Orchid: Image Latent Diffusion for Joint Appearance and Geometry Generation

Supplementary Material

In this supplementary material, we provide additional details on our datasets, model architecture, ablations and training methodology. We also provide a runtime analysis, user-study, and additional qualitative results from Orchid for text conditioned color-depth-normal generation as well as image conditioned depth-normal prediction, and joint inpainting tasks, including comparisons to more baselines. We conclude with a discussion of our limitations, and scope for future work. Novel-view synthesis videos of 3D reconstructions using predictions from Orchid are provided on the web page *https://orchid3d.github.io*, along with a discussion in Section 5.5.

1. Orchid details

1.1. Architecture

For our VAE, we use a convolutional encoder and decoder with a latent dimension of 8, with $8\times$ spatial downsampling. The VAE has 7 input channels: 3 for RGB, 1 for depth, and 3 for surface normals. The discriminator (used only during VAE training) is a small ConvNet + MLP.

Once the VAE is trained, we keep it frozen when training the latent diffusion model. The latent diffusion model itself is a UNet transformer similar to *Stable Diffusion* [13] which is conditioned on both time and text embeddings. It has approximately 2B parameters.

1.2. Training

We use a combination of RGB, depth, and normal losses when training the VAE, with weights as explained in Section 3.1 of our paper. Here, we provide the values of the weights we used for our model. For $L_{\mathbf{x}}$, we use $w_1^{\mathbf{x}} = 1, w_2^{\mathbf{x}} = 0.1, w_3^{\mathbf{x}} = 0.1, w_4^{\mathbf{x}} = 1$. For $L_{\mathbf{d}}$, we use $w_1^{\mathbf{d}} = 1, w_2^{\mathbf{d}} = 0.5$. We use $w^{\mathbf{n}} = 1$ for $L_{\mathbf{n}}$. We also use $w^{distill} = 10^{-}6$ for $L_{distill}$ and $w^{KL} = 10^{-}3$ for L_{KL}. Our choice of loss components and their weights for $L_{\mathbf{x}}$ and $L_{\mathbf{KL}}$ are based on standard training recipes for VAEs used in latent diffusion models. For losses we introduce, i.e, $L_{\mathbf{d}}$, $L_{\mathbf{n}}$, and $L_{distill}$, we obtained similar results with weights of similar orders of magnitude, but dropping them completely worsens quality (as shown in our ablations).

On 16 NVIDIA A100 GPUs, we take approximately 5 days to train the VAE, 2 days to finetune our LDM starting from a color LDM, and 8-12 hours to finetune our image-conditioned model.

1.3. Dataset construction

We provide details of the dataset we use for VAE and LDM training in Table 1. When training the VAE, we sample more heavily from the high-quality real world datasets, rather than our dataset with teacher model predictions. Whereas for the text-conditional LDM training, we sample more heavily from the distillation dataset which contains text-captions. For image-conditioned LDM finetuning, we ignore the text captions, and sample from both real-world and distillation data. While predictions from teacher models are not perfect, models distilled from multiple teachers have performed better in previous work[16]. We remove a few rare examples where depth and normal teacher models disgree (high depth-normal inconsistency) for significant parts of the image.

Dataset	Size	Text	Depth	Normals
Hypersim	60k	✗	✓	✓
Virtual KITTI	21k	✗	✓	✗
Replica + GSO (Omnidata)	100k	✗	✓	✓
Taskonomy (Omnidata)	2M	✗	✓	✓
DIODE	25k	✗	✓	✓
Pseudo-labeled (ours)	110M	✓	✓	✓

Table 1. **Dataset details:** We use all the above datasets for training the VAE, but only the pseudo-labeled text-image dataset, Hypersim, and Replica + GSO for finetuning our LDM.

2. Runtime analysis

Model	Diff + Diff	Diff + FF	Orchid
Inference time (s / img)	4.2*	1.3*	**1.2**

Table 2. **Runtime analysis:** Orchid is the fastest way to generate color, depth, and normals. A fair runtime comparison is hard since these methods vary in memory usage. Baselines using multiple models (*) cannot store all models on a GPU and need added weight I/O time that is not included here.

We provide an analysis of the runtime taken for the different approaches discussed in Table 1 in our main paper in Table 2. We report inference times for on a single H100 for all three methods. Our joint generation of color, depth, and normals is significantly faster than generating them with 3 different diffusion models. It is also faster than using discriminative models for depth and normals after an image diffusion process - 1.2 vs 1.3 s per image. Although this difference may seem less significant, please note that we do not include the time taken to move model parameters to/from the GPU, which is required when using multiple models. This I/O time is significantly greater than the inference time for discriminative models.

3. Ablation details

This section provides details for some of the ablations provided in our paper.

Unified appearance-geometry diffusion baseline with disentangled latents: Orchid uses a unified joint latent space for color-depth-normal generation. An alternative design to enable a unified color-depth-normal diffusion model would be to explicitly encode all three modalities using separate latents (all produced by the same VAE), and finetune the LDM to denoise a higher dimensional concatenation of all three latents. We find that while this is a feasible approach, the quality of generated images is significantly worse than that of using a joint latent. Our hypothesis is that this is likely due to a significant mismatch of the latent space from the color image-only pretraining stage, as opposed to a joint latent space that is similar in structure (due to the distillation loss) and dimensionality to the pretrained LDM's latent space. Quantitatively, Table 5 in our paper shows that this disentangled latents model has a lower CLIP-similarity score when evaluated on COCO captions. It does however have a slightly higher LPIPS, likely because it uses the same latent dimension to store color information alone. Our joint latent however is significantly better on image-conditioned prediction tasks, indicating that the model is able to learn an effective joint latent representation of all three modalities.

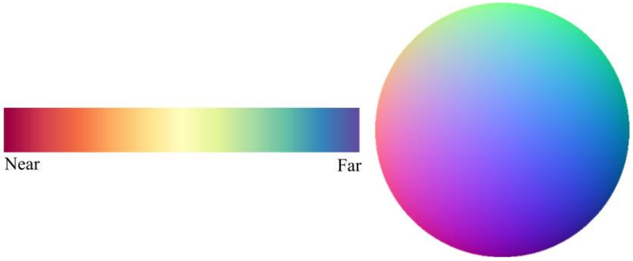

Figure 1. Colormap for depth (left) and surface normal on a unit hemisphere (right) used for all qualitative results in this paper.

4. Color generation quality

Method	RGB generation metrics		User preference (%)	
	CLIP (↑)	LPIPS (↑)	Aesthetics	Text adherence
RGB LDM (ours-PT)	**0.319**	0.741	32.1	31.0
Orchid	0.316	**0.764**	**47.4**	**46.9**
No notable difference	-	-	20.5	22.1

Table 3. Quantitative evaluation and user study for RGB quality.

While the focus of our work is not to improve the quality of generated color images, we evaluated how the quality of Orchid's text-conditioned color generations compare to the pretrained RGB-only diffusion model that we finetune from. In Table 3, we report the commonly used CLIP score and LPIPS for both models on the MS-COCO dataset, together with the findings of a user study we conducted. We generated images from both models using different captions, and asked users to pick from 3 options - Orchid's image, the base LDM's image, or notable difference. The users were asked to vote on two different aspects: aesthetics (overall quality of the image), text adherence (closeness to the text caption). We surveyed 40 users with 25 images each (1000 votes across both aspects in total). The quantitative metrics show that Orchid's generations are comparable to the color-only baseline, while the user study indicates that Orchid's generations are slightly better, with about 20% votes indicating no notable difference between the two. These metrics depend significantly on the pretraining data and color-only model being used; Orchid maintains the pretrained generation quality while enabling joint color-depth-normal generation.

4.1. Note on depth-normal redundancy

Using a joint latent for color, depth, and normals minimizes redundancy in our latent space, in comparison to using separate latents for each modality. Depth and normals are highly inter-dependent, as normals can be derived from (metric) depth. A joint latent avoids the need for separate latents, resulting in highly consistent predictions. To further validate this redundancy, we performed a PCA analysis on concatenated (separate) depth and normal latents (8 dimensions each, 1000 samples). Only 8 PCA bases (out of the full 16 dimensions) were needed to explain ¿ 95% variance, confirming the strong depth-normal redundancy.

5. Qualitative results

We provide additional qualitative results and comparisons for the experiments in our paper. Colormaps used to visualize the depth and surface normal predictions is shown in Figure 1.

5.1. Note on depth map visualization

Orchid predicts affine-invariant inverse depth, unlike other baselines Marigold [7] and GeoWizard [3] that predict affine invariant depth normalized to [0, 1]. To compare our depth qualitatively when ground truth depth is available (Figures 5, 6, 7, 8, 9), we align all predictions to the ground truth by estimation a shift and scale offset using least squares. When ground truth is not available (Figures 3 and 4), we inverted inverse-depth produced for our method, while using the predicted depth for [7] and [3], which may appear different due to an unknown inverse-depth shift. We use the colormap in Figure 1.

5.2. Text conditioned joint generation

We show color-depth-normals generated by our model for different text prompts in Figure 2. Figure 3 compares the results from our model to a baseline that uses a color-only LDM to first generate color, and then depth and normal diffusion models to generate depth and surface normals. The results from a single pass of our model are comparable to

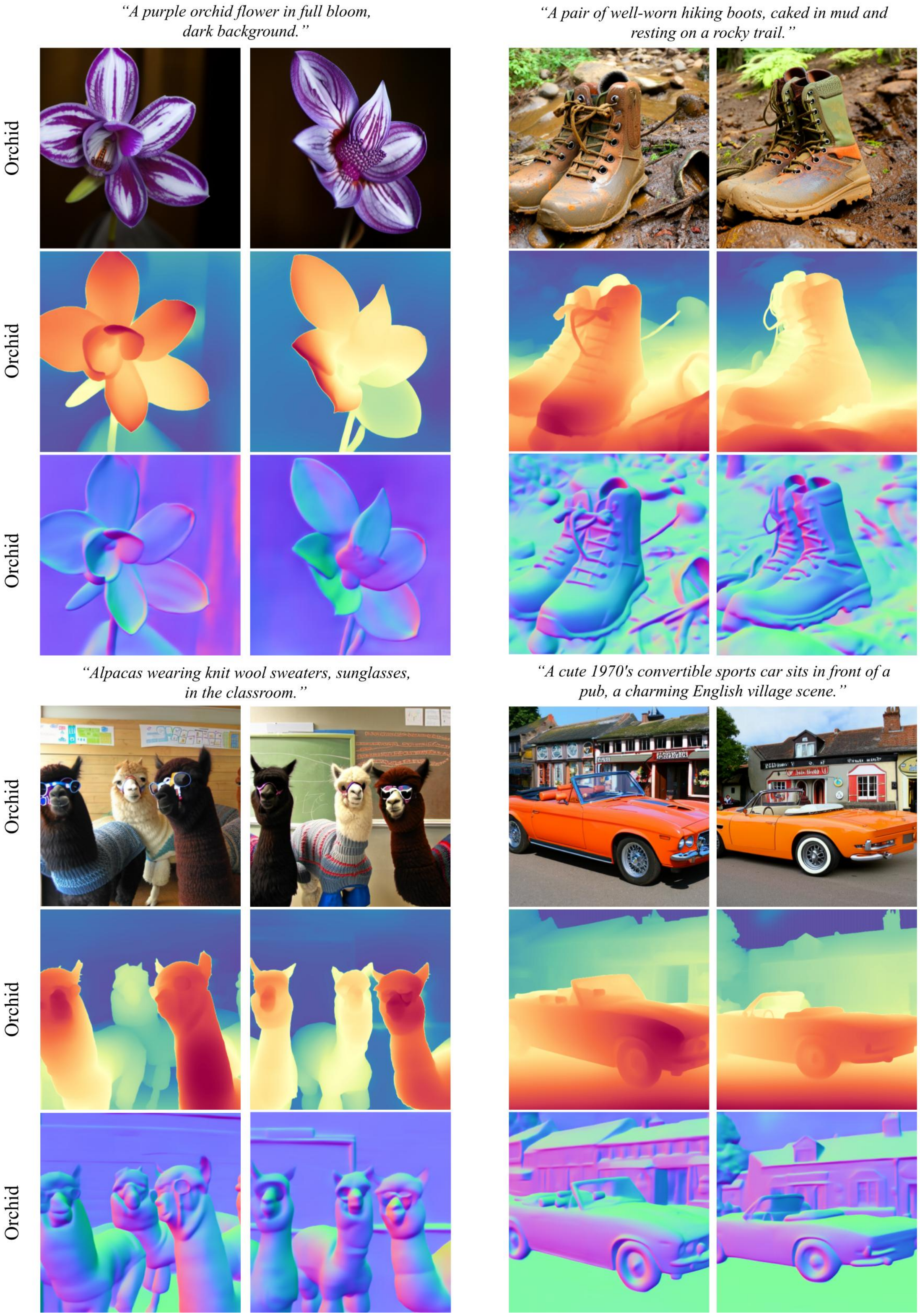
"A purple orchid flower in full bloom, dark background."
"A pair of well-worn hiking boots, caked in mud and resting on a rocky trail."
Orchid
Orchid
Orchid
"Alpacas wearing knit wool sweaters, sunglasses, in the classroom."
"A cute 1970's convertible sports car sits in front of a pub, a charming English village scene."
Orchid
Orchid
Orchid

Figure 2. **Text conditioned generation**: We show color, depth and normals generated by Orchid for different text prompts. We show two results for each prompt.

these results. Figure 4 compares the depth and normals generated by our model to those predicted by depth and normal prediction baselines [1, 7, 17] on our images (generated along with depth and normals). In examples regions where depth is ambiguous for generated images (*e.g.* background structure in Figures 3, 4), predictions from our model are qualitatively better. In other cases, our generated depth is comparable to those of baselines [7] while our normals are significantly better.

5.3. Monocular depth and normal estimation

Internet images: We show more depth and normal predictions on in-the-wild images produced by Orchid in Figures 5, 6, 7, and 8. Figures 5 and 6 compare our joint predictions to those from GeoWizard [3]. We find that our depth and normals are more accurate (with fewer errors on large sections), even though GeoWizard's predictions more detailed in many cases. In Figures 7, and 8, we compare Orchid's predictions to Marigold [7]. We find that Orchid has better depth estimates at longer ranges, and significantly better normal estimates overall. Note that we need different Marigold weights to predict depth and normals (unlike our joint prediction model). When comparing colorized depth maps on these datasets without ground truth depth, please refer to the note in Section 5.1.

Zero-shot benchmark images: We show more depth and normal predictions on the zero-shot depth and normal estimation benchmarks used in Section 4 of our paper in Figures 10, 9, and 11. Figure 10 shows that Orchid is competitive with diffusion-based depth prediction baselines Marigold [7] and GeoWizard [3], while being slightly better in some cases. Both [7] and [3] have a common failure mode where depth estimates are sensitive to image discontinuities, which our model is significantly less sensitive to. Figure 9 shows that our model is significantly better are depth estimation in outdoor environments, especially at longer ranges. Figure 11 shows that our model is significantly better at surface normals estimation, particularly on objects with curved surfaces.

5.4. Joint inpainting

Section 4.4 of our paper explains how our model can be used to jointly inpaint color-depth-normals. For this task, we use as input paired color, depth, and normal images, and a user-provided mask for the region to be inpainted. In cases where only a color image is available, depth and normals can be generated using the image-conditioned Orchid. We then generate the latents in the masked region, using Orchid to iteratively denoise them, while using noise-free latents encoded from the inputs for the unmasked region. This is similar to the approach proposed in RePaint [10]. We provide qualitative results in Figure 12. We show multiple inpainting results for the same input. We find that Orchid is able to generate very realistic images, with different semantically and geometrically consistent color, depth, and normals for the masked regions. We compare this to a baseline that first inpaints color (Stable-Diffusion XL-inpainting [12]), then inpaints depth (Marigold-DC [15]), and predicts normals using Marigold/Lotus [5, 7]. The baseline performs significantly worse than Orchid, with several geometric inconsistencies in the generated color image (edges of objects or walls not intersecting, mismatch in vanishing directions, etc.). It also appears more unrealistic. The baseline uses conditional prediction on the full inpainted image for normals instead of inpainting them, as there are no publicly available normal-inpainting diffusion baselines.

5.5. 3D reconstruction from single view

The image-conditioned Orchid can jointly generate depth and normals from an input image. These color, depth, and surface normals can be used to reconstruct the 3D scene using either Gaussian Splatting methods (3DGS [8], 2DGS [6]) or Poisson surface reconstruction. The novel-view synthesis videos of reconstructions produced from the generated color and geometry are provided on our web page *https://orchid3d.github.io*.

6. Limitations and future work

Orchid is not without limitations. In terms of geometry prediction accuracy, there is some scope for improvement on surfaces with high frequency edges (eg.: grass, fur, or hair). Some of these undesirably smooth predictions are apparent in our qualitative results on images in-the-wild. Future work can focus on further scaling unified appearance and geometry diffusion models, incorporating more recent developments in color diffusion models such as DiTs and flow-matching schedules. We also anticipate unified appearance-geometry diffusion models to be applied to many downstream reconstruction settings that are beyond the scope of our work: 3D scene completion, novel-view synthesis, text-conditioned full 3D generation, 3D inpainting, etc.

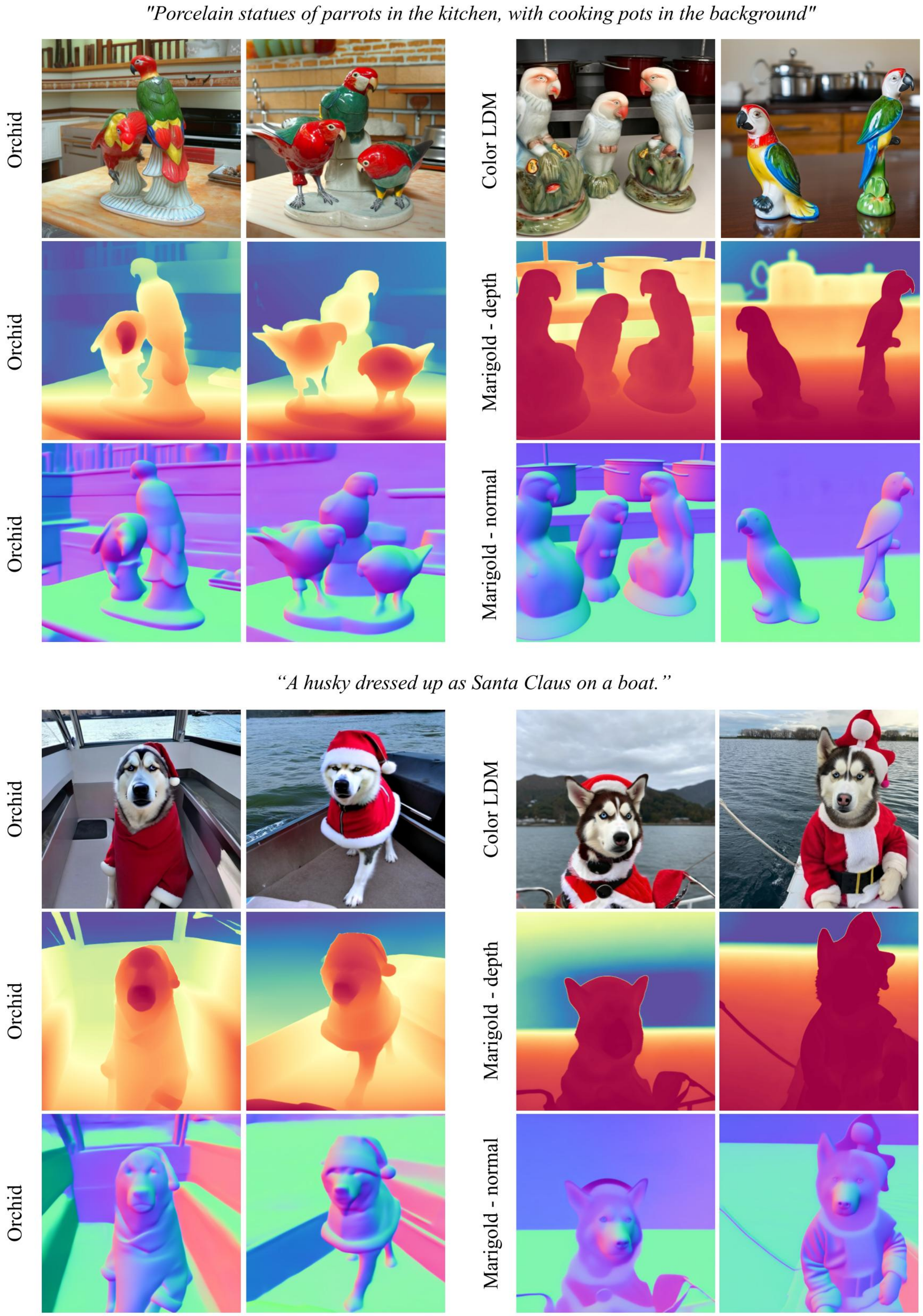
"Porcelain statues of parrots in the kitchen, with cooking pots in the background"
Orchid
Orchid
Orchid
Color LDM
Marigold - depth
Marigold - normal
"A husky dressed up as Santa Claus on a boat."
Orchid
Orchid
Orchid
Color LDM
Marigold - depth
Marigold - normal

Figure 3. **Text conditioned color-depth-normal generation:** We show two predictions from Orchid for each text prompt. We qualitatively compare these to the alternative: generate color, depth and normals from a separate diffusion model for each. For this baseline, we use a color-only LDM for color, and separate Marigold [7] models for depth and normals. When comparing results, please refer to our note on depth map visualization (Section 5.1).

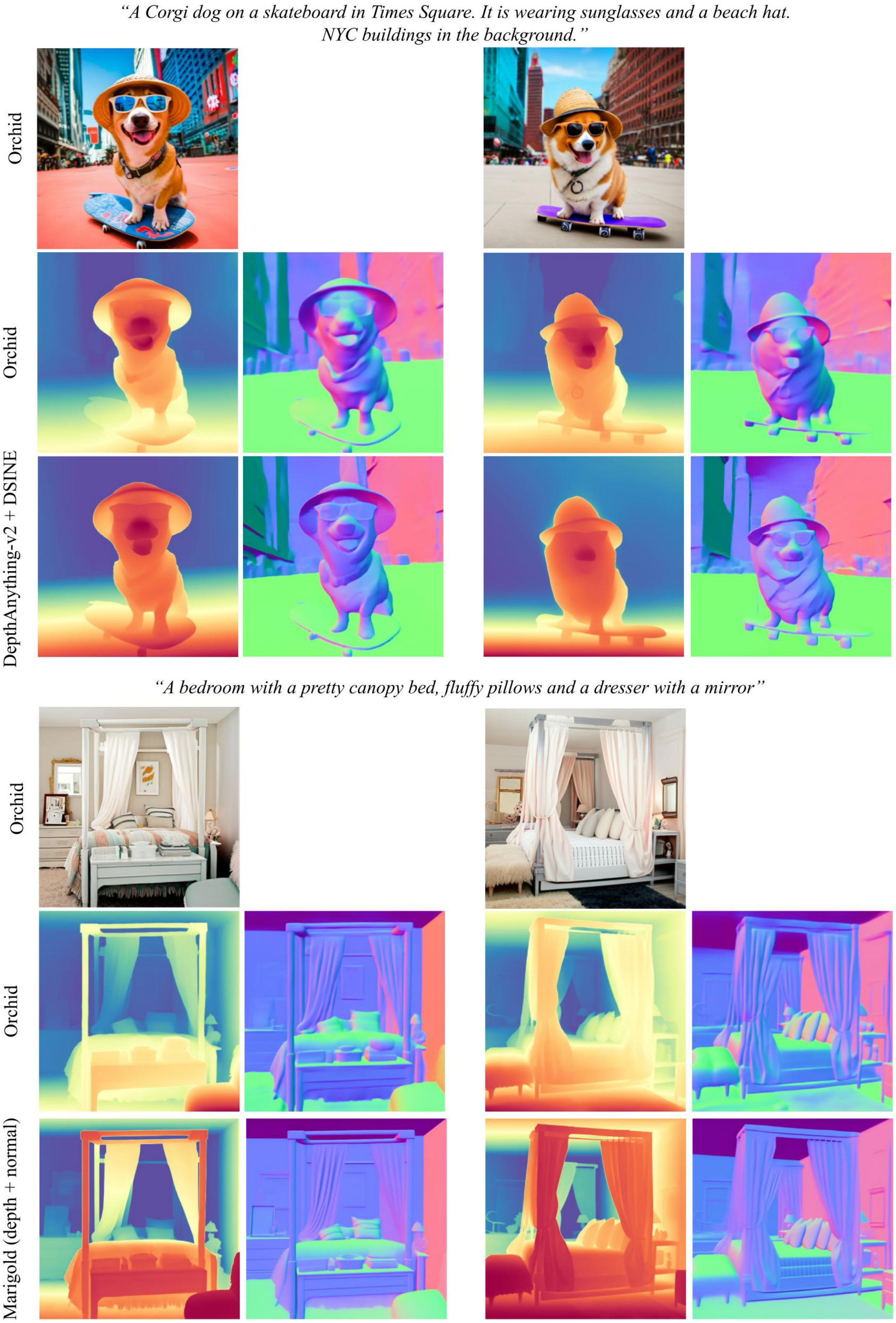
"A Corgi dog on a skateboard in Times Square. It is wearing sunglasses and a beach hat. NYC buildings in the background."
Orchid
Orchid
DepthAnything-v2 + DSINE
"A bedroom with a pretty canopy bed, fluffy pillows and a dresser with a mirror"
Orchid
Orchid
Marigold (depth + normal)

Figure 4. **Text conditioned color-depth-normal generation:** We show two predictions from Orchid for each text prompt. We compare the geometry predicted by our model to Marigold (separate) depth and normal models [7], and to the DepthAnything-v2 + DSINE combination[1, 17]. We find Orchid's geometry predictions to be qualitatively better, especially on structures / people in the background in the Corgi image. Color-conditional models may be inaccurate in such cases where geometry is ambiguous. When comparing results, please refer to our note on depth map visualization (Section 5.1).

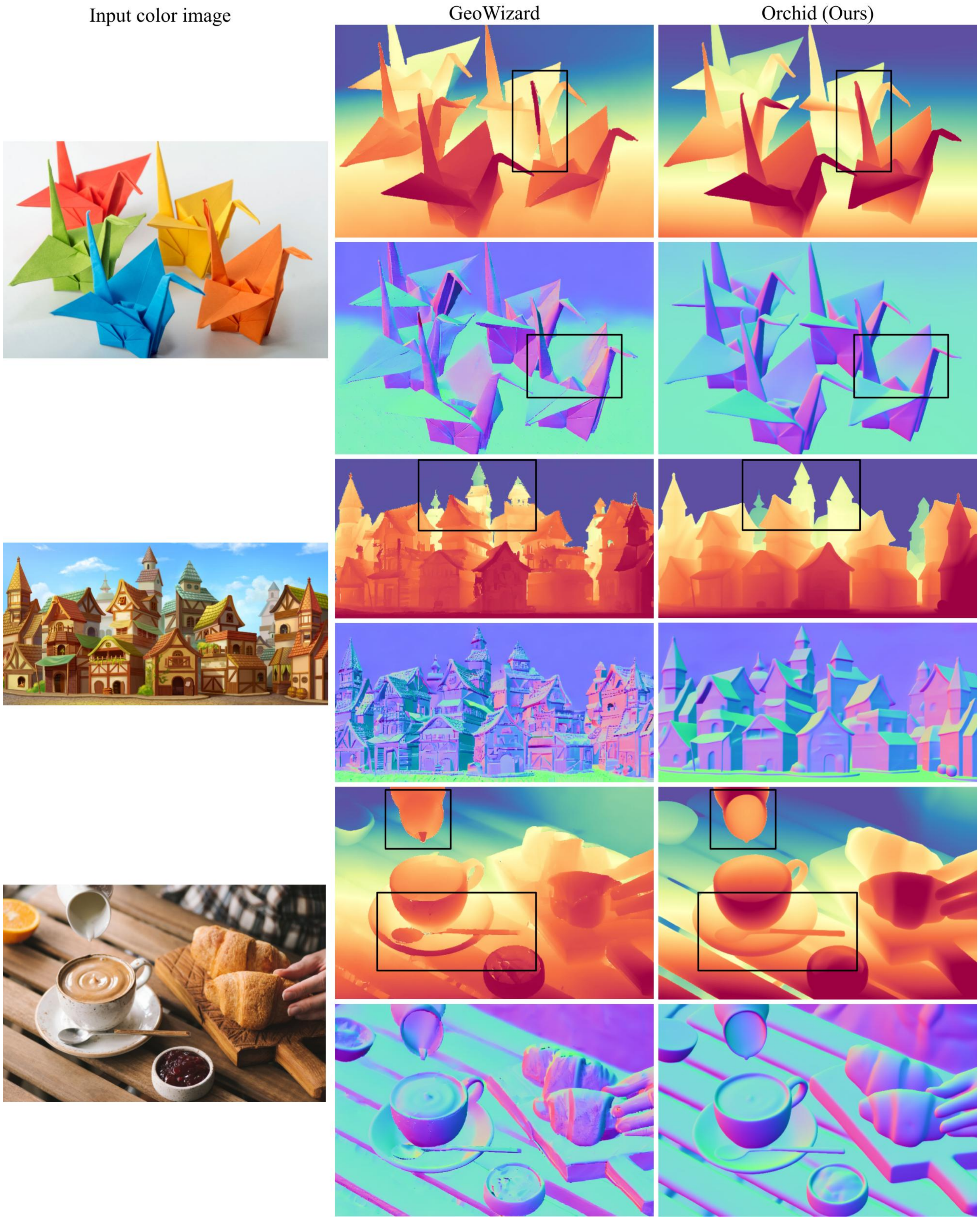

Figure 5. Comparison of GeoWizard [3] and Orchid for depth and normal estimation on in-the-wild input images. We can see that unlike GeoWizard, results from Orchid have correct depth and normal predictions while still having sharp boundaries. Some of these areas have been highlighted in the images shown above. In particular, Orchid shows less discontinuities in the Origami surfaces in both depth and normals, and more accurate depth predictions of the hollow objects pictured (milk pitcher, coffee mug and saucer).

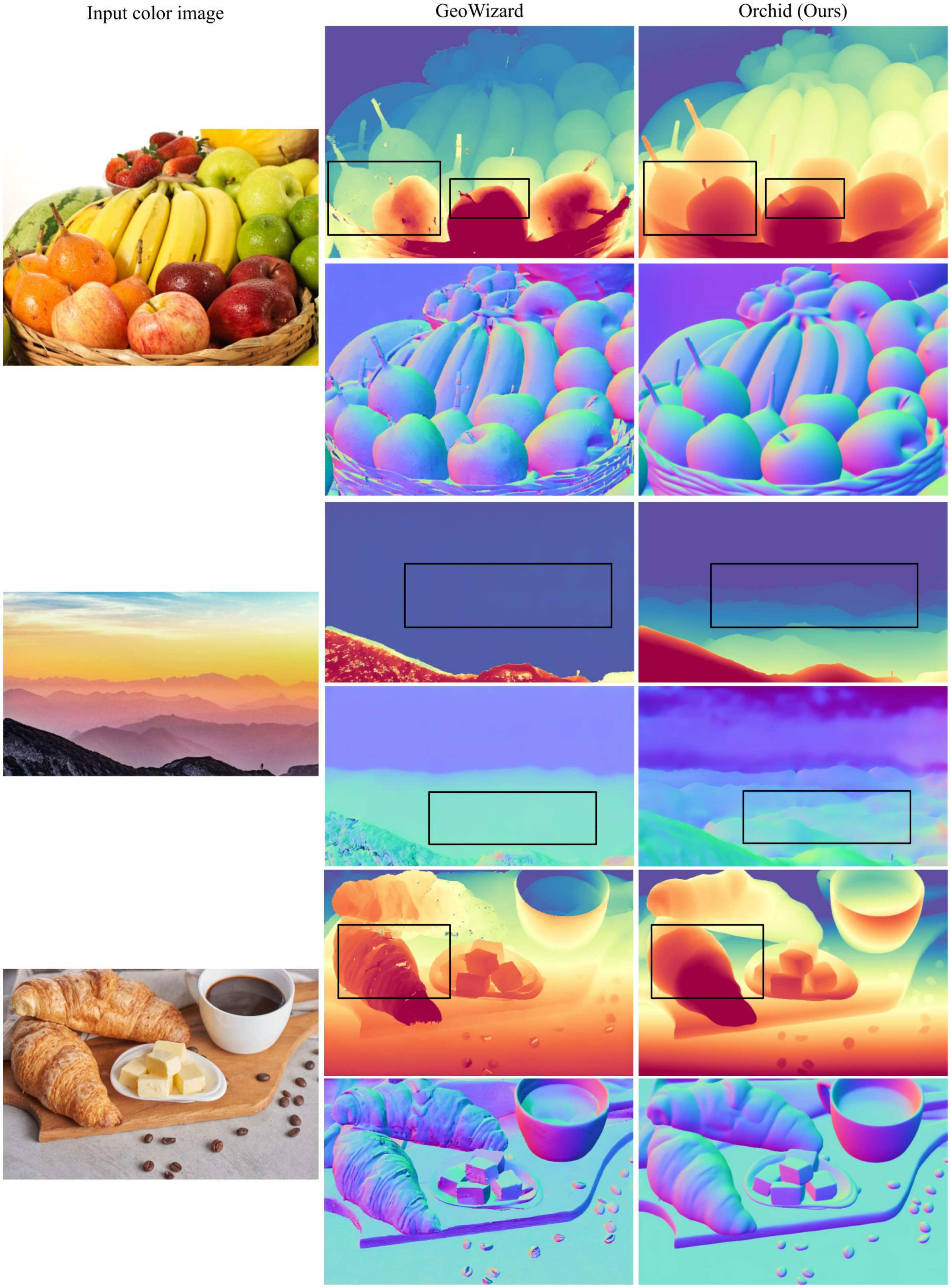

Figure 6. Comparison of GeoWizard [3] and Orchid on in-the-wild input images. Some areas with larger differences have been highlighted. In particular, we observe that high-frequency parts of the image can manifest themselves in noisy depth and normal predictions by GeoWizard (highlights on the fruits, texture of the croissants), whereas Orchid correctly predicts smooth surfaces. In far-away layered scenes we also observe that GeoWizard's predictions do not cover background (mountain range example).

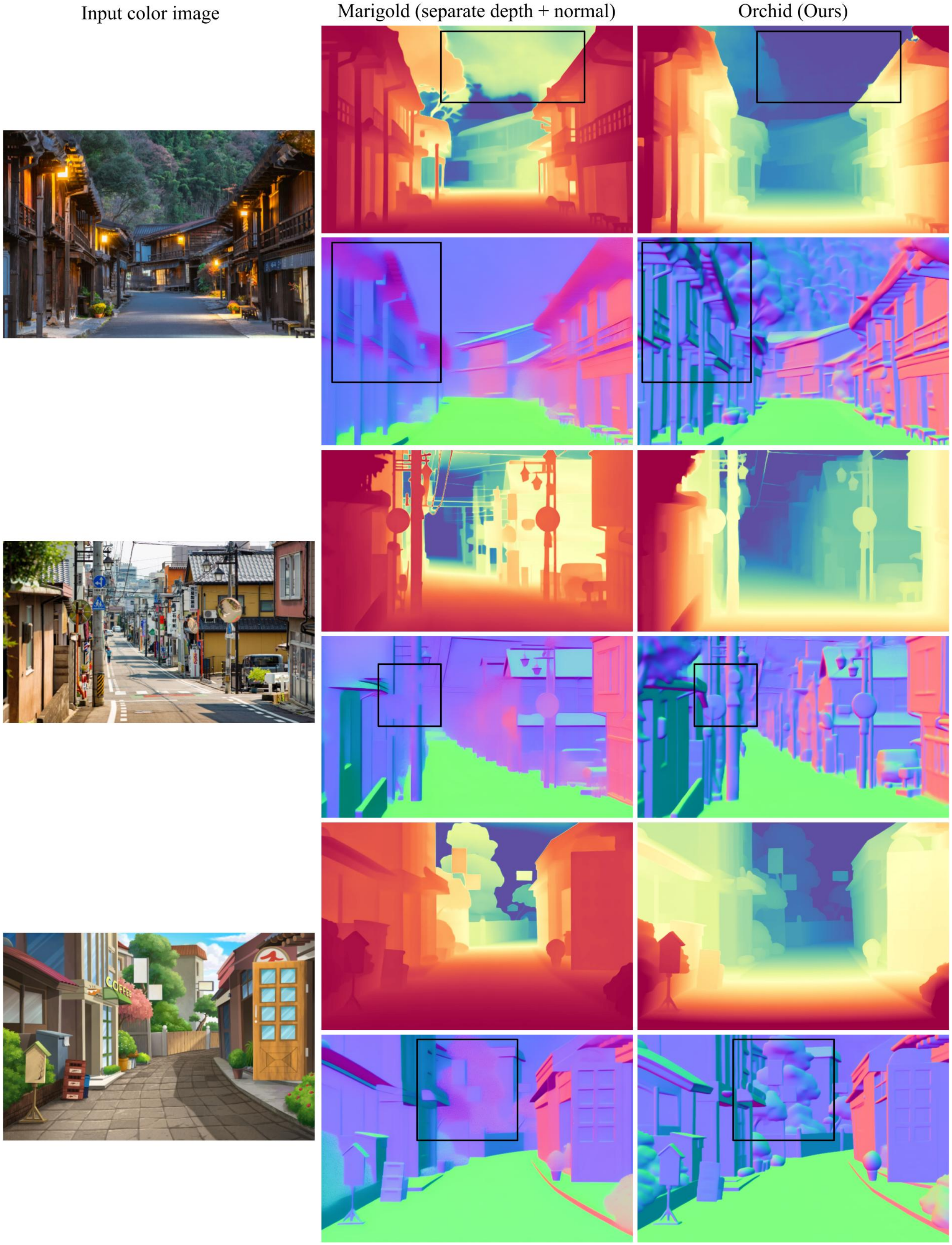

Figure 7. Comparison of Marigold [7] and Orchid on some in-the-wild input images. We use separate Marigold models to predict depth and normals. Orchid's joint predictions are better, especially for surface normals. Some notable differences are highlighted above.

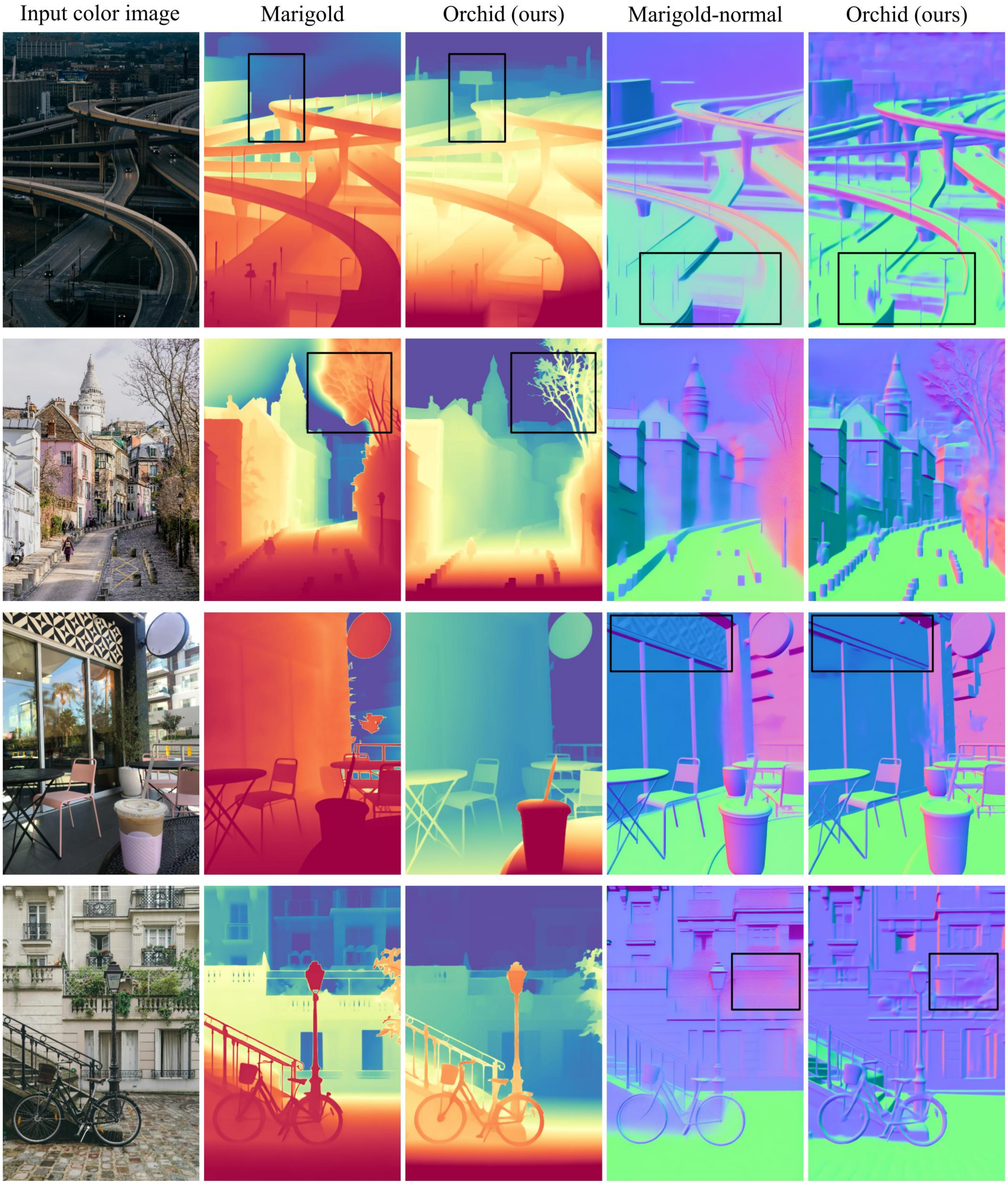

Figure 8. Comparison of Marigold [7] and Orchid on some in-the-wild input images. We can clearly see that our model Orchid can correctly predicts depth and surface normal of both far-away and nearby objects. Depth-maps from Orchid also has sharper and more accurate boundaries near pixels with depth discontinuities (*e.g*. between narrow tree branches and sky). Some of these are highlighted in the figure above.

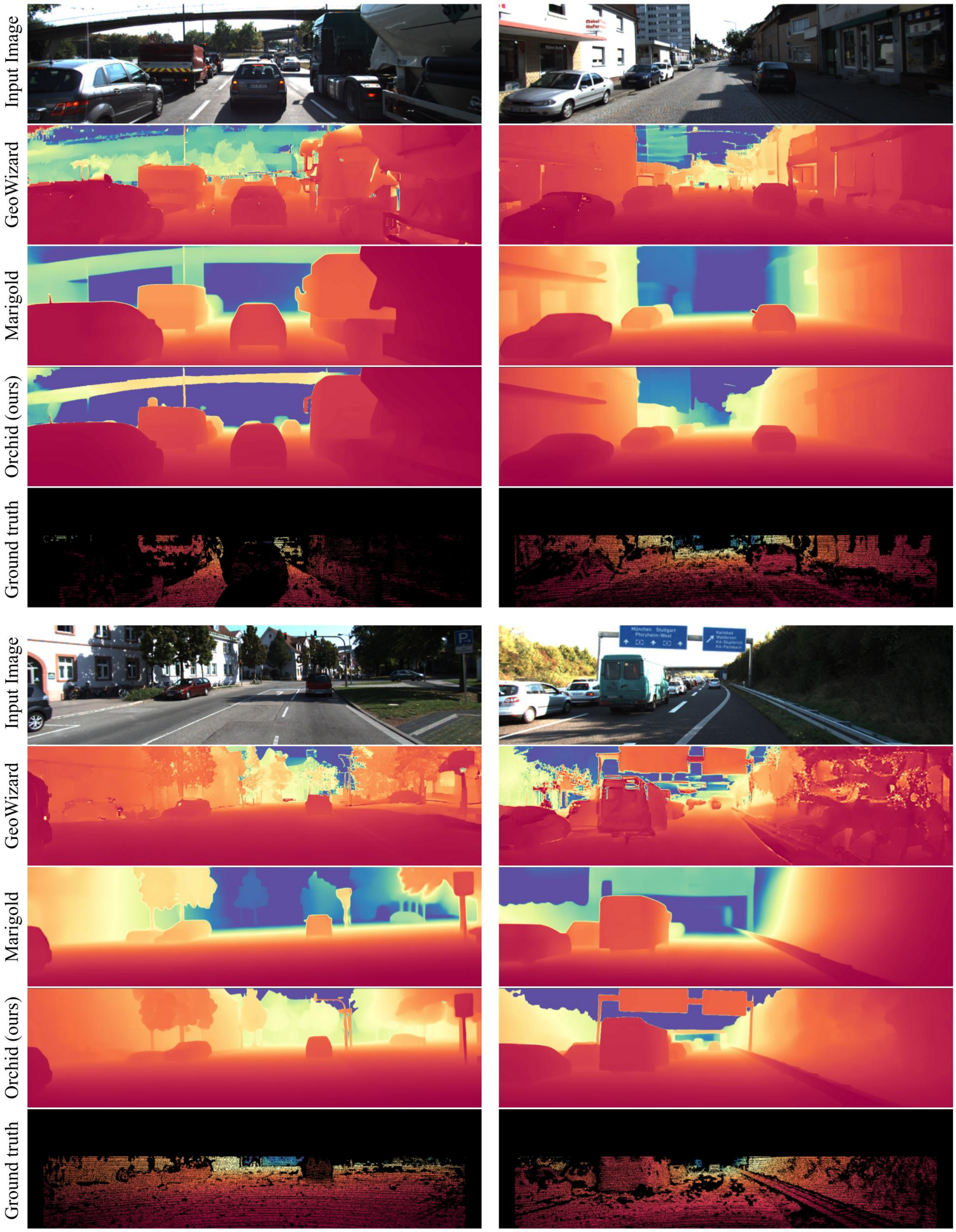

Figure 9. Qualitative comparison of monocular depth prediction on KITTI [4] dataset between GeoWizard [3], Marigold [7] and Orchid. Ground-truth depth (from lidar) are shown in the bottom row. Pixels without valid ground-truth depth are colored black. Orchid's predictions are significantly better, especially at longer ranges.

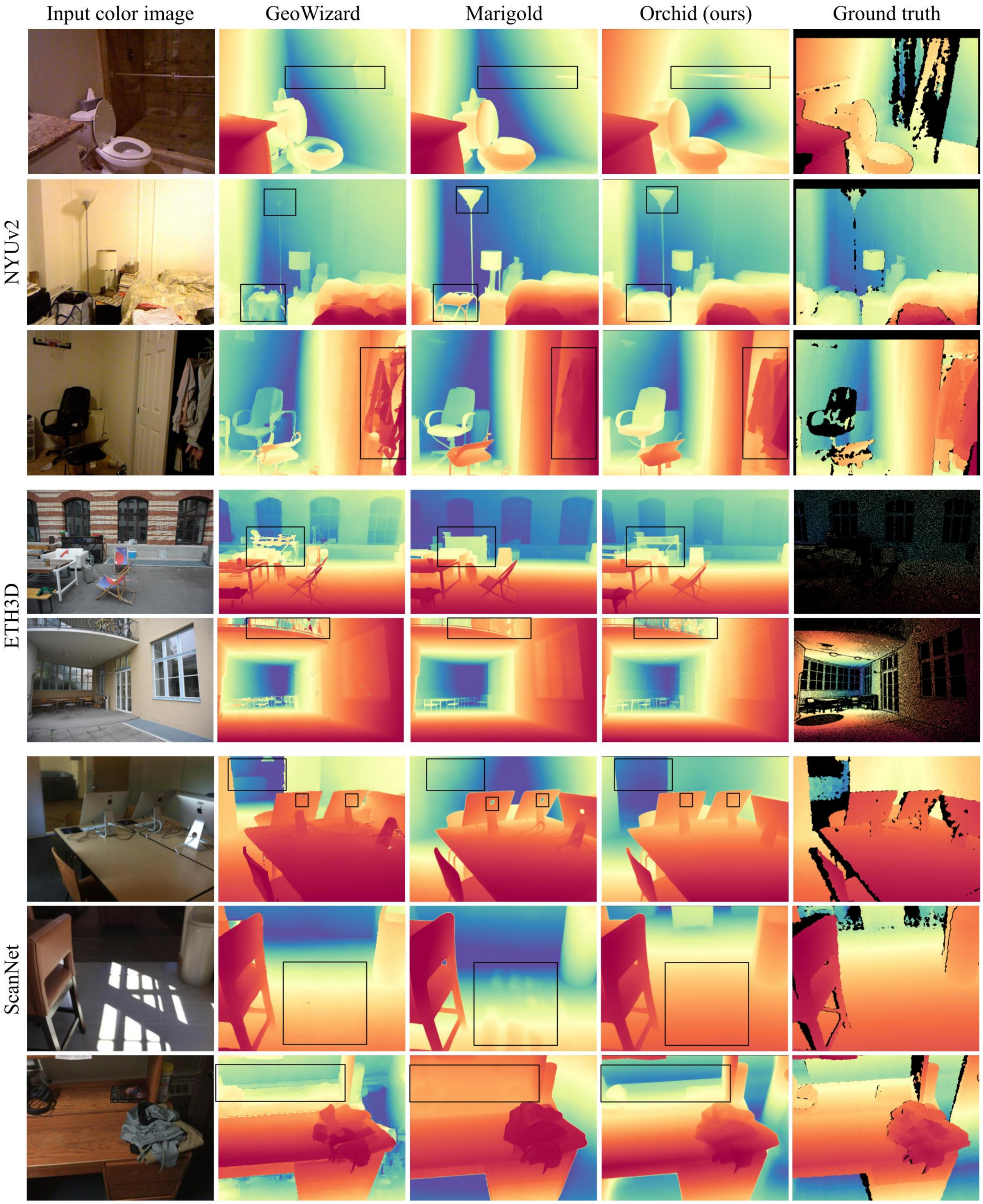

Figure 10. Comparison of monocular depth prediction results by GeoWizard [3], Marigold [7] and Orchid on NYUv2 [11], ETHD3D [14], and ScanNet [2] datasets. Ground-truth depth are shown in the rightmost column. Pixels without valid ground-truth depth are colored black. Our model Orchid has better depth predictions. Some notable differences are highlighted.

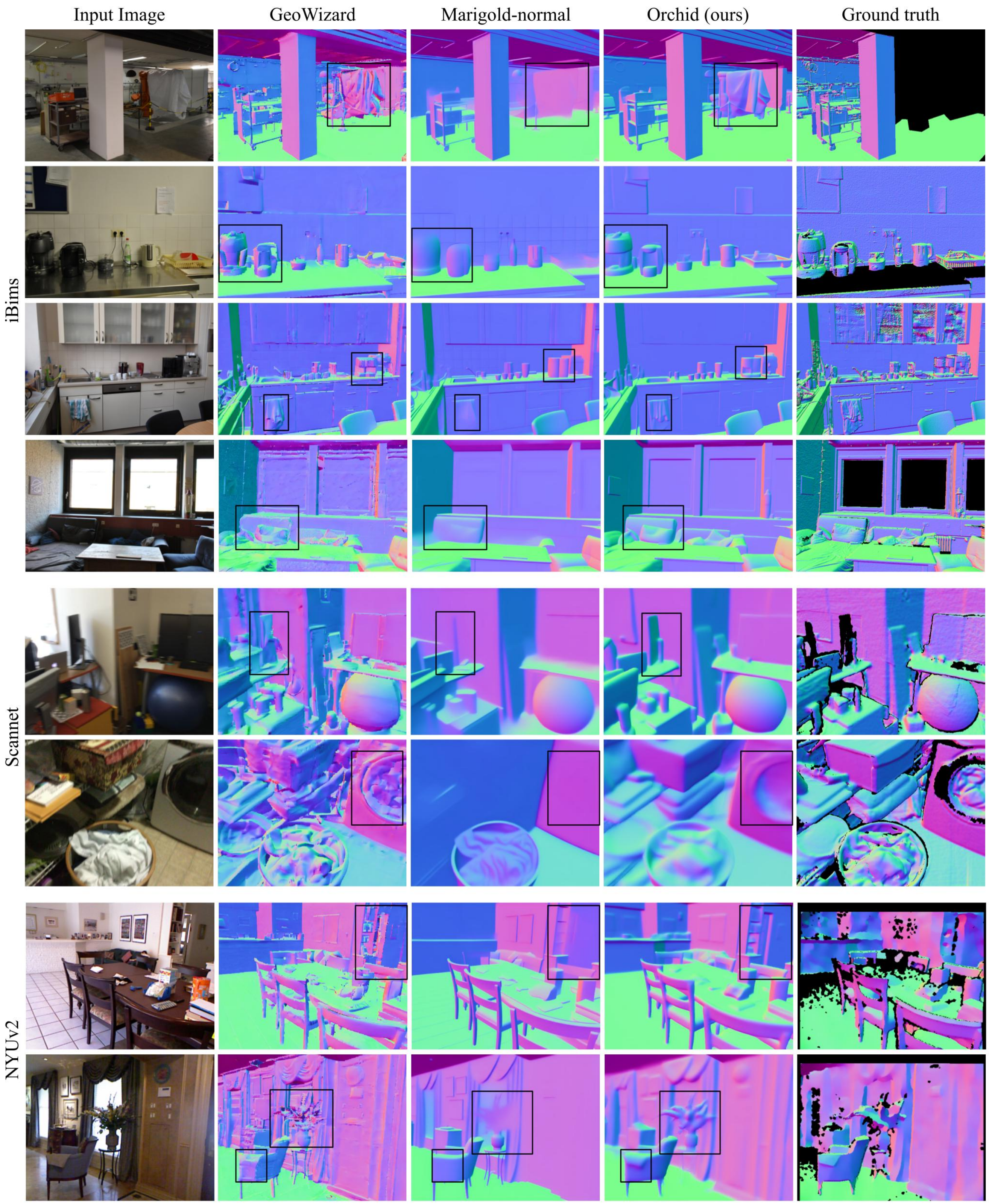

Figure 11. We compare single color image to surface-normal prediction methods of GeoWizard [3], Marigold [7] and Orchid on iBims [9], and ScanNet [2], and NYUv2 [11] datasets. Ground-truth normal are shown in the rightmost column. Pixels without valid ground-truth normal are colored black. Some notable differences are highlighted. Orchid's normals are significantly better than baselines.

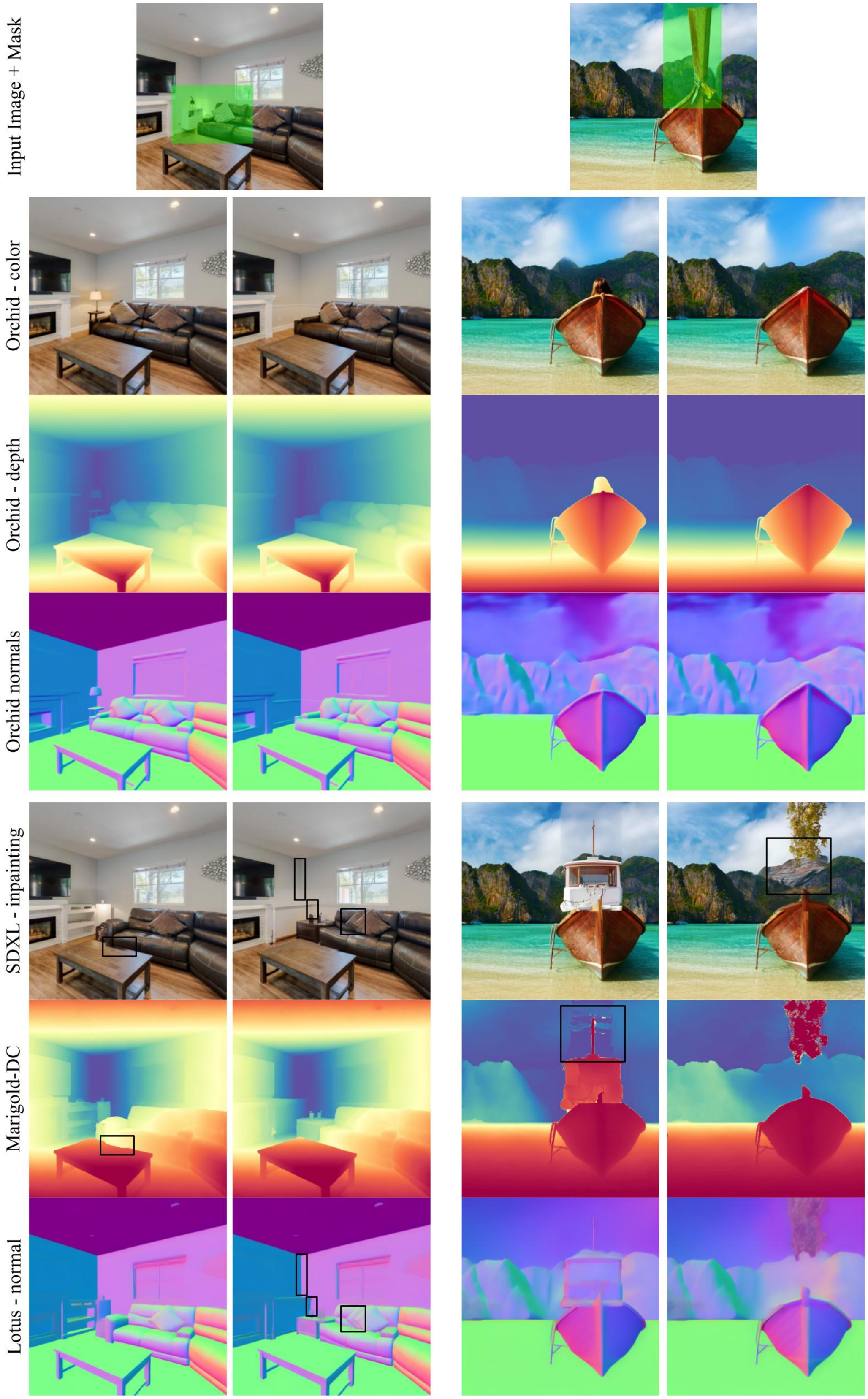

Figure 12. **Joint color-depth-normal inpainting**: Given color-depth-normal images with masked regions, our model inpaints them jointly. Masked-out pixels are shown with green overlays on the input images. Inpainted outputs from Orchid look very realistic. For *e.g.*, the edge of the wall is a continuous straight line, unlike the inpainting generated by a color-inpainting SDXL model. The inpainted results are also diverse (*e.g.* the table lamp, the shape of the canoe).

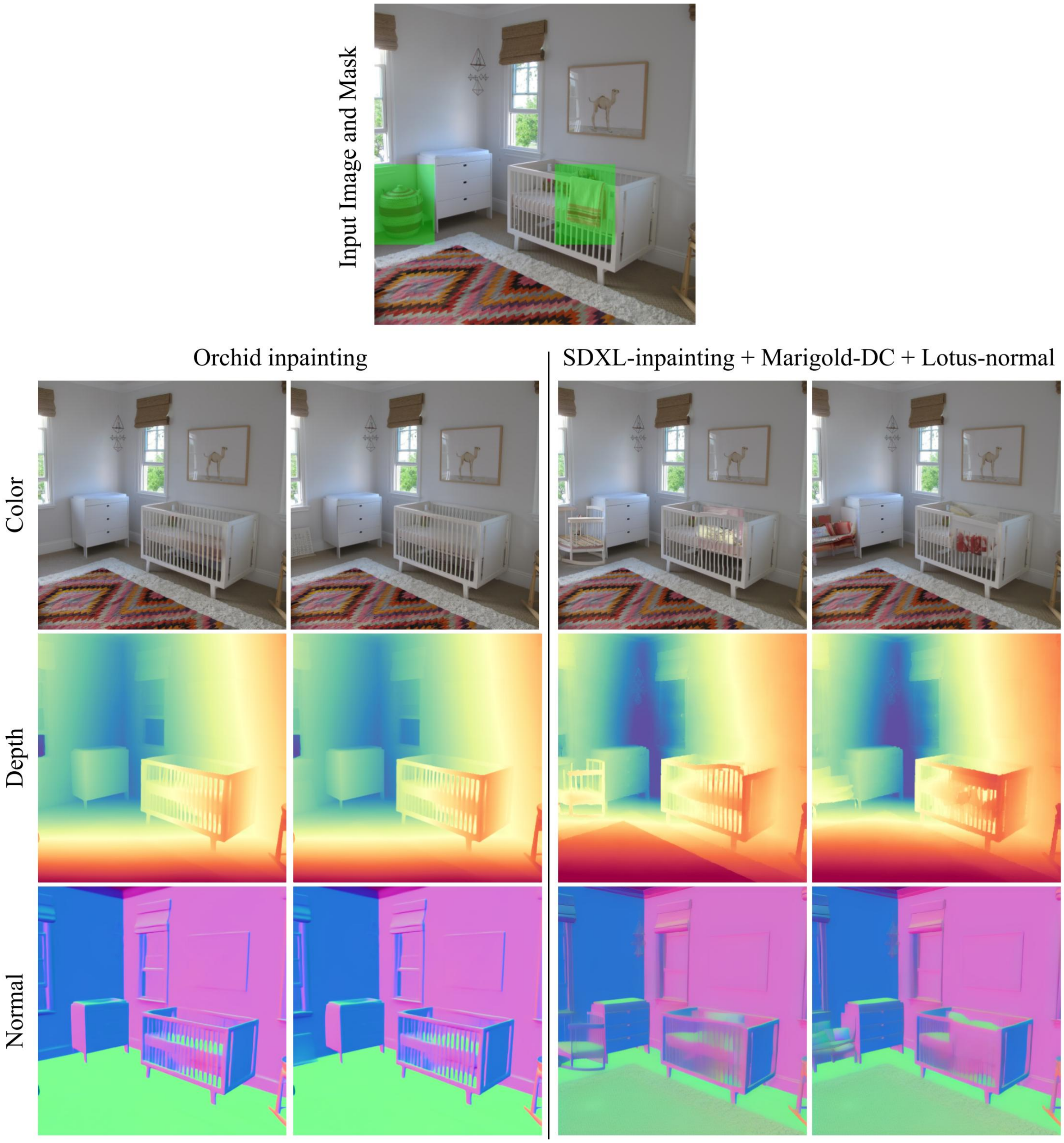

Figure 13. **Joint color-depth-normal inpainting**: (contd. from Figure 12) Our model inpaints color-depth-normals them jointly. Masked-out pixels are shown with green overlays on the input image. Inpainted outputs from Orchid are much more realistic, including geometric details such as the shape of the cradle. On the other hand, multimodal inpainting using existing baselines produce geometric artifacts and unrealistic results. When comparing results, please refer to our note on depth map visualization (Section 5.1).

References

[1] Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *CVPR*, 2024. 4, 6

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 12, 13

[3] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024. 2, 4, 7, 8, 11, 12, 13

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 11

[5] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Liu, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 4

[6] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 4

[7] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024. 2, 4, 5, 6, 9, 10, 11, 12, 13

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 4

[9] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *ECCV Workshops*, pages 0–0, 2018. 13

[10] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 4

[11] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 12, 13

[12] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 4

[13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1

[14] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, pages 3260–3269, 2017. 12

[15] Massimiliano Viola, Kevin Qu, Nando Metzger, Bingxin Ke, Alexander Becker, Konrad Schindler, and Anton Obukhov. Marigold-dc: Zero-shot monocular depth completion with guided diffusion, 2024. 4

[16] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding, 2024. 1

[17] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 4, 6