UniK3D: Universal Camera Monocular 3D Estimation

Supplementary Material

This supplementary material offers further insights into our work. In Sec. A we describe the network architecture in more detail, necessarily Sec. A overlaps with Sec. 3. Moreover, we analyze the complexity of UniK3D and compare it with other methods in Sec. A.1. Also, we provide further alternatives to our design choices and ablate them in Sec. A.2. Sec. B outlines the training pipeline and hyperparameters chosen in Sec. B.1, altogether with training and validation data in Sec. B.2, and the camera augmentations in Sec. B.3 for completeness and reproducibility. Furthermore, Sec. C provides a more detailed quantitative evaluation with per-dataset evaluation in Sec. C.2 The results corresponding to UniK3D finetuned on KITTI and NYUv2 are reported in Sec. C.1. In Sec. D, we provide answers to possible questions that may arise. Eventually, additional visualizations are provided in Sec. E.

A. Architecture

Encoder. Our model architecture employs a Vision Transformer (ViT) [13] as the encoder, demonstrating its effectiveness across different scales, from Small to Large. The ViT backbones were originally developed for classification tasks, and as such, we modify them by removing the final three layers: the pooling layer, the fully connected layer, and the $\mathrm{softmax}$ layer. We extract feature maps and class tokens from the last four layers of the modified ViT backbone. These outputs are flattened and processed using LayerNorm [4] followed by a linear projection layer. The linear layer maps the features and class tokens to a common channel dimension, which is set to 512, 384, and 256 for Large, Base, and Small ViT variants, respectively. Importantly, the normalization and linear layer weights are distinct and are not shared between the different feature resolutions and the class tokens. The dense feature maps are subsequently passed to the Radial Module, while the class tokens are directed to the Angular Module.

Angular Module. The four class tokens extracted from the encoder are first projected to dimensions of $3D$, $3D$, $5D$, and $7D$, respectively. These are then divided into chunks based on the channel dimension d, yielding token groups of size 3, 3, 5, and 7. These token groups serve as the initialization for domain tokens, representing the spherical harmonics (SH) coefficients: 1st-degree, 2nd-degree, and 3rd-degree, respectively. In total, there are 18 tokens ($\mathbf{T}$), which are processed through two layers of a Transformer Encoder. Each Transformer Encoder layer consists of self-attention with eight heads and a Multi-Layer Perceptron (MLP) that has a single hidden layer of dimension $4C$ and uses the Gaussian Error Linear Unit (GELU) activation function [19]. Both self-attention and MLP layers include residual connections to improve learning stability. Each of the 18 tokens is then projected to a scalar dimension. The first three tokens specifically define the domain for the spherical harmonics. The first token determines the horizontal field of view (HFov), calculated as $2\pi \cdot \sigma(\mathbf{T}_0)$, where σ denotes the sigmoid function. The second and third tokens represent the poles of the spherical harmonics, *i.e.* the center of projection relative to the image shape, computed as $c_x = \frac{\sigma(\mathbf{T}_1)W}{2}$ and $c_y = \frac{\sigma(\mathbf{T}_2)H}{2}$, respectively, where H and W are the image height and width. The vertical FoV is derived under the assumption of square pixels: $\mathrm{HFov} \times \frac{H}{W}$. Using this domain definition, we compute the spherical harmonics up to the 3rd degree, excluding the constant component, yielding 15 harmonic tensors of size $\mathbb{R}^{H\times W\times 3}$. The pencil of rays $\mathbf{C}$ is then constructed as a linear combination of these harmonics and the corresponding 15 processed tokens ($\mathbf{T_{3:18}}$).

Radial Module. The sine-encoded camera rays $\mathbf{C}$ are used to condition each resolution level of the dense feature maps $\mathbf{F}$ via a Transformer Decoder layer. In this setup, the dense features $\mathbf{F}$ serve as the *query*, while the sine-encoded camera rays provide the *keys* and *values*. The cross-attention mechanism includes a residual connection without any learnable gain factors, such as LayerScale. The conditioned features are then refined in a Feature Pyramid Network (FPN) manner: the deepest features are processed through two Residual Convolution blocks [18], followed by bilinear upsampling and a projection step that halves the channel dimension. These upsampled features are then combined with the features from the next layer, which are similarly projected to match channel dimension and upsampled using a single 2x2 transposed convolution. This process continues until all remaining three feature maps are consumed. The final output features are upsampled to the input image resolution and projected to a single-channel dimension, yielding the log-radius $\mathbf{R}_{\log}$. The same projection, architectural-wise but with separate weights, is used to generate the log-confidence $\mathbf{\Sigma}_{\log}$. The final radius and confidence values are obtained by exponentiating these tensors element-wise, transforming them from log-space to the original space.

A.1. Complexity

We perform a detailed analysis of the computational cost of UniK3D, presented in Table 7, and compare it to other state-of-the-art methods. To ensure a fair and consistent comparison, we use input sizes that are as similar as possible across all models. However, this approach introduces certain challenges. DepthPro, for instance, has an entangled and multi-resolution architecture, which complicates tuning the input

Table 7. **Parameters and efficiency comparison.** Comparison of performance of methods based on input size, latency, and number of trainable parameters. Tested on RTX3090 GPU, 16-bit precision float, and synchronized timers. The last two rows correspond to the Angular and Radial Modules evaluated independently. All models are based on ViT-L backbone.

Method	Input Size	Latency (ms)	Parameters (M)
ZoeDepth [7]	512×512	144.8	345.9
DepthAnything v2 [55]	518×518	78.1	334.7
UniDepth [42]	518×518	146.4	347.0
Metric3Dv2 [21]	518×518	135.6	441.9
MASt3R [21]	512×512	154.7	668.6
DepthPro [9]	1536×1536	808.1	952.0
UniK3D	518×518	88.4	358.8
Radial Module	-	21.9	38.2
Angular Module	-	3.1	12.1

size consistently across methods. Its architectural design does not easily allow for adjustments, making it difficult to align with a standardized input size. Additionally, the performance of models like DepthPro and Metric3D, as evaluated in our main experiments in Sec. 4, shows a significant drop when tested with image shapes that differ from those used during training. This sensitivity highlights a fundamental limitation: these methods are heavily optimized for specific image resolutions, and deviations from these resolutions can lead to substantial performance degradation. Consequently, while we strive to measure computation under the most equitable conditions, it is essential to acknowledge that these models are not well-suited for resolutions that differ from their training setup. In contrast, UniK3D is designed to be flexible w.r.t. image shape, maintaining robust performance across different resolutions. For our experiments, we chose the same input shape as DepthAnything v2, as it provides a balanced trade-off between computational efficiency and performance. Furthermore, to account for the asynchronous nature of CUDA kernel threading, we ensure precise inference time measurements by enabling proper synchronization and utilizing CUDA event recording. This approach guarantees an accurate reflection of computational cost, avoiding any misrepresentation caused by asynchronous operations. As shown in Table 7, UniK3D is among the most efficient models. The primary differences in computational cost, especially when compared to DepthAnything v2, stem from the inclusion of our Angular Module and Scale components. These components are essential for our model to handle absolute metric depth and camera-specific adjustments, features that relative depth estimation networks do not require. Despite this additional complexity, our model's efficiency remains competitive, underscoring its design's effectiveness in addressing diverse camera geometries while maintaining high performance.

A.2. Architectural Alternatives

Despite the camera conditioning has been proven superior in UniDepth [42], we ablate alternative architectural choices

Table 8. **Ablation on camera conditioning design.** *Camera Cond.* corresponds to the type of camera conditioning employed to condition the depth features with camera ones. *Add* refers to a simple addition in the feature space. *Cat* represents a simple concatenation and projection from $2C$ to C channel dimension. *Prompt* is our attention-based conditioning.

	Camera Cond.	S.FoV		S.FoV$_{\text{Dist}}$		L.FoV		Pano	
		$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$	$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$	$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$	$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$
1	Add	53.0	78.9	26.3	41.6	45.0	58.5	42.5	18.2
2	Cat	54.7	79.0	28.7	44.6	46.6	58.1	42.3	18.1
3	Prompt	57.3	79.8	44.6	59.3	53.5	64.8	58.6	26.3

Table 9. **Ablation on camera tokens processing.** *T-Enc.* indicates if the camera tokens are processed in the Angular Module either via the transformer encoder layer or not, in the latter case the tokens are fed directly to the final projections.

	T-Enc	S.FoV		S.FoV$_{\text{Dist}}$		L.FoV		Pano	
		$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$	$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$	$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$	$\mathrm{F_A}\uparrow$	$\rho_\mathrm{A}\uparrow$
1	✗	55.7	77.3	43.2	56.6	50.9	63.2	54.9	20.7
2	✓	57.3	79.8	44.6	59.3	53.5	64.8	58.6	26.3

for both the Transformer Encoder and Decoder components. In particular, we have chosen the most typical alternatives for conditioning: a simple addition or concatenation in place. While the camera tokens processing "alternative" involves an identity that shortcuts the camera tokens to the final projection layers. Table 9 shows how the camera tokens processing, via the encoder layer, does not present large changes, showing how the class tokens from different layers are already informative. However, Table 8 clearly shows how the simpler conditioning alternatives, such as addition or concatenation, underperform our attention-based conditioning. This highlights how conditioning plays an important role in final performance and how strongly designed conditioning is paramount to achieving proper generalization.

B. Training Details

B.1. Hyperparameters.

The training parameters, *i.e*. those for optimization, scheduling, and augmentations, are described in Table 10. The losses utilized, with the input and corresponding weights, are outlined in Table 11.

B.2. Data

Details of training and validation datasets are presented in Table 12 and Table 13.

Training Datasets. The datasets utilized for training are a mixture of different cameras and domains as shown in Table 12. The sequence-based datasets are sub-sampled during collection in a way that the interval between two consecutive frames is not smaller than half a second. No post-processing is applied. The total amount of training samples accounts for more than 8M samples. The datasets are sampled in each batch with a probability corresponding to the values in *Sampling* column in Table 12. This probability is related to the

Table 10. **Training Hyperparamters.** All training hyperparameters with corresponding values are presented.

Hyperparameter	Value
Steps	250k
Batch Size	128
LR	$5 \cdot 10^{-5}$
LR Encoder	$5 \cdot 10^{-6}$
Optimizer	AdamW [36]
(β_1, β_2)	$(0.9, 0.999)$
Weight Decay	0.1
Gradient Clip Norm	1.0
Precision	16-bit Float
LR Scheduler	Cosine to 0.1 start after 75k iters
EMA	0.9995 start after 75k iters
Color jitter prob	80%
Color jitter intensity	$[0.0, 0.5]$
Gamma prob	80%
Gamma intensity	$[0.5, 1.5]$
Horizontal flip prob	50%
Greyscale prob	20%
Gaussian blur prob	20%
Gaussian blur sigma	$[0.1, 2.0]$
Random zoom	$[0.5, 2.0]$
Random translation	$[-0.05, 0.05]$
Image ratio	$[1:2, 2:1]$
Resolution	0.28MP $[0.2\text{MP}, 0.6\text{MP}]$ last 50k iters

Table 11. **Training Losses.** Training losses with corresponding weight and input.

Loss	Inputs	Weight	Parameters
L1	Radius (log)	2.0 (η)	-
L1-asymmetric	Polar	0.75	$\alpha = 0.7$
L1	Azimuth	0.25	-
L1	Confidence (log), Radius error (detached)	0.1 (γ)	-

number of scenes present in each dataset. However, probabilities are changed based on a simple qualitative data inspection, such that the most diverse datasets are sampled more. Most of the datasets involve pinhole images or rectified cameras, *e.g.* MegaDepth [31] or NianticMapFree [3], other datasets provide only the pinhole calibration despite being clearly distorted, *i.e.* Mapillary [1], there the entire samples are masked out in the camera loss computation.

Validation Datasets. Table 13 presents all the validation datasets and splits them into 3 groups: small FoV, large FoV, and Panoramic. As per standard practice, KITTI Eigen-split corresponds to the corrected and accumulated GT depth maps with 45 images with inaccurate GT discarded from the original 697 images. The small FoV with distortion presented in Sec. 3 and used for evaluation is obtained based on synthesized cameras from ETH3D, Diode (Indoor), and IBims-1, all distorted images and cameras are manually checked for realism, after being generated with the pipeline presented in Sec. B.3.

Table 12. **Training Datasets.** List of the validation datasets: number of images, scene type, acquisition method, and sampling frequency are reported. SfM: Structure-from-Motion. MVS: Multi-View Stereo. Syn: Synthetic. Rec: Mesh reconstruction. KB: Kannala-Brandt [22]. Equi: Equirectangular

Dataset	Images	Scene	Acquisition	Camera	Sampling
A2D2 [17]	78k	Outdoor	LiDAR	Pinhole	2.5%
aiMotive [37]	178k	Outdoor	LiDAR	Mei [38]	0.3%
Argoverse2 [51]	403k	Outdoor	LiDAR	Pinhole	7.6%
ARKit-Scenes [5]	1.75M	Indoor	LiDAR	Pinhole	1.3%
ASE [14]	2.72M	Indoor	Syn	Fisheye624	10.1%
BEDLAM [8]	24k	Various	Syn	Pinhole	2.0%
BlendedMVS [56]	114k	Outdoor	MVS	Pinhole	2.5%
DL3DV [34]	306k	Outdoor	SfM	KB [22]	4.7%
DrivingStereo [53]	63k	Outdoor	MVS	Pinhole	2.5%
DynamicReplica [23]	120k	Indoor	Syn	Pinhole	1.3%
EDEN [27]	368k	Outdoor	Syn	Pinhole	2.5%
FutureHouse [32]	28.3	Indoor	Syn	Equi	2.5%
HOI4D [35]	59k	Egocentric	RGB-D	KB [22]	1.7%
HM3D [43]	540k	Indoor	Rec	Pinhole	5.2%
Matterport3D [11]	10.8k	Indoor	Rec	Equi	2.0%
Mapillary PSD [1]	742k	Outdoor	SfM	Pinhole	2.0%
MatrixCity [30]	190k	Outdoor	Syn	Pinhole	5.0%
MegaDepth [31]	273k	Outdoor	SfM	Pinhole	8.0%
NianticMapFree [3]	25k	Outdoor	SfM	Pinhole	2.0%
PointOdyssey [61]	33k	Various	Syn	Pinhole	1.7%
ScanNet [12]	83k	Indoor	RGB-D	Pinhole	5.0%
ScanNet++ [57]	39k	Indoor	Rec	Pinhole	3.0%
TartanAir [50]	306k	Various	Syn	Pinhole	5.5%
Taskonomy [60]	1.94M	Indoor	RGB-D	Pinhole	6.0%
Waymo [46]	223k	Outdoor	LiDAR	Pinhole	7.5%
WildRGBD [52]	1.35M	Indoor	RGB-D	Pinhole	7.5%

Table 13. **Validation Datasets.** List of the validation datasets: number of images, scene type, acquisition method, and max evaluation distance are reported. 1st group: small FoV, 2nd group: large FoV, 3rd: Panoramic. Rec: Mesh reconstruction.

Dataset	Images	Scene	Acquisition	Max Distance
KITTI [16]	652	Outdoor	LiDAR	80.0
NYU [39]	654	Indoor	RGB-D	10.0
IBims-1 [25]	100	Indoor	RGB-D	25.0
Diode [48]	325	Indoor	LiDAR	25.0
ETH3D [44]	454	Outdoor	RGB-D	50.0
NuScenes [10]	3.6k	Outdoor	LiDAR	80.0
ScanNet++ [57]	779	Indoor	Rec	10.0
ADT [40]	469	Indoor	Rec	20.0
KITTI360 [33]	527	Outdoor	LiDAR	80.0
Stanford-2D3D [2]	1413	Indoor	Rec	10.0

B.3. Camera Augmentations

To address the limited diversity of distorted camera data, we augment images captured with pinhole cameras by artificially deforming them, thereby simulating images from distorted camera models, *e.g.* Fisheye624 or radial Kannala-Brandt [22]. The augmentation process involves two main steps. First, we compute a deformation field. This starts with unprojecting the 2D depth map obtained from a pin-

Table 14. **Camera Sampling for S.FoV$_{\text{Dist}}$ generation.** The parameters to generate S.FoV$_{\text{Dist}}$ images are listed. We employed different camera models with different parameter ranges. The sampling is uniform sampling within the ranges. The seed is 13.

Model	Probability	Parameter	Range
EUCM	0.1	α β	$[0, 1]$ $[0.25, 4]$
Fisheye624	0.35	$\{k_i\}_{i=1}^{6}$ $\{t_i\}_{i=1}^{2}$ $\{s_i\}_{i=1}^{4}$	$[0.6, 0.8]$ $[-0.01, 0.01]$ $[-0.01, 0.01]$
Fisheye624	0.35	$\{k_i\}_{i=1}^{6}$ $\{t_i\}_{i=1}^{2}$ $\{s_i\}_{i=1}^{4}$	$[-0.6, -0.4]$ $[-0.01, 0.01]$ $[-0.01, 0.01]$
Fisheye624	0.2	$\{k_i\}_{i=1}^{6}$ $\{t_i\}_{i=1}^{2}$ $\{s_i\}_{i=1}^{4}$	$[-0.2, 0.2]$ $[-0.05, 0.05]$ $[-0.05, 0.05]$

Table 15. **Camera Sampling for Camera Augmentation.** The parameters to generate an augmented camera during training images are listed. We employed different camera models with different parameter ranges. The sampling is uniform sampling within the ranges. When some parameters are not listed, *e.g.* $\{k_i\}_{i=4}^{6}$ for Kannala-Brandt model, they are set to 0.

Model	Probability	Parameter	Range
EUCM	0.1	α β	$[0, 1]$ $[0.25, 4]$
Fisheye624	0.15	$\{k_i\}_{i=1}^{6}$ $\{t_i\}_{i=1}^{2}$ $\{s_i\}_{i=1}^{4}$	$[0.1, 0.5]$ $[-0.005, 0.005]$ $[-0.01, 0.01]$
Fisheye624	0.15	$\{k_i\}_{i=1}^{6}$ $\{t_i\}_{i=1}^{2}$ $\{s_i\}_{i=1}^{4}$	$[-0.5, -0.1]$ $[-0.005, 0.005]$ $[-0.01, 0.01]$
Kannala-Brandt	0.2	$\{k_i\}_{i=1}^{3}$ $\{t_i\}_{i=1}^{2}$	$[-0.05, 0.05]$ $[-0.02, 0.02]$
Kannala-Brandt	0.4	$\{k_i\}_{i=1}^{3}$ $\{t_i\}_{i=1}^{2}$	$[-0.5, 0.5]$ $[-0.001, 0.001]$

hole camera into a 3D point cloud. We then project these 3D points onto the image plane of a randomly sampled distorted camera model to obtain the new 2D coordinates. The deformation field is defined as the distance between the original 2D image coordinates and the newly projected 2D coordinates. This flow indicates how the original image should be warped to mimic the appearance of a distorted camera view. Next, we warp the image using softmax-based splatting [45], a technique that projects pixels based on the computed deformation field while preserving image details. To ensure the warping process does not create artifacts like holes, we use an "importance" metric, which is the inverse of the depth value for each pixel. This metric prioritizes closer points, ensuring that details and correct parallax are maintained during the warping. For non-synthetic images, where ground-truth depth maps are unavailable, we generate depth predictions in an inference-only mode to compute the deformation. To ensure these predictions are accurate enough to create realistic deformations, we apply this augmentation only after the model has been trained for 10,000 steps. By this point, the model has learned a decently reliable (scale-invariant) depth representation. The specific camera parameters used to sample the new random camera are listed in Table 15.

Table 16. **Comparison on NYU validation set.** All models are trained on NYU. The first four are trained only on NYU. The last four are fine-tuned on NYU.

Method	δ_1	δ_2	δ_3	A.Rel	RMS	Log_{10}
	Higher is better			*Lower is better*		
BTS [28]	88.5	97.8	99.4	10.9	0.391	0.046
AdaBins [6]	90.1	98.3	99.6	10.3	0.365	0.044
NeWCRF [59]	92.1	99.1	99.8	9.56	0.333	0.040
iDisc [41]	93.8	99.2	99.8	8.61	0.313	0.037
ZoeDepth [7]	95.2	99.5	99.8	7.70	0.278	0.033
Metric3Dv2 [21]	**98.9**	**99.8**	**100**	4.70	0.183	0.020
DepthAnythingv2 [55]	98.4	**99.8**	**100**	5.60	0.206	0.024
UniK3D	**98.9**	**99.8**	**100**	**4.43**	**0.173**	**0.019**

Table 17. **Comparison on KITTI Eigen-split validation set.** All models are trained on KITTI E-ign-split training and tested on the corresponding validator split. The first are trained only on KITTI. The last 4 are fine-tuned on KITTI.

Method	δ_1	δ_2	δ_3	A.Rel	RMS	$\text{RMS}_{\log}$
	Higher is better			*Lower is better*		
BTS [28]	96.2	99.4	99.8	5.63	2.43	0.089
AdaBins [6]	96.3	99.5	99.8	5.85	2.38	0.089
NeWCRF [59]	97.5	99.7	99.9	5.20	2.07	0.078
iDisc [41]	97.5	99.7	99.9	5.09	2.07	0.077
ZoeDepth [7]	96.5	99.1	99.4	5.76	2.39	0.089
Metric3Dv2 [58]	98.5	**99.8**	**100**	4.40	1.99	**0.064**
DepthAnythingv2 [55]	98.3	**99.8**	**100**	4.50	1.86	0.067
UniK3D	**99.0**	**99.8**	99.9	**3.69**	**1.68**	**0.060**

Validation datasets generation. Generating validation datasets for testing models on distorted images with reduced fields of view presents an additional challenge, as most distortions are typically associated with large fields of view. To simulate this, we use synthetic camera parameters to deform RGB images from datasets such as ETH3D [44], IBims-1 [25], and Diode (Indoor) [48]. These datasets are chosen because they provide nearly complete ground-truth depth maps, making the deformation process well-posed and realistic. Any small gaps or holes in the depth maps are filled using inpainting. Importantly, the 3D ground-truth data remains unchanged, as it is invariant to the camera model used. To ensure realism, we manually validate each deformed image and will release both the code for data generation and the resulting validation data.

C. Additional Quantitative Results

C.1. Fine-tuning

We evaluate the fine-tuning capability of UniK3D by resuming training with either KITTI or NYU as the sole training dataset. The fine-tuning process starts from the weights and optimizer states obtained after the large-scale pretraining phase, ensuring a fair and consistent initialization. The standard SILog loss is used as the training objective, with a batch size of 16, and the model is trained for an additional 40,000 steps. To focus the evaluation on the impact of in-domain data, we disable all augmentations except for horizontal flipping and omit the asymmetric component of the angular loss during fine-tuning. For evaluation, we adhere to the standard practices for both datasets to ensure comparability with prior work. KITTI results are reported using the Garg [15] evaluation crop, and the maximum evaluation depths for KITTI and NYU are set to 80 and 10 meters, respectively mportantly, we do not apply any test-time augmentations or tuning, such as varying the input size, to maintain consistency and avoid introducing additional confounding factors. Our results demonstrate that UniK3D benefits significantly from in-domain fine-tuning. Table 17 highlights the model's ability to perform exceptionally well on highly structured and calibrated datasets like KITTI, even though UniK3D is inherently designed for flexibility and cross-domain generalization. This suggests that the model can effectively adapt to well-structured data when fine-tuned. This fine-tuning analysis highlights the adaptability of UniK3D to diverse settings while maintaining its primary design focus on flexibility. Similarly, Table 16 shows that UniK3D remains competitive when fine-tuned on less structured domains like NYU, which represent typical indoor environments. These results reinforce the importance of in-domain data for achieving optimal performance, particularly on datasets with distinct properties or domain-specific challenges. In addition, the results underline the robustness of our model, as it achieves strong performance across significantly different dataset characteristics.

Table 18. **Comparison on zero-shot evaluation for NYUv2.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel ↓	RMSE ↓	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	97.8	-	-
DepthAnythingv2 [55]	-	-	-	97.7	-	-
Metric3D†‡ [58]	68.1	44.2	1.23	89.0	-	-
Metric3Dv2†‡ [21]	93.4	9.1	0.399	98.1	-	-
ZoeDepth† [7]	94.2	8.2	0.305	98.0	-	-
UniDepth [42]	**98.0**	**7.3**	**0.230**	**99.0**	**83.1**	**99.2**
MASt3R [29]	83.9	13.0	0.435	94.8	69.6	90.7
DepthPro [9]	92.2	10.1	0.357	97.2	73.0	93.1
UniK3D-Small	90.4	11.2	0.351	97.4	69.1	83.0
UniK3D-Base	93.1	10.3	0.325	97.9	75.4	89.1
UniK3D-Large	96.5	7.4	0.259	98.2	82.5	91.2

Table 19. **Comparison on zero-shot evaluation for KITTI.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel ↓	RMSE ↓	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	88.5	-	-
DepthAnythingv2 [55]	-	-	-	88.4	-	-
Metric3D†‡ [58]	3.3	49.8	10.35	97.0	-	-
Metric3Dv2†‡ [21]	2.3	56.3	12.81	96.7	-	-
ZoeDepth† [7]	93.6	8.2	3.24	96.7	-	-
UniDepth [42]	**98.0**	**4.8**	**2.14**	**98.3**	**85.8**	**97.5**
MASt3R [29]	2.8	58.2	11.88	90.9	10.9	77.7
DepthPro [9]	78.2	17.2	5.27	94.8	62.4	80.9
UniK3D-Small	92.1	11.6	3.76	96.4	77.7	85.6
UniK3D-Base	93.1	12.6	3.84	97.3	76.6	82.7
UniK3D-Large	81.2	17.4	4.77	96.8	71.4	79.3

Table 20. **Comparison on zero-shot evaluation for IBims-1.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel ↓	RMSE ↓	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	97.0	-	-
DepthAnythingv2 [55]	-	-	-	98.0	-	-
Metric3D†‡ [58]	75.1	19.3	0.633	96.2	-	-
Metric3Dv2†‡ [21]	68.4	20.7	0.700	**98.8**	-	-
ZoeDepth† [7]	49.8	21.5	0.989	95.8	-	-
UniDepth [42]	15.7	41.0	1.25	98.1	30.3	**76.6**
MASt3R [29]	61.0	19.7	0.883	95.1	55.7	76.0
DepthPro [9]	82.3	17.0	0.573	98.0	62.8	75.9
UniK3D-Small	87.7	13.0	0.484	97.7	67.3	74.6
UniK3D-Base	87.6	12.5	0.452	98.0	67.5	73.4
UniK3D-Large	**91.9**	**10.4**	**0.406**	98.5	**69.8**	75.4

Table 21. **Comparison on zero-shot evaluation for ETH3D.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel ↓	RMSE ↓	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	93.2	-	-
DepthAnythingv2 [55]	-	-	-	93.3	-	-
Metric3D†‡ [58]	19.7	136.8	10.45	81.1	-	-
Metric3Dv2†‡ [21]	**90.0**	**12.7**	**1.85**	89.7	-	-
ZoeDepth† [7]	33.8	54.7	3.45	86.1	-	-
UniDepth [42]	18.5	53.3	3.50	93.9	27.6	42.6
MASt3R [29]	21.4	45.3	4.43	91.3	28.4	**92.2**
DepthPro [9]	39.7	65.2	36.31	81.1	41.2	77.4
UniK3D-Small	53.6	60.0	4.89	94.2	44.3	80.7
UniK3D-Base	68.4	28.5	3.77	95.8	**53.8**	82.0
UniK3D-Large	68.7	23.6	2.63	**95.9**	53.6	81.3

C.2. Per-dataset Evaluation

We present results for each of the validation datasets independently in Table 18 (NYUv2), Table 19 (KITTI), Table 20 (IBims-1), Table 21 (ETH3D), Table 22 (Diode Indoor), Table 23 (nuScenes), Table 24 (IBims-1_{Dist}), Table 25

Table 22. **Comparison on zero-shot evaluation for Diode (Indoor).** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	97.5	-	-
DepthAnythingv2 [55]	-	-	-	97.6	-	-
Metric3D†‡ [58]	40.4	61.1	2.34	91.3	-	-
Metric3Dv2†‡ [21]	**94.0**	**9.3**	**0.399**	**98.5**	-	-
ZoeDepth† [7]	34.9	33.6	2.07	91.8	-	-
UniDepth [42]	76.2	17.2	0.954	97.2	**63.0**	**96.1**
MASt3R [29]	52.6	27.9	1.68	92.3	48.8	70.2
DepthPro [9]	67.1	19.9	0.900	93.9	50.3	71.5
UniK3D-Small	57.2	21.4	0.968	96.1	49.3	92.5
UniK3D-Base	55.1	19.6	0.859	97.4	50.1	91.2
UniK3D-Large	71.3	16.1	0.767	97.9	53.8	79.5

Table 23. **Comparison on zero-shot evaluation for nuScenes.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	79.0	-	-
DepthAnythingv2 [55]	-	-	-	79.4	-	-
Metric3D†‡ [58]	75.4	23.7	8.94	64.0	-	-
Metric3Dv2†‡ [21]	84.1	23.6	9.40	64.8	-	-
ZoeDepth† [7]	33.8	42.0	7.41	64.8	-	-
UniDepth [42]	84.6	**12.7**	**4.56**	83.1	64.4	97.7
MASt3R [29]	2.7	65.6	13.76	63.5	13.6	78.3
DepthPro [9]	56.6	28.7	11.29	59.1	46.5	79.1
UniK3D-Small	80.9	18.9	8.43	83.8	59.4	95.8
UniK3D-Base	**84.9**	16.7	9.15	86.7	**65.5**	**97.8**
UniK3D-Large	84.0	18.9	10.83	**87.0**	60.3	86.9

Table 24. **Comparison on zero-shot evaluation for $\mathbf{IBims\text{-}1_{Dist}}$.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	97.1	-	-
DepthAnythingv2 [55]	-	-	-	93.4	-	-
Metric3D†‡ [58]	56.8	26.5	0.947	93.3	-	-
Metric3Dv2†‡ [21]	61.3	22.1	0.940	93.3	-	-
ZoeDepth† [7]	30.0	28.0	1.28	94.5	-	-
UniDepth [42]	48.7	23.0	0.966	97.2	53.3	69.3
MASt3R [29]	31.8	31.9	1.30	92.8	44.1	69.7
DepthPro [9]	27.2	47.6	1.86	83.0	32.4	69.5
UniK3D-Small	67.2	17.1	0.726	97.6	62.6	71.5
UniK3D-Base	66.0	17.9	0.695	98.3	59.8	72.7
UniK3D-Large	**70.9**	**15.0**	**0.615**	**98.6**	**67.9**	**77.3**

Table 25. **Comparison on zero-shot evaluation for $\mathbf{ETH3D_{Dist}}$.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	91.8	-	-
DepthAnythingv2 [55]	-	-	-	83.9	-	-
Metric3D†‡ [58]	19.6	123.6	11.05	80.9	-	-
Metric3Dv2†‡ [21]	42.8	104.3	9.87	83.5	-	-
ZoeDepth† [7]	25.4	45.9	4.12	86.1	-	-
UniDepth [42]	27.6	43.8	4.69	90.1	38.5	67.5
MASt3R [29]	14.6	51.8	5.37	87.7	32.0	78.5
DepthPro [9]	16.1	72.8	18.77	72.7	29.1	69.9
UniK3D-Small	42.1	125.3	12.14	92.9	49.9	68.4
UniK3D-Base	47.9	36.5	3.54	95.1	53.5	67.1
UniK3D-Large	**67.0**	**22.1**	**2.75**	**95.5**	**63.6**	**83.1**

Table 26. **Comparison on zero-shot evaluation for $\mathbf{Diode_{Dist}}$ (Indoor).** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	94.2	-	-
DepthAnythingv2 [55]	-	-	-	89.3	-	-
Metric3D†‡ [58]	26.4	124.0	4.08	89.7	-	-
Metric3Dv2†‡ [21]	**34.1**	35.2	1.61	91.6	-	-
ZoeDepth† [7]	24.0	39.8	2.32	90.1	-	-
UniDepth [42]	30.2	34.8	1.85	94.7	**37.2**	74.8
MASt3R [29]	20.6	46.0	2.41	89.3	29.5	83.0
DepthPro [9]	24.7	56.5	2.31	86.0	26.5	75.7
UniK3D-Small	27.6	33.4	1.48	95.0	33.0	82.6
UniK3D-Base	31.6	30.0	1.35	96.1	37.0	85.1
UniK3D-Large	26.9	**30.0**	**1.33**	**97.5**	36.1	**85.4**

Table 27. **Comparison on zero-shot evaluation for ScanNet++ (DSLR).** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_\mathrm{A} \uparrow$
DepthAnything [54]	-	-	-	51.4	-	-
DepthAnythingv2 [55]	-	-	-	52.3	-	-
Metric3D†‡ [58]	16.5	180.5	1.83	51.2	-	-
Metric3Dv2†‡ [21]	5.2	237.0	2.51	71.3	-	-
ZoeDepth† [7]	2.0	158.5	1.45	71.2	-	-
UniDepth [42]	0.6	162.9	1.59	71.0	9.1	20.2
MASt3R [29]	5.8	114.8	1.07	73.0	21.0	16.6
DepthPro [9]	9.6	95.8	0.928	74.1	24.4	30.9
UniK3D-Small	6.2	92.8	0.931	78.1	23.5	35.1
UniK3D-Base	55.4	33.1	0.340	86.6	53.9	65.1
UniK3D-Large	**65.1**	**25.3**	**0.285**	**90.8**	**59.1**	**70.0**

($\mathrm{ETH3D_{Dist}}$), Table 26 (Diode $\mathrm{Indoor_{Dist}}$), Table 27 (ScanNet++ DSLR), Table 28 (ADT), and Table 29 (KITTI360). Note that we do not report results for the "Pano" group, as it only consists of a single dataset, Stanford-2D3D. Our results show that performance on pinhole camera models has reached a saturation point, yet UniK3D achieves the highest average metric overall, even though it does not always rank first on every individual dataset. This demonstrates the strong generalization ability of UniK3D, attributed to its flexible design and large-scale training, which enables robust performance across diverse domains without overfitting to any specific one. We report additional and more typical metrics such as absolute relative error as $\mathrm{A.Rel}$ as a percentage and root-means-squared error RSME using meter as unit.

Table 28. **Comparison on zero-shot evaluation for ADT.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_{\mathrm{A}} \uparrow$
DepthAnything [54]	-	-	-	81.7	-	-
DepthAnythingv2 [55]	-	-	-	82.6	-	-
Metric3D†‡ [58]	72.5	26.2	0.560	85.3	-	-
Metric3Dv2†‡ [21]	75.6	21.9	0.433	92.4	-	-
ZoeDepth† [7]	11.0	81.4	1.36	83.5	-	-
UniDepth [42]	13.3	76.0	1.37	90.8	27.1	32.1
MASt3R [29]	44.8	40.1	0.717	86.7	52.5	51.4
DepthPro [9]	33.6	45.1	0.902	81.3	47.9	48.0
UniK3D-Small	89.8	13.4	0.323	93.8	82.9	92.2
UniK3D-Base	93.5	10.3	0.288	95.0	88.1	**93.8**
UniK3D-Large	**94.6**	**9.3**	**0.275**	**95.6**	**89.5**	93.7

Table 29. **Comparison on zero-shot evaluation for KITTI360.** Missing values (-) indicate the model's inability to produce the respective output. †: ground-truth camera for 3D reconstruction. ‡: ground-truth camera for 2D depth map inference.

Method	$\delta_1 \uparrow$	A.Rel $\downarrow$	RMSE $\downarrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_{\mathrm{A}} \uparrow$
DepthAnything [54]	-	-	-	9.5	-	-
DepthAnythingv2 [55]	-	-	-	11.3	-	-
Metric3D†‡ [58]	0.2	1366.2	34.78	39.7	-	-
Metric3Dv2†‡ [21]	0.1	1655.3	40.32	43.9	-	-
ZoeDepth† [7]	0.7	1200.2	24.71	41.2	-	-
UniDepth [42]	29.4	152.2	4.23	44.0	14.6	7.1
MASt3R [29]	16.5	312.8	7.17	41.7	15.7	7.4
DepthPro [9]	5.5	103.8	7.35	38.0	5.9	17.5
UniK3D-Small	74.9	39.8	2.58	81.6	59.5	**82.8**
UniK3D-Base	73.3	33.8	2.62	80.8	61.2	80.9
UniK3D-Large	**81.7**	**24.4**	**2.40**	**85.3**	**66.4**	82.5

Table 30. **Comparison with UniDepth.** All models use ViT-S backbone and the same training data. Test set grouping as in the main paper. Best viewed on a screen and zoomed in.

Method	Small FoV			Small $\mathrm{FoV_{Dist}}$			Large FoV			Panoramic		
	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_{\mathrm{A}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_{\mathrm{A}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_{\mathrm{A}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\mathrm{F_A} \uparrow$	$\rho_{\mathrm{A}} \uparrow$
UniDepth [42]	89.0	54.7	77.8	92.7	35.4	45.6	71.8	41.9	48.8	34.9	1.5	1.2
UniK3D	**89.1**	**57.3**	**79.8**	**93.1**	**44.6**	**59.3**	**79.8**	**53.5**	**64.8**	**64.3**	**58.6**	**26.3**

D. Q&A

Here we list possible questions that might arise after reading the paper. We structure the section in a discursive question-and-answer fashion.

- **What is the importance of data for generalization w.r.t. scene scale?**
  Data diversity is crucial for generalizing depth estimation, especially for monocular methods that heavily rely on semantic cues and are sensitive to domain gaps. Scale prediction in monocular metric depth estimation is inherently ill-posed, making it highly dependent on the training domain and its distribution coverage. Excessive diversity can hurt performance in narrow, specialized domains like KITTI, where models trained on large, diverse datasets often underperform compared to those trained on domain-specific data. Conversely, these models perform better in broader domains like NYU. Scale prediction is typically noisy and sensitive to domain shifts, but this issue can be mitigated through in-domain fine-tuning. For example, a few hundred optimization steps can largely resolve the "scale gap" when fine-tuning on KITTI.
- **The camera representation is superior to pinhole or fully non-parametric camera model, but you did not compare it to some common camera models, why so?**
  We initially experimented with explicit parametric camera models but encountered significant drawbacks. Most standard camera models rely on backprojection operations which are not differentiable and, thus cannot be used in a standard deep learning pipeline. Addressing this limitation requires either (i) using differentiable parametric models, such as EUCM [24] or DoubleSphere [47], (ii) approximating polynomial inversions with differentiable functions, or (iii) supervising only the model parameters without direct camera supervision. All these approaches suffer from the inherent instability of parametric models, where parameter variations need to be considered jointly on their actual output, namely the pencil of rays. This compounding effect, where small compounded changes lead to large output variations, often leads to unstable optimization. Furthermore, parametric models limit the expressiveness of the backprojection operation and constrain applicability to only those cameras the model can represent. In contrast, our representation avoids these limitations and provides greater flexibility and stability.
- **DUSt3R / MASt3R architecture directly predicts point maps, are they unable to work with generic cameras?**
  While DUSt3R and MASt3R networks can theoretically represent any camera model, our studies revealed that fully non-parametric approaches struggle when trained on diverse datasets and tested on edge cases or distribution tails. Additionally, the test-time point cloud global alignment technique used in DUSt3R [49] and MASt3R [29] explicitly requires a pinhole camera, further limiting their applicability to generic cameras.
- **What is the role of the confidence prediction?**
  Confidence prediction is included primarily for its utility in downstream tasks and also for legacy reasons. It is worth noting that, like most regression tasks, confidence prediction is vulnerable to domain gaps, which can render it unreliable in strong out-of-domain scenarios.
- **What is the rationale of camera augmentations?**
  Camera augmentations were employed to address the lack of diverse real-camera data. While our simple augmentation pipeline resulted in minor improvements, we observed that many generated cameras are unrealistic and fall outside the distribution of real-world cameras. However, softmax-based warping proved effective in creating realistic images. We hypothesize that a more sophisticated camera sampling procedure, considering the realism of the output rays instead of the singled-out parameters, could

significantly enhance the robustness and generalization across real and practical camera models.

- **What are the differences with UniDepth?**
UniDepth [42] and UniK3D differ in camera modeling and 3D representation, both ablated in Tabs. 3, 4, and 5. UniDepth relies on the *pinhole model* by predicting the calibration matrix (cf. [60, Sec. 3.2]), thus not being able to predict *any* camera. In addition, [42] represents the 3rd dimension as *depth* (z) [60, Sec. 3.1]. These two aspects force [42] to model *only* pinhole and to output FoV $< 180°$. In contrast, UniK3D uses spherical harmonics (SH) to approximate *any* camera model and it exploits *radial distance* (r) as 3rd dimension. UniDepth projects the predicted *pinhole ray map* [60, Sec. 3.1] onto a high-dimensional space $\mathbf{E}$ using SH, whereas UniK3D directly *predicts the SH coefficient* used to generate the ray map $\mathbf{C}$ via inverse transform (L230-262). This key methodological difference leads to modeling any camera. Table 30 (row 1 *vs*. row 3) shows its impact, as UniK3D consistently outperforms [42] also when trained on identical data.
- **Has someone done something similar before?**
Yes, there are a few works [20, 26] which tried to remove the pinhole assumption for depth estimation. However, they are different for two important reasons: (i) those works focused on single-domain scenarios, leading to a simpler setting and (ii) the task is self-supervised depth estimation, where the camera is needed to define the warping-based photometric loss, inherently needing the camera, rather than supervised large-scale monocular 3D estimation.
- We provide here the δ_1^{SSI} scores of row 3 and 4 of Tab. 5: 92.1 and 92.2, respectively. This score similarity, along with $\mathrm{F_A}$ and ρ_A drops (Tab. 5), spotlights angular module's role. In fact, radial- and SH-based model (row 4) overestimates FoV of images with lens distortions. Retraining with stronger distortion augmentation for small FoV leads to $(\mathrm{F_A}, \rho_\mathrm{A}) = (43.1, 62.3)$, validating our assumption.

E. Additional Qualitative Results

We provide here more qualitative comparisons, in particular from validation domains not presented in the main paper and with distorted cameras, namely ScanNet++ (DSLR), IBims-1_{Dist}, and $\mathrm{Diode_{Dist}}$ (Indoor), in Fig. 5. In addition, we test our model on complete in-the-wild scenarios, for instance, frames from movies, TV series, YouTube, or animes. All images depicted in Fig. 6 and Fig. 7 present deformed cameras or unusual points of view. The visualization here presented, both from the validation sets and the in-the-wild ones are casually selected and not cherry-picked.

References

[1] Manuel Lopez Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *The European Conference on Computer Vision (ECCV)*, pages 589–604. Springer International Publishing, 2020. 3

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 3

[3] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[5] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Advances in Neural Information Processing Systems (NIPS)*, 2021. 3

[6] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4008–4017, 2020. 4

[7] 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. 2, 4, 5, 6, 7

[8] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, 2023. 3

[9] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 2, 5, 6, 7

[10] 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)*, 2020. 3

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017. 3

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at

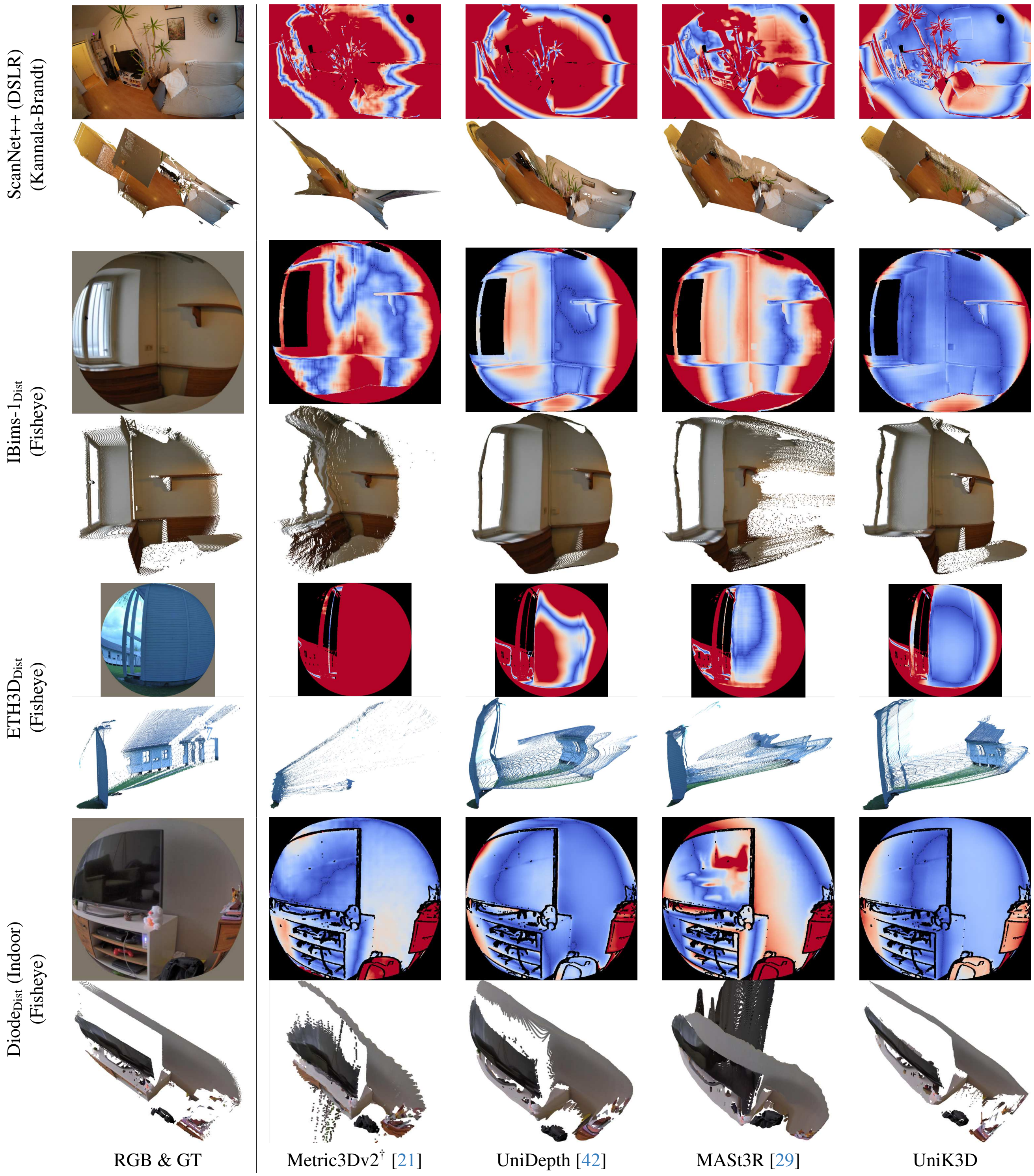

Figure 5. **Qualitative comparisons.** Each pair of consecutive rows represents one test sample. Each odd row displays the input RGB image and the 2D error map, color-coded with the *coolwarm* colormap based on absolute relative error with blue corresponding to 0% error and red to 25%. To ensure a fair comparison, errors are calculated on GT-based shifted and scaled outputs for all models. Each even row shows the ground truth and predictions of the 3D point cloud. All samples are randomly selected and not picked. †: GT-camera unprojection.

scale. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021. 1

Figure 6. **Qualitative in-the-wild 3D results.**UniK3D is fed solely each single image in the left column and it outputs the corresponding point cloud in the right column, the point of view is slightly tilted to better appreciate the 3D. The images are video frames respectively from Poor Things (movie), The Revenant (movie), Eminem (music video), and YouTube (egocentric GoPro). The frames present a variety of camera types and unusual viewpoints.

Figure 7. **Qualitative in-the-wild 3D results.** UniK3D is fed solely each single image in the left column and it outputs the corresponding point cloud in the right column, the point of view is slightly tilted to better appreciate the 3D. The images are video frames respectively from Trainspotting (movie), YouTube (doorbell camera), Naruto (anime), and Breaking Bad (TV series). The frames present a variety of camera types and unusual viewpoints.

[14] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023. 3

[15] Ravi Garg, B. G. Vijay Kumar, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *Lecture Notes in Computer Science*, 9912 LNCS:740–756, 2016. 5

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3

[17] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2D2: Audi Autonomous Driving Dataset. *arXiv preprint arXiv:2004.06320*, 2020. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016-December:770–778, 2015. 1

[19] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. 1

[20] Noriaki Hirose and Kosuke Tahara. Depth360: Self-supervised learning for monocular depth estimation using learnable camera distortion model. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 317–324, 2021. 8

[21] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 2, 4, 5, 6, 7, 9

[22] Juho Kannala and Sami Sebastian Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI*, 28:1335–1340, 2006. 3

[23] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[24] Bogdan Khomutenko, Gaëtan Garcia, and Philippe Martinet. An enhanced unified camera model. *IEEE Robotics and Automation Letters (RA-L)*, 1:137–144, 2016. 7

[25] Tobias Koch, Lukas Liebel, Marco Körner, and Friedrich Fraundorfer. Comparison of monocular depth estimation methods using geometrically relevant metrics on the IBims-1 dataset. *Computer Vision and Image Understanding (CVIU)*, 191:102877, 2020. 3, 4

[26] Varun Ravi Kumar, Senthil Kumar Yogamani, Markus Bach, Christian Witt, Stefan Milz, and Patrick Mäder. Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8177–8183, 2020. 8

[27] Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. Eden: Multimodal synthetic dataset of enclosed garden scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1579–1589, 2021. 3

[28] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *CoRR*, abs/1907.10326, 2019. 4

[29] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 5, 6, 7, 9

[30] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3205–3215, 2023. 3

[31] 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)*, pages 2041–2050, 2018. 3

[32] Zhen Li, Lingli Wang, Xiang Huang, Cihui Pan, and Jiaqi. Yang. Phyir: Physics-based inverse rendering for panoramic indoor images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[33] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022. 3

[34] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22160–22169, 2024. 3

[35] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, 2022. 3

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*, 2017. 3

[37] Tamas Matuszka, Ivan Barton, Ádám Butykai, Péter Hajas, Dávid Kiss, Domonkos Kovács, Sándor Kunsági-Máté, Péter Lengyel, Gábor Németh, Levente Pető, Dezső Ribli, Dávid Szeghy, Szabolcs Vajna, and Balint Viktor Varga. aimotive dataset: A multimodal dataset for robust autonomous driving with long-range perception. In *International Conference on Learning Representations (ICLR) Workshop on Scene Representations for Autonomous Driving*, 2023. 3

[38] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. *Proceed-*

ings of the IEEE International Conference on Robotics and Automation (ICRA), pages 3945–3950, 2007. 3

[39] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *The European Conference on Computer Vision (ECCV)*, 2012. 3

[40] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20133–20143, 2023. 3

[41] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. iDisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[42] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10106–10116, 2024. 2, 5, 6, 7, 8, 9

[43] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Advances in Neural Information Processing Systems (NIPS)*, 2021. 3

[44] Thomas Schöps, Johannes L. Schönberger, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 4

[45] Niklaus Simon and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[46] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 3

[47] Vladyslav C. Usenko, Nikolaus Demmel, and Daniel Cremers. The double sphere camera model. *International Conference on 3D Vision (3DV)*, pages 552–560, 2018. 7

[48] Igor Vasiljevic, Nicholas I. 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*, abs/1908.00463, 2019. 3, 4

[49] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024. 7

[50] 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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 3

[51] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Advances in Neural Information Processing Systems*, 2021. 3

[52] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22378–22389, 2024. 3

[53] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[54] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10371–10381, 2024. 5, 6, 7

[55] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 2, 4, 5, 6, 7

[56] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1790–1799, 2020. 3

[57] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[58] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9043–9053, 2023. 4, 5, 6, 7

[59] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3906–3915. IEEE, 2022. 4

[60] Amir R Zamir, Alexander Sax, William B Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 3

[61] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale

synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19855–19865, 2023. 3