Learning Depth Estimation for Transparent and Mirror Surfaces

Alex Costanzino* Pierluigi Zama Ramirez* Matteo Poggi*

Fabio Tosi Stefano Mattoccia Luigi Di Stefano

CVLAB, Department of Computer Science and Engineering (DISI)

University of Bologna, Italy

{alex.costanzino, pierluigi.zama, m.poggi, fabio.tosi5}@unibo.it

https://cvlab-unibo.github.io/Depth4ToM/

Abstract

Inferring the depth of transparent or mirror (ToM) surfaces represents a hard challenge for either sensors, algorithms, or deep networks. We propose a simple pipeline for learning to estimate depth properly for such surfaces with neural networks, without requiring any ground-truth annotation. We unveil how to obtain reliable pseudo labels by in-painting ToM objects in images and processing them with a monocular depth estimation model. These labels can be used to fine-tune existing monocular or stereo networks, to let them learn how to deal with ToM surfaces. Experimental results on the Booster dataset show the dramatic improvements enabled by our remarkably simple proposal.

1. Introduction

In our daily lives, we often interact with several objects of various appearances. Among them are those made of transparent or mirror surfaces (ToM), ranging from the glass windows of buildings to the reflective surfaces of cars and appliances. These might represent a hard challenge for an autonomous agent leveraging computer vision to operate in unknown environments. Specifically, among the many tasks involved in Spatial AI, accurately estimating depth information on these surfaces remains a challenging problem for both computer vision algorithms and deep networks [64], yet necessary for proper interaction with the environment in robotic, autonomous navigation, picking, and other application fields. This difficulty arises because ToM surfaces introduce misleading visual information about scene geometry, which makes depth estimation challenging not only for computer vision systems but even for humans – e.g., we might not distinguish the presence of a glass door in front of us due to its transparency. On the one hand, the definition of depth itself might appear ambiguous in such cases:

---

*These authors contributed equally to this work.
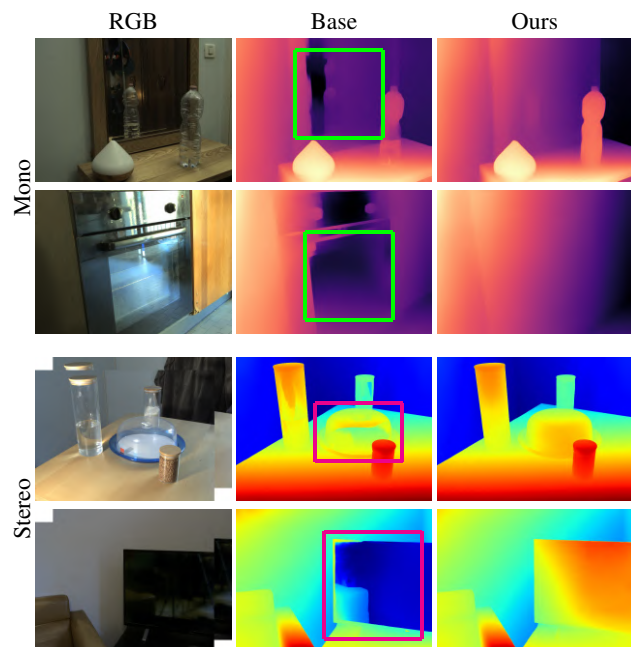
Figure 1. **Depth estimation on ToM surfaces.** Two examples for both monocular (top) and stereo (bottom) images. In the central column, the depth/disparity maps predicted by DPT [37] and CREStereo [22] original weights. In the rightmost column, the depth/disparity maps predicted by the models after being fine-tuned by our strategy without exploiting any ground-truth depth.

is *depth* the distance to the scene behind the glass door or to the door itself? Nonetheless, from a practical point of view, we argue that the actual definition depends on the task itself – e.g., a mobile robot should definitely be aware of the presence of the glass door. On the other hand, as humans can deal with this through experience, depth sensing techniques based on deep learning, e.g., monocular [38, 37] or stereo [26, 22] networks, hold the potential to address this challenge given sufficient training data [64].

Unfortunately, light reflection and refraction over ToM

surfaces violate also the working principles of most active depth sensors, such as Time-of-Flight (ToF) cameras or devices projecting structured-light patterns. This has two practical consequences: i) it makes active sensors unsuited to deal with ToM objects in real-world applications, and ii) prevents the use of these sensors for collecting and annotating data to train deep neural networks to deal with ToM objects. As evidence of this, very few datasets featuring transparent objects provide ground-truth depth annotations, which have been obtained through very intensive human intervention [64], graphical engines [40], or based on the availability of CAD models [5] for ToM objects.

In short, accurately perceiving the presence (and depth) of ToM objects represents an open challenge for both sensing technologies and deep learning frameworks. Purposely, this paper proposes a simple yet effective strategy for obtaining training data and, thereby, dramatically boosting the accuracy of learning-based depth estimation frameworks dealing with ToM surfaces. Driven by the observation that ToM objects alone are responsible for misleading recent monocular networks [38, 37], which would otherwise generalize well to most unseen environments, we argue that *replacing* them with equivalent, yet opaque objects would allow restoring an environment layout in which such networks could accurately estimate the depth of the scene. To this end, we mask ToM objects in images by in-painting them with arbitrary uniform colors. Then, we employ a monocular depth network to generate a *virtual* depth map out of the modified image. By repeating this process on a variety of images featuring ToM objects, we can easily and effectively annotate a dataset and then use it to train the same monocular network used to distill labels, which will now process the not-in-painted images. As a result, the trained monocular network will learn to handle ToM objects, producing consistent depth even in their presence.

Our main contributions can be resumed as follows:

- We propose a simple yet very effective strategy to deal with ToM objects. We trick a monocular depth estimation network by replacing ToM objects with virtually textured ones, inducing it to hallucinate their depths.

- We introduce a processing pipeline for fine-tuning a monocular depth estimation network to deal with ToM objects. Our pipeline exploits the network itself to generate virtual depth annotations and requires only segmentation masks delineating ToM objects – either human-made or predicted by other networks [54, 60] – thus getting rid of the need for any depth annotations.

- We show how our strategy can be extended to other depth estimation settings, such as stereo matching. Our experiments on the Booster dataset [64] prove how monocular and stereo networks dramatically improve their prediction on ToM objects after being fine-tuned according to our methodology.

Fig. 1 highlights some specific regions where monocular (top) and stereo (bottom) models struggle (middle column), and how they learn to handle ToM surfaces thanks to our strategy (rightmost column).

2. Related Work

Monocular Depth Estimation. Early methods used CNNs for pixel-level regression [10, 11]. More recent approaches such as AdaBins [2], DPT [37], and MiDaS [38] use adaptive bins and vision transformers for depth regression and leverage large-scale depth training by mixing multiple datasets. Self-supervised methods use view synthesis for image reconstruction, where predicted depth is combined with known or estimated camera pose to establish correspondences between adjacent images, exploiting either stereo pairs [10, 11] or monocular videos [70, 12]. Recent works aim to improve the robustness of the photometric loss based on SSIM and L1 [68] by incorporating photometric uncertainty [57, 35], feature descriptors [66, 44, 46], 3D geometric constraints [29], proxy supervision [52, 49], optical flow [62, 50], or adversarial losses [1, 34]. Others propose architecture changes as in [69, 14, 33, 18, 13]. Except for some works that address non-Lambertian depth estimation using depth completion approaches and sparse depth measurements from active sensors [8, 40], to the best of our knowledge, we are not aware of any previous single-view depth estimation network that can handle ToM surfaces.

Stereo Matching. Traditional algorithms [42] utilize handcrafted features to estimate a disparity map [63, 17, 59, 58, 24, 47, 21, 3]. Then, deep learning methods replaced traditional matching cost computation, as demonstrated in [65], and, eventually, end-to-end approaches became the most effective solution for disparity estimation. These networks can be mainly categorized into 2D and 3D architectures, with the former adopting an encoder-decoder design [30, 32, 25, 39, 45, 56, 61, 48] and the latter building a feature cost volume from extracted features on the image pair [19, 4, 20, 67, 6, 7, 9, 55, 51, 16, 43]. A thorough review of these works can be found in [36]. Recent papers exploit iterative refinement paradigms [26, 22] or rely on Vision Transformers [23, 15]. However, due to its inherently ill-posed nature, dealing with non-Lambertian surfaces, such as ToM objects, remains a very challenging problem for any kind of existing stereo approach.

Non-Lambertian Object Perception. Due to the relevance of dealing with ToM objects, some recent datasets focus on them. Trans10K [54] and MSD [60] consist of over 10 000 and 4 000 real in-the-wild images of transparent objects and mirrors, respectively. Both datasets provide manually annotated segmentations of ToM materials, though
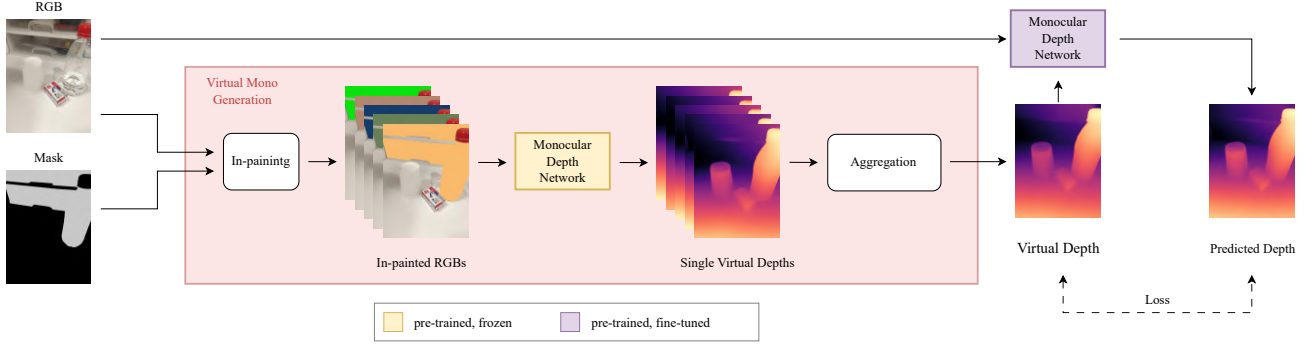
Figure 2. **Monocular Distillation pipeline.** Given an RGB and a segmentation mask, we in-paint pixels belonging to transparent and mirror surfaces with a random uniform color and process these augmented images with a pre-trained monocular network. The obtained virtual depths are aggregated to obtain a pseudo-labeled dataset for fine-tuning the network itself.

none of them provide depth labels. Others provide depth annotations: ClearPose [5] includes over $350\,000$ labeled real-world RGB-D frames of 63 household objects. Clear-Grasp [40] consists of over $50\,000$ synthetic RGB-D images of transparent objects, as well as real-world test benchmark with 286 RGB-D images. In addition, Booster [64] focuses on stereo matching, providing high-resolution depth labels and stereo pairs acquired in indoor scenes with specular and transparent surfaces. TOD [28] contains 15 transparent objects, labeled with relevant 3D keypoints, comprising $48\,000$ stereo and RGBD images. StereOBJ-1M [27] also deals with stereo vision, but focuses on pose estimation for ToM objects and do not provides depth ground truths. Obtaining depth labels for these kinds of datasets is expensive, challenging, and time-consuming since it requires either CAD models for ToM objects [5], painting such objects in the scene [40, 64, 28] or relies on a complex multi-camera setup [53]. In contrast, our proposal effectively sidesteps these challenges, by demonstrating that monocular and stereo networks can learn to deal with these objects in the absence of depth annotations.

3. Method

Our goal is to generate depth annotations for images featuring ToM objects in a cheap and scalable manner. This allows for training deep networks to properly estimate their depth as the distance of the closest surface in front of the camera, rather than the distance of the scene content refracted/reflected through it. Our strategy is simple yet dramatically effective and relies on the availability of recent pre-trained monocular depth estimation models [38, 37], which are capable of strong generalization across a variety of scenes though struggling to deal with ToM surfaces. Based on the above state of affairs, we argue that ToM objects are often the sole elements harming the reliability of recent pre-trained monocular depth estimation networks.

Therefore, by virtually replacing these objects with textured artifacts that resemble their very same shapes, the monocular model may be possibly tricked and induced into estimating the depth of an opaque object, ideally placed at the very same spot in the scene. This methodology can be realized by delineating ToM objects, through manual annotations or a segmentation network, masking them from the image and then in-painting virtual textures within the masked areas. On the one hand, since a proper detection of ToM objects is crucial to our methodology, manual labeling indisputably results in the most accurate choice, though it comes with significant annotation costs. On the other hand, relying on a segmentation network would alleviate this cost: one would need some initial human annotations for training, but this would then allow to segment a large number of images for free. Unfortunately, the overall effectiveness of our methodology would be inevitably affected by the accuracy of the trained segmentation model. However, we reckon that annotating images with segmentation masks requires, definitely, a vastly lower effort compared to depth annotation [64, 27]. Hence, we settled on exploring both the aforementioned approaches.

The reader may argue that, as a consequence of our intuition, training a depth network to deal with ToM objects might be unnecessary – indeed, it would be sufficient to segment and in-paint such objects at deployment time before estimating depth. However, we retort that such a methodology would rely heavily on the actual accuracy of the model trained to segment ToM objects, which is not granted to generalize. Moreover, it would add non-negligible computational cost – i.e., the inference by a second network. On the contrary, an offline training or fine-tuning procedure allows for exploiting human-made annotation – if available – and, potentially, enable the trained network to learn how to properly estimate depth on ToM surfaces and to get rid of the second network, as well as design advanced strategies for other depth estimation frameworks, e.g. deep stereo net-
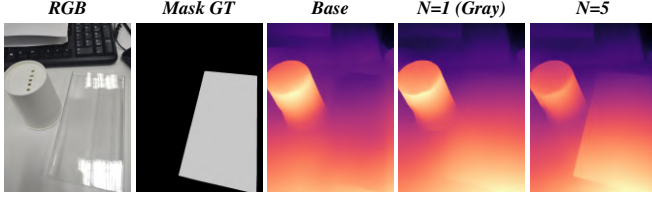
Figure 3. **Virtual depth generation alternatives.** From left to right: RGB, ground-truth segmentation, DPT predictions on the RGB image, on the gray-masked input, and the median of five predictions on images masked with random colors.

Figure 4. **Stereo Distillation pipeline.** Given a stereo pair and a segmentation mask for the left image, we merge predictions from a pre-trained stereo network with virtual depths obtained by a monocular network with our strategy. We merge the monocular or stereo maps, by taking values belonging to either ToM or other surfaces from mono or stereo, respectively. These final merged depth labels are used to fine-tune the original stereo network.

works. Our experiments will highlight that the former strategy results ineffective, while we achieve a large boost in accuracy by fine-tuning depth models with our approach.

In the remainder, we describe our methodology to deal with ToM objects. Given a dataset of images $\mathcal{I}$, our pipeline sketched in Fig. 2 builds as follows: i) surface labeling, ii) in-painting and distillation, and iii) fine-tuning of the depth network on virtual labels. Additionally, we show how it can be revised to fine-tune also deep stereo networks.

Surface Labeling. For any image $I_k \in \mathcal{I}$, we produce a segmentation mask M_k classifying each pixel p as

$$M_k(p) = \begin{cases} 1 & \text{if } I_k(p) \in \text{ToM surfaces} \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

by labeling pixels as either 1 or 0 if they belong to a ToM surface or not, respectively. Such a segmentation mask can be obtained either through manual annotation or by means of a segmentation network Θ as $M_k = \Theta(I_k)$.

In-painting and Distillation. Given an image I_k and its corresponding segmentation mask M_k, we generated an augmented image $\tilde{I}_k$ applying an in-painting operation to replace the pixels belonging to ToM objects with a color c:

$$\tilde{I}_k(p) = \begin{cases} c & \text{if } M_k(p) = 1 \\ I_k & \text{otherwise} \end{cases} \quad (2)$$

Then, a virtual depth $\tilde{D}_k$ for image I_k is obtained by forwarding $\tilde{I}_k$ to a monocular depth network Ψ as $\tilde{D}_k = \Psi(\tilde{I}_k)$. Colors are randomly sampled for every single frame I_k. However, depending on the image content, certain colors might result ineffective and increase the scene ambiguity – e.g., by in-painting white pixels into a transparent object located in front of a white wall. To discourage these occurrences, we sample a set of N custom colors $c_i, i \in [0, N-1]$, and in-paint I_k using each of these custom colors, so as to generate a set of N augmented images $\tilde{I}_k^i$. Then, we obtain the final, *Virtual Depth* $\tilde{D}_k$ by computing the per-pixel median between the N depth maps

$$\tilde{D}_k^* = \text{med}\left\{\Psi(\tilde{I}_k^i), i \in [0, N-1]\right\} \quad (3)$$

As depicted in Fig. 3, in some cases, the in-painted color might be similar to the background – e.g., the transparent object disappears when a single gray mask is used – while it is visible by aggregating multi-color in-painting.

Fine-Tuning on Virtual Labels. The steps outlined so far allow for labeling a dataset $\mathcal{I}$ with virtual depth labels that are not influenced by the ambiguities of ToM objects. Then, our newly annotated dataset can be used to train or fine-tune a depth estimation network, thereby enabling it to handle the aforementioned difficult objects robustly. Specifically, during training, the original images I_k are forwarded to the network, and the predicted depth $\hat{D}_k$ is optimized with respect to the distilled virtual ground-truth map $\tilde{D}_k^*$ obtained from in-painted images. This simple pipeline can dramatically improve the accuracy of monocular depth estimation networks when dealing with ToM objects, as we will show in our experiments.

Extension to Deep Stereo. Our pipeline can be adapted to fine-tune deep stereo models as well, as shown in Fig. 4. Again, we argue that state-of-the-art stereo architectures [26, 22] already expose outstanding generalization capabilities while struggling with ToM objects, due to the task of matching pixels belonging to non-Lambertian surfaces being inherently ambiguous. Consequently, we exploit a monocular depth estimation network to obtain virtual depth annotations solely for these objects. Given a dataset $\mathcal{S}$ consisting of stereo pairs (L_k, R_k), we distill virtual depth labels $\tilde{D}_k^*$ from L_k and triangulate them into disparities $\tilde{d}_k^*$ according to extrinsic parameters of the stereo rig. Then,

Category	Method	MiDaS [37]								DPT [38]							
		$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)	$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)
All	Base	94.56	91.72	85.68	74.00	**50.12**	90.82	**0.07**	120.51	96.79	94.45	89.71	79.00	56.26	75.35	0.06	100.68
All	Virtual Depth N=1	**96.08**	93.23	87.68	75.23	49.26	83.88	**0.07**	112.03	97.59	95.88	92.06	82.75	60.14	64.99	**0.05**	85.96
All	Virtual Depth N=5	96.04	**93.51**	**87.93**	**75.70**	49.36	**82.98**	**0.07**	**110.52**	**98.43**	**96.74**	**92.86**	**83.42**	**60.18**	**62.46**	**0.05**	**82.06**
ToM	Base	87.44	83.40	72.71	59.63	36.28	122.33	0.12	140.31	92.77	88.77	80.98	62.46	37.70	113.14	0.10	136.28
ToM	Virtual Depth N=1	**94.11**	**91.99**	**84.12**	68.40	41.17	76.69	0.09	86.46	96.00	93.67	88.88	75.79	45.26	65.58	0.07	78.24
ToM	Virtual Depth N=5	93.87	91.64	83.66	**68.76**	**43.65**	**76.65**	**0.08**	**86.01**	**98.94**	**97.19**	**92.24**	**77.52**	**45.97**	**57.19**	**0.06**	**66.86**
Other	Base	94.57	91.81	85.99	74.01	**50.28**	91.08	**0.07**	119.86	97.10	94.84	90.08	79.76	57.31	73.19	0.06	95.63
Other	Virtual Depth N=1	95.62	92.50	86.77	74.63	48.76	88.30	**0.07**	116.78	97.63	95.96	92.09	83.11	**61.01**	66.10	**0.05**	87.37
Other	Virtual Depth N=5	**95.66**	**92.93**	**87.31**	**75.42**	49.06	**86.49**	**0.07**	**114.48**	**98.29**	**96.50**	**92.57**	**83.51**	60.90	**64.17**	**0.05**	**84.06**

Table 1. **Virtual depth distillation by varying** N. Results on Booster train set at quarter resolution. All networks use the official weights [37, 38] without further training. Different masking strategies are applied to the RGB input image. Best results in **bold**.

we predict a *Base* disparity map d_k by forwarding (L_k, R_k) to the stereo network we aim to fine-tune. Eventually, we replace the disparities for ToM objects with those from d_k according to M_k, this latter produced over L_k this time. Formally, this operation namely *Merging*, is defined as:

$$d_k(p) = \begin{cases} d_k(p) & \text{if } M_k(p) = 0 \\ \alpha_k \hat{d}_k^*(p) + \beta_k & \text{otherwise} \end{cases} \quad (4)$$

with α_k, β_k being scale and shift factors, as monocular predictions are up to an unknown scale factor. Following [38], α_k, β_k are estimated through Least Square Estimation (LSE) regression over d_k for pixels not belonging to any ToM object, i.e., having $M_k(p) = 0$:

$$(\alpha_k, \beta_k) = \arg\min_{\alpha, \beta} \sum_{p | M_k(p) = 0} \left(\alpha \hat{d}_k^*(p) + \beta - d_k(p) \right)^2 \quad (5)$$

4. Experimental Settings

Implementation Details. We employ MiDaS [38] and DPT [37] as our monocular networks using the official pre-trained weights, given their excellent in-the-wild generalization performance. To fine-tune them, we iterate for 20 epochs with batch size 8 and a learning rate of 10^{-7} with exponential decay with gamma 0.95. We use random color and brightness and random horizontal flip augmentations. We pad/crop and resize images to match the pre-training resolution, i.e., 384 pixels for the long or short side, preserving aspect ratio with mirror pad or square crop, for MiDaS or DPT, respectively. We normalize images as the original networks do. Regarding stereo networks, we employ RAFT [26] and CREStereo [22], using the official pre-trained weights, since they achieve the top rankings in the Middlebury dataset [41] among published methods. To fine-tune them, we run 20 epochs, with batch size 2, fixed learning rate 10^{-5}. Following [64], we randomly resize images to half or quarter of the original dataset resolution, randomly crop to 456×884 and 448×880 for RAFT and CREStereo respectively, and further randomly scale images and disparities by a factor $\in [0.9, 1.1]$. We assume

22 and 10 iterations during training for RAFT-Stereo and CREStereo, respectively. During testing, we run 32 and 20 iterations. When creating virtual labels with our masking strategy, we fix the random seed of color sampling to 0.

Datasets. Among the datasets, we selected Trans10K [54], MSD[60], and Booster[64] as they focus on ToM surfaces and contain images acquired in many realistic environments. Trans10K contains 5 003, 1 003, 4 431 images for the training, validation, and test set, respectively, featuring common transparent objects and stuff. It provides segmentation masks with pixels categorized into 12 different classes that we collapse into 2 – ToM (classes 1 to 11) or not. MSD contains 3 066, and 958 images and binary segmentation masks for the training and test set, respectively, featuring mirrors. Booster contains 228, and 191 images for training and testing, respectively. The dataset provides disparity and segmentation maps for the training set, where the segmentation maps are categorized into 4 classes, which we group into 2 – classes 2-3 into "ToM" category, classes 0-1 into "Other" materials. We fine-tune on Trans10K and MSD for monocular models and on the Booster training split for stereo networks, without using any depth ground truths.

Evaluation Protocol. We evaluate the accuracy of the monocular networks using several metrics, including absolute error relative to the ground-truth value (ABS Rel.), the percentage of pixels having the maximum between the prediction/ground-truth and ground-truth/prediction ratios lower than a threshold (δ_i, with i being 1.05, 1.10, 1.15, 1,20, and 1.25), the mean absolute error (MAE) and Root Mean Squared Error (RMSE). Additionally, we evaluate stereo networks using the metrics defined in Booster [64], i.e. bad-2, bad-4, bad-6, bad-8, MAE, RMSE. Results are reported on all valid pixels (*All*) or for those belonging to either ToM or other objects, in order to assess the impact of our strategy on the different kinds of surfaces. For any metrics considered for stereo networks, the lower, the better – annotated with ↓ in tables. The same applies to metrics used for monocular networks except for δ_i, resulting in the higher, the better – with ↑ being reported in tables. As the predictions by monocular networks are up to an unknown scale factor, we rescale them according to the LSE criterion [38] defined in Eq. 5, using all valid pixels here. Monocu-

		MiDaS [37]								DPT [38]							
Category	Method	$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)	$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)
All	Base	94.56	91.72	85.68	74.00	50.12	90.82	**0.07**	120.51	96.79	94.45	89.71	79.00	56.26	75.35	0.06	100.68
All	Ft. Base	94.09	91.28	85.29	73.55	49.14	93.34	0.07	124.54	96.83	94.60	90.14	79.46	56.89	76.00	0.06	100.98
All	Ft. Virtual Depth	**95.07**	**92.31**	**86.39**	**75.20**	**50.57**	**88.83**	0.07	**118.23**	**97.99**	**96.65**	**93.55**	**83.94**	**60.46**	**64.93**	**0.05**	**85.93**
ToM	Base	87.44	83.40	72.71	59.63	36.28	122.33	0.12	140.31	92.77	88.77	80.98	62.46	37.70	113.14	0.10	136.28
ToM	Ft. Base	86.30	82.96	72.84	61.04	37.91	126.71	0.12	145.69	92.69	89.49	81.12	63.62	37.95	118.84	0.11	141.27
ToM	Ft. Virtual Depth	**91.81**	**89.12**	**81.68**	**70.75**	**47.85**	**93.66**	**0.09**	**104.80**	**96.68**	**95.96**	**92.23**	**79.96**	**54.67**	**70.68**	**0.06**	**83.06**
Other	Base	94.57	91.81	85.99	74.01	50.28	91.08	**0.07**	119.86	97.10	94.84	90.08	79.76	57.31	73.19	0.06	95.63
Other	Ft. Base	94.27	91.49	85.87	73.77	49.38	92.48	**0.07**	122.46	97.15	95.02	90.97	80.77	58.05	72.45	0.06	94.07
Other	Ft. Virtual Depth	**95.33**	**92.53**	**86.80**	**75.43**	**50.46**	**89.17**	**0.07**	**119.58**	**98.07**	**96.64**	**93.52**	**84.30**	**61.19**	**64.70**	**0.05**	**85.57**

Table 2. **Monocular networks fine-tuning - ground-truth segmentation.** Training on all MSD and Trans10K, results on the Booster train set at quarter resolution. All models start from the official weights [37, 38]. Best results in **bold**.

		MiDaS [37]								DPT [38]							
Category	Method	$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)	$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)
All	Base	94.56	91.72	85.68	74.00	50.12	90.82	**0.07**	120.51	96.79	94.45	89.71	79.00	56.26	75.35	0.06	100.68
All	Virtual Depth (Proxy)	91.78	87.52	79.57	66.00	42.21	105.67	0.09	140.00	93.23	89.43	81.98	68.06	43.62	98.09	0.08	128.36
All	Ft. Virtual Depth (GT)	94.99	92.12	86.32	74.99	49.91	88.63	**0.07**	117.40	98.09	**96.85**	**93.91**	**83.50**	58.74	65.52	**0.05**	86.41
All	Ft. Virtual Depth (Proxy)	**95.00**	**92.17**	**86.48**	**75.46**	**50.24**	**88.10**	0.07	**117.10**	**98.11**	96.68	93.48	83.00	58.13	66.43	**0.05**	87.18
ToM	Base	87.44	83.40	72.71	59.63	36.28	122.33	0.12	140.31	92.77	88.77	80.98	62.46	37.70	113.14	0.10	136.28
ToM	Virtual Depth (Proxy)	86.37	79.35	69.91	54.84	32.31	112.92	0.12	124.10	89.75	85.26	75.17	56.57	31.75	110.74	0.11	123.28
ToM	Ft. Virtual Depth (GT)	**91.85**	**89.93**	**83.25**	70.63	**47.55**	92.05	**0.09**	103.54	**96.95**	**96.26**	**93.27**	**81.84**	**55.49**	**67.95**	**0.06**	**80.88**
ToM	Ft. Virtual Depth (Proxy)	91.12	88.36	81.18	69.15	45.02	96.33	**0.09**	107.59	96.82	96.05	92.57	80.62	53.74	70.67	**0.06**	83.44
Other	Base	94.57	91.81	85.99	74.01	50.28	91.08	**0.07**	119.86	97.10	94.84	90.08	79.76	57.31	73.19	0.06	95.63
Other	Virtual Depth (Proxy)	91.80	87.47	79.41	65.86	41.95	108.09	0.09	143.64	93.22	89.39	82.07	68.30	43.88	98.64	0.08	129.48
Other	Ft. Virtual Depth (GT)	**95.27**	92.30	86.60	75.43	49.95	88.81	**0.07**	118.48	98.19	**96.85**	**93.85**	83.70	58.93	65.65	**0.05**	86.30
Other	Ft. Virtual Depth (Proxy)	**95.27**	**92.42**	**86.84**	**76.02**	**50.48**	**88.07**	**0.07**	**117.99**	**98.20**	96.67	93.35	83.20	58.40	66.51	**0.05**	87.06

Table 3. **Monocular networks fine-tuning – ground-truth vs proxy segmentation.** Training on only the test set of MSD and Trans10K, results on the Booster train set at quarter resolution. All models start from the official weights [37, 38]. Best results in **bold**.
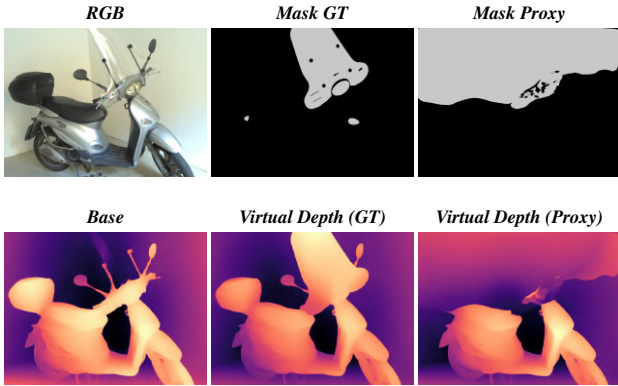

Figure 5. **Virtual Depth Qualitatives – GT vs Proxy.** From left to right - Top: RGB, ground-truth and proxy segmentations; Bottom: prediction with DPT on the RGB image prediction with DPT on the median of five predictions by in-painting with either the ground truth or semantic proxy masks on Booster.

lar networks are evaluated on the Booster training set, while stereo models are evaluated on the Booster test set.

5. Experiments

In this section, we report the outcome of our findings concerning both monocular and stereo depth estimation.

5.1. Monocular Depth Estimation.

Number of In-Paintings. We investigate the quality of the virtual depth labels by varying N. When using $N = 1$ we generate a single in-painted image that is forwarded to the monocular network, while with $N = 5$ we generate virtual depths from 5 masked images with different colors which are then aggregated by selecting the pixel-wise depth median. In Tab. 1, we report the accuracy of depth maps produced by the two strategies, together with those of the *Base* architectures, i.e. without applying any in-painting strategy. Firstly, with both MiDaS and DPT, both in-painting strategies obtain virtual depths that are much more accurate for ToM regions w.r.t. the *Base* architecture. Secondly, $N=5$ maps yield slightly better results in most metrics, especially when looking at DPT performance. We ascribe it to the higher robustness of the second strategy. For the remaining experiments, we fix $N = 5$ as we did not observed any further improvement with larger values.

Fine-tuning Results (GT Segmentation). In Tab. 2, we report results on the Booster train set, obtained after fine-tuning MiDaS and DPT on all available data from Trans10K and MSD. In the *Base* row, we report the results of the network using the officially released weights without any further training, and we compare with those in row *Ft. Virtual Depth*, i.e., the results of our method. We notice that the accuracy on *All* pixels is improved with our approach. In particular, we achieve a significant boost in performances for ToM surfaces, of 4.37, 5.72, 8.97, 11.12 and 11.57%, 28.67mm, 0.03%, 35.51mm for MiDaS [38], and 3.91, 7.19, 11.25, 16.8 and 16.97%, 42.46mm, 0.04%, 53.255mm for DPT[38] in the $\delta_{1.25}$, $\delta_{1.20}$, $\delta_{1.15}$, $\delta_{1.10}$, $\delta_{1.05}$, MAE, Abs.Rel, and RMSE, respectively. We highlight that, after fine-tuning, the accuracy on *ToM* is only slightly worse than on *Other*. Moreover, class *Other* metrics are also slightly

Category	Method	MiDaS [37]								DPT [38]							
		$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)	$\delta < 1.25$ ↑(%)	$\delta < 1.20$ ↑(%)	$\delta < 1.15$ ↑(%)	$\delta < 1.10$ ↑(%)	$\delta < 1.05$ ↑(%)	MAE ↓(mm)	Abs. Rel ↓	RMSE ↓(mm)
All	Base	**91.68**	**87.13**	**79.17**	**64.77**	**39.04**	**27.17**	0.09	**40.16**	**91.78**	**87.61**	**80.44**	**67.42**	**42.60**	**26.42**	0.09	**40.10**
All	Ft. Virtual Depth	90.38	85.89	77.97	63.74	38.57	29.40	0.09	44.37	90.97	86.74	79.56	66.45	42.12	27.72	0.09	42.56

Table 4. **Monocular networks fine-tuning – results on NYU-V2 [31].** After training on MSD and Trans10K, both MiDaS and DPT show a negligible drop in accuracy on non-ToM surfaces of existing datasets.

Category	Method	RAFT-Stereo [26]						CREStereo [22]					
		bad-2 ↓(%)	bad-4 ↓(%)	bad-6 ↓(%)	bad-8 ↓(%)	MAE ↓(px)	RMSE ↓(px)	bad-2 ↓(%)	bad-4 ↓(%)	bad-6 ↓(%)	bad-8 ↓(%)	MAE ↓(px)	RMSE ↓(px)
All	Base	17.42	13.49	11.59	10.11	4.07	8.63	15.13	10.70	8.91	7.57	3.15	7.40
All	Ft. Base	18.44	14.04	12.02	10.49	4.33	8.92	14.65	10.14	8.39	6.97	2.86	6.64
All	Ft. Virtual Depth	16.81	12.34	9.98	8.09	2.60	6.04	18.85	14.35	12.42	10.87	5.11	9.83
All	Ft. Merged	**14.68**	**9.63**	**7.35**	**5.57**	**1.95**	**4.58**	**10.86**	**6.13**	**4.41**	**3.13**	**1.51**	**3.64**
ToM	Base	56.77	44.38	38.43	33.31	13.45	16.56	51.83	37.88	32.86	28.19	12.42	15.60
ToM	Ft. Base	58.56	44.40	38.02	32.99	13.83	16.47	49.45	34.49	29.00	24.24	10.25	13.01
ToM	Ft. Virtual Depth	57.09	42.89	34.70	27.61	8.47	10.75	49.70	35.45	29.84	24.79	11.99	13.94
ToM	Ft. Merged	**47.54**	**30.56**	**22.83**	**16.61**	**5.83**	**7.43**	**36.96**	**20.79**	**15.38**	**10.67**	**5.04**	**6.73**
Other	Base	8.48	5.74	4.52	3.83	1.58	3.79	8.11	4.83	3.50	2.78	1.14	2.77
Other	Ft. Base	9.22	6.23	4.91	4.16	1.71	3.99	7.02	4.20	3.15	2.53	1.00	2.56
Other	Ft. Virtual Depth	7.97	5.03	3.81	3.15	1.19	3.20	11.45	8.24	7.03	6.30	2.73	5.45
Other	Ft. Merged	**7.14**	**4.08**	**2.87**	**2.22**	**0.96**	**2.63**	**5.27**	**2.73**	**1.82**	**1.34**	**0.68**	**1.70**

Table 5. **Stereo networks fine-tuning – ground truth segmentation.** Results on Booster test set at quarter resolution. All models start from the official weights [26, 22] and are fine-tuned according to different strategies. Best results in **bold**.

better, probably because of the enhanced features extracted by the network, which has a better understanding of the scene context. Finally, we have reported in *Ft. Base* the fine-tuning results obtained by self-training the networks on their own predictions without any in-painting strategy. As expected, without the appropriate virtual depth labels, the networks cannot effectively learn from the new dataset, yielding results comparable to the *Base* architecture. Experiments on additional datasets are in the supplement.

Fine-tuning Results (Proxy Segmentation). Even though obtaining semantic labels is cheaper than collecting depth ground truths, using the predictions of a segmentation network as proxy semantic annotations would accelerate the dataset collection process. Thus, we investigate the impact of replacing manually annotated masks in our pipeline with the predictions of Trans2Seg[54] and MirrorNet[60], pre-trained on the training set of Trans10K and MSD, respectively, on the unseen test set of each dataset. We use weights made available by the authors. For a fair comparison, we also re-train again the models exploiting GT segmentations only on the test sets of the two datasets. Tab. 3 highlights that both models, using either GT - *Ft. Virtual Depth (GT)* - or proxy segmentations - *Ft. Virtual Depth (Proxy)*, achieve much more accurate results compared to the *Base* network. Interestingly, the two networks yield comparable results in the class *Other*, while the one using GTs is slightly better than the other in the class *ToM*, yet still comparable. Finally, in row *Virtual Depth (Proxy)*, we report the results of our in-painting methodology (i.e., without fine-tuning) but coloring pixels according to the proxy segmentations. We note that performances are even worse than the *Base* method. Indeed, the segmentation network struggles to gen-
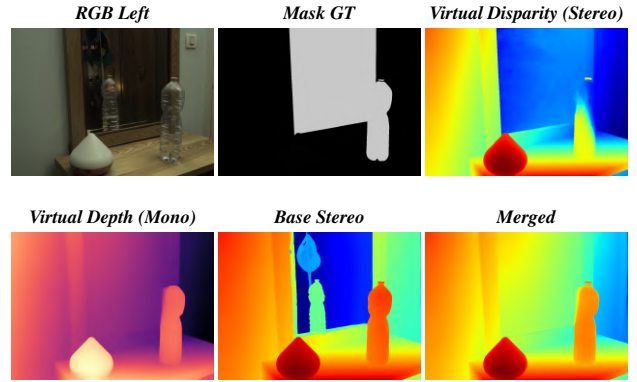

Figure 6. **Qualitative comparison – disparity virtual labels.** On top: left RGB image, ground truth segmentation mask, and virtual disparities by CREStereo processing masked stereo pairs. At bottom: depth by DPT on the in-painted left image, disparities by CREStereo, and their final merged labels.

eralize to the Booster dataset, making the depth model incapable of estimating the correct values, e.g., due to some overextended in-painted ToM areas, as shown in Fig. 5. Yet, depth networks, fine-tuned on the test set of MSD and Trans10K (row *Ft. Virtual Depth (Proxy)*), generalize properly on Booster.

Performance on non-ToM surfaces. To prove the effectiveness of the fine-tuned models on scenes with only a few ToM surfaces, we test the MiDaS and DPT trained over our proxy labels on the NYU-V2 dataset [31], reporting the results in Tab. 4. We notice only negligible drops in performances – the strictest metric $\delta_{1.05}$ drops by less than 0.5%, which we consider a small price to pay for the large improvement on ToM surfaces observed before.

Category	Method	RAFT-Stereo [26]						CREStereo [22]					
		bad-2 ↓(%)	bad-4 ↓(%)	bad-6 ↓(%)	bad-8 ↓(%)	MAE ↓(px)	RMSE ↓(px)	bad-2 ↓(%)	bad-4 ↓(%)	bad-6 ↓(%)	bad-8 ↓(%)	MAE ↓(px)	RMSE ↓(px)
All	Base	17.42	13.49	11.59	10.11	4.07	8.63	15.13	10.70	8.91	7.57	3.15	7.40
All	Ft. Merged (Proxy)	19.56	13.53	10.53	8.28	2.51	5.37	12.60	7.42	5.24	3.64	**1.41**	**3.33**
All	Ft. Merged (GT)	**14.68**	**9.63**	**7.35**	**5.57**	**1.95**	**4.58**	**10.86**	**6.13**	**4.41**	**3.13**	1.51	3.64
ToM	Base	56.77	44.38	38.43	33.31	13.45	16.56	51.83	37.88	32.86	28.19	12.42	15.60
ToM	Ft. Merged (Proxy)	51.85	35.32	27.17	20.88	6.56	8.04	40.85	23.66	16.83	11.10	**4.18**	**5.61**
ToM	Ft. Merged (GT)	**47.54**	**30.55**	**22.81**	**16.62**	**5.83**	**7.43**	**36.96**	**20.79**	**15.38**	**10.67**	5.04	6.73
Other	Base	8.48	5.74	4.52	3.83	1.58	3.79	8.11	4.83	3.50	2.78	1.14	2.77
Other	Ft. Merged (Proxy)	11.95	7.36	5.31	4.15	1.41	3.41	6.56	3.69	2.53	1.88	0.82	2.02
Other	Ft. Merged (GT)	**7.14**	**4.08**	**2.87**	**2.22**	**0.96**	**2.63**	**5.27**	**2.73**	**1.82**	**1.34**	**0.68**	**1.70**

Table 6. **Stereo networks fine-tuning – ground truth vs proxy segmentation.** Results on Booster test set at quarter resolution. All models start from the official weights [26, 22] and are fine-tuned according to different strategies. Best results in **bold**.

5.2. Stereo Depth Estimation

Virtual Disparity Generation Alternatives. We inquire about two main alternatives to generate virtual disparities: i) *Virtual Disparity*: masking both left and right images according to material segmentation masks – as materials annotations are provided for the left image only, we warp it over the right image according to ground-truth disparity – and then processing the two with the stereo network we are going to fine-tune similar to Monocular networks, ii) *Merged*: merging disparity labels produced by the stereo model itself with those generated by original DPT weights [37], as detailed in Eq. 4. Although the former might appear as the natural extension of our proposal from the monocular to the stereo case, we will demonstrate its ineffectiveness.

Fine-tuning Results (GT Segmentation). Tab. 5 collects the results obtained by fine-tuning RAFT-Stereo and CREStereo through our technique. From top to bottom, we report the results achieved by the original models (*Base*) as well as the instances fine-tuned on their own predictions (*Ft. Base*) or on pseudo labels obtained according to the two strategies (*Ft. Virtual Depth*, *Ft. Merged*).

Not surprisingly, fine-tuning the networks on their own predictions is harmful (RAFT-Stereo) or scarcely effective (CREStereo). Applying the first of the two strategies sketched before yields just a negligible improvement over the original models on ToM classes. This evidence confirms that our pipeline designed for the monocular case cannot naïvely be extended to the stereo case by in-painting the two images since masking ToM objects with constant colors does not ease matching – on the contrary, it introduces textureless regions, which are likely to be labeled as planar surfaces by stereo models. Conversely, the second strategy consistently improves the predictions with both RAFT-Stereo and CREStereo. In particular, the former achieves 9.23, 13.83, 15.62, and 16.69% absolute reductions on bad-2, bad-4, bad-6, and bad-8, respectively, as well 7.62 and 9.13 reductions on MAE and RMSE on ToM regions. CREStereo obtains 14.93, 17.13, 17.48, and 17.54% on bad metrics, and 7.40 and 8.91 reductions on MAE and RMSE. Moreover, the accuracy over *Other* pixels is also improved,
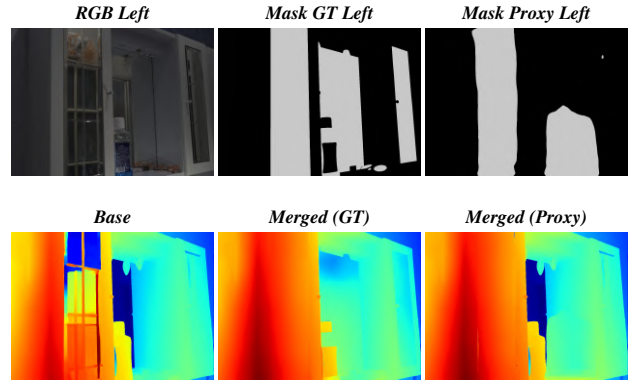

Figure 7. **Stereo depth merging with GT or Proxy semantic labels.** From left to right: RGB left image, ground-truth semantic mask, proxy semantic mask, prediction by CREStereo on the RGB images, and the final merged labels using either the GT or Proxy segmentation masks.

although with minor margins. Fig. 6 provides a qualitative comparison between the labels obtained by the two strategies. The former produces a planar surface for the mirror completely misaligned with respect to the wall, whereas the latter combines the virtual depth labels from DPT on masked images with disparity labels at best.

Fine-tuning Results (Proxy Segmentation). Finally, we replace the manually annotated segmentation masks with those predicted by Trans2Seg and MirrorNet and then distill virtual disparities for fine-tuning both stereo networks. As pointed out before, both Trans2Seg and MirrorNet have not been trained on Booster. Thus, *Merging* produces significant differences with respect to the use of manually annotated masks, as shown in Fig. 7. Nevertheless, Tab. 6 shows that our pipeline improves the performance of both RAFT-Stereo and CREStereo on ToM objects, even in the case of extremely noisy proxy semantic annotations. More precisely, CREStereo seems to benefit more from the Proxy segmentation configuration than RAFT-Stereo. Indeed, on the one hand, we can notice how RAFT-Stereo improves on ToM regions at the expense of accuracy on other pixels when using Proxy segmentations. This yields,
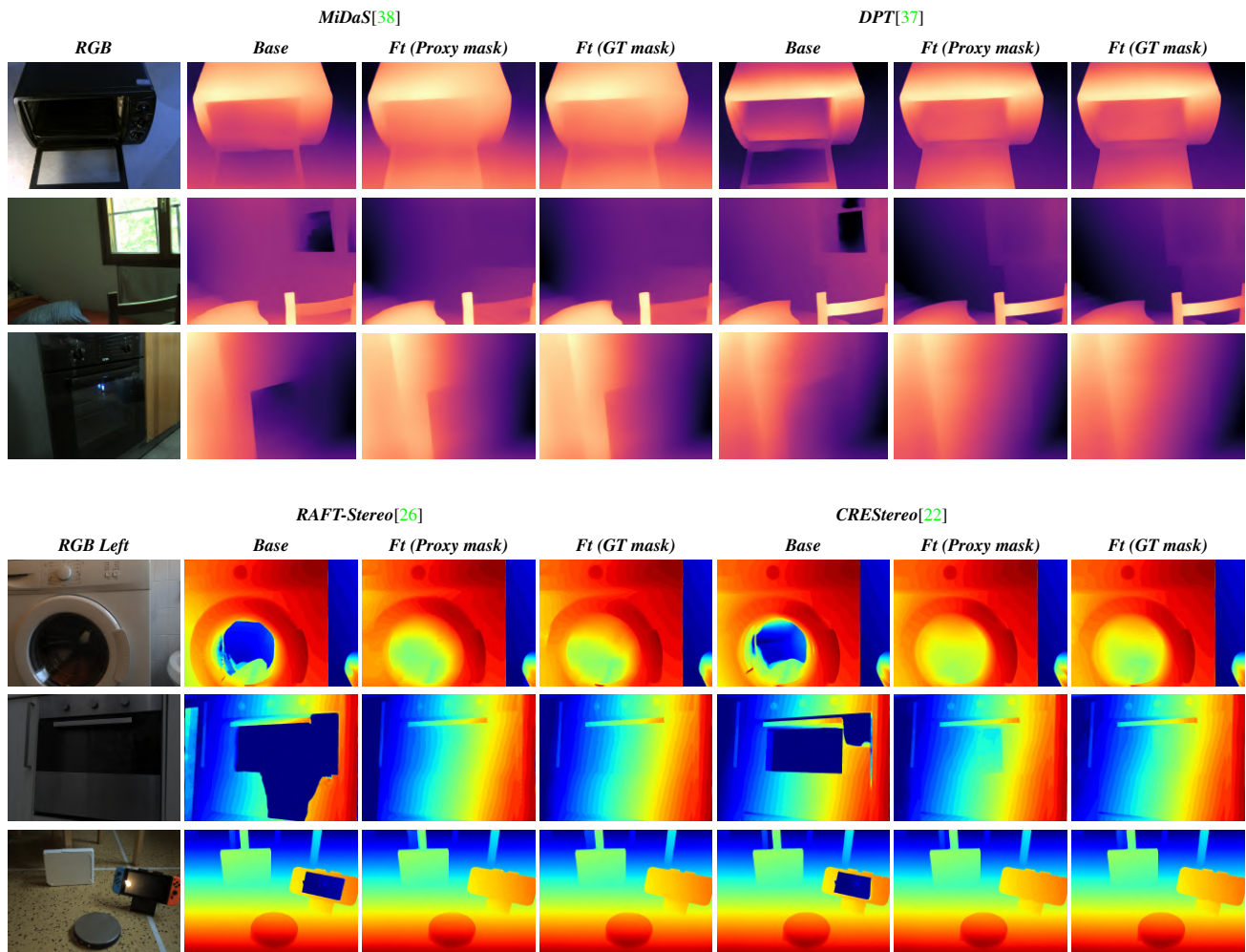
Figure 8. **Qualitative post fine-tuning results.** Examples of predictions by MiDaS and DPT (top), RAFT-Stereo and CREStereo (bottom). For each model, we show results achieved by the original model and by fine-tuned instances using proxy or GT segmentation masks.

on All pixels, an increase in the bad-2 and bad-4 error rates, whereas bad-6, bad-8, MAE, and RMSE remain lower. On the other hand, CREStereo seems capable of exploiting fine-tuning much better, yielding more accurate results on any metric with both Proxy or GT masks. This outcome proves that our pipeline is effective for fine-tuning stereo models even without manually annotated masks. Nonetheless, segmenting images through human labeling unleashes its full potential, whose cost is much lower compared to that that would be required to annotate depths.

5.3. Qualitative Results

To conclude, Fig. 8 shows the effect of the fine-tuning carried out according to our proposal, with two examples for monocular (top) and stereo (bottom) networks from the Booster train and test sets respectively. We highlight how MiDaS, DPT, RAFT-Stereo, and CREStereo learn to deal with ToM surfaces either when relying on proxy segmenta-

tion masks provided by neural networks or accurately annotated by humans. More qualitative examples are reported in the supplementary material.

6. Conclusion

We have proposed an effective methodology for training depth estimation networks to deal with transparent and mirror surfaces. By in-painting these surfaces on RGB images, we can quickly annotate a dataset with virtual depth labels, that can be used to fine-tune both monocular and stereo networks, with outstanding results. A promising future direction would be to extend our technique to instance segmentation masks to get better virtual depth maps in the presence of multiple ToM objects in the same scene.

References

[1] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 2

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 2

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018. 2

[5] Xiaotong Chen, Huijie Zhang, Zeren Yu, Anthony Opipari, and Odest Chadwicke Jenkins. Clearpose: Large-scale transparent object dataset and benchmark. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 381–396. Springer, 2022. 2, 3

[6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 2

[7] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[8] Jaehoon Choi, Dongki Jung, Yonghan Lee, Deokhwa Kim, Dinesh Manocha, and Donghwan Lee. Selfdeco: Self-supervised monocular depth completion in challenging indoor environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 467–474. IEEE, 2021. 2

[9] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4384–4393, 2019. 2

[10] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 740–756. Springer, 2016. 2

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 2

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *The International Conference on Computer Vision (ICCV)*, October 2019. 2

[13] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33:12626–12637, 2020. 2

[14] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2485–2494, 2020. 2

[15] Weiyu Guo, Zhaoshuo Li, Yongkui Yang, Zheng Wang, Russell H Taylor, Mathias Unberath, Alan Yuille, and Yingwei Li. Context-enhanced stereo transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[16] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 2

[17] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 2

[18] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 4756–4765, 2020. 2

[19] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[20] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018. 2

[21] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004. 2

[22] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022. 1, 2, 4, 5, 7, 8, 9

[23] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6197–6206, 2021. 2

[24] Chia-Kai Liang, Chao-Chung Cheng, Yen-Chieh Lai, Liang-Gee Chen, and Homer H Chen. Hardware-efficient belief propagation. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(5):525–537, 2011. 2

[25] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning

for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[26] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021. 1, 2, 4, 5, 7, 8, 9

[27] Xingyu Liu, Shun Iwase, and Kris M Kitani. Stereobj-1m: Large-scale stereo image dataset for 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10870–10879, 2021. 3

[28] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[29] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5667–5675, 2018. 2

[30] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[31] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7

[32] Jiahao Pang, Wenxiu Sun, Jimmy SJ. Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[33] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9250–9256. IEEE, 2019. 2

[34] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. In *2018 international conference on 3D vision (3DV)*, pages 587–595. IEEE, 2018. 2

[35] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 2

[36] Matteo Poggi, Fabio Tosi, Konstantinos Batsos, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5314–5334, 2022. 2

[37] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021. 1, 2, 3, 5, 6, 7, 8, 9

[38] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 1, 2, 3, 5, 6, 7, 9

[39] Tonmoy Saikia, Yassine Marrakchi, Arber Zela, Frank Hutter, and Thomas Brox. Autodispnet: Improving disparity estimation with automl. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1812–1823, 2019. 2

[40] Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3642. IEEE, 2020. 2, 3

[41] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 5

[42] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.*, 47(1-3):7–42, 2002. 2

[43] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13906–13915, June 2021. 2

[44] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, pages 572–588. Springer, 2020. 2

[45] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *ACCV*, 2018. 2

[46] Jaime Spencer, Richard Bowden, and Simon Hadfield. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14402–14413, 2020. 2

[47] Tatsunori Taniai, Yasuyuki Matsushita, and Takeshi Naemura. Graph cut based continuous stereo matching using locally shared labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1613–1620, 2014. 2

[48] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14362–14372, June 2021. 2

[49] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019. 2

[50] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mat-

toccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4654–4665, 2020. 2

[51] Yan Wang, Zihang Lai, Gao Huang, Brian H Wang, Laurens Van Der Maaten, Mark Campbell, and Kilian Q Weinberger. Anytime stereo image depth estimation on mobile devices. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5893–5900, 2019. 2

[52] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2162–2171, 2019. 2

[53] Thomas Whelan, Michael Goesele, Steven J Lovegrove, Julian Straub, Simon Green, Richard Szeliski, Steven Butterfield, Shobhit Verma, and Richard Newcombe. Reconstructing scenes with mirror and glass surfaces. *ACM Transactions on Graphics (TOG)*, 37(4):102, 2018. 3

[54] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. *arXiv preprint arXiv:2003.13948*, 2020. 2, 5, 7

[55] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019. 2

[56] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *ECCV*, pages 636–651, 2018. 2

[57] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1281–1292, 2020. 2

[58] Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1458–1465. IEEE, 2010. 2

[59] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewénius, and David Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE transactions on pattern analysis and machine intelligence*, 31(3):492–504, 2008. 2

[60] Xin Yang, Haiyang Mei, Ke Xu, Xiaopeng Wei, Baocai Yin, and Rynson W.H. Lau. Where is my mirror? In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 5, 7

[61] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6044–6053, 2019. 2

[62] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Pro-

ceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018. 2

[63] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Third European Conference on Computer Vision (Vol. II)*, 3rd European Conference on Computer Vision (ECCV), pages 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. 2

[64] Pierluigi Zama Ramirez, Fabio Tosi, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Open challenges in deep stereo: the booster dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. CVPR. 1, 2, 3, 5

[65] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, 2016. 2

[66] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 340–349, 2018. 2

[67] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided aggregation net for end-to-end stereo matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[68] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, 63(9):1612–1627, 2020. 2

[69] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. *arXiv preprint arXiv:2208.03543*, 2022. 2

[70] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 2