On-Site Adaptation for Monocular Depth Estimation with a Static Camera

Huan Li
huan.li3@unibo.it
Matteo Poggi
m.poggi@unibo.it
Fabio Tosi
fabio.tosi5@unibo.it
Stefano Mattoccia
stefano.mattoccia@unibo.it

Department of Computer Science and Engineering (DISI), University of Bologna, Italy

Abstract

We introduce a novel technique for easing the deployment of an off-the-shelf monocular depth estimation network in unseen environments. Specifically, we target a very diffused setting with a fixed camera mounted higher over the ground to monitor an environment and highlight the limitations of state-of-the-art monocular networks deployed in such a setup. Purposely, we develop an on-site adaptation technique capable of 1) improving the accuracy of estimated depth maps in the presence of moving subjects, such as pedestrians, cars, and others; 2) refining the overall structure of the predicted depth map, to make it more consistent with the real 3D structure of the scene; 3) recovering absolute metric depth, usually lost by state-of-the-art solutions. Experiments on synthetic and real datasets confirm the effectiveness of our proposal.

1 Introduction

Estimating the depth of a single image [38] represents a fascinating challenge in computer vision. In addition to the scientific charm, such an approach is desirable from a practical point of view, allowing for unconstrained depth sensing in almost any scenario without requiring cumbersome and expensive active sensors such as LiDARs [14] nor multiple, synchronized [34] cameras / a single, moving one [39]. Indeed, a single color camera represents the most common setup for several practical applications such as video surveillance [40], road traffic monitoring [33], or, more recently, social distancing [2, 31]. More importantly, these applications raise some crucial privacy concerns, and the possibility of estimating (accurate) depth maps in a surveillance setting also has the potential to improve this aspect. For instance, depth maps could be computed on edge and sent to the cloud to be processed by higher-level applications; under this assumption, the edge node would avoid sharing the color images containing more sensitive information. However, this comes at the cost of dealing with a highly ill-posed problem since the absence of triangulation cues from multiple views prevents the computation of a unique solution explaining the 3D geometry out of a single image.

© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.
Monocular depth estimation – before and after adaptation. On fixed-camera settings (a), state-of-the-art depth estimation models [36] might fail in unseen environments or camera settings (b). Our adaptation scheme allows us to improve their reliability (c).

The advent of deep learning enabled the development of the first-ever solutions [10, 21, 29] making it possible to face such a problem, thanks to the increasing availability of images annotated with depth labels [14, 32] to be used for training, or to the introduction of alternative, self-supervised regimes replacing such annotations with synchronized stereo pairs [13, 15] or monocular video sequences [51]. These approaches, deploying Convolutional Neural Networks (CNNs) or, more recently, Transformers, learn to infer the distance from the camera for objects in the scene based on visual cues [7] such as shadows, perspective, vanishing lines, and more. As proof of this, by acting on some of these cues – e.g., manually shifting the height of the horizon in the image or simulating camera tilting with respect to the ground plane [7] – estimated depth for the very same scene might be sensibly altered.

Indeed, obtaining a network capable of predicting accurate depth maps in any environment remains challenging, even in the availability of a vast amount of annotated data – e.g., millions of images [36, 37] collected from very different datasets. Moreover, the scale ambiguity intrinsic in single images also plays a role, making even state-of-the-art monocular depth estimation networks capable of predicting accurate relative depth, yet up to an unknown scale factor. As a consequence, despite the recent progress in cross-dataset generalization [36, 37], these models are still subject to failures in specific settings that are under-represented in the training data, e.g., on ambiguous objects such as mirrors [49] or, more commonly, when dealing with images taken from a perspective rarely – or never – observed during the training process [7]. Among them, we report an example in Fig. 1, showing a widespread surveillance setting (a), with the camera positioned high over the ground and slanted with respect to it. Although this configuration represents the perfect ground for deploying single image depth estimation – i.e., because of the lack of camera motion or multiple synchronized devices – existing approaches are not ready for unconstrained use there (b), often failing at properly estimating depth for common agents such as pedestrians and cars. We argue that adaptation techniques [4, 18] could attenuate this problem. In particular, online techniques [41] allow the monocular network to improve its accuracy in a new environment right at deployment time, without any prior assumption on it or requiring any sample beforehand. However, existing online strategies suited for monocular networks rely on the assumption that the camera is moving in the scene [17, 19, 43, 50] to exploit the same principles over which self-supervised approaches build upon [51], and thus cannot be
exploited in the static-camera setting mentioned above. Moreover, using stereo images [17] to adapt a monocular network would have little practical sense – i.e., stereo networks [34] would be used instead.

To cope with these limitations, we propose a novel technique for the on-site adaption of an off-the-shelf, monocular depth estimation network when deployed in unseen environments. In contrast to the approaches mentioned above, requiring the camera to move and any other objects in the scene to be static, we exploit the opposite behavior. As we aim at running adaptation on fixed-camera installations, we identify any agents in the scene and use their motion to detect the ground plane over which they move. From these basic cues, we can extract pseudo depth labels for the agents themselves that can be used for a lightweight fine-tuning of the original depth network and exploit the detected ground plane for a test-time refinement step to better align the predicted depth with the actual 3D structure of the environment. Furthermore, by having access to simple priors about the specific camera installation – i.e., the camera height over the ground – we can recover the metric scale for depth maps predicted by the monocular network. To validate our proposal, we run experiments on a subset of the KITTI dataset featuring static camera sequences and two novel datasets composed of synthetic frames rendered through CARLA [8] and real images. In the latter case, the dataset frames indoor and outdoor scenes.

The main contributions of this work are:

• A novel, on-site adaptation scheme for monocular depth estimation networks working in fixed-camera setups, consisting of 1) a lightweight fine-tuning procedure aimed at correcting gross errors on moving agents, 2) a scene alignment step enabled by the moving agents in the scene identifying the ground plane, and 3) metric scale recovery by simply knowing the camera height over the ground.

• Two novel datasets with dense, ground-truth depth labels used to validate the effectiveness of our proposal, available at https://sites.google.com/view/staticdepth-dataset.

2 Related Work

Monocular Depth Estimation. After early attempts at learning for monocular depth estimation with classical machine learning [20, 38], this task attracted increasing interest with the rise of deep learning. Eigen et al. [9, 11] proposed a pivotal multi-stage, coarse-to-fine network for single image depth prediction, Laina et al. [21] developed a fully convolutional architecture with skip connections. DORN [12] deploys a densely connected backbone and casts depth prediction through ordinal regression, while BTS [22] uses local planar guidance to improve accuracy. More recent approaches estimate depth by predicting probability distributions and discrete bins [5, 27, 40], or by introducing self-attention mechanisms [1, 23, 26, 36]. Among the latest works, NeWCRFs [48] and VA-DepthNet [28], respectively, resumed CRFs within the multi-head mechanism of transformers and first-order variational constraints to set the current state-of-the-art. However, any of the previous frameworks always focus on single domains – i.e., training and evaluating over indoor (NYU v2 [32]) and outdoor (KITTI [42]) data separately. A parallel research trend consists of training a single model for generalizing across different domains. MegaDepth [25] represents the first attempt in this direction, followed by MiDaS [37] and DPT [36]. Nonetheless, these models
still fail when processing images from uncommon viewpoints or yield blurred predictions missing some of the agents in the scene, as already highlighted in Fig. 1.

Domain Adaptation for Monocular Depth Estimation. This research topic arose to deal with the inherent difficulties of obtaining a monocular depth estimation network capable of generalizing. At first, the focus has been on synthetic to real adaptation, exploiting image style transfer [4] or GANs [18] for the purpose, yet needing some samples from the target domain to be available beforehand, and focusing exclusively on outdoor [4] or indoor [18] environments – whereas state-of-the-art solutions [36] are nowadays capable of good generalization across the two. A more practical solution consists of directly adapting the model online during deployment [17, 19, 43, 50]. They build on the image reprojection principle at the core of self-supervised monocular depth estimation approaches [16] by exploiting consecutive frames acquired over time [51]. However, this strategy requires the camera to move constantly, with still objects, to obtain reliable self-supervision.

In contrast, our proposal aims to deal with the opposite setting, where the camera placement is fixed and moving agents appear in the scene.

3 Method

We now introduce our adaptation strategy to address three issues encountered when deploying monocular solutions [36, 37] in the wild: 1) incorrect/blurred predictions for some subjects in the scene, e.g., pedestrians, cars, etc. 2) inaccurate global structure, being not properly aligned with the real scene and 3) the predictions being up to an unknown scale factor. The three are dealt with by different steps in our pipeline, as spotlighted in Fig. 2.
3.1 Lightweight Fine-tuning with Pseudo Labels

Although modern monocular depth networks exhibit powerful generalization capabilities [36, 37], they sometimes fail when used in ever-seen environments. In particular, in the case of a fixed-camera installation with a viewpoint substantially different from those observed during training, these solutions might often miss the presence of agents in the scene, such as pedestrians or cars, as shown in Fig. 1. Purposely, we design a lightweight fine-tuning procedure to improve the perception of such agents by the monocular network by relying on pseudo labels obtained in two steps.

**Pseudo labels initialization.** We initialize the pseudo labels with the depth values estimated by monocular depth network [36, 37] considering its excellent generalization performance except for the weaknesses mentioned earlier.

**Agents rectification and fine-tuning.** In order to recover the miss-estimation of subjects in the scene or the blurred estimates, we distill proper depth labels. Purposely, we first detect possibly moving agents in the scene, for instance, through an instance segmentation network such as MaskRCNN from the Detectron2 framework [46]. Then, by assuming each agent is standing or moving over the ground plane, we generate pseudo labels by replacing the depth of each instance with the depth value of the lowest pixel in the instance itself – i.e., the contact point with the ground. For complex agents, such as bicycles or motorcycles, we approximate the riders’ depth to that of the vehicles. For bags and hand-holding items, e.g. umbrellas, the depth will match that of the closest pedestrian. This process ignores other semantic classes. At deployment time, this allows for rapidly collecting a small set of samples for fine-tuning the original model, considerably improving the depth accuracy for such subjects, as shown in Fig. 1.

3.2 Ground Plane Estimation and Scene Alignment

After correcting the depth for agents in the scene, there are still evident errors between scene structures reconstructed by the fine-tuned depth model and the sensed environment. As illustrated by Fig. 3 on the right, the ground plane reconstructed from the predicted depth map (red) is not properly aligned with the real 3D plane in the scene (green). This behaviour is probably a consequence of the very different viewpoints in training images [3], yielding a degradation of the predicted relative depth and scale recovery process (when feasible). To solve this issue, we aim to estimate the real ground plane in the scene and use it to restore the proper structure of the scene in the predicted depth map.

**Ground Plane Estimation.** According to the perspective principle, we can model a 3D plane as in the left part of Eq.1:

\[
\begin{align*}
    a_g X + b_g Y + c_g Z &= d_g \\
    X &= \frac{Z x}{f}, \\
    Y &= \frac{Z y}{f}, \\
    H &= \frac{Z h}{f}
\end{align*}
\]

where \((x,y)\) and \((X,Y)\) denote, respectively, the pixel coordinates and projected 3D coordinates, \(Z\) the depth, \((a_g, b_g, c_g, d_g)\) the ground plane parameters, \(f\) the camera focal length, \(H\) and \(h\) the actual height and pixel height of an object in the scene. From it, we can derive (right side of Eq.1) the plane equation as a function of the 2D coordinates and height of a known object.

Inspired by [45], we use the moving agents previously detected as probes in the scene to estimate plane coefficients \((a_g, b_g, c_g)\). Specifically, by detecting a single agent in more than
Figure 3: **Structural misalignment between predicted depth and real scene.** For a single scene with pedestrians walking around (a), we report ground normals \((a, b, c)\) obtained from predicted depth (red), our ground plane estimation method (blue), and ground truth (green). On the right, we visualize the misalignment between predicted and real planes.

three frames, we record their corresponding pixel height \(h\) and standing point coordinates \((x,y)\). Then, assuming no co-linear positions, coefficients \((a_g, b_g, c_g)\) can be estimated using the least squares method, with \(c\) and the actual height \(H\) of the agent regarded as constants. Once \((a_g, b_g, c_g)\) is given, the relative depth \(Z\) of any ground point can be estimated as a function of \(d_g H\) – i.e., up to an unknown scale factor. Despite this, this cue is enough to proceed with the alignment step discussed next.

**Scene Alignment.** We can exploit the estimated ground plane to align the predicted depth map, making it more congruent with the 3D structure of the scene. Given the predicted depth, a common practice for aligning it to a set of known depth values in metric scale consists of using the least squares algorithm \([6, 36, 37]\) or a non-linear model, i.e. a CNN \([45]\). However, this would not fit with the priors we derive from the moving agents since we can estimate the ground plane model without precisely segmenting it from the rest of the scene. Hence, we introduce an alternative approach to align the ground plane in the predicted depth map with the one modeled by our method.

Any 3D point \((X, Y, Z)\) in the estimated depth map can be projected onto the ground plane according to the ground parameters \((a_p, b_p, c_p, d_p)\), as formulated in Eq.\(2\).

\[
X_g = \frac{a_p d_p - a_p c_p Z - a_p b_p Y + b_p^2 X}{a_p^2 + b_p^2} \quad Y_g = \frac{b_p d_p - b_p c_p Z - a_p b_p X + a_p^2 Y}{a_p^2 + b_p^2} \quad Z_g = Z \tag{2}
\]

Thus, we can align the projections we obtain according to the ground plane model extracted from predicted depth maps with those yielded by the model obtained according to the moving agents’ motion. To calculate the surface normal from depth predictions, we can fit \(a_g X + b_g Y + c_g Z = d_g\) using the least square algorithm. It requires identifying a portion of the scene belonging to the ground plane by manually annotating a single image captured after installation or directly during deployment according to agents’ motion.

After projecting all the 3D points onto the plane, i.e. \((X_g, Y_g, Z)\), the 3D plane will be re-projected into the image plane according to the camera intrinsic parameters, yielding new pixel coordinates \((x', y')\). Finally, we can adjust the prediction to fit the ground plane model estimated through agents’ motion substituting \((x', y')\) in Eq. 1.
3.3 Absolute Scale Recovery

Although accurate in terms of relative depth, predictions of state-of-the-art models [36, 37] are often up to an unknown scale factor, whereas knowing the absolute depth is often necessary for practical applications. According to [30, 47], the missing scale can be restored by knowing the camera height over the ground. Unfortunately, they rely on the assumption that the Z axis of the camera is roughly parallel to the ground plane, a condition not always met in practice.

To remove this constraint, we design a custom method to deal with arbitrarily oriented cameras, by estimating the depth for the bottom-most pixel in the center of the image, assumed as anchor point. Fig. 4 shows that in the $O_1$ setup, in which the Z-axis of the camera is parallel to the ground, the closest point laying on it can be represented as $P$, where the Z-axis projection $D_{min}$ can be calculated from camera height $H_c$, image height $h_i$ and focal $f$ as they compose a similar triangle. In the $O_2$ setup, with an arbitrarily titled camera, point $P$ is now positioned at $P'$, and the closest point has changed to $P''$, which is the intersection between the origin-$P'$ ray with the plane. Given the current ground normal and original normal $N(0,1,0)$, the pose matrix $R$ from $O_1$ to $O_2$ can be estimated to achieve the coordinate transformation from $P$ to $P'$. After transforming $P''$ to the system $O_2$ by multiplying with $R^{-1}$, the $D'_{min}$ for the anchor point in the depth map will be estimated in metric scale. This latter will be used, together with predicted depth for this very same point, for restoring metric scale on the entire depth map, instead of using the median rescaling technique based on ground truth depth. This strategy assumes the anchor point is on the ground plane and not occluded by moving agents.

4 Experimental Results

4.1 Datasets

We run our experiments on a mixture of synthetic and real datasets, with a static camera mounted over the scene pointing toward roads, sidewalks, or pedestrian areas. A total of seven sequences are used – Fig. 5 shows an example for each – grouped into three categories:

**Synthetic data (CARLA).** We generate three sequences using CARLA simulator [8],
Table 1: Quantitative results – on-site adaptation. We report error metrics on the seven sequences, for original DPT [36], DPT after lightweight fine-tuning (DPT-ft) and with test-time scene alignment (DPT-ft-align). We highlight first, second, and third best results.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Method</th>
<th>SLog_2</th>
<th>RMSE_2</th>
<th>Abs rel_2</th>
<th>Sq rel_2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DPT [36]</td>
<td>0.051</td>
<td>4.098</td>
<td>0.184</td>
<td>0.707</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.044</td>
<td>4.012</td>
<td>0.171</td>
<td>0.656</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.018</td>
<td>3.421</td>
<td>0.103</td>
<td>0.399</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.026</td>
<td>3.677</td>
<td>0.131</td>
<td>0.512</td>
</tr>
<tr>
<td>C1</td>
<td>DPT [36]</td>
<td>0.061</td>
<td>5.164</td>
<td>0.221</td>
<td>1.183</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.029</td>
<td>3.476</td>
<td>0.144</td>
<td>0.594</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.011</td>
<td>2.301</td>
<td>0.041</td>
<td>0.307</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.009</td>
<td>2.201</td>
<td>0.036</td>
<td>0.288</td>
</tr>
<tr>
<td>C2</td>
<td>DPT [36]</td>
<td>0.056</td>
<td>2.355</td>
<td>0.183</td>
<td>1.274</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.042</td>
<td>2.052</td>
<td>0.131</td>
<td>0.594</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.018</td>
<td>1.282</td>
<td>0.112</td>
<td>0.154</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.014</td>
<td>1.103</td>
<td>0.098</td>
<td>0.115</td>
</tr>
<tr>
<td>C3</td>
<td>DPT [36]</td>
<td>0.069</td>
<td>5.705</td>
<td>0.219</td>
<td>1.724</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.057</td>
<td>3.666</td>
<td>0.101</td>
<td>0.483</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.019</td>
<td>3.666</td>
<td>0.101</td>
<td>0.483</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.019</td>
<td>3.666</td>
<td>0.101</td>
<td>0.483</td>
</tr>
<tr>
<td>K1</td>
<td>DPT [36]</td>
<td>0.149</td>
<td>4.718</td>
<td>0.337</td>
<td>2.289</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.048</td>
<td>2.948</td>
<td>0.158</td>
<td>0.658</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.057</td>
<td>3.686</td>
<td>0.235</td>
<td>1.441</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.045</td>
<td>2.876</td>
<td>0.152</td>
<td>0.552</td>
</tr>
<tr>
<td>K2</td>
<td>DPT [36]</td>
<td>0.052</td>
<td>1.626</td>
<td>0.151</td>
<td>0.546</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.049</td>
<td>1.579</td>
<td>0.148</td>
<td>0.507</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.033</td>
<td>0.999</td>
<td>0.099</td>
<td>0.293</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.036</td>
<td>1.075</td>
<td>0.107</td>
<td>0.337</td>
</tr>
<tr>
<td>R1 (Indoor)</td>
<td>DPT [36]</td>
<td>0.054</td>
<td>3.786</td>
<td>0.198</td>
<td>0.748</td>
</tr>
<tr>
<td></td>
<td>DPT-ft</td>
<td>0.052</td>
<td>3.654</td>
<td>0.189</td>
<td>0.723</td>
</tr>
<tr>
<td></td>
<td>DPT-align</td>
<td>0.051</td>
<td>3.604</td>
<td>0.167</td>
<td>0.671</td>
</tr>
<tr>
<td></td>
<td>DPT-ft-align</td>
<td>0.041</td>
<td>3.136</td>
<td>0.159</td>
<td>0.542</td>
</tr>
</tbody>
</table>

4.2 Implementation details
Given its outstanding generalization performance, we adopt DPT [36] as the baseline monocular depth estimation network in our experiments, on a single 3090 GPU. We adapt it for each of the seven sequences through the pipeline introduced earlier. Concerning the lightweight fine-tuning process, we select the first 342, 463, and 406 frames from the three synthetic sequences, the first 273 and 107 frames from the KITTI sequences, and the first 684 and 679 frames from the real scenes. On top of them, we generate pseudo labels and fine-tune DPT for 30 epochs. This strategy simulates an on-site adaptation carried out on the first
frames collected after installation. Then, we evaluate the effectiveness of our pipeline on the remaining frames by enabling test-time scene alignment as well. We compute standard metrics concerning the monocular depth estimation task [11, 42], such as SiLog, RMSE, Abs Rel, and Sq Rel. The evaluation is performed both by rescaling predictions using ground truth depth itself [36], as well as by restoring absolute scale following our approach.

### 4.3 Quantitative results

**Lightweight fine-tuning and scene alignment.** We start by evaluating the effectiveness of the first two steps in our pipeline. Table 1 gathers the results achieved by the original DPT model and those obtained after performing the lightweight fine-tuning and scene alignment steps. In this experiment, absolute scale is restored according to median rescaling [51]. Starting from the former, we refer to DPT-ft for the model fine-tuned on pseudo labels without alignment. We can notice how the lightweight fine-tuning improves the accuracy compared to the original model. In particular, by correcting several errors in correspondence of the moving pedestrians or vehicles. By performing scene alignment on the predictions by the original DPT model (DPT-align) we improve the accuracy as well, often with a major gain with respect to what is achieved by the light-weight finetuning – since this latter only acts on moving agents, representing a minority of the pixels in the scene. Finally, combining fine-tuning and alignment – DPT-ft-align entries – further decreases the error in most cases, except on C1 and R1 in which the ground plane covers the largest portion of the scene compared to other sequences.

**Absolute scale recovery.** To conclude, we evaluate the effectiveness of our scale recovery strategy. For this purpose, we first compare our approach with alternative techniques
[30, 47] exploiting knowledge of the camera height but assuming the camera to be parallel to the ground plane. Table 2 reports the average depth value for the anchor points estimated by the different techniques, followed by their error compared to ground truth depth. For KITTI, the anchor is replaced by the closest pixel with available ground truth. We can notice how our approach consistently restores the absolute scale more accurately. Although the two are substantially equivalent on KITTI, where the camera is almost parallel to the ground, our solution results superior in the real datasets featuring a significant camera tilt.

Finally, we compare the accuracy of depth maps predicted by DPT after having recovered scale through our technique with the results obtained by a model trained directly on-place with supervision from a stereo camera in the scene, i.e., with knowledge about the metric scale of the scene. We assume this configuration sets an upper bound at the performance a monocular network could achieve by having perfect knowledge of the scale during training. Table 3 shows how the outcome of this experiment, with MonoDepth2 [16] used for the comparison. After our scale recovery step, DPT-ft-align often yields results close to those by MonoDepth2 – outperforming it a few times – confirming the proposal’s effectiveness.

4.4 Qualitative Results

To conclude, Fig. 6 shows how our whole framework dramatically reduces the error being the primary source of failure – i.e., in the presence of agents such as pedestrians and cars, or in the farthest parts of the scene, where the ground plane misalignment between predicted and real depth becomes more prominent. After processing, the error remains slightly higher on objects farther from the ground plane – e.g., structures on the sidewalk (left) or the obstacle in the very foreground (center) – where no optimization is performed by our method.

5 Conclusion

This paper proposes a novel pipeline for the on-site adaptation of a monocular depth network specifically designed to deal with fixed-camera installations. Our method allows for a lightweight fine-tuning of the model and a test-time scene alignment of the predicted depth maps by leveraging the presence of agents moving freely in the scene, as well as for recovering the metric scale of the scene by only knowing the height at which the camera is. Our experiments show how a few images collected just after deployment allow for improving the results achieved by the DPT network thanks to our solution.

Acknowledgment. We sincerely thank the scholarship supported by China Scholarship Council (CSC).
References


