089	0.199
	GBDF [3]	0.899	0.105	0.226	0.897	0.096	0.207	0.925	0.089	0.199
	Ours	**0.905**	**0.100**	**0.218**	**0.904**	**0.092**	**0.199**	**0.930**	**0.086**	**0.191**

Table 1: **Comparisons with one-stage methods.** As prior arts [14, 4, 3], we conduct evaluations with different depth predictors [30, 51, 1]. For each predictor, we report the initial metrics and results of refinement methods. Best performances with each depth predictors [30, 51, 1] are in boldface.

regions. Ω should be larger in those high-frequency areas $\{G_S > t_{1-a}\}$ and smaller in the flat regions $\{G_S < t_a\}$. This suggests that G_S and Ω should be synchronized with similar data distribution. Thus, if we define the lower quantile of Ω as T_a, *i.e.*, $P(X < T_a | X \in \Omega) = a$, an arbitrary pixel $i \in \{G_S < t_a\}$ in flat regions should also belong to $\{\Omega < T_a\}$ with a lower weight for F_H, while the pixel $i \in \{G_S > t_{1-a}\}$ in high-frequency areas should be contained in $\{\Omega > T_{1-a}\}$ for more detailed information. The edge-based fusion loss can be depicted as follows:

$$\mathcal{L}_{fusion} = \frac{1}{N_w N_p} \sum_{n=1}^{N_w} \sum_{i=1}^{N_p} \begin{cases} \max(0, \Omega_i - T_{n*a}), & i \in \{G_S < t_{n*a}\}\,, \\ \max(0, T_{1-n*a} - \Omega_i), & i \in \{G_S > t_{1-n*a}\}\,, \end{cases} \tag{9}$$

where N_p is the pixel number. We supervise the distribution of Ω with lower quantiles T_{n*a} and $T_{1-n*a}, n \in \{1, 2, \cdots, N_w\}$. Therefore, pixels with larger deviations between G_S and Ω will be penalized more heavily. Taking the worst case as an example, if $i \in \{G_S < t_{N_w*a}\}$ but $i \notin \{\Omega < T_{N_w*a}\}$, the error for the pixel will be accumulated for N_w times from a to $N_w * a$. $\mathcal{L}_{fusion}$ enforces SDDR with consistent structures (low ϵ_{cons} noise) in $I - \Omega$ and accurate edges (low ϵ_{edge} noise) in Ω. The visualizations of quantile-sampled G_S and Ω are presented in Fig. 4.

Finally, combining $\mathcal{L}_{grad}$ and $\mathcal{L}_{fusion}$ as edge-based guidance for self-distillation, the overall loss $\mathcal{L}$ for training $\mathcal{N}_r$ is calculated as Eq. 10. $\mathcal{L}_{gt}$ supervises the discrepancy between D_0 and ground truth D^*_{gt} with affinity-invariant loss [30, 29]. See Appendix A for implementation details of SDDR.

$$\mathcal{L} = \mathcal{L}_{gt} + \lambda_1 \mathcal{L}_{grad} + \lambda_2 \mathcal{L}_{fusion}\,. \tag{10}$$

4 Experiments

To prove the efficacy of Self-distilled Depth Refinement (SDDR) framework, we conduct extensive experiments on five benchmarks [34, 52, 32, 15, 40] for indoor and outdoor, synthetic and real-world.

Experiments and Datasets. Firstly, we follow prior arts [3, 25, 14] to conduct zero-shot evaluations on Middlebury2021 [34], Multiscopic [52], and Hypersim [32]. To showcase our superior generalizability, we compare different methods on DIML [15] and DIODE [40] with diverse natural scenes. Moreover, we prove the higher efficiency of SDDR and undertake ablations on our specific designs.

Evaluation Metrics. Evaluations of depth accuracy and edge quality are necessary for depth refinement models. For edge quality, we adopt the ORD and D^{3}R metrics following Boost [25]. For depth accuracy, we adopt the widely-used REL and δ_i $(i = 1, 2, 3)$. See Appendix B for details.

4.1 Comparisons with Other Depth Refinement Approaches

Comparisons with One-stage Methods. For fair comparisons, we evaluate one-stage [14, 4, 3] and two-stage tile-based [25, 21] approaches separately. The one-stage methods predict refined depth

Predictor	Method	Middlebury2021			Multiscopic			Hypersim		
		δ_1↑	REL↓	ORD↓	δ_1↑	REL↓	ORD↓	δ_1↑	REL↓	ORD↓
MiDaS	MiDaS [30]	0.868	0.117	0.384	0.839	0.130	0.292	0.781	0.169	0.344
	Boost [25]	0.870	0.118	0.351	0.845	0.126	0.282	0.794	0.161	0.332
	Ours	**0.871**	**0.115**	**0.303**	**0.858**	**0.120**	**0.263**	**0.799**	**0.154**	**0.322**
LeReS	LeReS [51]	0.847	0.123	0.326	0.863	0.111	0.272	0.853	0.123	0.279
	Boost [25]	0.844	0.131	0.325	0.860	0.112	0.278	**0.865**	**0.118**	0.272
	Ours	**0.861**	**0.123**	**0.309**	**0.870**	**0.109**	**0.268**	0.858	0.123	**0.271**
ZoeDepth	ZoeDepth [1]	0.900	0.104	0.225	0.896	0.097	0.205	0.927	0.088	0.198
	Boost [25]	0.911	0.099	0.210	**0.910**	0.094	0.197	0.926	0.089	0.193
	PatchFusion [21]	0.887	0.102	0.211	0.908	0.095	0.212	0.881	0.116	0.258
	Ours	**0.913**	**0.096**	**0.202**	0.908	**0.091**	**0.197**	**0.933**	**0.083**	**0.189**

Table 2: **Comparisons with two-stage methods.** PatchFusion [21] only adopts ZoeDepth [1] as the fixed baseline, while other approaches are pluggable for different depth predictors [30, 51, 1].

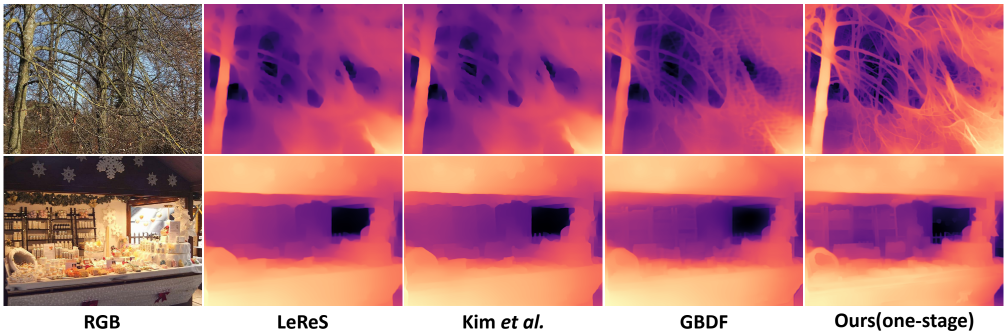


Figure 5: **Qualitative comparisons of one-stage methods on natural scenes.** LeReS [51] is used as the depth predictor. SDDR predicts sharper depth edges and more meticulous details than prior arts [3, 14], *e.g.*, fine-grained predictions of intricate branches. Better viewed when zoomed in.

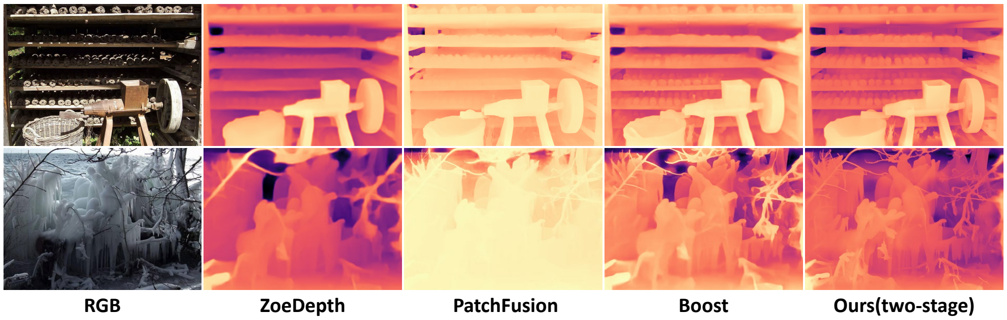


Figure 6: **Qualitative comparisons of two-stage methods on natural scenes.** ZoeDepth [1] is adopted as the depth predictor. The SDDR with coarse-to-fine edge refinement can predict more accurate depth edges and more consistent spatial structures than the tile-based methods [21, 25].

based on the whole image. SDDR conducts one-stage refinement without the coarse-to-fine manner during inference. Comparisons on Middlebury2021 [34], Multiscopic [52], and Hypersim [32] are shown in Table 1. As prior arts [14, 4, 3], we use three depth predictors MiDaS [30], LeReS [51], and ZoeDepth [1]. Regardless of which depth predictor is adopted, SDDR outperforms the previous one-stage methods [14, 4, 3] in depth accuracy and edge quality on the three datasets [34, 52, 32]. For instance, our method shows 6.6% and 20.7% improvements over Kim *et al.* [14] for REL and ORD with MiDaS [30] on Middlebury2021 [34], showing the efficacy of our self-distillation paradigm.

Comparisons with Two-stage Tile-based Methods. Two-stage tile-based methods [25, 21] conduct local refinement on numerous patches based on the global refined depth. SDDR moves away from the tile-based manner and utilizes coarse-to-fine edge refinement to further improve edges and details. As in Table 2, SDDR with the coarse-to-fine manner shows obvious advantages. For example, compared with the recent advanced PatchFusion [21], SDDR achieves 5.2% and 26.7% improvements for δ_1 and ORD with ZoeDepth [1] on Hypersim [32]. To be mentioned, PatchFusion [21] uses ZoeDepth [1] as the fixed baseline, whereas SDDR is readily pluggable for various depth predictors [30, 51, 1].

Generalization Capability on Natural Scenes. We prove the superior generalization capability of SDDR. In this experiment, we adopt LeReS [51] as the depth predictor. DIML [15] and DIODE [40]

Method	DIML				DIODE			
	$\delta_1\uparrow$	REL↓	ORD↓	D^{3}R↓	$\delta_1\uparrow$	REL↓	ORD↓	D^{3}R↓
LeReS [51]	0.902	0.101	0.242	0.284	0.892	0.105	0.324	0.685
Kim *et al.* [14]	0.902	0.100	0.243	0.301	0.889	0.105	0.325	0.713
Graph-GDSR [4]	0.901	0.101	0.243	0.300	0.890	0.104	0.326	0.690
GBDF [3]	0.906	0.100	0.239	0.267	0.894	0.105	0.322	0.673
Boost [25]	0.897	0.108	0.274	0.438	0.892	0.105	0.343	0.640
Ours	**0.926**	**0.098**	**0.221**	**0.220**	**0.900**	**0.098**	**0.293**	**0.637**

Table 3: **Comparisons of model generalizability.** We conduct zero-shot evaluations on DIML [15] and DIODE [40] datasets with diverse in-the-wild scenarios to compare the generalization capability. We adopt LeReS [51] as the depth predictor for all the compared methods in this experiment.

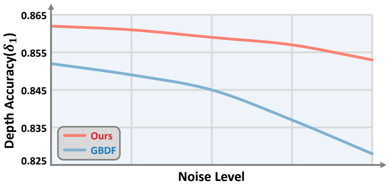


Figure 7: **Robustness against noises.** X-axis shows noise level of $\epsilon_{\text{cons}} + \epsilon_{\text{edges}}$. With higher noises, our SDDR is more robust with less performance degradation than the prior GBDF [3].

Method	$\delta_1\uparrow$	REL↓	ORD↓	D^{3}R↓
$S = 0$	0.859	0.125	0.313	0.235
$S = 1$	0.860	0.122	0.309	0.223
$S = 2$	0.860	0.120	0.307	0.219
$S = 3$	**0.862**	**0.120**	**0.305**	**0.216**

(a) Coarse-to-fine Edge Refinement

$\mathcal{L}_{gt}$	$\mathcal{L}_{grad}$	$\mathcal{L}_{fusion}$	$\delta_1\uparrow$	REL↓	ORD↓	D^{3}R↓
✓			0.854	0.124	0.313	0.240
✓	✓		0.858	0.122	0.307	0.220
✓		✓	0.859	0.120	0.311	0.229
✓	✓	✓	**0.862**	**0.120**	**0.305**	**0.216**

(b) Edge-based Guidance

Method	Training Data	$\delta_1\uparrow$	REL↓	ORD↓	D^{3}R↓
GBDF [3]	HRWSI [49]	0.852	0.122	0.316	0.258
Ours	HRWSI [49]	**0.860**	**0.121**	**0.309**	**0.222**

(c) Effectiveness

Method	$\delta_1\uparrow$	REL↓	ORD↓	D^{3}R↓
GBDF [3]	0.852	0.122	0.316	0.258
GBDF ($w/$ G_S)	**0.858**	**0.122**	**0.307**	**0.230**

(d) Transferability

Table 4: **Ablation Study.** All ablations are on Middlebury2021 [34] with depth predictor LeReS [51].

datasets are used for zero-shot evaluations, considering their diverse in-the-wild indoor and outdoor scenarios. As in Table 3, SDDR shows at least 5.7% and 9.0% improvements for REL and ORD on DIODE [40]. On DIML [15] dataset, our approach improves D^{3}R, ORD, and δ_1 by over 17.6%, 7.5%, and 2.0%. The convincing performance proves our strong robustness and generalizability, indicating the efficacy of our noisy Poisson fusion modeling and self-distilled training paradigm.

Qualitative Comparisons. We present visual comparisons of one-stage methods [14, 3] on natural scenes in Fig. 5. With our low-noise depth edge representation and edge-based guidance, SDDR predicts sharper depth edges and details, *e.g.*, the fine-grained predictions of intricate branches.

The visual results of two-stage approaches [25, 21] are shown in Fig. 6. Due to the excessive fusion of detailed information, tile-based methods [25, 21] produce structure disruption, depth inconsistency, or even noticeable artifacts, *e.g.*, disrupted and fuzzy structures of the snow-covered branches. By contrast, SDDR can predict more accurate depth edges and more consistent spatial structures.

Robustness against noises. As in Fig. 7, we evaluate SDDR and GBDF [3] with different levels of input noises. As the noise level increases, our method presents less degradation. The stronger robustness against the ϵ_{cons} and ϵ_{edges} noises is the essential reason for all our superior performance.

Model Efficiency. SDDR achieves higher efficiency. Two-stage tile-based methods [25, 21] rely on complex fusion of extensive patches with heavy computational overhead. Our coarse-to-fine manner noticeably reduces Flops per patch and patch numbers as in Fig. 1. For one-stage methods [4, 3, 14], SDDR adopts a more lightweight $\mathcal{N}_r$ with less parameters and faster inference speed over the previous GBDF [3] and Kim *et al.* [14]. See Appendix C.1 for detailed comparisons of model efficiency.

4.2 Ablation Studies

Coarse-to-fine Edge Refinement. In Table 4a, we adopt the coarse-to-fine manner with varied iterations. $S = 0$ represents one-stage inference. Coarse-to-fine refinement brings more fine-grained edge representations and refined depth. We set $S = 3$ for the SDDR with two-stage inference.

Edge-based Guidance. In Table 4b, we evaluate the effectiveness of edge-based guidance. $\mathcal{L}_{grad}$ focuses on consistent refinement of depth edges. $\mathcal{L}_{fusion}$ guides the adaptive feature fusion of low- and high-frequency information. With $\mathcal{L}_{gt}$ as the basic supervision of ground truth, adding $\mathcal{L}_{grad}$ and $\mathcal{L}_{fusion}$ improves D^{3}R by 10.0% and REL by 3.2%, showing the efficacy of edge-based guidance.

Effectiveness of SDDR Framework. As in Table 4c, we train SDDR with the same HRWSI [49] as GBDF [3] for fair comparison. Without the combined training data in Appendix B.1, SDDR still improves D^{3}R and ORD by 13.9% and 2.2% over GBDF [3], proving our superiority convincingly.

Transferability. We hope our depth edge representation G_S can be applicable to other depth refinement models. Therefore, in Table 4d, we directly train GBDF [3] combining the depth edge representation produced by the trained SDDR. The depth accuracy and edge quality are improved over the original GBDF [3], indicating the transferability of G_S in training robust refinement models.

5 Conclusion

In this paper, we model the depth refinement task as a noisy Poisson fusion problem. To enhance the robustness against local inconsistency and edge deformation noise, we propose Self-distilled Depth Refinement (SDDR) framework. With the low-noise depth edge representation and guidance, SDDR achieves both consistent spatial structures and meticulous depth edges. Experiments showcase our stronger generalizability and higher efficiency over prior arts. The SDDR provides a new perspective for depth refinement in future works. Limitations and broader impact are discussed in Appendix A.5.

Acknowledgement This work is supported by the National Natural Science Foundation of China under Grant No. 62406120.

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 1, 2, 3, 7, 8, 20, 21

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020. 2, 3

[3] Yaqiao Dai, Renjiao Yi, Chenyang Zhu, Hongjun He, and Kai Xu. Multi-resolution monocular depth map fusion by self-supervised gradient-based composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 488–496, 2023. 1, 2, 3, 4, 6, 7, 8, 9, 10, 15, 16, 18, 19, 20, 21

[4] Riccardo De Lutio, Alexander Becker, Stefano D'Aronco, Stefania Russo, Jan D Wegner, and Konrad Schindler. Learning graph regularisation for guided super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1979–1988, 2022. 1, 7, 8, 9, 19, 20

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 3, 4

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, volume 27, 2014. 2

[7] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2482–2491, 2020. 2, 3

[8] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979. 6, 15

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012. 3

[10] Yiwen Hua, Puneet Kohli, Pritish Uplavikar, Anand Ravi, Saravana Gunaseelan, Jason Orozco, and Edward Li. Holopix50k: A large-scale in-the-wild stereo image dataset. *arXiv preprint arXiv:2003.11172*, 2020. 2, 18

[11] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2821–2830, 2018. 2, 3, 4, 18

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017. 18

[13] John Kessenich, Graham Sellers, and Dave Shreiner. *OpenGL Programming Guide: The official guide to learning OpenGL, version 4.5 with SPIR-V*. Addison-Wesley Professional, 2016. 1

[14] Soo Ye Kim, Jianming Zhang, Simon Niklaus, Yifei Fan, Simon Chen, Zhe Lin, and Munchurl Kim. Layered depth refinement with mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3855–3865, 2022. 1, 3, 4, 6, 7, 8, 9, 14, 16, 19, 20, 21

[15] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE Transactions on Image Processing*, 27(8):4131–4144, 2018. 2, 7, 8, 9, 18, 20, 21

[16] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of cnn-based single-image depth estimation methods. In Laura Leal-Taixé and Stefan Roth, editors, *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, volume 11131, pages 331–348. Springer, 2018. 18

[17] Guchong Li. An iggm-based poisson multi-bernoulli filter and its application to distributed multisensor fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 58(4):3666–3677, 2022. 3

[18] Jiaqi Li, Yiran Wang, Zihao Huang, Jinghong Zheng, Ke Xian, Zhiguo Cao, and Jianming Zhang. Diffusion-augmented depth prediction with sparse annotations. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2865–2876, 2023. 3

[19] Jing Li, Hongtao Huo, Chenhong Sui, Chenchen Jiang, and Chang Li. Poisson reconstruction-based fusion of infrared and visible images via saliency detection. *IEEE Access*, 7:20676–20688, 2019. 3

[20] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 4

[21] Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. *arXiv preprint arXiv:2312.02284*, 2023. 1, 2, 3, 4, 5, 7, 8, 9, 14, 19, 20, 21

[22] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1925–1934, 2017. 16

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 16, 17

[24] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 1

[25] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9685–9694, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 16, 18, 19, 20, 21

[26] Antigoni Panagiotopoulou and Vassilis Anastassopoulos. Super-resolution image reconstruction techniques: Trade-offs between the data-fidelity and regularization terms. *Information Fusion*, 13(3):185–195, 2012. 3

[27] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16283–16292, 2022. 1

[28] Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. Mpib: An mpi-based bokeh rendering framework for realistic partial occlusion effects. In *European Conference on Computer Vision (ECCV)*, pages 590–607. Springer, 2022. 1

[29] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 4, 7

[30] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(03):1623–1637, 2020. 1, 2, 3, 4, 7, 8, 20

[31] Haoyu Ren, Amin Kheradmand, Mostafa El-Khamy, Shuangquan Wang, Dongwoon Bai, and Jungwon Lee. Real-world super-resolution using generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 436–437, 2020. 3

[32] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. 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. 1, 2, 3, 4, 7, 8, 18

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1

[34] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pages 31–42. Springer, 2014. 2, 7, 8, 9, 17, 19, 20, 21

[35] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 2, 3, 4

[36] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580. IEEE, 2012. 2, 3

[37] Zhanghao Sun, Wei Ye, Jinhui Xiong, Gyeongmin Choe, Jialiang Wang, Shuochen Su, and Rakesh Ranjan. Consistent direct time-of-flight video depth super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5075–5085, 2023. 1

[38] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 839–846. IEEE, 1998. 5

[39] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8942–8952, 2021. 2, 3, 4, 18

[40] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, 2019. 2, 7, 8, 9, 18, 20

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017. 16

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 5998–6008, 2017. 16

[43] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *IEEE International Conference on 3D Vision (3DV)*, pages 348–357. IEEE, 2019. 2, 3, 18

[44] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE Computer Society, 2021. 2, 3, 18

[45] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 2, 3, 18

[46] Yiran Wang, Min Shi, Jiaqi Li, Chaoyi Hong, Zihao Huang, Juewen Peng, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Nvds^{+}: Towards efficient and versatile neural stabilizer for video depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2024. 4

[47] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9466–9476, 2023. 2, 3, 4, 18

[48] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 311–320, 2018. 2

[49] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 608–617, 2020. 2, 3, 4, 9, 18, 19, 20, 21

[50] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. 16

[51] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. 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. 1, 2, 3, 4, 7, 8, 9, 19, 20, 21

[52] Weihao Yuan, Yazhan Zhang, Bingkun Wu, Siyu Zhu, Ping Tan, Michael Yu Wang, and Qifeng Chen. Stereo matching by self-supervision of multiscopic vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5702–5709. IEEE, 2021. 2, 7, 8, 17

[53] Junrui Zhang, Jiaqi Li, Yachuan Huang, Yiran Wang, Jinghong Zheng, Liao Shen, and Zhiguo Cao. Towards robust monocular depth estimation in non-lambertian surfaces. *arXiv preprint arXiv:2408.06083*, 2024. 3

[54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023. 1

[55] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 2, 3, 18

[56] Changzhong Zou and Youshen Xia. Bayesian dictionary learning for hyperspectral image super resolution in mixed poisson–gaussian noise. *Signal Processing: Image Communication*, 60:29–41, 2018. 3

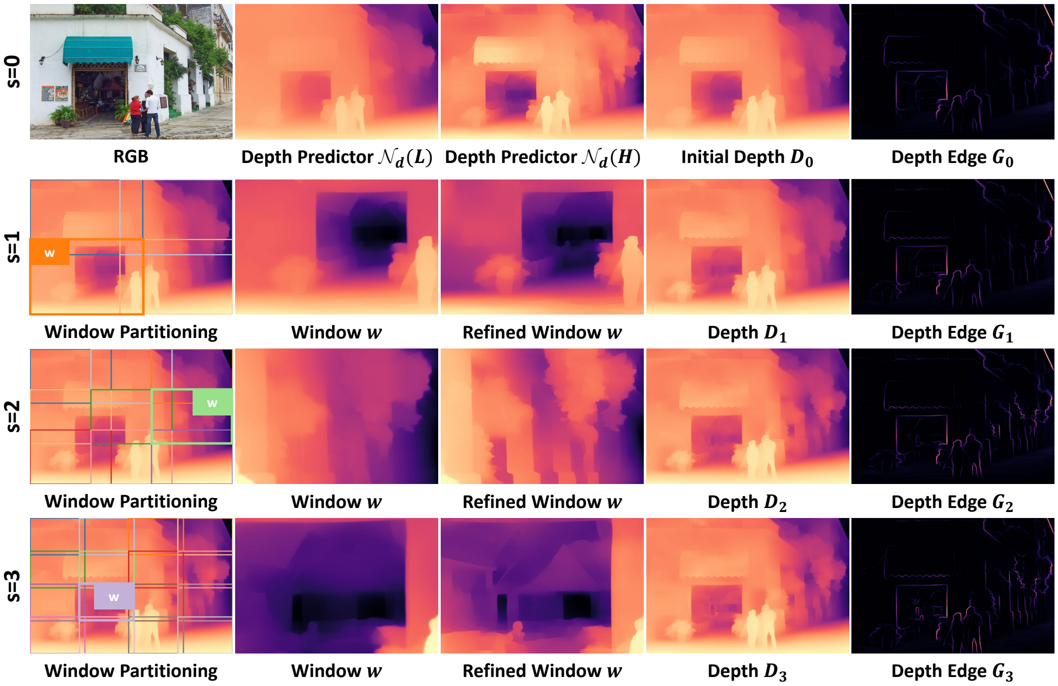


Figure 8: **Visualizations of coarse-to-fine edge refinement.** We present coarse-to-fine results of steps $s = 0, 1, 2, 3$. For $s = 0$, we showcase the low- and high-resolution predictions $\mathcal{N}_d(L)$ and $\mathcal{N}_d(H)$ of the depth predictor, along with the initial refined depth D_0 and edge representation G_0. For $s = 1, 2, 3$, we present the window partitioning on the previous D_{s-1}, the previous depth D^w_{s-1} on a certain window w, refined depth D^w_s on the window w, refined depth D_s of the whole image, and the depth edge representation G_s generated on the current step.

A More Details on SDDR Framework

A.1 Depth Edge Representation

Coarse-to-fine Edge Refinement. In Sec. 3.2, line 169 of main paper, we propose the coarse-to-fine edge refinement to generate accurate and fine-grained depth edge representation G_S. Here, we provide visualizations of the refinement process in Fig. 8. For the initial global refinement stage $s = 0$, we showcase the results of the depth predictor at low and high inference resolutions, *i.e.*, $\mathcal{N}_d(L)$ and $\mathcal{N}_d(H)$. Our refined depth D_0 presents both depth consistency and details. For $s = 1, 2, 3$, the refined depth maps and edge representations are noticeably improved with finer edges and details. The final depth edge representation G_S ($S = 3$) with lower local inconsistency noise and edge deformation noise is utilized as pseudo-label for the self-distillation training process.

Adaptive Resolution Adjustment. Adaptive resolution adjustment is applied to the low and high-resolution input L and H. We denote the resolutions of L and H as l and h, which play a crucial role in refined depth and need to be chosen carefully. Higher resolutions will bring finer details but could lead to inconsistent depth structures due to the limited receptive field of $\mathcal{N}_d$. Previous works [14, 25, 21] upscale images or patches to excessively high resolutions for more details, resulting in evident artifacts in their refined depth maps with higher levels of inconsistency noises ϵ_{cons}. On the other hand, if h is too low, edge and detailed information cannot be sufficiently preserved in $\mathcal{N}_d(H)$, leading to exacerbation of edge deformation noise ϵ_{edge} with blurred details in the refined depth. Such errors and artifacts are unacceptable in depth edge representations for training models. Therefore, we adaptively adjust resolutions l and h, considering both the density of depth edges and the training resolution of depth predictor $\mathcal{N}_d$.

For image window I^w_s, we generally set the low-resolution input L^w_s as the training resolution $\hat{r}$ of $\mathcal{N}_d$. If we denote the original resolution of I^w_s as r^w_s, SDDR adaptively adjusts the high resolution

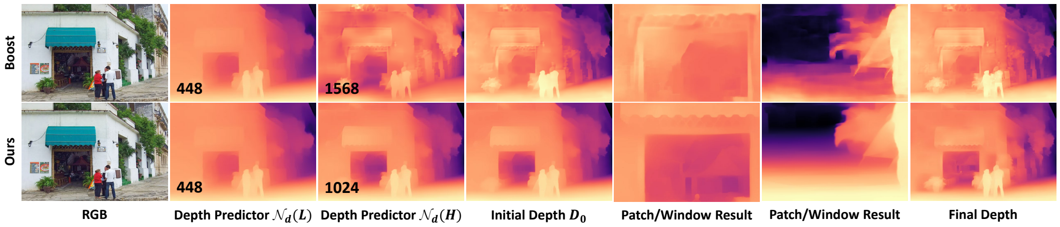

Boost
Ours
448
1568
448
1024
RGB
Depth Predictor $\mathcal{N}_d(\boldsymbol{L})$
Depth Predictor $\mathcal{N}_d(\boldsymbol{H})$
Initial Depth $\boldsymbol{D_0}$
Patch/Window Result
Patch/Window Result
Final Depth

Figure 9: **Adaptive resolution adjustment.** We compare the effects of inference resolutions with Boost [25]. The numbers in the corner of the second and third columns represent the chosen inference resolution. We relieve the artifacts in Boost [25] by adaptive resolution adjustment.

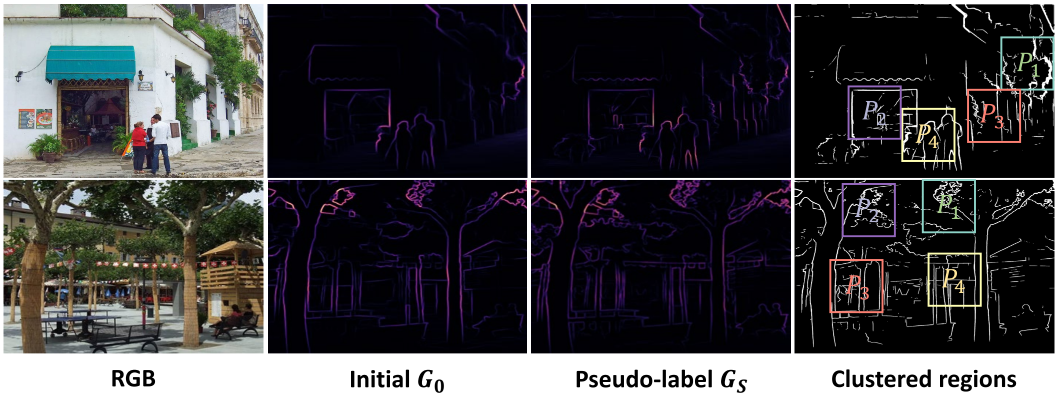


Figure 10: **Edge-guided gradient error.** $\mathcal{L}_{grad}$ focuses on high-frequency areas P_n extracted by clustering with more details. The flat regions are not constrained to preserve depth consistency.

h_s^w for the certain window as follows:

$$h_s^w = mean(\hat{r}, r_s^w) * \frac{mean(|\nabla \mathcal{N}_d(L_s^w)|)}{\alpha} * \frac{mean(|\nabla D_{s-1}^w|)}{mean(|\nabla D_{s-1}|)}\,, \tag{11}$$

where α is a priori parameter for depth predictor $\mathcal{N}_d$, averaging the gradient magnitude of the depth annotations on its sampled training data. The second term embodies adjustments according to depth edges. Assuming $mean(|\nabla \mathcal{N}_d(L_s^w)|) < \alpha$, it indicates that the current window area contains lower edge intensity or density than the training data of $\mathcal{N}_d$. In this case, we will appropriately decrease h_s^w from $mean(\hat{r}, r_s^w)$ to maintain the similar density of detailed information as the training stage of the depth predictor. The third term portrays adjustments based on the discrepancy of edge intensity between the window area and the whole image. To be mentioned, for the generation of the initial edge representation G_0, the third term is set to ineffective as one. L_0^w is equivalent to L with the whole image as the initial window w.

We present visual results with different resolutions to prove the effectiveness of our design. As shown in Fig. 9, considering the training data distribution and the edge density, the inference resolution is adaptively adjusted to a smaller one compared to Boost [25] (1024 versus 1568). In this way, our SDDR achieves better depth consistency and alleviates the artifacts produced by prior arts [3, 25].

A.2 Edge-based Guidance

Edge-guided Gradient Error. In line 192, Sec 3.3 of the main paper, we mention that we use clustering to obtain several high-frequency local regions to compute our edge-guided gradient loss. Here, we elaborate on the details. K-means clustering [8] is utilized to obtain the edge-dense areas. Specifically, we binarize the edge pseudo-label, setting the top 5% pixels to one and the rest to zero. Next, we employ k-means clustering on the binarized labels to get several edge-dense areas with the centroid value as one. The clustered areas are shown in the fourth column of the Fig. 10. Our edge-guided gradient loss supervises these high-frequency regions to improve depth details. The depth consistency in flat areas can be preserved without the constraints of depth edges.

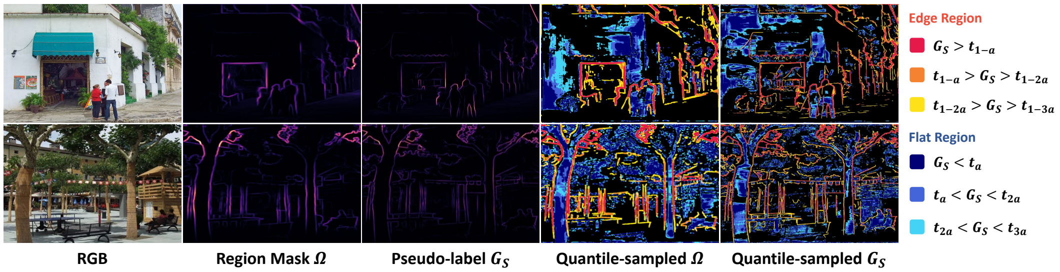


Figure 11: **Edge-based fusion error.** We present the region mask Ω and pseudo-label G_S before and after quantile sampling. Different colors on the right represent the range of pixel values. Guiding the Ω with G_S ensures that our model can predict balanced consistency and details by the simple one-stage inference. The use of Ω as a learnable soft mask achieves more fine-grained integration on the feature level, enhancing the accuracy of $\mathcal{N}_r$. This also leads to more accurate edge representation G_S in the iterative coarse-to-fine refinement process.

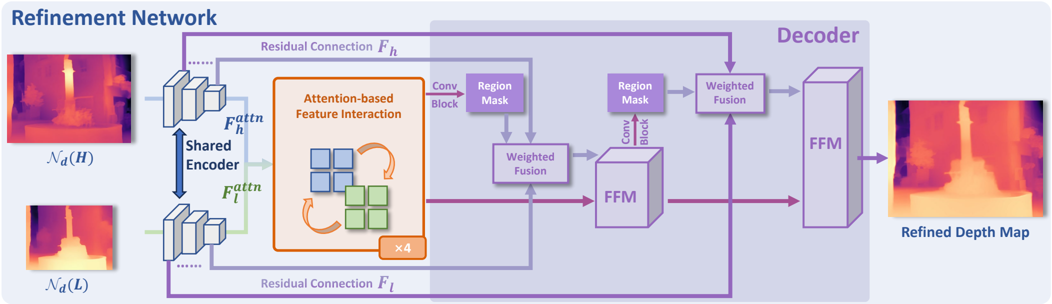


Figure 12: **Architecture of refinement network.** Some decoder layers are omitted for simplicity.

Edge-based Fusion Error. The proposed edge-based fusion loss aligns the data distribution of the learnable region mask Ω and the pseudo-label G_S by quantile sampling (Sec 3.3, line 205, main paper). Here, we provide additional visualizations for intuitive understanding. As shown in Fig. 11, we visualize the soft region mask Ω of high-frequency areas and the pseudo-label G_S with the same color map in the second and third columns. The regions highlighted in G_S with stronger depth edges and more detailed information naturally correspond to larger values in Ω to emphasize features from high-resolution inputs. We perform quantile sampling on Ω and G_S, as depicted in the fourth and fifth columns. The legends on the right indicate the percentile ranking of the pixel values in the whole image. Our edge-based fusion loss supervises that Ω and G_S have consistent distribution for each color. In this way, Ω tends to have smaller values in flat regions for more information from low-resolution input, while the opposite is true in high-frequency regions. This is advantageous for the model to balance the depth details and spatial structures.

A.3 Refinement Network

We provide the detailed model architecture of the refinement network $\mathcal{N}_r$. As shown in Fig. 12, the refinement network adopts the U-Net architecture similar to prior arts [25, 3, 14]. The depth maps from the depth predictor $\mathcal{N}_d$ predicted in different resolutions are up-sampled to a unified input size. A shared Mit-b0 [50] serves as the encoder to extract feature maps of different resolutions. The decoder gradually outputs the refined depth map with feature fusion modules (FFM) [22, 23] and skip connections. We make two technical improvements to the refinement network, including attention-based feature interaction and adaptive weight allocation.

Attention-based Feature Interaction. To predict refined depth maps in high resolution (*e.g.*, 2048 × 2048), prior arts [25, 3, 14] adopt a U-Net with numerous layers (*e.g.*, 10 layers or more) as the refinement network for sufficient receptive field. This leads to heavy computational overhead. In our case, we leverage the self-attention mechanism [41] to address this issue.

The features of low- and high-resolution inputs extracted by the encoder [50] are denoted as F_l^{attn} and F_h^{attn}. We stack F_l^{attn} and F_h^{attn} to obtain F^{in} for attention calculation. Positional embeddings [42]

PE_x, PE_y are added to F^{in} for the height and width dimensions. An additional PE_f is used to distinguish the low- and high-resolution inputs. The attention-based feature interaction process can be expressed as follows:

$$\begin{gathered} F^{in} = \text{Stack}(F_l^{attn}, F_h^{attn}) + PE_x + PE_y + PE_f \,, \\ K = W^k \cdot F^{in}, Q = W^q \cdot F^{in}, V = W^v \cdot F^{in}, \\ F^{out} = \text{Softmax}\Big(K^T Q/\sqrt{d}\Big) V + F^{in} \,. \end{gathered} \tag{12}$$

Four attention layers are included in $\mathcal{N}_r$. The interacted feature F^{out} is fed to the decoder to predict refined depth. Attention-based feature interaction achieves large receptive field with fewer layers, reducing model parameters and improving efficiency.

Adaptive Weight Allocation. The refinement network adopts adaptive weight allocation for the fusion of low- and high-resolution features with the learnable mask Ω. In each decoder layer, the feature go through a convolutional block to generate Ω with a single channel. The fused features F (line 212, main paper) and the feature from the previous layer are fused by the FFM module [23].

A.4 Noise Implementation.

For our local inconsistency noise, we segment the ideal depth D^* into regular patches of size 64×64, with an overlap of half the patch size. Considering the depth discontinuities on the edges, instead of applying a linear transformation to the entire patch, we extract the edges from D^* and apply a linear transformation to each connected domain to simulate the local depth inconsistency. For edge deformation noise, we first down-sample D^* to the inference resolution and then restore it to the original resolution. Subsequently, we optimize a certain number of Gaussian distributions around the edges of D^* to fit the edge deformation and blurring.

The local inconsistency noise and edge deformation noise can effectively model the degradation of network prediction results compared to ideal depth maps. An additional experiment on the Middlebury2021 [34] dataset also proves this point. We optimize the local inconsistency noise with the least squares method and $50{,}000$ position-constrained Gaussian distributions as edge deformation noise by gradient descent. The PSNR between the noisy depth $(D^* + \epsilon_{\text{cons}} + \epsilon_{\text{edge}})$ and model predicted depth D is over 40 dB, which indicates that the difference between D and $(D^* + \epsilon_{\text{cons}} + \epsilon_{\text{edge}})$ is very small. The result further demonstrates that the noises can accurately model depth prediction errors (Eq. 1, main paper), similar to the visualizations in Fig. 2 of the main text.

A.5 Broader Impacts and Limitations

Although SDDR works well in general, it still has limitations. For example, more advanced mechanisms and structures can be explored for the refinement network in future work. For inputs under conditions with specular surfaces, low light, or weak textures, the depth predictor tends to yield sub-optimal results. Although SDDR improves upon these results, the outcomes are still not perfect. Our approach exclusively utilizes publicly available datasets during the training process, thereby having no broad societal impact, not involving AI ethics, and not involving any privacy-sensitive data.

B Detailed Experimental Settings

B.1 Datasets

Evaluation Datasets. We use five different benchmarks with diverse scenarios for comparisons. The descriptions of our evaluation datasets are as follows:

- **Middlebury2021** [34] comprises 48 RGB-D pairs from 24 real indoor scenes for evaluating stereo matching and depth refinement models. Each image in the dataset is annotated with dense 1920×1080 disparity maps. We use the whole set of Middlebury2021 [34] for testing.
- **Multiscopic** [52] includes a test set with 100 synthetically generated indoor scenes. Each scene consists of RGB images captured from 5 different viewpoints, along with corresponding disparity annotations. The resolution of images is 1280×1080. We adopt its official test set for testing.

- **Hypersim** [32] is a large-scale synthetic dataset. In our experiment, we follow the test set defined by GBDF [3] for fair comparison, utilizing tone-mapped 286 images generated by their released code. Evaluation is performed using the corresponding 1024×768 depth annotations.
- **DIML** [15] contains RGB-D frames from both Kinect v2 [55] and Zed stereo camera with different resolutions. We conduct the generalization evaluation using the official test set, which includes real indoor and outdoor scene images along with corresponding high-resolution depth annotations.
- **DIODE** [40] contains high-quality 1024×768 LiDAR-generated depth maps of both indoor and outdoor scenes. We use the whole validation set (771 images) for generalization testing.

Training Datasets. Our training data is sampled from diverse datasets, which can be categorized into synthetic and natural-scene datasets. The synthetic datasets consist of TartanAir [45], Irs [44], UnrealStereo4K [39] and MVS-Synth [11]. Among these, the resolutions of TartanAir [45] and Irs [44] are below 1080p, while MVS-Synth [11] and UnrealStereo4K [39] reach resolutions of 1080p and 4k, respectively. Irs [44] and MVS-Synth [11] contain limited types of scenes, whereas others include both indoor and outdoor scenes, some of which [45, 39] present challenging conditions like poor lighting. To enhance the generalization to natural scenes, we also sample from four high-resolution real-world datasets, Holopix50K [10], iBims-1 [16], WSVD [43], and VDW [47]. IBims-1 [16] contains a small number of indoor scenes but provides high-precision depth annotations from the capturing device. The remaining three datasets include large-scale diverse scenes, but their depth annotations, obtained from stereo images [12], lack ideal edge precision.

B.2 Training Recipe

We leverage diverse training data to achieve strong generalizability. For each epoch, we randomly choose 20,000 images from natural-scene data [10, 47, 43, 16] and 20,000 images from synthetic datasets [45, 44, 39, 11]. For each sample, we adopt similar data processing and augmentation as GBDF [3]. To enhance training stability, we first train $\mathcal{N}_r$ for one epoch only with $\mathcal{L}_{gt}$. In the next two epochs, we involve $\mathcal{L}_{grad}$ and $\mathcal{L}_{fusion}$ for self-distillation. The a and N_w in $\mathcal{L}_{fusion}$ are set to 0.02 and 4. The learning rate is $1e-4$. λ_1 and λ_2 in Eq. 10 are 0.5 and 0.1. All training and inference are conducted on a single NVIDIA A6000 GPU.

B.3 Evaluation Metrics

Depth Accuracy. M denotes numbers of pixels with valid depth annotations, while d_i and d_i^* are estimated and ground truth depth of pixel i. We adopt the widely-used depth metrics as follows:

- **Absolute relative error (Abs Rel):** $\frac{1}{|M|}\sum_{d\in M}|d-d^*|/d^*$;
- **Square relative error (Sq Rel):** $\frac{1}{|M|}\sum_{d\in M}\|d-d^*\|^2/d^*$
- **Root mean square error (RMSE):** $\sqrt{\frac{1}{|M|}\sum_{d\in M}\|d-d^*\|^2}$;
- **Mean absolute logarithmic error ($\log_{10}$):** $\frac{1}{|M|}\sum_{d\in M}|\log(d)-\log(d^*)|$;
- **Accuracy with threshold t:** Percentage of d_i such that $max(\frac{d_i}{d_i^*},\frac{d_i^*}{d_i})=\delta<t\in\left[1.25,1.25^2,1.25^3\right]$.

Edge Quality. For the edge quality, we follow prior arts [25, 3, 49] to employ the ordinal error (ORD) and depth discontinuity disagreement ratio (D^{3}R). The ORD metric is defined as:

$$
\begin{gathered}
ORD=\frac{1}{N}\sum_i\phi(p_{i,0}-p_{i,1}),\\
\phi(p_{i,0}-p_{i,1})=\begin{cases}\log\left(1+\exp\left(-l\left(p_{i,0}-p_{i,1}\right)\right)\right), & l\neq 0,\\ (p_{i,0}-p_{i,1})^2, & l=0,\end{cases}\\
l=\begin{cases}+1, & p_{i,0}^*/p_{i,1}^*\geq 1+\tau,\\ -1, & p_{i,0}^*/p_{i,1}^*\leq\frac{1}{1+\tau},\\ 0, & otherwise,\end{cases}
\end{gathered}
\tag{13}
$$

Method	FLOPs (G)	Params (M)	Time (s)
GBDF [3]	10.377	201.338	0.112
Kim *et al.* [14]	1138.342	61.371	0.128
Graph-GDSR [4]	397.355	32.533	0.832
Ours (one-stage)	16.733	16.763	0.035
Boost [25]	286.13 × 63	79.565	2.183
PatchFusion [21]	810.813 × 177	42.511	5.345
Ours (two-stage)	16.733 × 30	16.763	1.050

Table 5: **Model efficiency.** We evaluate FLOPs, model parameters, and inference time of different methods. The first four rows contain one-stage methods [3, 14, 4], while the last three rows are for two-stage approaches [25, 21]. FLOPs and inference time are tested on a 1024 × 1024 image with one NVIDIA RTX A6000 GPU. For the two-stage methods [25, 21], their FLOPs are reported by multiplying FLOPs per patch with the required patch numbers for processing the image.

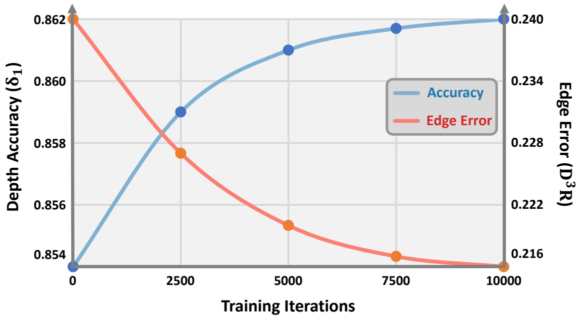


Figure 13: **Iterations for self-distillation.** We report the depth accuracy and edge error metrics of our SDDR model in the self-distillation training process.

Method	δ_1↑	REL↓	ORD↓	D^3R↓
Ours ($w/\ D_S$)	0.855	0.129	0.317	0.237
Ours ($w/\ G_S$)	**0.862**	**0.120**	**0.305**	**0.216**

Table 6: **Formats of Pseudo-labels.** We compare the self-distilled training with refined depth D_S and depth edge representation G_S as pseudo-labels. The experiment is conducted on Middlebury2021 [34] dataset with LeReS [51] as the depth predictor.

where $p_{i,0}$ and $p_{i,1}$ represent pairs of edge-guided sampling points. $p^*_{i,0}$ and $p^*_{i,1}$ are the ground truth values at corresponding positions. l is used to represent the relative ordinal relationship between pairs of points. ORD characterizes the quality of depth edges by sampling pairs of points near extracted edges using a ranking loss [49]. On the other hand, D^3R [25] uses the centers of super-pixels computed with the ground truth depth and compares neighboring super-pixel centroids across depth discontinuities. It directly focuses on the accuracy of depth boundaries.

C More Experimental Results

C.1 Model Efficiency Comparisons.

In line 277 of the main paper, we mention that our method achieves higher model efficiency than prior arts [4, 3, 14, 25, 21]. Here, we provide detailed comparisons of model efficiency in Table 5. For one-stage methods [4, 3, 14], SDDR adopts a more lightweight refinement network, reducing model parameters by 12.5 times than GBDF [3] and improving inference speeds by 3.6 times than Kim *et al.* [14]. Compared with two-stage tile-based methods [25, 21], our coarse-to-fine edge refinement reduces the Flops per patch by 50.6 times and the patch numbers by 5.9 times than PatchFusion [21].

C.2 More Quantitative and Qualitative Results

Training Iterations of Self-distillation We investigate the iteration numbers of self-distillation in Fig. 13. The iteration number of zero indicates the model after the training of the first epoch only with

Predictor	Method	Depth							Edge	
		Abs Rel↓	Sq Rel↓	RMSE↓	$\log_{10}$↓	δ_1↑	δ_2↑	δ_3↑	ORD↓	D^{3}R↓
MiDaS	MiDaS [30]	0.117	0.576	3.752	0.052	0.868	0.973	0.992	0.384	0.334
	Kim *et al.* [14]	0.120	0.562	**3.558**	0.053	0.864	0.973	0.994	0.377	0.382
	Graph-GDSR [4]	0.121	0.566	3.593	0.053	0.865	0.973	0.994	0.380	0.398
	GBDF [3]	0.115	0.561	3.685	0.052	0.871	0.973	0.993	0.305	0.237
	Ours	**0.112**	**0.545**	3.668	**0.050**	**0.879**	**0.979**	**0.994**	**0.299**	**0.220**
LeReS	LeReS [51]	0.123	0.464	3.040	0.052	0.847	0.969	0.992	0.326	0.359
	Kim *et al.* [14]	0.124	0.474	3.063	0.052	0.846	0.969	0.992	0.328	0.387
	Graph-GDSR [4]	0.124	0.467	3.052	0.052	0.847	0.969	0.992	0.327	0.373
	GBDF [3]	0.122	**0.444**	**2.963**	0.051	0.852	0.969	0.992	0.316	0.258
	Ours	**0.120**	0.452	2.985	**0.050**	**0.862**	**0.971**	**0.993**	**0.305**	**0.216**
Zoedepth	Zoedepth [1]	0.104	0.433	2.724	0.043	0.900	0.970	0.993	0.225	0.208
	Kim *et al.* [14]	0.107	0.469	2.766	0.044	0.896	0.970	0.992	0.228	0.243
	Graph-GDSR [4]	0.103	0.431	2.725	0.044	0.901	0.971	0.993	0.226	0.233
	GBDF [3]	0.105	0.430	2.732	0.044	0.899	0.970	0.993	0.226	0.200
	Ours	**0.100**	**0.406**	**2.674**	**0.042**	**0.905**	**0.973**	**0.994**	**0.218**	**0.187**

Table 7: **Comparisons with one-stage refinement approaches on Middlebury2021.**

Predictor	Method	Depth							Edge	
		Abs Rel↓	Sq Rel↓	RMSE↓	$\log_{10}$↓	δ_1↑	δ_2↑	δ_3↑	ORD↓	D^{3}R↓
MiDaS	MiDaS [30]	0.117	0.576	3.752	0.052	0.868	0.973	0.992	0.384	0.334
	Boost [25]	0.118	**0.544**	3.758	0.053	0.870	**0.979**	**0.997**	0.351	0.257
	Ours	**0.115**	0.563	**3.710**	**0.052**	**0.871**	0.973	0.993	**0.303**	**0.248**
LeReS	LeReS [51]	0.123	0.464	3.040	0.052	0.847	0.969	**0.992**	0.326	0.359
	Boost [25]	0.131	0.487	3.014	0.054	0.844	0.960	0.989	0.325	**0.202**
	Ours	**0.123**	**0.459**	**3.005**	**0.052**	**0.861**	**0.969**	0.991	**0.309**	0.214
Zoedepth	Zoedepth [1]	0.104	0.433	2.724	0.043	0.900	0.970	0.993	0.225	0.208
	Patchfusion [21]	0.102	0.385	**2.406**	0.042	0.887	0.977	**0.997**	0.211	0.139
	Boost [25]	0.099	**0.349**	2.502	0.042	0.911	**0.979**	0.995	0.210	0.140
	Ours	**0.096**	0.350	2.432	**0.041**	**0.913**	0.977	0.995	**0.202**	**0.125**

Table 8: **Comparisons with two-stage tile-based methods on Middlebury2021.** PatchFusion [21] can only adopt ZoeDepth [1] as the fixed baseline, while other approaches are reconfigurable and pluggable for different depth predictors [1, 51, 30].

Dataset	Method	Depth							Edge	
		Abs Rel↓	Sq Rel↓	RMSE↓	$\log_{10}$↓	δ_1↑	δ_2↑	δ_3↑	ORD↓	D^{3}R↓
DIML	LeReS [51]	0.101	45.607	325.191	0.043	0.902	0.990	0.998	0.242	0.284
	Kim *et al.* [14]	0.100	45.554	325.155	0.042	0.902	0.990	0.998	0.243	0.301
	Graph-GDSR [4]	0.101	45.993	326.320	0.043	0.901	0.989	0.998	0.243	0.300
	GBDF [3]	0.100	44.038	**318.874**	0.042	0.906	**0.991**	0.998	0.239	0.267
	Boost [25]	0.108	50.923	341.992	0.046	0.897	0.987	0.998	0.274	0.438
	Ours	**0.098**	**41.328**	320.193	**0.042**	**0.926**	0.990	**0.998**	**0.221**	**0.230**
DIODE	LeReS [51]	0.105	1.642	9.856	0.041	0.892	0.968	**0.989**	0.324	0.685
	Kim *et al.* [14]	0.105	1.654	9.888	0.044	0.889	0.964	0.987	0.325	0.713
	Graph-GDSR [4]	0.104	1.626	9.876	0.044	0.890	0.967	0.988	0.326	0.690
	GBDF [3]	0.105	1.625	9.770	**0.041**	0.894	0.968	0.990	0.322	0.673
	Boost [25]	0.105	1.612	9.879	0.044	0.892	0.966	0.987	0.343	0.640
	Ours	**0.098**	**1.529**	**9.549**	0.042	**0.900**	**0.968**	0.988	**0.293**	**0.637**

Table 9: **Comparisons with previous refinement approaches on DIML and DIODE.**

ground truth for supervision, *i.e.*, before self-distillation. Clearly, with the proposed self-distillation paradigm, both the depth accuracy and edge quality are improved until convergence.

Formats of Pseudo-labels We compare the refined depth D_S and the proposed depth edge representation G_S as pseudo-labels. Using the accurate and meticulous depth D_S could be a straightforward idea. However, with depth maps as the supervision, the model cannot precisely focus on improving edges and details. Thus, G_S achieves stronger efficacy than D_S, proving the necessity of our designs.

Quantitative Comparisons. In the main paper, only δ_1, REL, ORD, and D^{3}R are reported. Here, we present the additional metrics of all the compared methods [14, 4, 3, 25, 21] on Middlebury2021 [34], DIML [15], and DIODE [40] datasets in Table 7, Table 8, and Table 9. Our method outperforms previous approaches on most evaluation metrics, showing the effectiveness of our SDDR framework.

Qualitative Comparison We provide more qualitative comparisons with one-stage [14, 3] and two-stage [21, 25] methods in Fig. 14 and Fig. 15. These visual results further demonstrate the excellent performance and generalization capability of SDDR on diverse scenes [34, 15, 49].

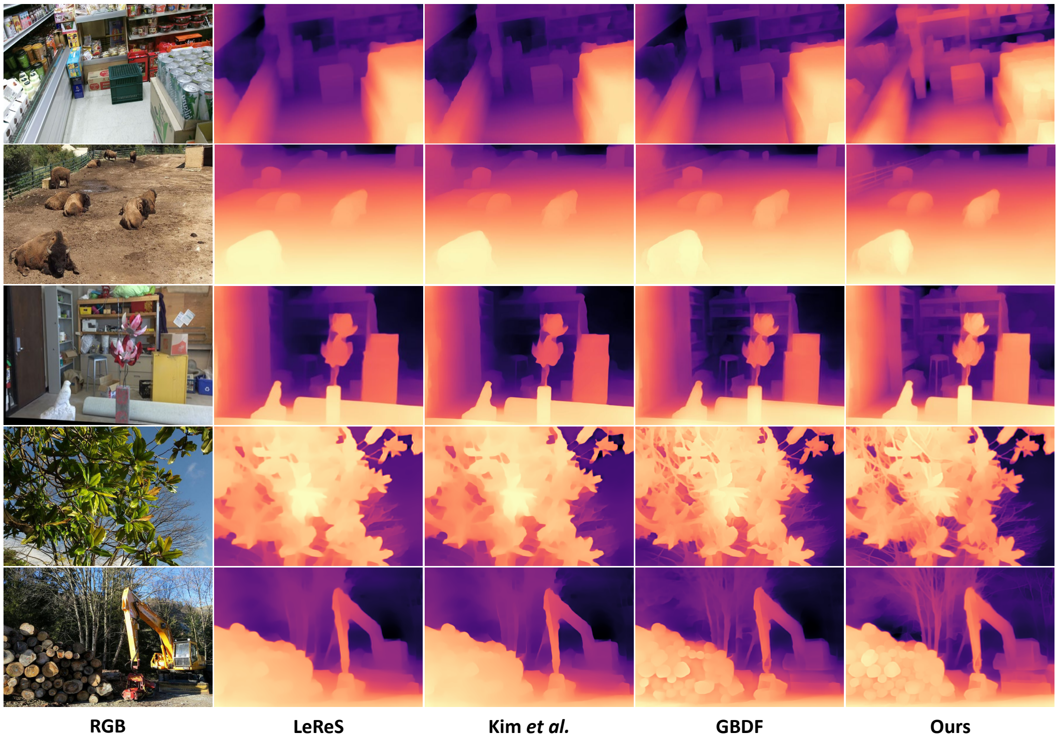


Figure 14: **Qualitative comparisond with one-stage methods [14, 3] on various datasets [15, 49, 34].** We adopt LeReS [51] as the depth predictor. Better viewed when zoomed in.

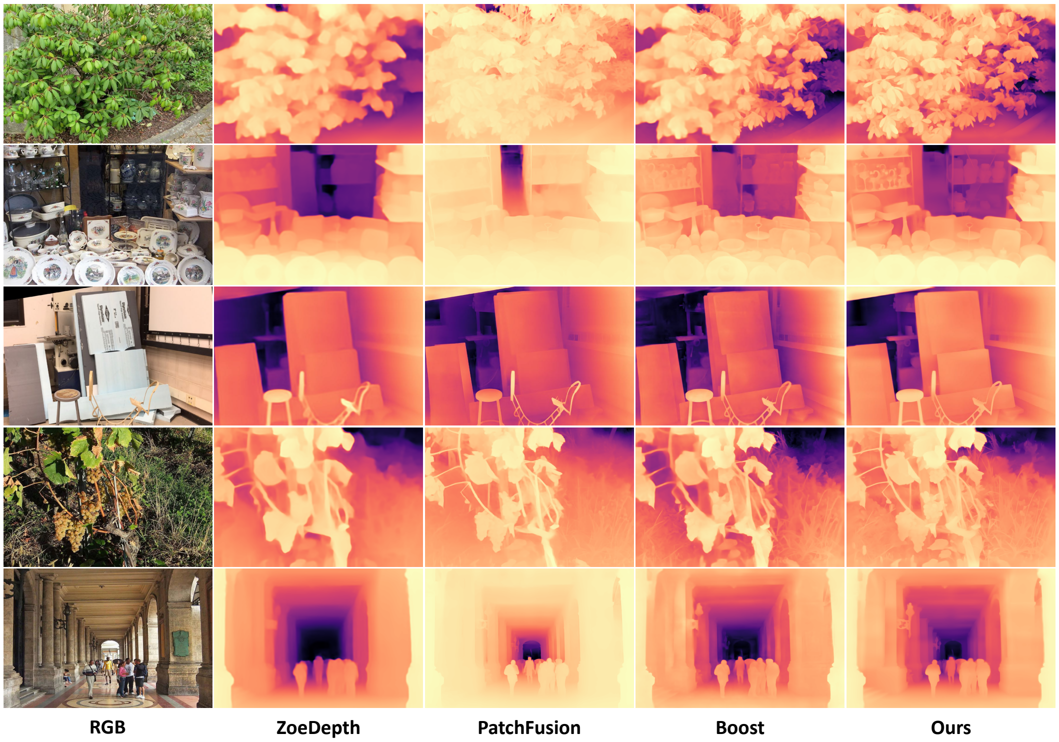


Figure 15: **Qualitative comparisons with two-stage methods [21, 25] on various datasets [15, 49, 34].** We adopt Zoedepth [1] as the depth predictor. Better viewed when zoomed in.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper accurately conveys the contributions and scope of this work in the abstract and introduction sections, and provides a bullet-point summary at the end.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the method in Appendix A.5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experiments presented in this paper are reproducible. We will release the code and model following the acceptance of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the code and model after the acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide a detailed description of our experimental setup and results in Sec. 4 of the main paper, as well as in Appendice B and C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments are stable across multiple runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Appendix C.1, we provide a detailed account of our computational overhead and model efficiency.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper adheres to the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: In Appendix A.5, we elaborate on the lack of societal impact of our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks as elaborated in Appendix A.5.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper employs publicly available datasets and code for training and comparative evaluation, adhering to all protocol restrictions that accompanied their release, and cites the relevant literature.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Upon acceptance of the paper, we will release our model and code under the CC BY-NC-SA 4.0 license.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.