RM-Depth: Unsupervised Learning of Recurrent Monocular Depth in Dynamic Scenes*

Tak-Wai Hui
H-1 Research
eetwhui@gmail.com

Abstract

Unsupervised methods have showed promising results on monocular depth estimation. However, the training data must be captured in scenes without moving objects. To push the envelope of accuracy, recent methods tend to increase their model parameters. In this paper, an unsupervised learning framework is proposed to jointly predict monocular depth and complete 3D motion including the motions of moving objects and camera. (1) Recurrent modulation units are used to adaptively and iteratively fuse encoder and decoder features. This improves the single-image depth inference without overspending model parameters. (2) Instead of using a single set of filters for upsampling, multiple sets of filters are devised for the residual upsampling. This facilitates the learning of edge-preserving filters and leads to the improved performance. (3) A warping-based network is used to estimate a motion field of moving objects without using semantic priors. This breaks down the requirement of scene rigidity and allows to use general videos for the unsupervised learning. The motion field is further regularized by an outlier-aware training loss. Despite the depth model just uses a single image in test time and 2.97M parameters, it achieves state-of-the-art results on the KITTI and Cityscapes benchmarks.

1. Introduction

Visual perception is an important ability for human to understand and perceive the world. As a consequence, research work on scene geometry has attracted a lot of attention over several decades. This promotes the deployment of technology to numerous applications such as autonomous vehicle, interactive robot, virtual and augmented reality, and more. The problem of scene geometry generally involves estimating depth, camera motion[1], and optical flow from an image sequence. The above computer vision tasks are often recovered together since they are coupled through geometric constraints [34, 44].

Unlike depth from triangulation, single-image depth estimation is inherently ill-posed because there are multiple possible 3D points along each light ray towards the camera center. Convolutional neural networks have demonstrated the ability to exploit the relationship between a captured image and the corresponding scene depth [9, 25]. Recently, unsupervised methods [13, 14, 34, 44, 47] have achieved appealing performance than the early supervised counterparts. Their successes primarily rely on the use of structure from motion. Given at least two images, a novel view generated from an image will be consistent with another image in the pair if depth and camera motion are correctly estimated. This strictly requires the training data to be captured in static scenes without moving objects, *i.e.* scene rigidity. To get rid of this requirement, stereo image sequences [13] and masking out dynamic objects [34, 47] are commonly adopted. Recent works tend to devise a multi-image approach [42], a large amount of model parameters [16], and semantic priors [39] for improving the depth accuracy.

In this paper, an unsupervised learning framework of recurrent monocular depth, dubbed RM-Depth, is proposed to jointly predict depth, camera motion, and motion field of moving objects without requiring static scenes. RM-Depth requires neither a large number of parameters nor semantic prior. Particularly, image pairs are used in training while only a single image is used for depth inference at test time. The contributions of this work are summarized as follows:

1. *Recurrent modulation unit* – Fusion of feature maps across encoder and decoder often appears in top-down approach [14, 35]. I propose to iteratively refine the fusion by adaptive modulating of the encoder features using the hidden state of the decoder. This in turn improves the single-image depth inference.

2. *Residual upsampling* – Conventionally, feature maps are upsampled using a single set of filters [37, 45]. I propose to use multiple sets of filters such that each

*This research work is not for commercial use unless a prior arrangement has been made with the author.

[1]The words, ego-motion, camera motion and pose, are interchangeably used throughout the paper.

set of them is specifically trained for upsampling some of the spectral components. This effectively improves upsampling along edges.

3. *Motion field of moving objects* – Besides camera motion, I propose to estimate a 3D motion field of moving objects in a coarse-to-fine framework through a warping approach. This breaks down the scene rigidity assumption and allows to use general videos for the unsupervised learning. The unsupervised learning of motion field is further improved by introducing an outlier-aware regularization loss.

With the above innovations, RM-Depth achieves state-of-the-art results on the KITTI and Cityscapes benchmarks. The depth model only requires 2.97M parameters, while it achieves 4.8 and 44 times reduction in model size comparing to Monodepth2 [14] and PackNet [16], respectively. Code and trained models are made publicly available at https://github.com/twhui/RM-Depth.

2. Related Work

2.1. Unsupervised Joint Learning of Depth and Ego-motion

Depth from a Single Image. A pioneer work from Zhou *et al*. [47] proposes an unsupervised learning framework for estimating depth and ego-motion. Based on [47], Godard *et al*. [14] introduce the per-pixel minimum reprojection loss, auto-masking of stationary pixels, and full-scale estimation loss for improving the unsupervised training. Mahjourian *et al*. [31] and Bian *et al*. [3] explore the consistencies of 3D point clouds and depth maps across consecutive frames, respectively. Wang *et al*. [40] devise to use direct visual odometry for pose estimation without requiring additional pose network. Recently, Guizilini *et al*. [16] utilize 3D convolutions for packing and unpacking feature maps. Johnston *et al*. [22] propose to estimate depth map using self-attention and disparity volume. Poggi *et al*. [33] impose depth uncertainty during unsupervised training. Unlike the prior works, RM-Depth introduces recurrent modulation units (RMU) and residual upsampling in the depth model. The proposed components lead to the improved performance while the depth model just requires a very small number of parameters (2.97M).

The previous works recover rigid flow[2] through the projection of estimated scene depth, but moving objects in the scene cannot be taken into account. To recover full flow, Yin *et al*. [44] propose to use a network cascade to estimate the residual flow accounting for moving objects. Ranjan *et al*. [34] propose a framework that facilitates the coordinated trainings of depth, ego-motion, and optical flow. Their method reasons about segmenting a scene into static and moving regions. Chen *et al*. [5] recover optical flow using a separated network and develop an online refinement scheme. Different from the prior works, RM-Depth recovers both camera and object motions. Full flow, but not rigid flow, is used for the unsupervised training of depth model. This eliminates the requirement of scene rigidity.

Depth from Multiple Images. Wang *et al*. [41] exploit the temporal correlation across consecutive frames by using convolutional long short-term memory (LSTM). Despite a 10-frame sequence is used, it just performs on par with Monodepth2 [14]. Li *et al*. [28] utilize the encoded feature resulting from a self-contained optical flow network as the input to each LSTM. However, their model requires 15 LSTM modules for a proper depth inference. Li *et al*. [27] propose a self-supervised online meta-learning that uses LSTM to aggregate spatial-temporal information in the past. Watson *et al*. [42] propose a cost volume based approach to fuse temporal information. Unlike LSTM or GRU [6], the proposed RMU uses features from a single static image as the input but not features from a time-varying image sequence.

2.2. Unsupervised Joint Learning of Depth, Ego-motion, and Object Motion

Video data is often captured in scenes involving dynamic objects. Therefore, the assumption of scene rigidity is violated. Most of the prior works rely on additional segmentation labels to assist the unsupervised learning of object motion. With semantic prior, Casser *et al*. [4] estimate the 3D motion of each dynamic object using a network similar to the one used for ego-motion. Gordon *et al*. [15] propose a network for estimating the motion field of moving objects. A pre-computed segmentation mask that pinpoints the locations of moving objects imposes regularization of the motion field. Li *et al*. [27] eliminate the use of semantic priors in [15] by introducing a sparsity loss. Gao *et al*. propose attentional CNN blocks to disentangle camera and object motion without semantic priors [10], but their experimental results are limited to the KITTI dataset. Lee *et al*. [26] propose an instance-aware photometric and geometric consistency loss that imposes self-supervisory signals for static and moving object regions. RM-Depth estimates the motion field of moving objects without using semantic priors. A warping-based network is proposed for the motion field estimation. An outlier-aware training loss is further exploited for regularizing the motion field. RM-Depth outperforms the prior works because of the proposed innovations.

2.3. Unsupervised Learning of Depth Using Stereo Training Data

The scene rigidity requirement limits unsupervised methods to use monocular data without involving dynamic

[2] A component of optical flow that is solely due to camera motion without considering moving objects in the scene.

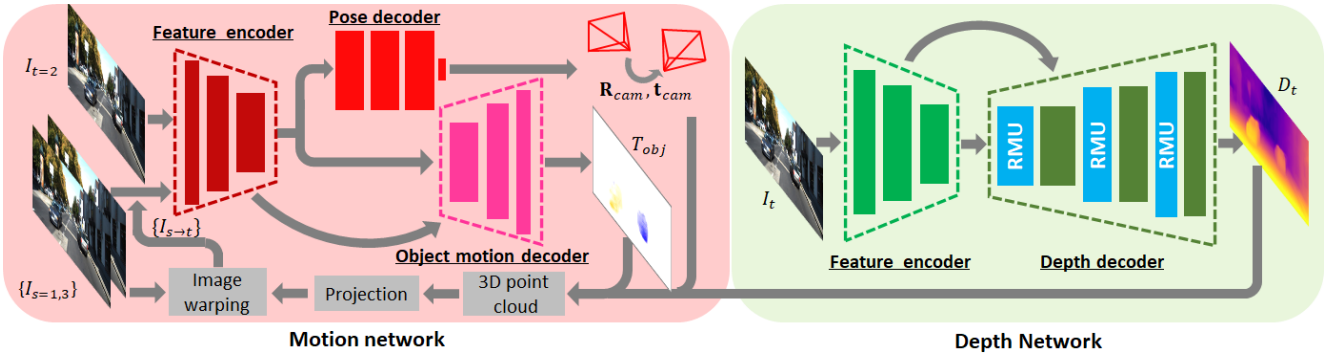


Figure 1. An overview of the unsupervised learning framework. For brevity, a 3-level design is shown. Given an image sequence $\{I_1, I_2, I_3\}$, define $I_{t=2}$ as the target image and the rest $\{I_{s=1,3}\}$ as the source images. Depth map and motion field are estimated in a coarse-to-fine framework. For the motion network, $\{I_s\}$ are warped towards I_t in accordance to the image projection computed by Eq. (2) using motion field T_{obj}, camera pose $(\mathbf{R}_{cam}, \mathbf{t}_{cam})$, and scene depth D_t. For the depth network, encoder and decoder features are adaptively and iteratively fused by RMUs. More details of the depth and motion networks are presented in Sects. 3.2 and 3.4, respectively.

objects in scenes. Since the left and right images in a stereo rig are captured simultaneously, stereo data provides an alternative option for the unsupervised training. Garg *et al*. [11] propose to use the photometric difference between images in each stereo pair for governing the learning of monocular depth estimation. Godard *et al*. [13] explore the consistency between the disparities produced relative to the left and right images. Zhan *et al*. [46] devise the temporal and spatial clues in stereo image sequences for improving the unsupervised training. Yang *et al*. [43] aligns the illumination of the training images and model the photometric uncertainties of pixels on the input images.

3. Depth from a Single Image

Unsupervised learning of single-image depth estimation is achieved by training two networks together [14, 47]. The primary (depth) network takes an image as the input and gradually predicts scene depth (up to a scale factor) with increasing spatial resolutions. The secondary (pose) network estimates camera motion for each image pair. Source frames in a given image sequence are warped towards the target frame by projecting the computed 3D point cloud to the target frame. The difference between the target and each synthesized frame is used as the driving force for the unsupervised training. In this work, an unsupervised learning framework RM-Depth is proposed for joint learning of depth, ego-motion, and object motion in general scenes. An overview of the learning framework is shown in Fig. 1. In more details, the depth network utilizes Recurrent Modulation Units (RMU) to adaptively and iteratively combine encoder and decoder features (Sec. 3.2). Residual upsampling (Sec. 3.3) is used to facilitate the learning of edge-aware filters. Furthermore, a 3D motion field of moving objects (Sec. 3.4) is recovered. As it will show in Sec. 4, the proposed innovations lead to the improved depth accuracy despite not using any segmentation labels.

3.1. Preliminaries

Perspective Projection. Denote O as the camera coordinate system associated with image I and $\Omega \subset \mathbb{R}^2$ as the image domain. Suppose $D : \Omega \to \mathbb{R}$ is the depth map. A point $\mathbf{x} \in \Omega$ on I is the image projection from a 3D point $\mathbf{p} \in \mathbb{R}^3$. Once $D(\mathbf{x})$ (*i.e*. z-coordinate of $\mathbf{p}$) is given, $\mathbf{p}$ can be recovered by back-projection as follows:

$$\mathbf{p} = D(\mathbf{x})\mathbf{K}^{-1}\begin{pmatrix}\mathbf{x} & 1\end{pmatrix}^{\intercal}, \tag{1}$$

where $\mathbf{K}$ denotes a 3×3 camera intrinsic matrix.

Novel View Synthesis. Suppose an image sequence $\{I_1, I_2, ..., I_N\}$ is given. In the following, subscripts t and s will be used to denote variables that are defined in the target and source views, respectively. Let's consider one of the frames I_t being the target view and the rest being the source views $I_s (1 \le s \le N, s \ne t)$. The transformation from O_t to O_s is governed by a 3×3 rotation matrix $\mathbf{R}$ and a 3D translation vector $\mathbf{t}$. Using Eq. (1), the image projection of $\mathbf{p}_t$ onto I_s is given by:

$$\begin{pmatrix}\mathbf{x}_s & 1\end{pmatrix}^{\intercal} \cong \mathbf{K}\Big(\mathbf{R}D_t(\mathbf{x}_t)\mathbf{K}^{-1}\begin{pmatrix}\mathbf{x}_t & 1\end{pmatrix}^{\intercal} + \mathbf{t}\Big), \tag{2}$$

where "$\cong$" denotes equality up to a positive scale factor and D_t is the depth map at the target view. I_s is warped towards I_t to form a novel view $I_{s\to t}$ in accordance with the visual displacement $\mathbf{x}_s - \mathbf{x}_t$.

3.2. Recurrent Depth Network

Top-down approach [14, 47] often adopts U-Net architecture [35] for depth inference. Fig. 2a provides an overview of the network architecture. The upsampled decoder feature x from the previous level is fused with the corresponding encoder feature $\mathcal{F}$ through a concatenation followed by a convolution layer. The feature fusion can be represented as follows:

$$h = \theta(\text{conv}([x, \mathcal{F}])), \tag{3}$$

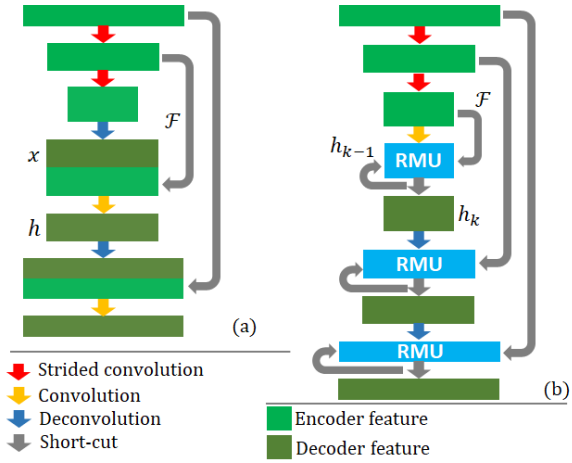


Figure 2. The network architectures of different depth models: (a) Conventional method [14, 47] and (b) RMU-based model. For the ease of representation, a 3-level design is illustrated.

where "θ" and "conv" represent an activation function and a convolution layer, respectively. Since the convolution kernels are fixed, the fusion cannot be adapted for different inputs. This limits the performance of depth inference.

It is desired to make the feature fusion to be adaptive. Intuitively, the decoder feature can be augmented with a modulated encoder feature. To this end, the encoder feature is adaptively transformed according to the current hidden state of the decoder. This is equivalent to change the feed-forward behavior of the encoder despite using the same input. Besides, recurrent CNN has been shown useful in improving network performance [23]. Taking these inspirations, Recurrent Modulation Unit (RMU) is devised for dynamic and iterative feature fusion in the depth network. Fig. 2b provides an overview of the proposed network. This design leads to the improved depth accuracy (Sec. 4). In the following, when the operations are discussed in a pyramid level, the same operations are applicable to other levels.

Recurrent Modulation Unit (RMU). There are two components inside a RMU, namely *modulation* and *update*. Fig. 3 shows the details. At iteration step k, the encoder feature $\mathcal{F}$ is adaptively modulated according to the previous fused feature h_{k-1} (*i.e.* the hidden state at iteration $k-1$) through an affine transformation[3] consisting of weight and bias terms (w_k, b_k) as follows (*modulation phase*):

$$w_k, b_k = \text{convs}([h_{k-1}, \mathcal{F}]), \tag{4a}$$

$$\mathcal{F}'_k = \tanh(\text{conv}(w_k \odot \mathcal{F} + b_k)), \tag{4b}$$

where "convs" and "$\odot$" denote convolutions and the Hadamard product, respectively. Eq. (4a) can be re-written to a residual form as $\text{conv}(\text{conv}(h_{k-1}) + \text{conv}(\mathcal{F}))$. Since $\mathcal{F}$ is fixed, the second term can be pre-computed to reduce the computational complexity. The hidden state h_{k-1} is combined with the modulated encoder feature $\mathcal{F}'_k$ for the feature fusion according to an element-wise adaptive scalar z_k as follows (*update phase*):

$$z_k = \sigma(\text{conv}([h_{k-1}, \mathcal{F}'_k])), \tag{5a}$$

$$h_k = (1 - z_k) \odot h_{k-1} + z_k \odot \mathcal{F}'_k, \tag{5b}$$

where "σ" denotes a sigmoid function. Particularly, the conventional feature fusion in Eq. (3) is static while the proposed feature fusion is both dynamic and iterative.

Comparing to GRU [6], RMU uses features from a single static image as the input but not features from a time-varying image sequence. GRU uses an extra memory state that depends on the input at the current time for the update. As a whole, GRU uses two sigmoid gates while RMU uses one sigmoid gate.

Hidden State Initialization. Instead of initializing the first hidden state h_0 with zero, $\mathcal{F}$ resulting from the top level of the encoder is converted to h_0 as follows:

$$h_0 = \tanh(\text{convs}(\mathcal{F})). \tag{6}$$

Depth Inference. Depth map D_t is inferred from the last hidden state. To prevent numerical issues during back-propagation, D_t is bounded by $[D_{min}, D_{max}]$ as follows:

$$\hat{D}_t = \sigma\big(\text{convs}(h_k)\big), \tag{7a}$$

$$D_t = D_{min}(1 - \hat{D}_t) + D_{max}\hat{D}_t. \tag{7b}$$

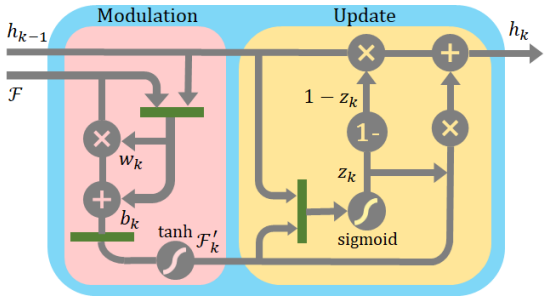


Figure 3. The details of a RMU. At iteration k, the encoder feature $\mathcal{F}$ is modulated to $\mathcal{F}'_k$. The new hidden state h_k is a weighted average between $\mathcal{F}'_k$ and the previous hidden state h_{k-1} according to the element-wise adaptive scalar z_k.

3.3. Residual Upsampling

Upsampling decoder feature is required when passing from a low-resolution to a high-resolution level in top-down approach [14, 47]. A feature map x is upsampled to x' by a upsampling function f (such as deconvolution [45] or sub-pixel convolution [37]). The process can be represented by

$$x' = \theta\big(f(x; \mathsf{W})\big), \tag{8}$$

[3] There could be other choices for the modulation function, affine transformation is selected because of its low computational complexity.

where "θ" denotes an activation function. Since a feature map like a colour image consists of different spectral components, a single filter W is not universal enough to perform well on all regions. It is desired to use different upsampling filters on different regions (flat region: averaging filter, edge region: high-pass filter). To this end, a generic upsampling layer that uses multiple filters $\{\mathsf{W}_i\}$ is proposed as follows:

$$x' = \theta\Big(\sum_i f_i(x; \mathsf{W}_i)\Big). \tag{9}$$

Particularly, each upsampling operator f_i is band-limited to some spectral components. The individual upsampled feature maps are summed before applying the activation. To compromise between accuracy and speed, RM-Depth is limited to use two kinds of upsampling operators, namely low-frequency f_l and high-frequency f_h ones, as follows:

$$x' = \theta\big(f_l(\mathrm{conv}_{1\times1}(x)) + f_h(x; \mathsf{W}_h)\big), \tag{10}$$

where a 1×1 convolution is used to squeeze x for matching the channel dimension of $f_h(\cdot)$. A bilinear upsampling is chosen as f_l. Besides the 1×1 convolution, there is no additional increase in model parameters or computational overhead in comparison to Eq. (8).

3.4. Object Motion

Unsupervised learning of depth relies on novel view synthesis as presented in Sec. 3.1. Prior works tend to jointly recover depth and camera motion but leaving out motions of moving objects [14,16]. As a result, the visual displacement that is computed by Eq. (2) is just a component of full flow (so-called *rigid flow*) inferred by the camera motion. The novel view is not correctly synthesized and in turn affects the unsupervised training. Artifacts often exist in moving objects when the object motion is not taken into consideration (see Fig. 6 in Sec. 4.2). To resolve this issue, both the camera and object motions are necessarily recovered. Since it is rare to have objects spinning on their owns with large magnitudes in street-view scenes, it can be assumed that the rotational motion of moving objects is nearly zero. An overview of the proposed motion network is shown in Fig. 4. More details are presented below.

Warping-Based Motion Field Inference. Motions of moving objects are estimated in form of a motion field $T_{obj} : \Omega \rightarrow \mathbb{R}^3$ in a coarse-to-fine framework as shown in Fig. 4. The motion field T_{obj} is combined with camera motion $\mathbf{t}_{cam}$ to form a complete motion field. Source images $\{I_s\}$ are warped towards the target image I_t in accordance with the full flow $\mathbf{u}_{full} = \mathbf{x}_s - \mathbf{x}_t$, where $\mathbf{x}_s$ is computed by Eq. (2). For the initialization, $\{I_s\}$ are warped towards I_t in accordance with the rigid flow by setting $T_{obj} = \mathbf{0}$. The warped source images $\{I_{s\rightarrow t}\}$ together with the target image I_t are fed into the motion encoder to generate a new

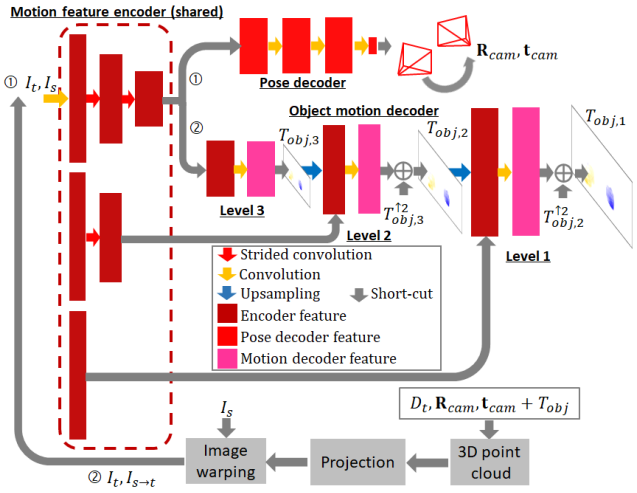


Figure 4. The architecture of the proposed motion network. The encoder is shared by the pose and object motion decoders. Object motion field T_{obj} is refined in a multi-scale framework by feedbackwarding the previous estimate to the encoder through novel view synthesis (see Sec. 3.4).

set of multi-scale encoder features $\{\mathcal{F}(I_t, I_{s\rightarrow t})\}$. The encoder features are more aligned to I_t since $\{I_s\}$ have been warped towards I_t. This in turn makes the generation of motion field easier as inspired by the feature warping proposed in LiteFlowNet series [19–21]. The object motion decoder refines the previous estimate $T_{obj,l+1}$ by augmenting with the encoder feature at the same scale as follows:

$$T_{obj,l} = \mathrm{convs}([T^{\uparrow 2}_{obj,l+1}, \mathcal{F}_l(I_t, I_{s\rightarrow t})]) + T^{\uparrow 2}_{obj,l+1}, \tag{11}$$

where "convs" represents several convolution layers and $(\cdot)^{\uparrow 2}$ denotes an upsampling operator by a factor of 2. Particularly, the encoder features are warped for the motion refinement. This is different from prior works [15,27] that use fixed encoder features.

Outlier-Aware Regularization Loss. Motion field is generally sparse since moving objects do not fully occupy a scene, *i.e.* $T_{obj}(\mathbf{x}) = \mathbf{0}$ when an image position $\mathbf{x}$ is not affected by non-rigid motion. This observation can impose a constraint on the unsupervised training and in turn improves the depth accuracy. A motion mask M is constructed by comparing full flow $\mathbf{u}_{full}$ (computed by Eq. (2) using depth, camera and object motions) against rigid flow $\mathbf{u}_{rig}$ (using only depth and camera motion). If there are no moving objects in the scene other than the moving camera itself, then $\mathbf{u}_{full} = \mathbf{u}_{rig}$. Otherwise, $\mathbf{u}_{full} \neq \mathbf{u}_{rig}$. This motivation allows us to segment image locations affected by non-rigid motions using the following condition:

$$M(\mathbf{x}) = [\|\mathbf{u}_{full} - \mathbf{u}_{rig}\|_2 < \alpha], \tag{12}$$

where $[\cdot]$ is the Iverson bracket. A thresholding approach is used to suppress outliers by setting $\alpha = 0.5$. When an image position $\mathbf{x}$ is affected by non-rigid motions, $M(\mathbf{x}) = 0$.

Otherwise, $M(\mathbf{x}) = 1$. With the motion mask, an outlier-aware regularization loss L_{reg} on the motion field is proposed as follows:

$$L_{reg}(T_{obj}) = \sum_{\mathbf{x}\in\Omega} g\big(M \cdot T_{obj}\big), \tag{13}$$

where $g(\cdot)$ is chosen to be the sparsity function [27] as it encourages more sparsity than L_1 norm. L_{reg} helps the motion network to properly learn T_{obj} by suppressing the growth of undesired object motion in rigid regions.

4. Experiments

4.1. Implementation Details

Network Architecture. The overviews of depth and motion networks can be referred to Figs. 2 and 4, respectively. A modified 6-level ResNet18 [17] that contains an additional convolution layer at level 1 and excludes the classification head is adopted as the encoders. Particularly, the top two levels are not used in the depth encoder. For ego-motion network, the decoder is adopted from [14]. Other unmentioned parts are custom-designed. 9 and 2 RMUs are assigned to level 4 and the remained levels, respectively. RMUs are not shared across different levels in order to maximize filter diversity for different scales.

Training Details. The system is implemented in TensorFlow [1]. Same augmentations are performed on the training data as [14], namely 50% horizontal flips, random brightness, contrast, saturation, and hue jitter. Following [47], the length of each image sequence is fixed to 3 frames. The central frame is treated as the target view. The depth and motion networks are jointly trained using Adam [24] with a batch size of 16 to 24. To address the stationary pixels and the occlusion problem, the auto-masking and the per-pixel minimum reprojection loss [14] are adopted. Depth map and motion field are regularized by an edge-aware smoothness loss [14] while the proposed outlier-aware regularization loss is further imposed on the object motion field. The self-supervision [38] is also adopted but no cropping is applied. All the loss weights keep the same as the suggested values. The overall network is trained for 25 epochs. A learning rate of 1e-4 for the first 15 epochs and reduce the learning rate to 1e-5 for the remained epochs. All the encoders have been pre-trained on ImageNet [36]. Other specific training details are available in the code package.

Dataset. The system is trained and validated on KITTI [12] and Cityscapes [7]. The image resolution is set to 640×192. For KITTI, the data split of Eigen *et al*. [8] is used excluding all the static and evaluation frames as Zhou *et al*. [47]. For Cityscapes, the standard training split is used and no static frames are neglected. The cropping scheme "A" defined in [42] is used for the evaluation.

4.2. Results

RM-Depth is compared against prior state-of-the-art methods such that they are also trained on monocular image sequences and perform single-image depth inference without using online refinement unless otherwise specified. Depth map is capped to 80m [13] and is normalized using median scaling [47]. Other experimental results (related to generalization on unseen dataset, visual odometry, and more) are available in the supplementary material [18].

Depth (KITTI). As shown in the upper half of Table 1, RM-Depth outperforms the compared methods. Examples of estimations are provided in Fig. 5. It can be observed that RM-Depth is superior in recovering thin structures and moving objects than GeoNet [44]. Monodepth2 [14] cannot correctly predict depth values on objects with reflective surface (the on-road train in the first example and the white car in the third example) while RM-Depth is free of such defects. PackNet [16] and RM-Depth recover depth maps with sharp discontinuities. However the moving car in the second example is not correctly estimated by PackNet.

Depth (Cityscapes). This dataset is more challenging since it involves more moving objects than KITTI. Only a few works report the evaluation results on Cityscapes. The bottom half of Table 1 summarizes the results. Despite RM-Depth does not use segmentation labels, it outperforms the prior works. Visual comparison is provided in Fig. 6. When object motion is neglected, holes (*i.e*. depth values tend to the maximum) often appear on moving objects.

Object Motion and Segmentation. The protocol as [34] is followed and the motion segmentation is evaluated on the KITTI 2015 dataset [32]. The results are summarized in Table 2. RM-Depth performs on par with Distilled Semantics [39] while RM-Depth neither uses semantic labels for training nor semantic network. Fig. 7 show examples of motion field and segmentation predictions.

Optical Flow. It is computed by Eq. (2) using depth, camera and object motions. As provided in Table 3, AEE is improved when object motion is considered. The performance is reasonable since no stand-alone optical flow network is constructed. Examples of optical flow are shown in Fig. 7.

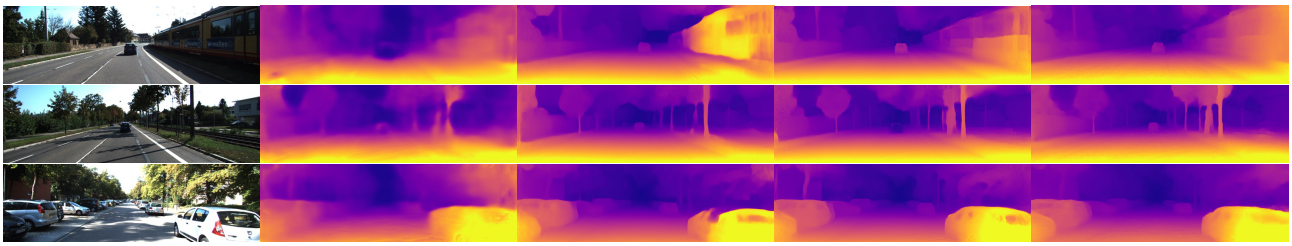
Model Size and Runtime. As shown in Fig. 8, RM-Depth just requires 2.97M parameters for the depth model while it outperforms the prior works even for those with semantics. RM-Depth runs at 40FPS for a single depth prediction on a machine equipped with a GeForce GTX 1080.

4.3. Ablation Study

The contributions of the proposed components are studied by evaluating different variants of RM-Depth. Since moving objects are limited on KITTI, the proposed components that are related to object motion are evaluated on

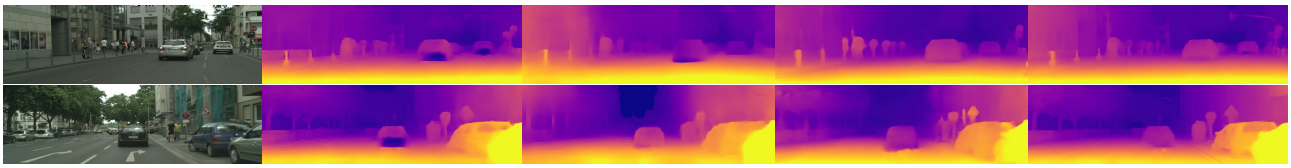
Table 1. Monocular depth results on the KITTI dataset (K) by the testing split of Eigen *et al*. [8] and the testing split of Cityscapes dataset (CS). Models that require explicit semantic data are highlighted. The best in each category is in bold and the second best is underlined.

Method	Semantics	Dataset	Error (lower is better)				Accuracy (higher is better)		
			AbsRel	SqRel	RMS	RMSlog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou *et al*. [47]		K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
GeoNet [44]		K	0.164	1.303	6.090	0.247	0.765	0.919	0.968
Mahjourian *et al*. [31]		K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
GeoNet (ResNet) [44]		K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DDVO [40]		K	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Li *et al*. [28]		K	0.150	1.127	5.564	0.229	0.823	0.936	0.974
DF-Net [48]		K	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Pilzer *et al*. [2]		K	0.142	1.231	5.785	0.239	0.795	0.924	0.968
EPC++ [29]		K	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2Depth [4]	•	K	0.141	1.026	5.291	0.215	0.816	0.945	0.979
CC [34]		K	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Bian *et al*. [3]		K	0.137	1.089	5.439	0.217	0.830	0.942	0.975
GLNet [5]		K	0.135	1.070	5.230	0.210	0.841	0.948	0.980
Li *et al*. [27]	•	K	0.130	0.950	5.138	0.209	0.843	0.948	0.978
Gordon *et al*. [15]	•	K	0.128	0.959	5.230	0.212	0.845	0.947	0.976
Distilled Semantics [39]	•	K	0.126	0.835	4.937	0.199	0.844	0.953	0.982
Monodepth2 [14]		K	0.115	0.882	4.701	0.190	0.879	0.961	0.982
PackNet [16]		K	0.111	0.785	4.601	0.189	0.878	0.960	0.982
PackNet [16] (weak velocity sup.)		K	0.111	0.829	4.788	0.199	0.864	0.954	0.980
Johnston *et al*. [22]		K	0.111	0.941	4.817	0.189	**0.885**	0.961	0.981
Monodepth2-*Boot+Self* [33]		K	0.111	0.826	4.667	0.184	0.880	0.961	**0.983**
Monodepth2-*Snap+Log* [33]		K	0.117	0.900	4.838	0.192	0.873	0.958	0.981
Lee *et al*. [26]	•	K	0.112	0.777	4.772	0.191	0.872	0.959	0.982
Gao *et al*. [10]		K	0.112	0.866	4.693	0.189	0.881	0.961	0.981
RM-Depth		K	**0.108**	**0.710**	**4.513**	**0.183**	0.884	**0.964**	**0.983**
Struct2Depth [4]	•	CS	0.145	1.737	7.280	0.205	0.813	0.942	0.976
GLNet [5] (with online refinement)		CS	0.129	1.044	5.361	0.212	0.843	0.938	0.976
Gordon *et al*. [15]	•	CS	0.127	1.330	6.960	0.195	0.830	0.947	0.981
Li *et al*. [27]		CS	0.119	1.290	6.980	0.190	0.846	0.952	0.982
Lee *et al*. [26]	•	CS	0.111	1.158	6.437	0.182	0.868	0.961	0.983
RM-Depth		CS	**0.100**	**0.839**	**5.774**	**0.154**	**0.895**	**0.976**	**0.993**



RGB image | GeoNet [44] | Monodepth2 [14] | PackNet [16] | RM-Depth

Figure 5. Examples of depth map predictions on KITTI.



RGB image | without object motion | Struct2Depth [4] | Gordon *et al*. [15] | RM-Depth

Figure 6. Examples of depth map predictions on Cityscapes.

Table 2. Motion segmentation results on the KITTI 2015 dataset.

Method	Semantics	Intersection over Union (IoU)		
		Overall	Static car	Moving car
EPC++ [29]		50.00	-	-
CC [34]	•	56.94	55.77	58.11
DS [39]	•	62.66	58.42	**66.89**
DS (semantic network) [39]	•	63.98	64.16	63.79
RM-Depth		**64.48**	**66.91**	62.04

Table 3. Average end-point error on the KITTI 2015 dataset.

Method	Flow net.	All	Fl
Distilled Semantics (ego-motion) [39]		13.50	51.22%
Distilled Semantics [39]	•	**11.61**	**25.78%**
GeoNet (DirFlowNetS) [44]	•	12.21	-
RM-Depth (ego-motion)		13.14	44.17%
RM-Depth (complete motion)		11.77	41.62%

Cityscapes. All the results are evaluated on their testing splits and are capped at 80m per standard practice.

RMU and Residual Upsampling. As shown in Table 4, the full model outperforms the baseline by a large margin. The proposed components are effective in improving the depth accuracy. By removing either the residual upsampling or

Figure 7. Examples of depth, object motion field, segmentation mask, and optical flow predictions on the KITTI 2015 dataset.

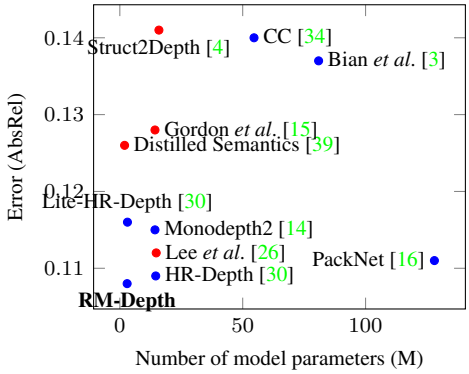


Figure 8. Error of depth models on KITTI against the number of model parameters. Red dots denote models requiring semantics.

Table 4. Ablation study of RM-Depth on KITTI.

Model	Error (lower is better)			
	AbsRel	SqRel	RMS	RMSlog
full	**0.1081**	**0.7100**	**4.5138**	**0.1831**
w/o residual upsampling	0.1097	0.7313	4.5269	0.1839
w/o RMU	0.1167	0.8186	4.7100	0.1895
w/o modulation	0.1165	0.7546	4.6623	0.1910
baseline (w/o my contributions)	0.1187	0.8382	4.7894	0.1927

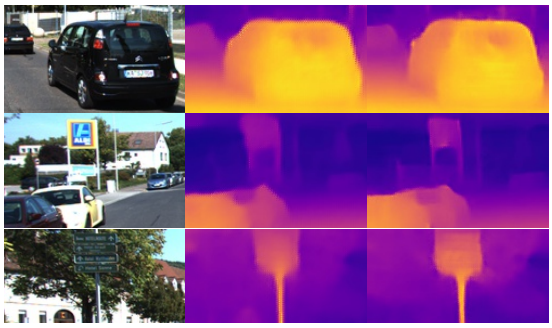


Figure 9. Depth map predictions using different upsamplings.

RMU, the depth error is increased. Thanks to the residual upsampling, depth edges are less dispersed comparing to the results using conventional upsampling [37] as demonstrated in Fig. 9. A RMU consists of modulation and update parts. When the modulation part is removed, the depth error is increased. This indicates that the depth improvement is largely benefited by the modulation since it adaptively modifies the feed-forward behavior of the encoder.

Object Motion. The full model performs the best among all the variants as summarized in Table 5. The proposed

Table 5. Ablation study of RM-Depth on Cityscapes.

Model	Error (lower is better)			
	AbsRel	SqRel	RMS	RMSlog
full	**0.1002**	**0.8387**	**5.7742**	**0.1540**
w/o warping	0.1022	0.9847	5.9272	0.1563
w/o outlier-aware regularization	0.1052	0.9291	6.1911	0.1638
with sparsity loss [27]	0.1108	1.1254	7.4494	0.1825
baseline (w/o my contributions)	0.1238	1.1508	6.5320	0.1780

Table 6. Ablation study of the number of RMUs on KITTI.

Number of RMUs	Error (lower is better)				Runtime
	AbsRel	SqRel	RMS	RMSlog	[ms]
3 (L4: 1, L3: 1, L2: 1)	0.1161	0.7713	4.6799	0.1906	14.99
6 (L4: 2, L3: 2, L2: 2)	0.1135	0.7490	4.6128	0.1877	20.40
8 (L4: 4, L3: 2, L2: 2)	0.1098	0.7251	4.5535	0.1845	22.07
13 (L4: 9, L3: 2, L2: 2)	**0.1081**	**0.7100**	**4.5138**	**0.1831**	24.78

components are effective in improving depth accuracy on non-rigid scenes. When warping is disabled, the source images are not warped towards the target image. There is a large "visual gap" between images in the pair, and hence the depth error is increased. By disabling the outlier-aware regularization, the depth accuracy is deteriorated. Comparing to the variant using the sparsity loss [27], the full model performs much better. When all of the proposed components are disabled, it has been experienced that the training becomes diverged after 6 epochs. Holes often appear on moving objects as revealed in Fig. 6.

Number of RMUs. Compromising accuracy and computational complexity, at most 2 RMUs are assigned for levels 2 – 3. As summarized in Table 6, depth accuracy and runtime increase with the number of RMUs.

5. Conclusion

RM-Depth, an unsupervised learning framework, is proposed for single-image depth estimation. Complete motion that includes camera and object motions is used to assist the unsupervised learning. This breaks down the scene rigidity requirement. The depth network utilizes recurrent modulation units for dynamic and iterative feature fusion. The use of residual upsampling enables specific upsampling of different spectral components. For the motion network, a warping-based approach has been devised to recover object motion. An outlier-aware regularization loss has also been exploited. With the proposed innovations, the depth network achieves promising results while it only requires 2.97M model parameters.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. 6

[2] S. Lathuilière Andrea Pilzer and, N. Sebe, and E. Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. *CVPR*, pages 9768–9777, 2019. 7

[3] J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *NeurIPS*, 2019. 2, 7, 8

[4] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *AAAI*, pages 8001–8008, 2019. 2, 7, 8

[5] Y. Chen, C. Schmid, and C. Sminchisescu. Self-supervised learning with geometric constraints in monocular video connecting flow, depth, and camera. *ICCV*, pages 7063–7072, 2019. 2, 7

[6] K. Cho, B. V. Merri´enboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv:1409.1259*, 2014. 2, 4

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *CVPR*, pages 3213–3223, 2016. 6

[8] D. Eigen and R. Fergu. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV*, pages 2650–2658, 2015. 6, 7

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014. 1

[10] F. Gao, J. Yu, H. Shen, Y. Wang, and H. Yang. Attentional separation-and-aggregation network for self-supervised depth-pose learning in dynamic scenes. *CoRL*, 2020. 2, 7

[11] R. Garg, V. Kumar BG, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *ECCV*, pages 740–756, 2016. 3

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013. 6

[13] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, pages 270–279, 2017. 1, 3, 6

[14] C. Godard, O. M. Aodha, M. Firman, and G. Brostow. Digging into self-supervised monocular depth estimation. *ICCV*, pages 3828–3838, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[15] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. *ICCV*, pages 8977–8986, 2019. 2, 5, 7, 8

[16] V. Guizilini, R. Ambruş, S. Pillai, A. Raventos, and A. Gaidon. 3D packing for self-supervised monocular depth estimation. *CVPR*, pages 2485–2494, 2020. 1, 2, 5, 6, 7, 8

[17] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *ECCV*, pages 630–645, 2016. 6

[18] T.-W. Hui. Supplementary material for RM-Depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. 2022. 6

[19] T.-W. Hui and C. C. Loy. LiteFlowNet3: Resolving correspondence ambiguity for more accurate optical flow estimation. 5

[20] T.-W. Hui, X. Tang, and C. C. Loy. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. *CVPR*, pages 8981–8989, 2018. 5

[21] T.-W. Hui, X. Tang, and C. C. Loy. A lightweight optical flow CNN – Revisiting data fidelity and regularization. *TPAMI*, 43(8):2555–2569, 2021. 5

[22] A. Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. *CVPR*, pages 4756–4765, 2020. 2, 7

[23] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. *CVPR*, pages 1637–1645, 2016. 4

[24] D. P. Kingma and J. Bar. Adam: A method for stochastic optimization. *arXiv*, 2014. 6

[25] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *3DV*, pages 239–248, 2016. 1

[26] S. Lee, S. Im, S. Lin, and I. S. Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. *AAAI*, pages 1863–1872, 2021. 2, 7, 8

[27] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova. Unsupervised monocular depth learning in dynamic scenes. *CoRL*, 2020. 2, 5, 6, 7, 8

[28] S. Li, F. Xue, X. Wang, Z. Yan, and H. Zha. Sequential adversarial learning for self-supervised deep visual odometry. *ICCV*, pages 2851–2860, 2019. 2, 7

[29] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding. *TPAMI*, 42:2624–2641, 2020. 7

[30] X. Lyu, L. Liu, M. Wang, X. Kong, L. Liu, Y. Liu, X. Chen, and Y. Yuan. HR-Depth: High resolution self-supervised monocular depth estimation. *AAAI*, 2021. 8

[31] R. Mahjourian, M. Wicke, and A. Angelov. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. *CVPR*, pages 5667–5675, 2018. 2, 7

[32] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. *CVPR*, pages 3061–3070, 2015. 6

[33] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. On the uncertainty of self-supervised monocular depth estimation. *CVPR*, pages 3227–3237, 2020. 2, 7

[34] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive Collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. *CVPR*, pages 12240–12249, 2019. 1, 2, 6, 7, 8

[35] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *MICCAI*, pages 234–241, 2015. 1, 3

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, A. C. Berg M. Bernstein, and F.-F. Li. ImageNet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 6

[37] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CVPR*, pages 1874–1883, 2016. 1, 4, 8

[38] A. Stone, D. Maurer, A. Ayvaci, A. Angelova, and R. Jonschkowski. SMURF: Self-teaching multi-frame unsupervised RAFT with full-image warping. *CVPR*, pages 3887–3896, 2021. 6

[39] F. Tosi, F. Aleotti, P. Z. Ramirez, M. Poggi, S. Salti, L. D. Stefano, and S. Mattoccia. Distilled semantics for comprehensive scene understanding from videos. *CVPR*, pages 4654–4665, 2020. 1, 6, 7, 8

[40] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. *CVPR*, pages 2022–2030, 2018. 2, 7

[41] Y. Wang, P. Wang, Z. Yang, C. Luo, Y. Yang, and W. Xu. UnOS: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. *CVPR*, pages 8071–8081, 2019. 2

[42] J. Watson, O. M. Aodha, V. Prisacariu, G. Brostow, and M. Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. *CVPR*, pages 1164–1174, 2021. 1, 2, 6

[43] N. Yang, L. Stumberg, R. Wang, and D. Cremers. D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. *CVPR*, pages 1281–1292, 2020. 3

[44] Z. Yin and J. Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. *CVPR*, pages 1983–1992, 2018. 1, 2, 6, 7

[45] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *ICCV*, pages 2018–2025, 2011. 1, 4

[46] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Rei. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. *CVPR*, pages 340–349, 2018. 3

[47] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *CVPR*, pages 1851–1858, 2017. 1, 2, 3, 4, 6, 7

[48] Y. Zou, Z. Luo, and J.-B. Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. *ECCV*, pages 38–55, 2018. 7