SfM-TTR: Using Structure from Motion for Test-Time Refinement of Single-View Depth Networks

Sergio Izquierdo Javier Civera
I3A, University of Zaragoza, Spain
{izquierdo, jcivera}@unizar.es

Abstract

Estimating a dense depth map from a single view is geometrically ill-posed, and state-of-the-art methods rely on learning depth's relation with visual appearance using deep neural networks. On the other hand, Structure from Motion (SfM) leverages multi-view constraints to produce very accurate but sparse maps, as matching across images is typically limited by locally discriminative texture. In this work, we combine the strengths of both approaches by proposing a novel test-time refinement (TTR) method, denoted as SfM-TTR, that boosts the performance of single-view depth networks at test time using SfM multi-view cues. Specifically, and differently from the state of the art, we use sparse SfM point clouds as test-time self-supervisory signal, fine-tuning the network encoder to learn a better representation of the test scene. Our results show how the addition of SfM-TTR to several state-of-the-art self-supervised and supervised networks improves significantly their performance, outperforming previous TTR baselines mainly based on photometric multi-view consistency. The code is available at https://github.com/serizba/SfM-TTR.

1. Introduction

Obtaining accurate and dense depth maps from images is a challenging research problem and an essential input in a wide array of fields, like robotics [67], augmented reality [36], endoscopy [42], or autonomous driving [22]. Single-view per-pixel depth estimation is even more challenging, as it is geometrically ill-posed in the general case. However, in the last decade, intense research on deep models applied to this task has produced impressive results, showing high promise for real-world applications.

Single-view depth learning was initially addressed as a supervised learning problem, in which deep networks were trained using large image collections annotated with ground truth depth from range (e.g., LiDAR) sensors [12, 30]. At present, this line of research keeps improving the accuracy of single-view depth estimates by better learning models and training methods, as illustrated for example by [6, 59].

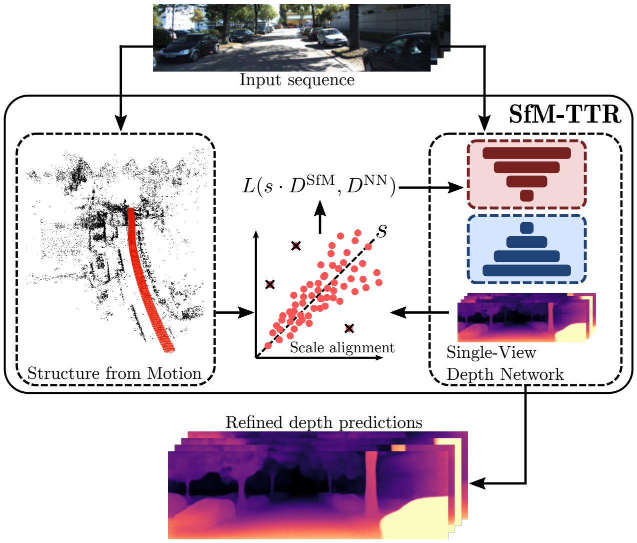
Input sequence
SfM-TTR
$L(s \cdot D^{\mathrm{SfM}}, D^{\mathrm{NN}})$
s
Structure from Motion
Scale alignment
Single-View Depth Network
Refined depth predictions

Figure 1. **SfM-TTR overview**. Our approach assumes an existing pre-trained depth network and an input sequence at test time. We estimate a SfM 3D reconstruction using the input sequence, and depth maps using a single-view depth network. We align the SfM point cloud with the network's depth to obtain a pseudo-ground truth to refine the network encoder, improving its representation of the test scene and producing significantly more accurate depth estimates.

In parallel to improving the learning side of the problem, several works are incorporating single- and multi-view geometric concepts to depth learning, extending its reach to more general setups. For example, [3, 15] propose camera intrinsics-aware models, enabling learning and predicting depths for very different cameras. More importantly, many other works (e.g. [20]) use losses based on multi-view photometric consistency, enabling self-supervised learning of depth and even camera intrinsics [21].

Incorporating single- and multi-view geometry into depth learning naturally links the field to classic research on Structure from Motion (SfM) [23, 45], visual odometry [14, 44] and visual SLAM [8, 9]. These methods typically produce very accurate but sparse or semi-dense reconstructions of high-gradient points using only multi-view geometry at test time. Among the many opportunities for cross-fertilization of both fields (e.g., using depth networks in visual SLAM [48] or SfM for training depth networks [28, 32, 56]), our work focuses on using SfM for refining single-view depth networks at test time.

As single-view depth applications typically include a moving camera, several recent works incorporate multiple views at inference or refine single-view depth networks with multi-view consistency cues [4, 11, 36, 37, 46, 49, 52]. Most approaches, however, rely mainly on photometric losses, similar to the ones used for self-supervised training. These losses are limited to be computed between close views, creating weak geometric constraints. Our contribution in this paper is a novel method that, differently from the others in the literature, uses exclusively a SfM reconstruction for TTR. Although SfM supervision is sparser than typical photometric losses, it is also significantly less noisy as it has been estimated from wider baselines. Our results show that our approach, which we denote as SfM-TTR, provides state-of-the-art results for TTR, outperforming photometric test-time refinement (Ph-TTR) for several state-of-the-art supervised and self-supervised baselines.

2. Related Work

Although there exists a large corpus of work on single-view depth under certain assumptions on the scene geometry, e.g. [2, 5, 24, 40, 47, 60], and on multi-view depth, e.g. [26, 62], we focus here on approaches that are mainly based on learning and target general scenes.

2.1. Supervised Single-View Depth Learning

Several early works addressed single-view depth learning either directly from the image [43] or via semantic labels [33] before the deep learning era. The seminal works by Eigen et al. [12, 13] significantly improved the prediction accuracy by training deep networks supervised with ground-truth depth from range sensors. Since then, single-view depth networks have received significant attention from the research community, focusing on improving the performance by using more sophisticated architectures and losses, e.g., [30, 31, 35, 38, 41, 53, 54]. A re-formulation of the problem as an ordinal regression has led to further improvement [6, 7, 17]. Recently, Bae et al. [4] fuse the single-view depths from multiple images, but differently from us without TTR of the network. Despite their remarkable progress, we show in this paper that the accuracy of state-of-the-art supervised depth networks is further improved by our SfM-TTR proposal.

2.2. Self-Supervised Single-View Depth Learning

As ground truth depth annotations are uncommon, self-supervised approaches emerged as an alternative, exploiting multi-view photometric consistency [19, 65]. Attracted by the convenience of training without depth labels, many works have further focused on addressing this paradigm, e.g., [25, 34, 46, 57, 58, 61, 66]. Close to our work, SfM has been used as supervisory signal during training, but limited to probabilistic networks [28], or using disparities [56] that require stereo images. Among self-supervised works, Monodepth2 [20], which proposed a robust loss to handle occlusions and discard invalid pixels, is of particular relevance. Monodepth2 is the base of most state-of-the-art approaches, and specifically of the baselines we chose to validate SfM-TTR: CADepth [55], that uses self-attention to capture more context, DIFFNet [64], that applies feature fusion to incorporate semantic information, and ManyDepth [52], that leverages more than one frame at inference to improve the predictions. All these networks use a typical encoder-decoder architecture and can be seamlessly refined with our SfM-TTR method.

2.3. Test-Time Refinement (TTR)

Multi-view consistency is the basis for both self-supervised depth learning and bundle adjustment [50], this last one naturally occurring at test time. Inspired by that, TTR was proposed [10, 11], updating the network with the same self-supervised losses from training. Similarly, McCraith et al. [37] showed the benefits of encoder-only fine-tuning and proposed two TTR modes: sequence- and instance-wise. Similar approaches were presented by Watson et al. [52], with multiple input images for the network, Shu et al. [46], with a feature-metric loss, and Kuznietsov et al. [29], using a replay buffer. All these TTR methods inherit the small baseline limitations from photometric losses, showing small improvements for medium and large depths for which close views produce small parallax. At these depths, our SfM-TTR introduces wide baseline cues, due to the higher invariance of features matching at wide baselines. This leads to significant improvements over the state of the art.

Tiwari et al. [49] iterates over optimizing the parameters of a single-view depth network and running pseudo-RGBD SLAM for pose estimation, but their alignment ignores the depth distributions, which results in smaller improvements compared to ours. Luo et al. work [36] is more related to ours, using SfM and optical flow as geometric constraints. However, despite heavy optimization (taking up to 40 minutes for a sequence of less than 250 frames), their TTR cannot improve over baseline networks on KITTI. Instead of defining derived constraints, we directly optimize the encoder using the sparse reconstruction as pseudo ground truth, resulting in a lighter and more effective pipeline.

3. SfM-TTR

Our SfM-TTR takes *any* single-view depth network, trained either supervised or self-supervisedly, and fine-tunes it for the test data by a three-stage process. As a brief summary, we first estimate a sparse feature-based reconstruction of the scene from multiple views (Section 3.1) and predict depth outputs with the network (Section 3.2). Then, we align the scale of the sparse point cloud and the network's depth (Section 3.3). Finally, we fine-tune the network using the depths of the aligned sparse point cloud as supervisory signal (Section 3.4).

3.1. Multi-View Depth from SfM

We perform a 3D reconstruction of the target scene using an off-the-shelf SfM algorithm. In our current implementation we use COLMAP [45], as it shows a high degree of accuracy and robustness in a wide variety of scenarios, although alternative SfM or visual SLAM implementations could also have been used [1,9,39].

From a set of images $\mathcal{I} = \{\mathbf{I}_1, \ldots, \mathbf{I}_K\}$, $\mathbf{I}_k \in \mathbb{R}^{w \times h \times 3}$ $\forall k \in \{1, \ldots, K\}$ of a scene, COLMAP returns a set of J 6-degrees-of-freedom poses $\mathcal{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_J\}$, $\mathbf{P}_j = \left(\begin{smallmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0} & 1 \end{smallmatrix}\right) \in \mathbf{SE}(3)$ $\forall j \in \{1, \ldots, J\}$, $J \leq K$, corresponding to the cameras that the method was able to register, and the set of 3D keypoints $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_I\}$, $\mathbf{X}_i \in \mathbb{R}^3$ $\forall i \in \{1, \ldots, I\}$ that were reconstructed, all of them in a common reference frame. The camera with pose $\mathbf{P}_j$ observes a subset of L_j points from the total set of 3D points $\mathcal{X}_j = \{\mathbf{X}_1, \ldots, \mathbf{X}_{L_j}\} \subset \mathcal{X}$. COLMAP final estimates are obtained by minimizing the sum of the squared reprojection errors $\sum_{j=1}^{J} \sum_{l=1}^{L_j} \mathbf{r}_{l,j}^2$.

The depth of each of the l^{th} point in the j^{th} camera frame is computed as

$$D_{l,j}^{\text{SfM}} = \mathbf{e}_3^\top \left(\mathbf{R}_j^\top \left(\mathbf{X}_l - \mathbf{t}_j \right) \right) \tag{1}$$

where $\mathbf{e}_3 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$ is the unit vector in the optical axis direction. We will group the depths for the sparse set of points $\mathcal{X}_j$ in the set $\mathcal{D}_j^{\text{SfM}} = \{D_{1,j}^{\text{SfM}}, \ldots, D_{L_j,j}^{\text{SfM}}\}$, $D_{l,j}^{\text{SfM}} \in \mathbb{R}_{>0}$ $\forall l \in \{1, \ldots, L_j\}$, and the depths for all images in $\mathcal{D}^{\text{SfM}} = \{\mathcal{D}_1^{\text{SfM}}, \ldots, \mathcal{D}_J^{\text{SfM}}\}$ $\forall j \in \{1, \ldots, J\}$.

3.2. Single-View Depth from Neural Networks

Our SfM-TTR method can be applied to any architecture, and hence its predicted depth $\mathbf{D}_j^{\text{NN}} \in \mathbb{R}^{w \times h}$ for an image $\mathbf{I}_j$ can be generally formulated as

$$\mathbf{D}_j^{\text{NN}} = h\left(g\left(\mathbf{I}_j, \boldsymbol{\theta}_g \right), \boldsymbol{\theta}_h \right) \tag{2}$$

where $h(\cdot)$ and $g(\cdot)$ stand respectively for the decoder and encoder parts of the deep networks, and $\boldsymbol{\theta}_h$ and $\boldsymbol{\theta}_g$ their respective weights, that have been trained either supervised or self-supervisedly.

Note that the depths $\mathcal{D}_j^{\text{SfM}}$ and $\mathbf{D}_j^{\text{NN}}$ correspond to the same image $\mathbf{I}_j$ but are respectively sparse and dense, having hence a different number of elements, and they may have different scales. The scale is unobservable by COLMAP and self-supervised networks, while it is learned from the training data by supervised networks.

In order to estimate the relative scale between $\mathcal{D}_j^{\text{SfM}}$ and $\mathbf{D}_j^{\text{NN}}$ and refine at inference time the deep network, we have to select from $\mathbf{D}_j^{\text{NN}}$ those elements corresponding to the sparse depth of $\mathcal{D}_j^{\text{SfM}}$. For a general element l, we use the sampling operator $[\cdot]$ to access the depth corresponding to the pixel coordinates $\mathbf{p}_{l,j}$

$$D_{l,j}^{\text{NN}} = \mathbf{D}_j^{\text{NN}} \left[\mathbf{p}_{l,j} \right] \tag{3}$$

where $\mathbf{p}_{l,j}$ is obtained from the coordinates of the 3D points $\mathbf{X}_l \in \mathcal{X}_j$ and the camera pose $\mathbf{P}_j$ and applying the pinhole projection function, that we will denote as $\pi(\cdot)$

$$\mathbf{p}_{l,j} = \begin{pmatrix} u & v \end{pmatrix}_{l,j}^\top = \pi \left(\mathbf{R}_j^\top \left(\mathbf{X}_l - \mathbf{t}_j \right) \right) \tag{4}$$

We finally group the depths predicted by the deep network for the sparse set of points $\mathcal{X}_j$ in a joint set $\mathcal{D}_j^{\text{NNs}} = \{D_{1,j}^{\text{NN}}, \ldots, D_{L_j,j}^{\text{NN}}\}$, $D_{l,j}^{\text{NN}} \in \mathbb{R}_{>0}$, and the depths for all images in $\mathcal{D}^{\text{NNs}} = \{\mathcal{D}_1^{\text{NNs}}, \ldots, \mathcal{D}_J^{\text{NNs}}\}$ $\forall j \in \{1, \ldots, J\}$.

3.3. Scale Alignment

Scale alignment is not trivial in our setup, as both $\mathcal{D}^{\text{SfM}}$ and $\mathcal{D}^{\text{NNs}}$ are affected by heteroscedastic (depth-dependent) inlier noise and contain a non-negligible rate of outliers. In addition, we are interested in removing outliers from $\mathcal{D}^{\text{SfM}}$, but we do want to keep them in $\mathcal{D}^{\text{NNs}}$, as then our SfM-TTR can reduce their errors. We developed a novel scale alignment method with two stages: we make a first fit with a strict inlier model to obtain an accurate relative scale, and then relax it in the second stage to select the points used for self-supervision from $\mathcal{D}^{\text{SfM}}$.

In the first stage we use RANSAC [16], computing 1D model instantiations, $s_{l,j} = D_{l,j}^{\text{NN}} / D_{l,j}^{\text{SfM}}$ and consider in the inlier set $\mathcal{D}^{\text{NNs}\checkmark} \subset \mathcal{D}^{\text{NNs}}$ and $\mathcal{D}^{\text{SfM}\checkmark} \subset \mathcal{D}^{\text{SfM}}$ all depths pairs $\{D_{l',j'}^{\text{NN}}, D_{l',j'}^{\text{SfM}}\}$ for which the following holds

$$\frac{\left(s_{l,j} \cdot D_{l',j'}^{\text{SfM}} - D_{l',j'}^{\text{NN}} \right)^2}{s_{l,j} \cdot D_{l',j'}^{\text{SfM}}} \leq \tau \tag{5}$$

where τ is the inlier threshold.

In most occasions, the distribution of depths in the image is highly unbalanced, with higher frequencies for closer depths. This, together with the heteroscedasticity of the depth errors (errors are smaller for closer depths), causes that the frequently used median scale [36] corresponds to close points, biasing the estimation. Using least squares with all the inlier set $\{\mathcal{D}^{\text{NNs}\checkmark}, \mathcal{D}^{\text{SfM}\checkmark}\}$ is not a good alternative either, the fit will be biased in this case towards large depths as they have larger errors. For these reasons, we use weighted least squares to obtain a refined estimate of s with the depths $D_{l,j}^{\text{NN}\checkmark} \in \mathcal{D}^{\text{NNs}\checkmark}$ and $D_{l,j}^{\text{SfM}\checkmark} \in \mathcal{D}^{\text{SfM}\checkmark}$

$$\hat{s} = \operatorname*{argmin}_s \sum_j \sum_l w_{l,j}^s \left(s \cdot D_{l,j}^{\text{SfM}\checkmark} - D_{l,j}^{\text{NN}\checkmark}\right)^2 \tag{6}$$

where $w_{l,j}^s$ is a per-pixel weight, that should be proportional to the inverse of the expected depth variance $\sigma_{l,j}^2$. Under the reasonable assumption of similar baselines and matching noises for all reconstructed points, it is well known that the variance grows with the depth squared [23] and hence we can use as weights

$$w_{l,j}^s = {}^1\!/\sigma_{l,j}^2 \approx {}^1\!/\left(D_{l,j}^{\text{NN}\checkmark}\right)^2 \tag{7}$$

Finally, we use s_j from the optimization in Equation 6 to obtain the final set of inliers $\{\mathcal{D}^{\text{NNs}\checkmark\checkmark}, \mathcal{D}^{\text{SfM}\checkmark\checkmark}\}$ that we will use for our SfM-TTR. We proceed similarly to Equation 5, but this time using the absolute value in the numerator, relaxing in this manner the model and favoring the inclusion of noisy depth predictions from the network depth set $\mathcal{D}^{\text{NNs}}$ in order to have the chance to improve them at test time.

3.4. Test-Time Refinement

We refine the target network for the selected scene by updating its parameters using the depths in the final inlier set $\mathcal{D}^{\text{SfM}\checkmark\checkmark}$ as supervision. As in [36], we optimize over the complete scene, thus obtaining a refined network with more consistent predictions across all views. This is different from other TTR works, such as [37], in which they refine a different network for each frame of the sequence.

Each batch update works as follows. We sample an image $\mathbf{I}_j$ from the sequence and do a feed-forward pass through the network to obtain the depth prediction $\mathbf{D}_j^{\text{NN}}$. Then we supervise the prediction with the sparse pseudo ground truth $\mathcal{D}_j^{\text{SfM}\checkmark\checkmark}$. This supervision is weighted according to the reliability of the reconstructed 3D points, that we approximate based on their reprojection errors as $w_{l,j}^{\boldsymbol{\theta}} = \exp(-\|\mathbf{r}_{l,j}\|_2^2)$.

$$\mathcal{L} = \frac{1}{|\mathcal{D}_j^{\text{SfM}\checkmark\checkmark}|} \sum_l w_{l,j}^{\boldsymbol{\theta}} \|\hat{s} \cdot D_{l,j}^{\text{SfM}\checkmark\checkmark} - D_{l,j}^{\text{NN}\checkmark\checkmark}\|_1 \tag{8}$$

As state-of-the-art depth networks already produce sharp predictions with well-defined object contours, we argue that our refinement should only optimize the internal understanding of the scene. Hence, we follow a similar approach as [37] and only update the encoder parameters during the TTR, keeping the rest of the network fixed. Our TTR optimization can be hence formulated as $\hat{\boldsymbol{\theta}}_g = \operatorname{argmin}_{\boldsymbol{\theta}_g} \mathcal{L}$. In this manner, the frozen decoder $h(\cdot)$ keeps producing sharp predictions, but now they stem from a more informed representation of the underlying scene.

4. Experimental Results

4.1. Implementation Details and Baselines

We validate our proposed SfM-TTR by applying it to different state-of-the-art baselines. Specifically, we provide evaluations with the baselines CADepth [55], DIFFNet [64], and ManyDepth [52] as representative of self-supervised approaches. We also implemented it on AdaBins [6] to benchmark SfM-TTR's performance also with a representative supervised model. The same set of hyperparameters was used for SfM-TTR with all baselines, achieving a substantial improvement in all of them without requiring individual tuning.

For the sparse reconstruction, we run COLMAP [45] with its default parameters, using a single pinhole camera model per sequence and sequential matching. Although we use all available images from a sequence to create the sparse reconstruction, the network is only optimized with the target frames of the evaluation. Regarding our scale alignment, we detect outliers running RANSAC for 20 iterations with inlier threshold $\tau = 0.5$. For the TTR optimization, we use Adam [27] applied to the encoder parameters, $\boldsymbol{\theta}_g$, with a learning rate of 10^{-4} for 200 steps.

For comparison, we also implemented the instance-wise photometric refinement (Ph-TTR) from ManyDepth [52][1], based on the work of McCraith et al. [37], which updates the weights of the network encoder during inference using the photometric loss from the training. Table 1 validates our implementation, showing similar performance as the one reported by the authors in [52].

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓
ManyDepth [52] Ph-TTR ⋄	0.087	0.696	4.183	0.167
ManyDepth [52] Ph-TTR ∗	0.088	0.681	4.122	0.168

Table 1. ManyDepth Ph-TTR [52] (⋄) and our own implementation (∗) obtain similar metrics.

4.2. Dataset

We run all evaluations on the KITTI dataset [18], the common benchmark for single- and multi-view depth learning. Regarding the KITTI ground truth for depth learning evaluations, the literature is split among those following Eigen et al. [13], with reprojected LIDAR point clouds, and those using the newer and improved ground truth [51], which aggregates 5 consecutive frames and handles dynamic objects. Given the higher reliability of the new ground truth, we used it to evaluate all the baselines on the Eigen test split with all the images that contain ground truth, a total of 652. We provide evaluation without and with the Eigen cropping, see Table 4 and Table 5. For fairness and

[1] The TTR code was not available in the authors' repository at the time of submission.

completeness, as some methods present results with the old ground truth, we also include an evaluation with the LiDAR reprojected depths, on the complete Eigen split with 697 images. We report additional results directly taken from the corresponding papers, see Table 6.

In a few of the KITTI test scenes the camera motion is insufficient for proper SfM convergence. Our SfM-TTR cannot refine the depth in those cases, but for a fair comparison, we included these sequences in the global metrics using the results of the network without SfM-TTR.

Note that although we have presented a novel scale alignment, for the sake of fairness we align the self-supervised predictions and the ground truth with the per-image median, as commonly done [20, 52]. Also following the common evaluation practices, we set a maximum depth of 80 meters.

4.3. Comparisons against Baselines

We demonstrate the benefits of our method by comparing the results of applying a photometric refinement (Ph-TTR) and ours (SfM-TTR) on the baseline networks. Table 4 shows how our SfM-TTR consistently and significantly improves the predictions of all networks, obtaining superior performance than the photometric refinement. Besides, Ph-TTR fails to improve over CADepth without TTR. The most likely reason is that it requires individual hyperparameter tuning, which was not required for our SfM-TTR.

The advantages of our proposed method are especially noticeable for large depths, where Ph-TTR cannot provide a good supervision signal due to the limited parallax between close frames. Our refinement, instead, leverages SfM, which triangulates points from the complete sequence. This produces better estimates for distant points and better supervision, resulting in a drastic reduction of the RMSE by up to 30%. This effect is clearly visible in Figure 2. Although smaller depths show comparable performance for Ph-TTR and SfM-TTR, the photometric loss does not help in areas with large depths. SfM-TTR, instead, provides a significant gain in performance in those areas.

The best results are obtained when applying our SfM-TTR to DIFFNet, even though the original DIFFNet without TTR performs slightly worse than ManyDepth. We believe that our TTR has a smaller effect on ManyDepth because it already leverages scene information by using multiple frames at inference time. SfM-TTR can also improve results on AdaBins, for which Ph-TTR cannot be implemented, as AdaBins does not provide a pose estimation module. This further demonstrates the effectiveness of directly optimizing for the 3D points from COLMAP.

Qualitatively, Figure 5 shows how predictions after SfM-TTR keep looking sharp with well-defined boundaries despite the sparsity of the pseudo-ground truth. We argue that optimizing the encoder enables a better understanding of the scene while freezing the decoder maintains the previously learned sharpness of the predictions. The error maps from Figure 4 reveal the differences between refinements, showing how our method can effectively reduce errors in regions where Ph-TTR cannot. The positive effect of SfM-TTR in distant points is visible in Figure 3, where large depths move closer to the ground truth after our refinement.

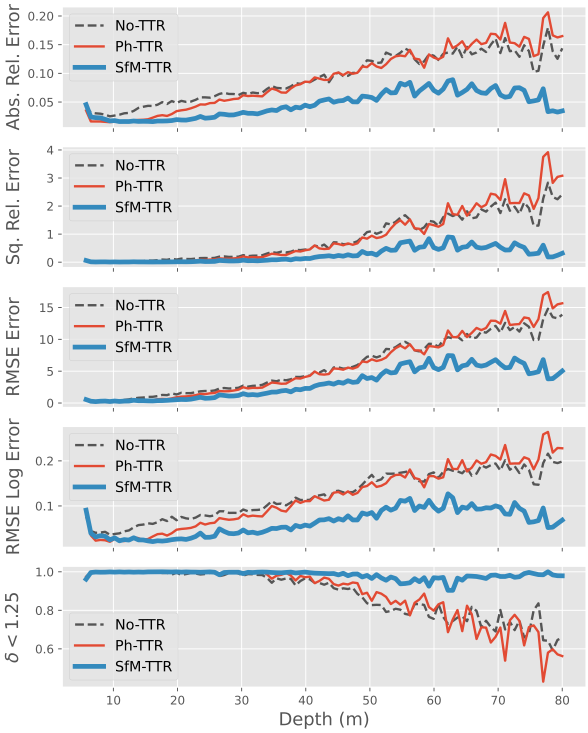

Figure 2. **Error metrics for different depths with DIFFNet.** Our SfM-TTR (thick blue) gives a substantial improvement over No-TTR (dashed black) and Ph-TTR (thin red) at medium and large depths. Ph-TTR offers some improvement over No-TTR at close depths, where the small baselines of photometric losses are informative, but it does not improve or it is slightly worse at medium and large depths. The metrics $\delta < 1.25^2$ and $\delta < 1.25^3$ are not plotted, as differences are small (see for example Table 4).

Regarding runtime efficiency, our method requires roughly 2 seconds per frame during the optimization, similar to Ph-TTR, and faster than other multi-view TTR that also use large baselines [36, 49].

4.4. Ablation Studies

To validate the relative importance of the individual components of our SfM-TTR, we perform ablation studies where we dispose some of our key components.

Table 2 shows a comparison between refining the complete network and only updating the encoder. Similar to [37], we obtain better results when only updating the encoder, further showing how light refinement schemes should only focus on improving the underlying representation of the network.

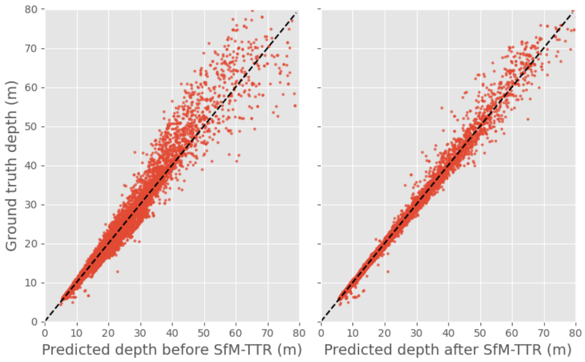

Figure 3. **Depth predictions and ground truth before and after SfM-TTR with DIFFNet.** The red dots stand for predicted pixel depths on a KITTI sequence with DIFFNet, the black dashed line stands for zero error. Note how after SfM-TTR the red dots gather closer to the dashed black line, illustrating that the predicted depths are closer to the ground truth ones.

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓
AdaBins [6]	0.072	0.325	3.134	0.112
AdaBins [6] + SfM-TTR (full model)	0.062	**0.204**	2.297	0.092
AdaBins [6] + SfM-TTR (encoder)	**0.060**	**0.204**	**2.260**	**0.091**
ManyDepth [52]	0.064	0.345	3.116	0.103
ManyDepth [52] + SfM-TTR (full model)	0.059	**0.293**	2.655	0.096
ManyDepth [52] + SfM-TTR (encoder)	**0.057**	0.294	**2.648**	**0.094**
CADepth [55]	0.078	0.403	3.432	0.119
CADepth [55] + SfM-TTR (full model)	0.069	**0.321**	2.824	**0.104**
CADepth [55] + SfM-TTR (encoder)	**0.068**	0.328	**2.821**	0.106
DIFFNet [64]	0.071	0.361	3.230	0.110
DIFFNet [64] + SfM-TTR (full model)	0.057	**0.273**	2.621	**0.092**
DIFFNet [64] + SfM-TTR (encoder)	**0.056**	**0.273**	**2.600**	0.093

Table 2. **Encoder vs. full network TTR.** Note how the best results are achieved with encoder-only TTR.

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓
AdaBins [6]	0.072	0.325	3.134	0.112
AdaBins [6] + SfM-TTR (median)	0.074	0.263	2.509	0.103
AdaBins [6] + SfM-TTR ($\mathcal{D}^{\text{NNs}\checkmark}, \mathcal{D}^{\text{SfM}\checkmark}$)	0.065	0.278	2.787	0.103
AdaBins [6] + SfM-TTR (Least Squares)	0.064	0.222	2.346	0.097
AdaBins [6] + SfM-TTR ($w^{\boldsymbol{\theta}}_{l,j} = 1$)	0.062	0.206	2.310	**0.091**
AdaBins [6] + SfM-TTR	**0.060**	**0.204**	**2.260**	**0.091**

Table 3. **Alignment ablation study.** Note the substantial improvement of our scaling approach (detailed in Section 3.3) over other alignments.

As shown in Table 3, using the mean of per-image medians [28, 36] alignment in our SfM-TTR, as well as other ablated versions of our method, worsens significantly the performance on AdaBins. The alignment is specially important for supervised models, as their scale is not corrected during the evaluation. With our alignment, we are accounting for outliers with RANSAC and for the heteroscedastic nature of the depth noise with weighted least squares, resulting in substantially more robust and accurate results.

5. Limitations

As our current implementation of SfM-TTR depends on COLMAP's output, it is inherently offline and its performance is bounded to the quality of the SfM results. Although we achieve good results in KITTI, a natural scenario and standard benchmark, more challenging setups for SfM (for example, dynamic objects, drastic appearance changes or low-parallax motion) are also problematic for SfM-TTR. Works addressing such SfM challenges [63] will also be beneficial for our method. Although we could easily replace COLMAP's reconstruction by that of an online real-time visual SLAM pipeline, e.g. [9], online and real-time refinement of deep models is not straightforward. We find these aspects relevant for our future work.

Although SfM-TTR excels at medium and large depths, we have noticed a comparable or slightly worse performance than Ph-TTR at very close depths, for which even the adjacent views used in Ph-TTR have sufficient parallax. Observe the metrics in Figure 2 for depths under 10 meters. This observation suggests a future line of research to combine the best from both Ph-TTR and SfM-TTR.

6. Conclusion

In this paper we have presented SfM-TTR, an effective test-time refinement for single-view depth networks that preserves the learned priors of supervised and self-supervised models while also leveraging wide-baseline multi-view constraints at inference. The key ingredient is formulating a TTR loss based on sparse SfM depths, which have been estimated from wider baselines than traditional photometric losses, that only consider adjacent frames. We propose a novel RANSAC-based method for scale alignment between SfM and the depth network that accounts for the depth outliers and its heteroscedastic noise. Very importantly, we use a fixed set of hyperparameters for our SfM-TTR for all experiments, without requiring per-architecture or per-sequence tuning.

Our experiments show that our SfM-TTR improves significantly the depth predictions of different state-of-the-art networks, supervised and self-supervised. We also outperform by a wide margin, in particular at medium and large depths, the common TTR approach that we denote as Ph-TTR, based on the use of photometric losses. These results validate our method as a general TTR approach easy to implement and use after all kinds of networks, current and future ones. Besides, as a more general comment, we believe that the presented contributions provide insights towards a further leverage of SfM in self-supervised depth learning, arising as a promising extension to the widely used photometry-based losses.

Acknowledgments

This work was supported by the Spanish Government (PID2021-127685NB-I00 and TED2021-131150B-I00), the Aragón Government (DGA FSE-T45 20R) and the scholarship FPU20/02342.

TTR	Method	Abs Rel $\downarrow$	Sq Rel $\downarrow$	RMSE $\downarrow$	RMSE log $\downarrow$	$\delta < 1.25$ $\uparrow$	$\delta < 1.25^2$ $\uparrow$	$\delta < 1.25^3$ $\uparrow$
✗	AdaBins [6] ∗†	0.072	0.325	3.134	0.112	0.941	0.990	**0.998**
✓	**AdaBins [6] + SfM-TTR**	**0.060**	**0.204**	**2.260**	**0.091**	**0.970**	**0.993**	**0.998**
✗	ManyDepth [52] ⋄	0.064	0.345	3.116	0.103	0.949	0.989	**0.997**
✓	ManyDepth [52] + Ph-TTR ⋄	**0.056**	0.322	3.034	0.096	0.961	**0.992**	**0.997**
✓	**ManyDepth [52] + SfM-TTR**	0.057	**0.294**	**2.648**	**0.094**	**0.963**	0.990	**0.997**
✗	CADepth [55] ∗	0.078	0.403	3.432	0.119	0.933	0.988	**0.997**
✓	CADepth [55] + Ph-TTR ∗	0.088	0.475	3.723	0.132	0.914	0.984	0.996
✓	**CADepth [55] + SfM-TTR**	**0.068**	**0.328**	**2.821**	**0.106**	**0.955**	**0.990**	0.996
✗	DIFFNet [64] ∗	0.071	0.361	3.230	0.110	0.946	0.990	0.997
✓	DIFFNet [64] + Ph-TTR ∗	0.057	0.285	2.900	0.095	0.961	**0.992**	**0.998**
✓	**DIFFNet [64] + SfM-TTR**	**0.056**	**0.273**	**2.600**	**0.093**	**0.969**	**0.992**	0.997

Table 4. **Quantitative results with new KITTI ground truth, Eigen split and no cropping.** Best results per model in **bold**, best results across all self-supervised models underlined. Experimental results are marked with ∗, results from original papers with ⋄. We compare different architectures without TTR, with Ph-TTR and with our SfM-TTR. † Results from AdaBins differ from [6], as in this table we do not crop during evaluation. For results using cropping, see Table 5.

TTR	Method	Abs Rel $\downarrow$	Sq Rel $\downarrow$	RMSE $\downarrow$	RMSE log $\downarrow$	$\delta < 1.25$ $\uparrow$	$\delta < 1.25^2$ $\uparrow$	$\delta < 1.25^3$ $\uparrow$
✗	AdaBins [6] ⋄ †	0.058	0.190	2.360	0.088	0.964	0.995	**0.999**
✓	**AdaBins [6] + SfM-TTR †**	**0.054**	**0.138**	**1.885**	**0.078**	**0.978**	**0.996**	**0.999**
✗	ManyDepth [52] ∗	0.059	0.297	2.960	0.097	0.954	0.991	**0.998**
✓	ManyDepth [52] + Ph-TTR ∗	**0.053**	**0.252**	2.774	**0.089**	0.962	**0.993**	**0.998**
✓	**ManyDepth [52] + SfM-TTR**	0.054	**0.252**	**2.510**	**0.089**	**0.966**	0.992	**0.998**
✗	CADepth [55] ∗	0.073	0.359	3.287	0.112	0.941	0.990	**0.997**
✓	CADepth [55] + Ph-TTR ∗	0.082	0.426	3.565	0.124	0.923	0.986	**0.997**
✓	**CADepth [55] + SfM-TTR**	**0.060**	**0.263**	**2.620**	**0.096**	**0.962**	**0.992**	**0.997**
✗	DIFFNet [64] ∗	0.066	0.318	3.078	0.103	0.953	0.992	**0.998**
✓	DIFFNet [64] + Ph-TTR ∗	0.053	0.252	2.778	0.090	0.965	0.993	**0.998**
✓	**DIFFNet [64] + SfM-TTR**	**0.052**	**0.229**	**2.444**	**0.085**	**0.973**	**0.994**	**0.998**

Table 5. **Quantitative results with new KITTI ground truth, Eigen split and Eigen cropping.** Best results per model in **bold**, best results across all self-supervised models underlined. Experimental results are marked with ∗, results from papers with ⋄. † Results from AdaBins + SfM-TTR follow the common KITTI Benchmark cropping from the supervised depth learning literature [6], and the AdaBins results without TTR are taken from the original paper.

TTR	Method	Abs Rel $\downarrow$	Sq Rel $\downarrow$	RMSE $\downarrow$	RMSE log $\downarrow$	$\delta < 1.25$ $\uparrow$	$\delta < 1.25^2$ $\uparrow$	$\delta < 1.25^3$ $\uparrow$
✗	AdaBins [6] ∗	**0.087**	0.480	3.637	0.168	0.917	0.970	**0.985**
✓	**AdaBins [6] + SfM-TTR**	0.088	**0.454**	**3.355**	**0.164**	**0.927**	**0.971**	**0.985**
✗	Monodepth2 (384x112) [20] ⋄	**0.128**	**1.040**	5.216	0.207	0.849	**0.951**	**0.978**
✓	Monodepth2 + TTR (from [36]) ⋄	0.130	2.086	**4.876**	**0.205**	**0.878**	0.946	0.970
✗	Monodepth2 [20] ⋄	0.115	0.903	4.863	0.193	0.877	0.9590	0.981
✓	Monodepth2 + TTR (from [49]) ⋄	0.113	**0.793**	4.655	0.188	0.874	0.960	**0.983**
✓	**Monodepth2 + SfM TTR**	**0.098**	0.858	**4.418**	**0.177**	**0.908**	**0.964**	0.981
✗	ManyDepth [52] ⋄	0.093	0.715	4.245	0.172	0.909	0.966	**0.983**
✓	ManyDepth [52] + Ph-TTR ⋄	**0.087**	**0.696**	4.183	**0.167**	**0.918**	**0.968**	**0.983**
✓	**ManyDepth [52] + SfM-TTR**	0.090	0.718	**4.040**	0.168	0.917	0.967	**0.983**
✗	CADepth [55] ⋄	0.102	0.734	4.407	0.178	0.898	**0.966**	**0.984**
✓	CADepth [55] + Ph-TTR ∗	0.110	0.802	4.648	0.187	0.878	0.962	0.983
✓	**CADepth [55] + SfM-TTR**	**0.095**	**0.703**	**4.073**	**0.173**	**0.912**	**0.966**	0.982
✗	DIFFNet [64] ⋄	0.097	0.722	4.345	0.174	0.907	0.967	**0.984**
✓	DIFFNet [64] + Ph-TTR ∗	**0.087**	0.667	4.138	0.167	0.920	0.968	**0.984**
✓	**DIFFNet [64] + SfM-TTR**	**0.087**	**0.660**	**3.948**	**0.165**	**0.925**	**0.969**	**0.984**

Table 6. **Quantitative results with Eigen (old) KITTI ground truth, Eigen split and Eigen cropping.** Best results per model in **bold**, best results across all self-supervised models underlined. Experimental results are marked with ∗, results from original papers with ⋄. Note how, with this different ground truth, we again outperform the results of the baselines in Tables 4 and 5 and we further demonstrate improvement over Monodepth2 [20] and the TTR approaches [36, 49] that were evaluated after such architecture in the original papers.

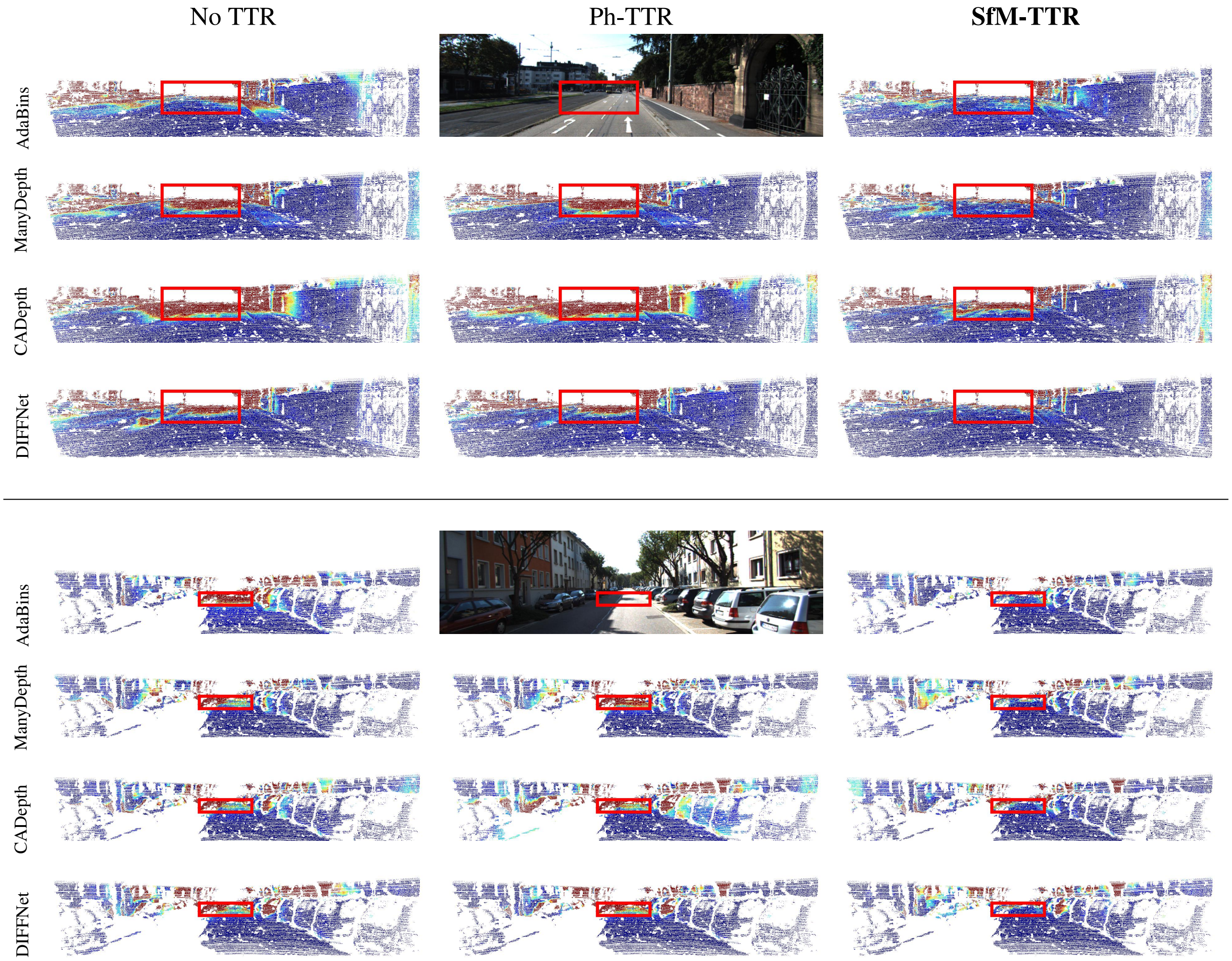

Figure 4. **RMSE maps for different baselines architectures (rows) and TTR (columns).** The input image is the center top image, as AdaBins cannot be refined with photometric loss. The benefit of our SfM-TTR is particularly noticeable for large depths (framed by red rectangles). Ph-TTR methods struggle in these areas as they use weak low-parallax constraints, while SfM leverages wider baselines and produces more accurate depth supervision. Figure best viewed in color.

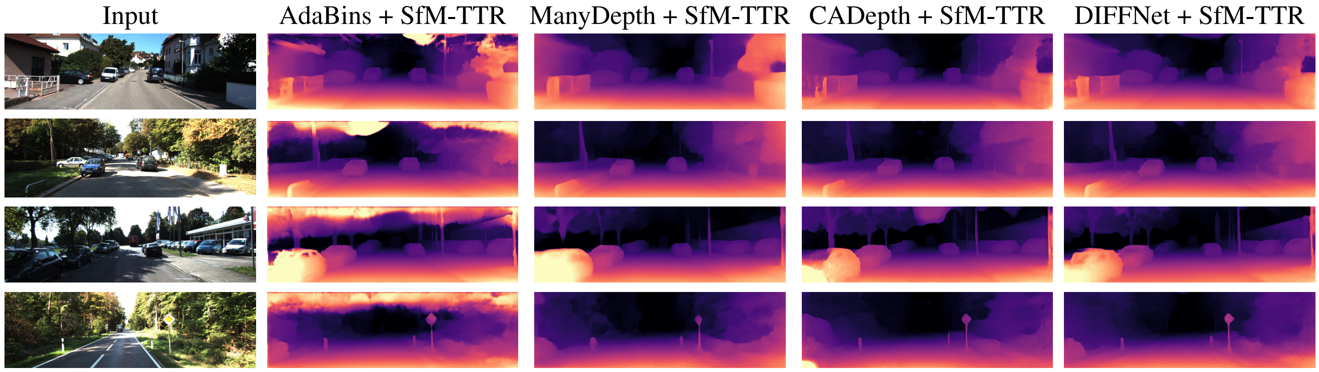

Figure 5. Qualitative depth maps for different architectures after SfM-TTR on KITTI.

References

[1] OpenSfM. https://github.com/mapillary/OpenSfM. 3

[2] Ahmed Ali, Ali Hassan, Afsheen Rafaqat Ali, Hussam Ullah Khan, Wajahat Kazmi, and Aamer Zaheer. Real-time vehicle distance estimation using single view geometry. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1111–1120, 2020. 2

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

[4] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2842–2851, 2022. 2

[5] Olga Barinova, Vadim Konushin, Anton Yakubenko, KeeChang Lee, Hwasup Lim, and Anton Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *European Conference on Computer Vision*, pages 100–113. Springer, 2008. 2

[6] 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. 1, 2, 4, 6, 7

[7] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. LocalBins: Improving Depth Estimation by Learning Local Distributions. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022. 2

[8] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016. 2

[9] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 2, 3, 6

[10] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8001–8008, 2019. 2

[11] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019. 2

[12] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 1, 2

[13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 2, 4

[14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 2

[15] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11826–11835, 2019. 1

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[17] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 2

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 4

[19] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. 2

[20] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1, 2, 5, 7

[21] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019. 1

[22] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters*, 7(2):5397–5404, 2022. 1

[23] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2, 4

[24] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 654–661. IEEE, 2005. 2

[25] 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

[26] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014. 2

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[28] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–713, 2018. 2, 6

[29] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2907–2917, 2021. 2

[30] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 1, 2

[31] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. Single-Image Depth Estimation Based on Fourier Domain Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 330–339, 2018. 2

[32] Zhengqi Li and Noah Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 2

[33] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1253–1260. IEEE, 2010. 2

[34] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural RGB-D Sensing: Depth and Uncertainty from a Video Camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10995, 2019. 2

[35] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015. 2

[36] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 1, 2, 3, 4, 5, 6, 7

[37] Robert McCraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Monocular depth estimation with self-supervised instance adaptation. *arXiv preprint arXiv:2004.05821*, 2020. 2, 4, 5

[38] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021. 2

[39] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open Multiple View Geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 3

[40] Jiyan Pan, Martial Hebert, and Takeo Kanade. Inferring 3D Layout of Building Facades From a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2918–2926, 2015. 2

[41] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3Depth: Monocular Depth Estimation with a Piecewise Planarity Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1610–1621, 2022. 2

[42] David Recasens, José Lamarca, José M Fácil, JMM Montiel, and Javier Civera. Endo-Depth-and-Motion: Reconstruction and Tracking in Endoscopic Videos using Depth Networks and Photometric Constraints. *IEEE Robotics and Automation Letters*, 6(4):7225–7232, 2021. 1

[43] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008. 2

[44] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. 2

[45] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2, 3, 4

[46] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, pages 572–588. Springer, 2020. 2

[47] Peter Sturm and Steve Maybank. A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images. In *The 10th British machine vision conference (BMVC'99)*, pages 265–274. The British Machine Vision Association (BMVA), 1999. 2

[48] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6243–6252, 2017. 2

[49] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo RGB-D for Self-Improving Monocular SLAM and Depth Prediction. In *European conference on computer vision*, pages 437–455. Springer, 2020. 2, 5, 7

[50] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. 2

[51] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity Invariant CNNs. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 4

[52] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist:

Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021. 2, 4, 5, 6, 7

[53] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 611–620, 2020. 2

[54] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3917–3925, 2018. 2

[55] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 464–473. IEEE, 2021. 2, 4, 6, 7

[56] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European conference on computer vision (ECCV)*, pages 817–833, 2018. 2

[57] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[58] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018. 2

[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*, pages 3916–3925, 2022. 1

[60] Aamer Zaheer, Maheen Rashid, Muhammad Ahmed Riaz, and Sohaib Khan. Single-view reconstruction using orthogonal line-pairs. *Computer Vision and Image Understanding*, 172:107–123, 2018. 2

[61] 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

[62] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on pattern analysis and machine intelligence*, 31(6):974–988, 2009. 2

[63] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 6

[64] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021. 2, 4, 6, 7

[65] 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

[66] Zhengming Zhou and Qiulei Dong. Self-distilled feature aggregation for self-supervised monocular depth estimation. In *European Conference on Computer Vision*, pages 709–726. Springer, 2022. 2

[67] Pengli Zhu, Siyuan Liu, Tao Jiang, Yancheng Liu, Xuzhou Zhuang, and Zhenrui Zhang. Autonomous reinforcement control of visual underwater vehicles: Real-time experiments using computer vision. *IEEE Transactions on Vehicular Technology*, 71(8):8237–8250, 2022. 1