

Supplementary Materials

A Overview

In this supplementary material, we will give the derivation for geometry guided diffusion model for stereo vision in Appendix B, introduce different metrics in Appendix C, study some interesting properties of our method in Appendix D, provide more implementation details in Appendix E, analyze different guidance modes in Appendix F, study alternative guidance in Appendix G, show more results of our methods in Appendix H, provide more data samples in our dataset **HISS** in Appendix I, and perform more experiments of mobile part manipulation in Appendix J.

B Geometry Guidance for Stereo Vision

To complement the main body of the paper, we provide the detailed derivation of the geometry guided diffusion model which appears in Equation 9 in the main text.

B.1 Stereo Vision

We define $y = \{I_l, I_r\}$ represents the conditioning stereo image pair and x_t is the noisy depth at time step t . By Bayes' theorem, we have

$$p(x_t|y) = \frac{p(x_t)p(y|x_t)}{p(y)} \quad (1)$$

$$\log p(x_t|y) = \log p(x_t) + \log p(y|x_t) - \log p(y) \quad (2)$$

Task derivative with respect to x_t on both sides of Equation 2:

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t) \quad (3)$$

Now, partition the second term $\log p(y|x_t)$ as

$$\begin{aligned} \log p(y|x_t) &= \log p(I_l, I_r|x_t) \\ &= \log p(I_l|x_t) + \log p(I_r|I_l, x_t) \\ &= \log p(x_t|I_l) + \log p(I_l) - \log p(x_t) + \log p(I_r|I_l, x_t) \end{aligned} \quad (4)$$

where we apply Bayes' theorem again in the third equation. Substitute Equation 4 back to Equation 3, we have

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(x_t|I_l) + \nabla_{x_t} \log p(I_r|I_l, x_t) \quad (5)$$

The first term is learned by the denoising network and the second term is the geometric guidance which can be calculated by stereo matching. In the experiments, we leverage more available data such as I_r and \tilde{D} in addition to I_l into the network during training:

$$\nabla_{x_t} \log p(x_t|y) = -\frac{1}{\sqrt{1-\alpha_t}} s \theta^*(x_t, t, y; \theta) + s \nabla_{x_t} \mathcal{L}_{\text{sm}}(I_l, I_r, x_t) \quad (6)$$

Here we empirically scale the geometry gradient with $s \in \mathbb{R}^+$ and set it to 1 in the experiments.

B.2 Extend to Active Stereo Vision

In addition to the left and right IR images, active stereo cameras provide another color image I_c captured from a third color camera. While the above derivation directly applies to active stereo cameras if we ignore the color image, we found that further feeding the color image into the network slightly improves the performance in DREDS [1]. However, most stereo datasets are *passive* and do not have additional color images. Therefore, during mixed dataset training, this additional color image is dropped. Here, we provide an active stereo version of derivation analogous to Equation 4:

$$\begin{aligned} \log p(y|x_t) &= \log p(I_c, I_l, I_r|x_t) \\ &= \log p(I_c|x_t) + \log p(I_l|I_c, x_t) + \log p(I_r|I_l, I_c, x_t) \\ &= \log p(I_c|x_t) + \log p(I_r|I_l, x_t) \\ &= \log p(x_t|I_c) + \log p(I_c) - \log p(x_t) + \log p(I_r|I_l, x_t) \end{aligned} \quad (7)$$

where the third equation assumes $p(I_l|I_c, x_t) = 1$. The I_c and I_l are already aligned and the only difference is the shadow pattern projected from the camera IR projector. The shadow pattern is irrelevant to the depth. Therefore, I_c is approximately the sufficient statistic of I_l . For the same argument, we have $\log p(I_r|I_l, x_t) = \log p(I_r|I_l, I_c, x_t)$. Likewise, the guidance for the active stereo camera can then be obtained by substituting Equation 7 into Equation 3:

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(x_t|I_c) + \nabla_{x_t} \log p(I_r|I_l, x_t) \quad (8)$$

In active stereo vision scenarios, we further train the network by conditioning it also on other available images. We set $y = \{I_l, I_r, I_c, \tilde{D}\}$.

C Baselines and Metrics

Baselines. NLSPN [2] is a depth completion work that uses an end-to-end non-local spatial propagation network to predict dense depth given sparse inputs. LIDF [3] proposes to learn an implicit density field that can recover missing depth given noisy RGB-D input. SwinDR [1] proposes a depth restoration framework based on SWIN transformer and is trained on a proposed table-top dataset with STD objects (DREDS). ASGrasp [4] proposes a stereo-depth estimation method based on Raft-Stereo to predict two-layer depths for tabletop grasping. Raft-Stereo [5] is the seminal deep stereo network. To this day, it is still the most adopted architecture in stereo vision.

Disparity Metric. End-Point Error (EPE) = $\frac{1}{H \times W} \sum |X - \hat{X}|$ is the mean absolute difference for all pixels between the ground truth and estimated disparity map.

Depth Metrics. We use the following depth metrics: 1) **RMSE** = $\sqrt{\frac{1}{H \times W} |D - \hat{D}|^2}$ is the root mean square error between ground truth and predicted depths, 2) **MAE** = $\frac{1}{H \times W} |D - \hat{D}|$ is the mean absolute depth error, 3) **REL** = $\frac{1}{H \times W} |D - \hat{D}|/D$ is the mean absolute relative difference, and 4) accuracy metric δ_i is the percentage of pixels satisfying $\max(\frac{d}{\hat{d}}, \frac{\hat{d}}{d}) < \delta_i$ where $\delta_i \in \{1.05, 1.10, 1.25\}$.

D Interesting Properties of Generative Stereo Vision

D.1 Uncertainty Estimation

Because our method is diffusion model based, we inherited the stochasticity in the reverse sampling process. To visualize the stochasticity, we run the same input 10 times. The uncertainty is obtained as the variance of the output disparity map. We conduct the experiments on DREDS and show the results in Figure 1. We observed that high uncertainty area corresponds to object edges where depth dramatically changes between foreground and background. Flat surfaces have lower uncertainty as the geometry is simpler. Such uncertainty could be used to filter outliers.

D.2 Generalization Comparisons with Monocular Methods

While our method works only in stereo cases, there are seminar works predicting depth given single RGB images. The attractive part of monocular depth estimation (MDE) is that more data is available for training. Therefore, these methods can be generalized well in the wild. While some monocular methods like ZeoDepth [6] propose to recover metric depth after a special training procedure, most monocular methods predict relative depth. The relative depth can be recovered with an absolute scale which can be obtained via other sensors like lidar or prior knowledge. However, our experiments (Figure 2) found that most monocular methods produce inferior quality depth even without considering the absolute scale.

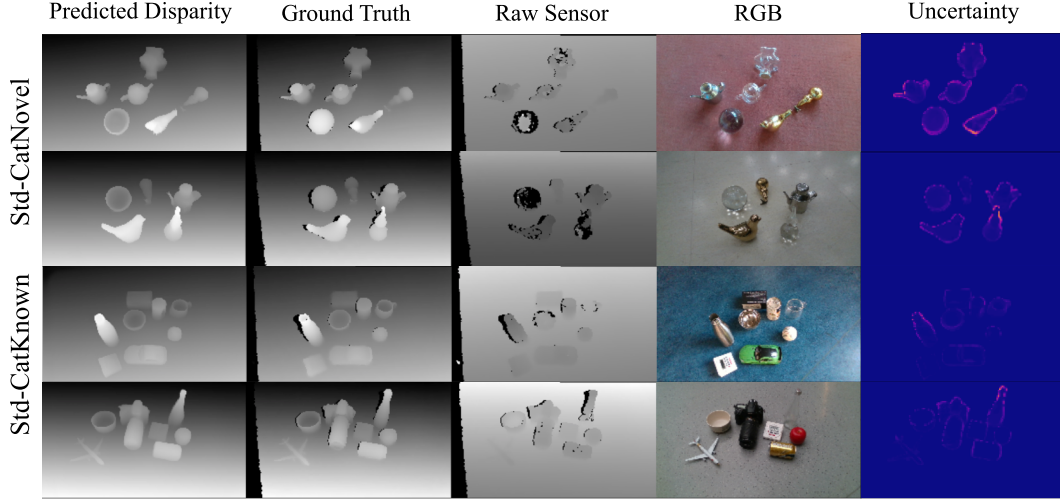


Figure 1: We visualize sample variance as uncertainty in the last column.

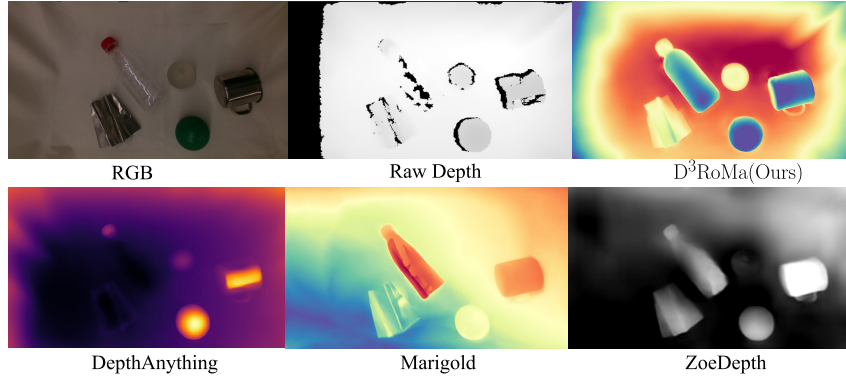


Figure 2: Generalization comparisons with State-of-the-art *monocular* depth estimation methods. All the results except ours are taken from their official web demo. Different methods used different color maps.

E Implementation Details

E.1 CUDA accelerated Semi-Global Matching

We used libSGM[7], a CUDA-accelerated, widely adopted implementation of the Semi-Global Matching (SGM)[8] algorithm. To seamlessly integrate libSGM into our pipeline, we utilized pybind11 to encapsulate the original codebase within our Python-based framework. This integration allows the adapted version of libSGM to achieve a performance of approximately 55 frames per second (FPS) at an input resolution of 960×540 , with around 380MB of memory allocated on an NVIDIA RTX 4090 GPU.

E.2 Network HyperParameters and Training

We implement our network using Hugging Face Diffusers [9] and pre-compute raw disparity maps using libSGM [7]. The network is trained 600 epochs with the batch size 6×8 and a constant learning rate 0.0001. All the images are randomly cropped to 320×240 and no other data augmentation is used during training. We use cosine scheduler [10] with 128 denoising time steps for β_t starting at 0.0001 and ending at 0.02. We use UNet as our denoising network. In the DREDS experiments, we have 6 downsampling ResNet blocks each layer has 128, 128, 256, 256, 512, and 512 channels.

84 The second-to-last channel is a downsampling block with spatial attention. We use MSE as our
 85 loss function. For the SceneFlow experiment, we scale down the original image resolutions from
 86 960×640 into 480×270 . We use a multi-resolution pyramid noise strategy as in [11]. We further
 87 use pretrained StableDiffusion v2 [12] in the grasping experiments and adapt the input Conv block
 88 accordingly to the conditioning inputs [11]. We also train the mixed datasets including DREDS,
 89 HISS, and SceneFlow at the batch level.

90 E.3 Grasping Implementation and Hardware Setup

91 In the grasping experiments, we mount the RealSense D415 on the wrist of the arm. After the camera
 92 captures a frame, we first acquire the depth map by $D = (f \cdot b)/X$. Then back project the depth into
 93 point cloud $\mathcal{P} = DK^{-1}P$, where $K \in \mathbb{R}^{3 \times 3}$ is the camera intrinsics and P are the homogeneous
 94 points in the image plane corresponding to each pixel. With the restored point cloud, we leverage
 95 GSNet [13] to predict 6 DoF grasping poses. To increase the grasping success rate for all baselines,
 96 we filter the grasping pose which has the angle between the grasping pose and the z (up) direction
 97 less than 30 degrees. We always select the grasping pose with the highest core and transform it into
 98 the robot base frame. Then we grasp the object with a motion planner like CuRobo [14]. *We did not*
 99 *perform workspace point cloud cropping operation as in the baseline ASGrasp [4] hence leading to*
 100 *an overall success rate drop in the main text compared with the numbers reported in ASGrasp.*

101 We use a wheeled mobile base mounted with two 7 DoF customized arms in the real mobile grasping
 102 experiments. Each arm attaches a parallel gripper. We only use the left arm in the experiments.
 103 Figure 3 displays the robot and the workplace.

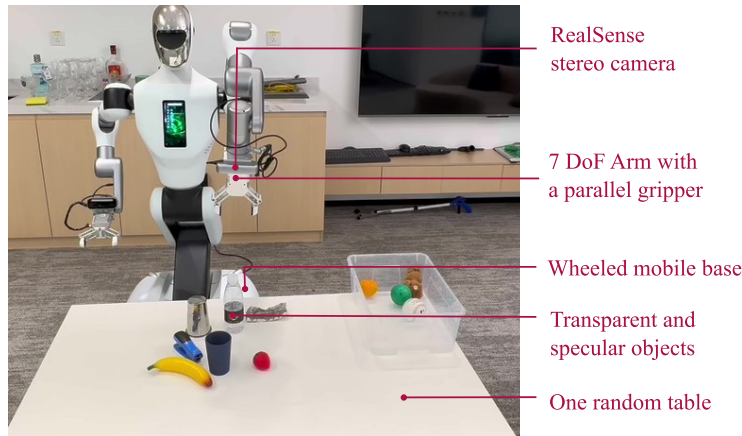


Figure 3: The robot used in the real mobile grasping experiments.

104 F Ablations on Network HyperParameters, Architectures and Samplers

105 F.1 Ablations on Network HyperParameters and Architectures

106 We provide ablation studies on the DREDS dataset in Table 1. The baseline is conditioned on the
 107 left, and right image and raw disparity. Its hyperparameters and network architecture are described
 108 in Appendix E.2. We also trained variants with different network architectures, loss functions, and
 109 noise strategies. We reduce the channels from 512 to 256 of the last two layers denoted as *reduced*
 110 *channels*. We also changed the loss function from MSE to L1 and used the default standard Gaussian
 111 noise.

112 F.2 Ablations on Different Samplers and Inference Time

113 The main factors of run time are the input image resolution and the number of denoising steps.
 114 We report the inference time of our method in Table 2. We also evaluate the effects of different

Table 1: Ablation Studies on Hyperparameters and Network Architectures.

Methods	RMSE ↓	REL ↓	MAE ↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
Baseline	0.0040	0.0014	0.0010	99.71	99.90	99.99
D ³ RoMa (reduced channels)	0.0048	0.0016	0.0011	99.60	99.85	99.98
D ³ RoMa (L1 loss)	0.0047	0.0008	0.0012	99.60	99.83	99.98
D ³ RoMa (randn noise)	0.0048	0.0017	0.0012	99.64	99.87	99.98

samplers in the real experiments where we used pretrained StableDiffusion [12]. The network total has about 865M parameters. We fixed the number of time steps during training to 1000 and used the same standard cosine scheduler [10], and all the samplers take 10 denoising steps during inference. We perform reverse sampling using different schedulers implemented by Diffusers [9]. All the samplers achieve similar qualitative results except *Euler Ancestral*. The results are shown in Figure 4. Empirically, we select DDPM with 10 denoising steps and a resolution of 640×360 in our real experiments.

Table 2: Runtime and memory consumption of our method during reverse sampling for single input. All times are reported on NVIDIA A100.

Disparity Resolutions	1280×720	640×360	480×270	320×180	224×126
5 Denoising steps	5.53	2.56	2.31	1.95	1.96
10 Denoising Steps	8.82	3.19	2.91	2.25	2.17
50 Denoising Steps	34.56	8.45	5.79	4.28	3.86
Peak Memory Usage	18.62G	7.87G	7.57G	6.94G	6.89G

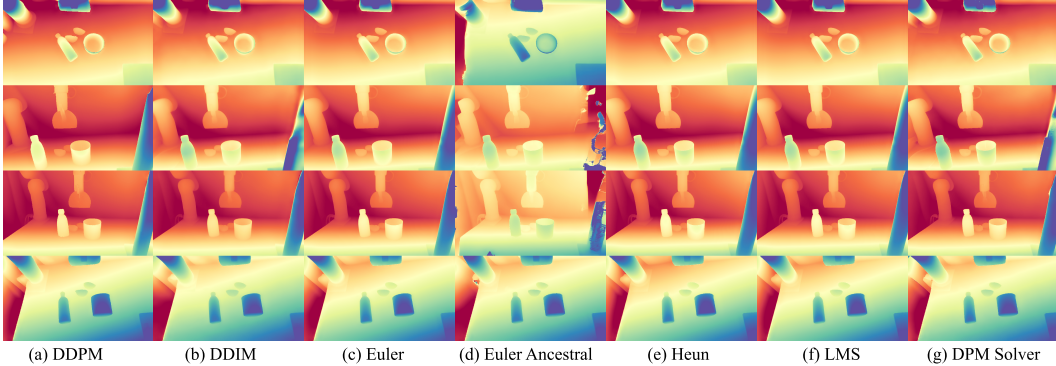


Figure 4: Comparisons of different samplers used during reverse sampling. We use DDPM [15] sampler with 10 steps in the experiments.

G Alternative Guidance with Raw Disparity

This section will study alternative guidance to the diffusion model during the reverse sampling processes. In the stereo vision case, the gradient of the photometric loss is obtained by checking the consistency of the left and right images. Mathematically, the gradient should also have the same direction with $x_0 - x_t$ where x_0 is the ground truth disparity. In test time, x_0 is unknown but can be approximated by an external less noisy measurement source such as a Lidar. The external depth measurement can be converted to the disparity space \tilde{x}_0 and is multiplied with a mask if it is sparse:

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) &= \nabla_{x_t} \log p(x_t|I_c) + \nabla_{x_t} \log p(I_r|I_l, x_t) \\ &\approx \nabla_{x_t} \log p(x_t|I_c) + \alpha \omega(u, v) \text{sign}(\tilde{x}_0 - x_t)\end{aligned}\quad (9)$$

We here experiment with raw depth guidance. The guidance \tilde{x}_0 is approximated by camera raw sensor depth. Therefore the $\text{sign}(\tilde{x}_0 - x_t)$ is the approximate gradient. We set mask $\omega(u, v) = (\tilde{x}_0 > 0)$ and α is a constant controls the guidance strength. We qualitatively study the guidance

of the approximate gradient in Figure 5. The benefits of guidance by the approximate gradient are limited when raw depth is highly noisy.

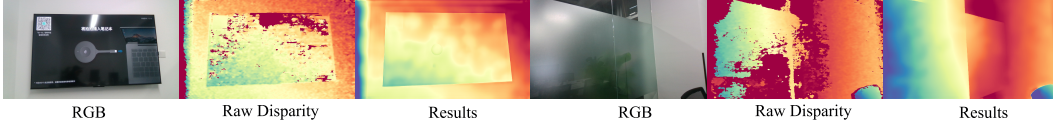


Figure 5: Guidance with raw disparity.

H Evaluations on HISS dataset

H.1 Results on HISS Test Split

In this section, we train other SOTA stereo methods from scratch and compare with our method on the HISS dataset. We further rendered 300 images in 5 new scenes different from our training dataset as the test set. The results are given in Table 3. We also show more real depth estimation results in Figure 7 and more comparisons in Figure 8, which we consider also attributed to the joint training on our dataset.

Table 3: Quantatives evaluations on HISS dataset.

Methods	EPE	RMSE ↓	REL ↓	MAE ↓	$\delta_{1.05} \uparrow$	$\delta_{1.10} \uparrow$	$\delta_{1.25} \uparrow$
Raft-Stereo	0.0721	0.0521	0.0092	0.0164	95.26	98.89	99.10
D ³ RoMa	0.0579	0.0378	0.0067	0.0084	97.86	99.22	99.76

H.2 Deonising Process

One of the motivations for using the diffusion model to predict depth is the multi-step reverse sampling process. It resembles the iterative solver which has been proven successful in RAFT [16] and its successors. In figure 6 we show an example of the denoising process trained on our HISS dataset. The total denoising steps is set to 128 and we visualize every 32 timesteps.

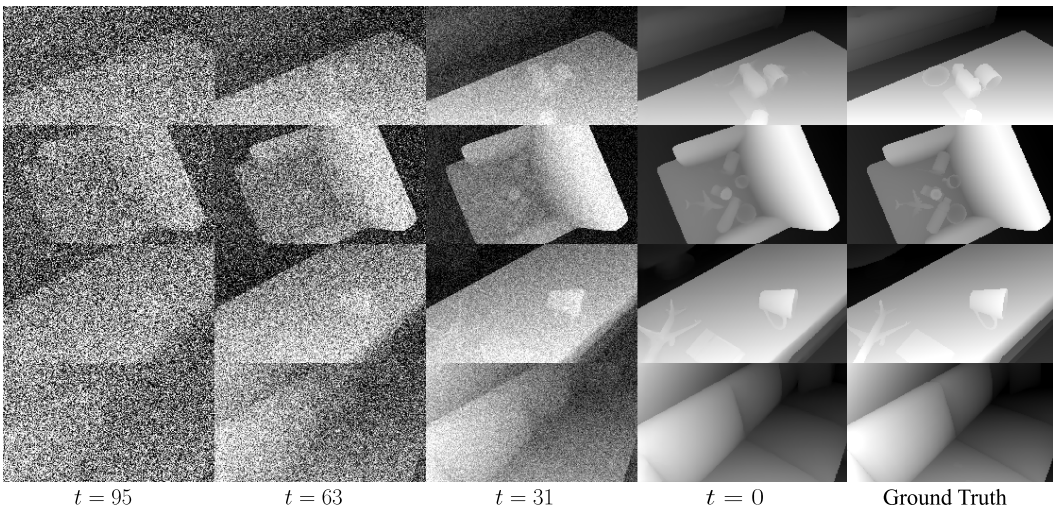


Figure 6: Visualization of the denoising process on the HISS dataset. The 4 left columns show the denoising steps every 2 time steps. The 2 right columns show the final output and ground truth disparity map respectively.

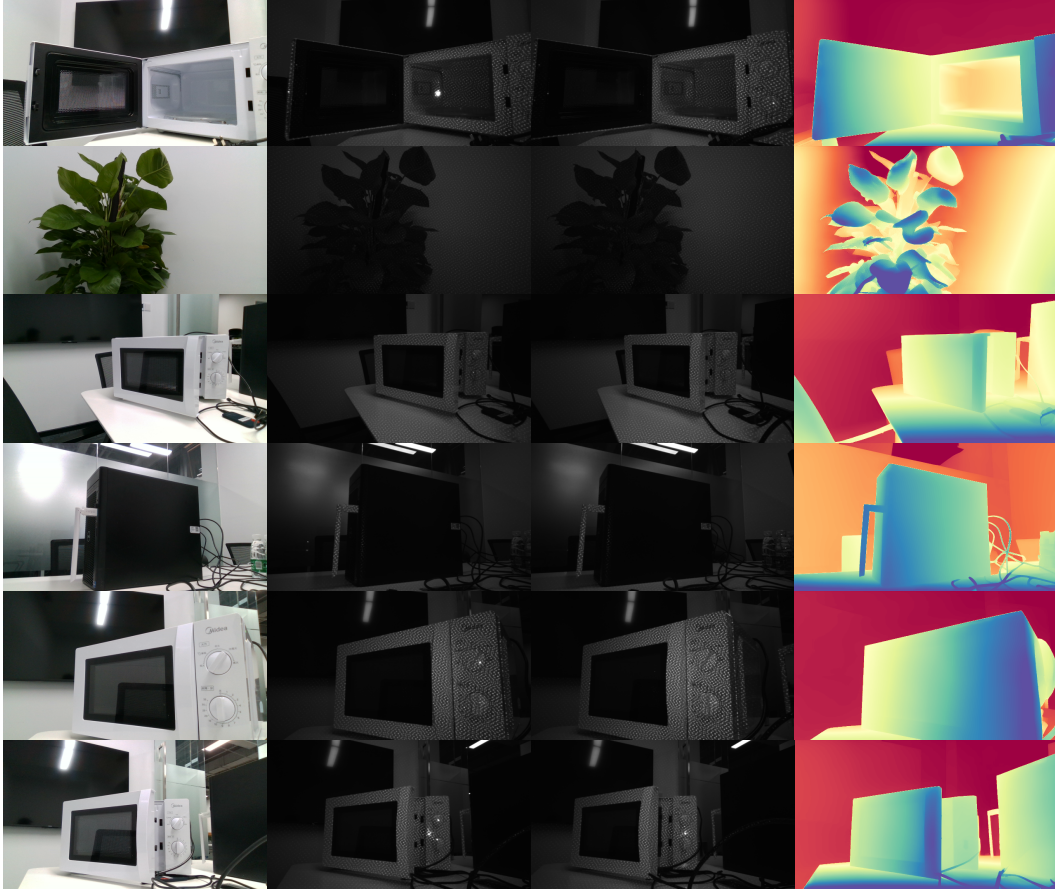


Figure 7: More in the wild examples.

I HISS Dataset

Training diffusion models for stereo depth estimation are more data-demanding due to lacking specialized network architectures and loss functions as in traditional deep stereo networks. Most existing stereo depth datasets are either synthetic (SceneFlow [17]) or with limited diversity, such as ClearGrasp[18], LIDF [3], DREDS [1]. While deep stereo methods [5, 19, 20, 21] are trained on a small number of stereo images and show strong generalizability performance, they are unable to predict correct depth for transparent objects, which are critical for many robotic tasks. Part of the reason is lacking data. Shi et al. [4] and Dai et al. [1] both created specially crafted transparent and specular objects but both datasets are table-top scenes. Based on the above motivations, we compensate existing datasets with more indoor room-level stereo images with domain randomization on object materials and scenes.

One aspect that characterizes our dataset is *scene-level* and *photo-realistic* rendering of the *specular*, *transparent*, and *diffuse* objects. We rendered over 350 objects in 168 different HSSD [22] scenes. The objects randomly fall onto the furniture, ground, and tables to simulate real-world object placements. The infrared (IR) images are rendered properly with seeing-through or specular lighting effects on Non-Lambertian surfaces. We provide some data samples in Figure 9.

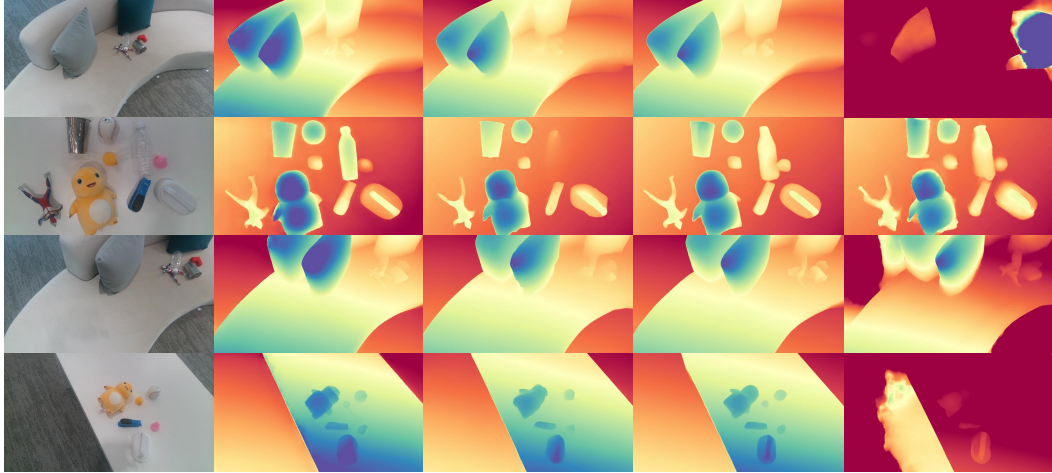


Figure 8: Qualitative comparisons with other state-of-the-arts. Each row (from left to right) displays rgb image and disparity map: RGB image, our method, pre-trained Raft Stereo, Raft Stereo fine-tuned on our dataset, and ASGrasp.

J Part Manipulation

J.1 Interaction Policy

Following [23, 24], we first do part segmentation and pose estimation using the perception method. Based on the predictions of the part poses, we move the robot arm toward the target part and turn the gripper in the direction suitable for grabbing. Finally, we move the gripper along the proposed trajectories toward the target position, following our GPart pose definition.

J.2 Experiment Setup

In the experiments, we use the Franka Emika Panda robot arm with CuRobo[14] motion planning and the end-effector trajectory just like GPartNet[23]. For manipulation tasks in the real world, a partial point cloud of the target object instance is acquired from our method. With the proposed network and manipulation heuristics in [23], the pose trajectory of the end-effector can be predicted. Then we use cuRobo[14] to solve the pose of Franka to follow our end-effector trajectory.



Figure 9: RGB Data samples from Our dataset HISS except the bottom row which shows a group of rendering of RGB image, (left) IR image, and normal.

References

- [1] Q. Dai, J. Zhang, Q. Li, T. Wu, H. Dong, Z. Liu, P. Tan, and H. Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *European Conference on Computer Vision*, pages 374–391. Springer, 2022.
- [2] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. So Kweon. Non-local spatial propagation network for depth completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 120–136. Springer, 2020.
- [3] L. Zhu, A. Mousavian, Y. Xiang, H. Mazhar, J. van Eenbergen, S. Debnath, and D. Fox. Rgb-d local implicit function for depth completion of transparent objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4649–4658, 2021.
- [4] J. Shi, Y. Jin, D. Li, H. Niu, Z. Jin, H. Wang, et al. Asgrasp: Generalizable transparent object reconstruction and grasping from rgb-d active stereo camera. *arXiv preprint arXiv:2405.05648*, 2024.
- [5] L. Lipson, Z. Teed, and J. Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021.
- [6] S. F. Bhat, R. Birkel, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.
- [7] O. Shingo, T. Akihiro, and H. Takaaki. libsgm. <https://github.com/fixstars/libSGM>, 2018.
- [8] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 2, pages 807–814. IEEE, 2005.
- [9] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.
- [10] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.
- [11] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler. Repurposing diffusion-based image generators for monocular depth estimation. *arXiv preprint arXiv:2312.02145*, 2023.
- [12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [13] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15964–15973, 2021.
- [14] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023.
- [15] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [16] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

- 217 [17] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large
218 dataset to train convolutional networks for disparity, optical flow, and scene flow estimation.
219 In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages
220 4040–4048, 2016.
- 221 [18] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song. Clear grasp: 3d shape
222 estimation of transparent objects for manipulation. In *2020 IEEE international conference on
223 robotics and automation (ICRA)*, pages 3634–3642. IEEE, 2020.
- 224 [19] P. Weinzaepfel, T. Lucas, V. Leroy, Y. Cabon, V. Arora, R. Brégier, G. Csurka, L. Antsfeld,
225 B. Chidlovskii, and J. Revaud. Croco v2: Improved cross-view completion pre-training for
226 stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference
227 on Computer Vision*, pages 17969–17980, 2023.
- 228 [20] G. Xu, X. Wang, X. Ding, and X. Yang. Iterative geometry encoding volume for stereo match-
229 ing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,
230 pages 21919–21928, 2023.
- 231 [21] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, and S. Liu. Practical stereo
232 matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the
233 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272,
234 2022.
- 235 [22] M. Khanna, Y. Mao, H. Jiang, S. Hareesh, B. Schacklett, D. Batra, A. Clegg, E. Undersander,
236 A. X. Chang, and M. Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d
237 scene scale and realism tradeoffs for objectgoal navigation. *arXiv preprint arXiv:2306.11290*,
238 2023.
- 239 [23] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang. Gapartnet: Cross-category
240 domain-generalizable object perception and manipulation via generalizable and actionable
241 parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recogni-
242 tion*, pages 7081–7091, 2023.
- 243 [24] H. Geng, S. Wei, C. Deng, B. Shen, H. Wang, and L. Guibas. Sage: Bridging semantic and
244 actionable parts for generalizable manipulation of articulated objects, 2024.