Distilled Semantics for Comprehensive Scene Understanding from Videos

Fabio Tosi* Filippo Aleotti* Pierluigi Zama Ramirez*
Matteo Poggi Samuele Salti Luigi Di Stefano Stefano Mattoccia
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
*{fabio.tosi5, filippo.aleotti2, pierluigi.zama}@unibo.it

Abstract

Whole understanding of the surroundings is paramount to autonomous systems. Recent works have shown that deep neural networks can learn geometry (depth) and motion (optical flow) from a monocular video without any explicit supervision from ground truth annotations, particularly hard to source for these two tasks. In this paper, we take an additional step toward holistic scene understanding with monocular cameras by learning depth and motion alongside with semantics, with supervision for the latter provided by a pre-trained network distilling proxy ground truth images. We address the three tasks jointly by a novel training protocol based on knowledge distillation and self-supervision and b) a compact network architecture which enables efficient scene understanding on both power hungry GPUs and low-power embedded platforms. We thoroughly assess the performance of our framework and show that it yields state-of-the-art results for monocular depth estimation, optical flow and motion segmentation.

1. Introduction

What information would an autonomous agent be keen to gather from its sensory sub-system to tackle tasks like navigation and interaction with the explored environment? It would need to be informed about the geometry of the surroundings and the type of objects therein, and likely better know which of the latter are actually moving and how they do so. What if all such cues may be provided by as simple a sensor as a single RGB camera?

Nowadays, deep learning is advancing the state-of-the-art in classical computer vision problems at such a quick pace that single-view holistic scene understanding seems to be no longer out-of-reach. Indeed, highly challenging problems such as monocular depth estimation and optical flow can nowadays be addressed successfully by deep neural networks, often through unified architectures [88, 3, 96]. Self-supervised learning techniques have yielded further major achievements [95, 58] by enabling effective training of deep networks without annotated images. In fact, labels are hard to source for depth estimation due to the need of active sensors and manual filtering, and are even more cumbersome in the case of optical flow. Concurrently, semi-supervised approaches [90, 16] proved how a few semantically labelled images can improve monocular depth estimation significantly. These works have also highlighted how, while producing per-pixel class labels is tedious yet feasible for a human annotator, manually endowing images with depth and optical flow ground-truths is prohibitive.

In this paper, we propose the first-ever framework for comprehensive scene understanding from monocular videos. As highlighted in Figure 1, our multi-stage network architecture, named ΩNet, can predict depth, semantics, optical flow, per-pixel motion probabilities and motion masks. This comes alongside with estimating the pose between adjacent frames for an uncalibrated camera, whose intrinsic parameters are also estimated. Our training methodology
leverages on self-supervision, knowledge distillation and multi-task learning. In particular, peculiar to our proposal and key to performance is distillation of proxy semantic labels gathered from a state-of-the-art pre-trained model [52] within a self-supervised and multi-task learning procedure addressing depth, optical flow and motion segmentation. Our training procedure also features a novel and effective self-distillation schedule for optical flow mostly aimed at handling occlusions and relying on tight integration of rigid flow, motion probabilities and semantics. Moreover, $\Omega$Net is lightweight, counting less than 8.5M parameters, and fast, as it can run at nearly 60 FPS and 5 FPS on an NVIDIA Titan Xp and a Jetson TX2, respectively. As vouched by thorough experiments, the main contributions of our work can be summarized as follows:

- The first real-time network for joint prediction of depth, optical flow, semantics and motion segmentation from monocular videos
- A novel training protocol relying on proxy semantics and self-distillation to effectively address the self-supervised multi-task learning problem
- State-of-the-art self-supervised monocular depth estimation, largely improving accuracy at long distances
- State-of-the-art optical flow estimation among monocular multi-task frameworks, thanks to our novel occlusion-aware and semantically guided training paradigm
- State-of-the-art motion segmentation by joint reasoning about optical-flow and semantics

2. Related Work

We review previous works relevant to our proposal.

**Monocular depth estimation.** At first, depth estimation was tackled as a supervised [24, 49] or semi-supervised task [48]. Nonetheless, self-supervision from image reconstruction is now becoming the preferred paradigm to avoid hard to source labels. Stereo pairs [25, 28] can provide such supervision and enable scale recovery, with further improvements achievable by leveraging on trinocular assumptions [64], proxy labels from SGM [76, 80] or guidance from visual odometry [2]. Monocular videos [95] are a more flexible alternative, although they do not allow for scale recovery and mandate learning camera pose alongside with depth. Recent developments of this paradigm deal with differentiable direct visual odometry [77] or ICP [57] and normal consistency [57]. Similarly to our work, [88, 96, 17, 3, 86, 56] model rigid and non-rigid components using the projected depth, relative camera transformations, and optical flow to handle independent motions, which can also be estimated independently in the 3D space [9, 83]. In [30], the authors show how to learn camera intrinsics together with depth and egomotion to enable training on any unconstrained video. In [29, 94, 6], reasoned design choices such as a minimum reprojection loss between frames, self-assembled attention modules and auto-mask strategies to handle static camera or dynamic objects proved to be very effective. Supervision from stereo and video have also been combined [91, 29], possibly improved by means of proxy supervision from stereo direct sparse odometry [84]. Uncertainty modeling for self-supervised monocular depth estimation has been studied in [63]. Finally, lightweight networks aimed at real-time performance on low-power systems have been proposed within self-supervised [62, 61] as well as supervised [81] learning paradigms.

**Semantic segmentation.** Nowadays, fully convolutional neural networks [55] are the standard approach for semantic segmentation. Within this framework, multi-scale context modules and proper architectural choices are crucial to performance. The former rely on spatial pyramid pooling [31, 93] and atrous convolutions [14, 13, 15]. As for the latter, popular backbones [47, 74, 32] have been improved by more recent designs [34, 18]. While for years the encoder-decoder architecture has been the most popular choice [70, 4], recent trends in Auto Machine Learning (AutoML) [52, 12] leverage on architectural search to achieve state-of-the-art accuracy. However, these latter have huge computational requirements. An alternative research path deals with real-time semantic segmentation networks. In this space, [60] deploys a compact and efficient network architecture, [89] proposes a two paths network to attain fast inferences while capturing high resolution details. DABNet [50] finds an effective combinations of depth-wise separable filters and atrous-convolutions to reach a good trade-off between efficiency and accuracy. [51] employs cascaded sub-stages to refine results while FCHardNet [11] leverages on a new harmonic densely connected pattern to maximize the inference performance of larger networks.

**Optical flow estimation.** The optical flow problem concerns estimation of the apparent displacement of pixels in consecutive frames, and it is useful in various applications such as, e.g., video editing [10, 43] and object tracking [82]. Initially introduced by Horn and Schunck [33], this problem has traditionally been tackled by variational approaches [8, 7, 69]. More recently, Dosovitskiy et al. [21] showed the supremacy of deep learning strategies also in this field. Then, other works improved accuracy by stacking more networks [38] or exploiting traditional pyramidal [65, 75, 35] and multi-frame fusion [67] approaches. Unfortunately, obtaining even sparse labels for optical flow is extremely challenging, which renders self-supervision from images highly desirable. For this reason, an increasing number of methods propose to use image reconstruction and spatial smoothness [41, 68, 73] as main signals to guide the training, paying particular attention to occluded regions [58, 85, 53, 54, 40, 37].

**Semantic segmentation and depth estimation.** Monocular depth estimation is tightly connected to the
semantics of the scene. We can infer the depth of a scene by a single image mostly because of context and prior semantic knowledge. Prior works explored the possibility to learn both tasks with either full supervision [78, 23, 59, 45, 92, 44, 22] or supervision concerned with semantic labels only [90, 16]. Unlike previous works, we propose a compact architecture trained by self-supervision on monocular videos and exploiting proxy semantic labels.

**Semantic segmentation and optical flow.** Joint learning of semantic segmentation and optical flow estimation has been already explored [36]. Moreover, scene segmentation [72, 5] is required to disentangle potentially moving and static objects for focused optimizations. Differently, [66] leverages on optical flow to improve semantic predictions of moving objects. Peculiarly w.r.t. previous work, our proposal features a novel self-distillation training procedure guided by semantics to improve occlusion handling.

**Scene understanding from stereo videos.** Finally, we mention recent works approaching stereo depth estimation with optical flow [1] and semantic segmentation [42] for comprehensive scene understanding. In contrast, we are the first to rely on monocular videos to this aim.

### 3. Overall Learning Framework

Our goal is to develop a real-time comprehensive scene understanding framework capable of learning strictly related tasks from monocular videos. Purposely, we propose a multi-stage approach to learn first geometry and semantics, then elicit motion information, as depicted in Figure 2.

#### 3.1. Geometry and Semantics

**Self-supervised depth and pose estimation.** We propose to solve a self-supervised single-image depth and pose estimation problem by exploiting geometrical constraints in a sequence of $N$ images, in which one of the frames is used as the target view $I_i$ and the other ones in turn as the source image $I_s$. Assuming a moving camera in a stationary scene, given a depth map $D_t$ aligned with $I_i$, the camera intrinsic parameters $K$ and the relative pose $T_{t \rightarrow s}$ between $I_t$ and $I_s$, it is possible to sample pixels from $I_s$ in order to synthesise a warped image $\tilde{I}_t$ aligned with $I_t$. The mapping between corresponding homogeneous pixels coordinates $p_t \in I_t$ and $p_s \in I_s$ is given by:

$$p_s \sim KT_tD_{ps}K^{-1}p_t$$  \hspace{1cm} (1)

Following [95], we use the sub-differentiable bilinear sampler mechanism proposed in [39] to obtain $\tilde{I}_t$. Thus, in order to learn depth, pose and camera intrinsics we train two separate CNNs to minimize the photometric reconstruction error between $\tilde{I}_t$ and $I_t$, defined as:

$$\mathcal{L}_{ap}^D = \sum_p \psi(I_t(p), \tilde{I}_t(p))$$  \hspace{1cm} (2)

where $\psi$ is a photometric error function between the two images. However, as pointed out in [29], such a formulation is prone to errors at occlusion/disocclusion regions or in static camera scenarios. To soften these issues, we follow the same principles as suggested in [29], where a minimum per-pixel reprojection loss is used to compute the photometric error, an autostark method allows for filtering-out spurious gradients when the static camera assumption is violated, and an edge-aware smoothness loss term is used as in [28]. Moreover, we use the depth normalization strategy proposed in [77]. See supplementary material for further details.

We compute the rigid flow between $I_t$ and $I_s$ as the difference between the projected and original pixel coordinates in the target image:

$$F_{t \rightarrow s}^{\text{rigid}}(p_t) = p_s - p_t$$  \hspace{1cm} (3)

**Distilling semantic knowledge.** The proposed distillation scheme is motivated by how time-consuming and cumbersome obtaining accurate pixel-wise semantic annotations is. Thus, we train our framework to estimate semantic segmentation masks $S_t$ by means of supervision from cheap proxy labels $S_s$ distilled by a semantic segmentation network, pre-trained on few annotated samples and capable to generalize well to diverse datasets. Availability of proxy semantic labels for the frames of a monocular video enables us to train a single network to predict jointly depth and semantic labels. Accordingly, the joint loss is obtained
by adding a standard cross-entropy term $\mathcal{L}_{sem}$ to the previously defined self-supervised image reconstruction loss $\mathcal{L}_{ap}^{D}$. Moreover, similarly to [90], we deploy a cross-task loss term, $\mathcal{L}_{edge}^{D}$ (see supplementary), aimed at favouring spatial coherence between depth edges and semantic boundaries. However, unlike [90], we do not exploit stereo pairs at training time.

3.2. Optical Flow and Motion Segmentation

Self-supervised optical flow. As the 3D structure of a scene includes stationary as well as non-stationary objects, to handle the latter we rely on a classical optical flow formulation. Formally, given two images $I_t$ and $I_s$, the goal is to estimate the 2D motion vectors $F_{t\rightarrow s}(p_t)$ that map each pixel in $I_t$ into its corresponding one in $I_s$. To learn such a mapping without supervision, previous approaches [58, 54, 88] employ an image reconstruction loss $\mathcal{L}_{ap}$ that minimizes the photometric differences between $I_t$ and the back-warped image $\tilde{I}_t$ obtained by sampling pixels from $I_s$ using the estimated 2D optical flow $F_{t\rightarrow s}(p_t)$. This approach performs well for non-occluded pixels but provides misleading information within occluded regions.

Pixel-wise motion probability. Non-stationary objects produce systematic errors when optimizing $\mathcal{L}_{ap}$ due to the assumption that the camera is the only moving body in an otherwise stationary scene. However, such systematic errors can be exploited to identify non-stationary objects: at pixels belonging to such objects the rigid flow $F_{t\rightarrow s}^{rigid}$ and the optical flow $F_{t\rightarrow s}$ should exhibit different directions and/or norms. Therefore, a pixel-wise probability of being to an object independently moving between frames $s$ and $t$, $P_t$, can be obtained by normalizing the differences between the two vectors. Formally, denoting with $\theta(p_t)$ the angle between the two vectors at location $p_t$, we define the per-pixel motion probabilities as:

$$P_t(p_t) = \max\{\frac{1 - \cos \theta(p_t)}{2}, 1 - \rho(p_t)\}$$

where $\cos \theta(p_t)$ can be computed as the normalized dot product between the vectors and evaluates the similarity in direction between them, while $\rho(p_t)$ is defined as

$$\rho(p_t) = \min\{\|F_{t\rightarrow s}(p_t)\|_2, \|F_{t\rightarrow s}^{rigid}(p_t)\|_2\}, \max\{\|F_{t\rightarrow s}(p_t)\|_2, \|F_{t\rightarrow s}^{rigid}(p_t)\|_2\}$$

i.e. a normalized score of the similarity between the two norms. By taking the maximum of the two normalized differences, we can detect moving objects even when either the directions or the norms of the vectors are similar. A visualization of $P_t(p_t)$ is depicted in Fig. 3(d).

Semantic-aware Self-Distillation Paradigm. Finally, we combine semantic information, estimated optical flow, rigid flow and pixel-wise motion probabilities within a final training stage to obtain a more robust self-distilled optical flow network. In other words, we train a new instance of the model to infer a self-distilled flow $SF_{t\rightarrow s}$, given the estimates $F_{t\rightarrow s}$ from a first self-supervised network and the aforementioned cues. As previously discussed and highlighted in Figure 3(c), standard self-supervised optical flow is prone to errors in occluded regions due to the lack of photometric information but can provide good estimates for the dynamic objects in the scene. On the contrary, the estimated rigid flow can properly handle occluded areas thanks to the minimum-reprojection mechanism [29]. Starting from these considerations, our key idea is to split the scene into stationary and potentially dynamics objects, and apply on them the proper supervision. Purposely, we can leverage several observations:

1. Semantic priors. Given a semantic map $S_t$ for image $I_t$, we can binarize pixels into static $M_t^s$ and potentially dynamic $M_t^d$ with $M_t^s \cap M_t^d = \emptyset$. For example, we expect that points labeled as road are static in the 3D world, while pixels belonging to the semantic class car may move. In $M_t^d$, we assign 1 for each potentially dynamic pixel, 0 otherwise, as shown in Figure 3(e).

2. Camera Motion Boundary Mask. Instead of using a backward-forward strategy [96] to detect boundaries occluded due to the ego-motion, we analytically compute a binary boundary mask $M_t^{f}$ from depth and ego-motion estimates as proposed in [57]. We assign a 0 value for out-of-camera pixels, 1 otherwise as shown in Figure 3(f).

3. Consistency Mask. Because the inconsistencies between the rigid flow and $F_{t\rightarrow s}$ are not only due to dynamic objects but also to occluded/inconsistent areas, we can leverage Equation (4) to detect such critical regions. Indeed, we define the consistency mask as:

$$M_t^{c} = P_t < \xi, \xi \in [0, 1]$$

This mask assigns 1 where the condition is satisfied, 0 otherwise (i.e. inconsistent regions) as in Figure 3(g).

Finally, we compute the final mask $M$, in Figure 3(h), as:

$$M = \min\{\max\{M_t^d, M_t^s\}, M_t^{f}\}$$

As a consequence, $M$ will effectively distinguish regions in the image for which we can not trust the supervision sourced by $F_{t\rightarrow s}$, i.e. inconsistent or occluded areas. On such regions, we can leverage our proposed self-distillation mechanism. Then, we define the final total loss for the self-distilled optical flow network as:

$$\mathcal{L} = \sum \alpha_r \phi(SF_{t\rightarrow s}, F_{t\rightarrow s}^{rigid}) \cdot (1 - M)$$

$$+ \alpha_d \phi(SF_{t\rightarrow s}, M) + \psi(I_t, \tilde{I}_t^{SF}) \cdot M$$
3.3. Motion Segmentation

At test time, from pixel-wise probability \( P_t \) computed between \( SF_{t\rightarrow s} \) and \( F_{t\rightarrow s}^{rigid} \), semantic prior \( M^d_t \) and a threshold \( \tau \), we compute a motion segmentation mask by:

\[
M^{mot}_t = M^d_t \cdot (P_t > \tau), \tau \in [0,1]
\]  

Such mask allows us to detect moving objects in the scene independently of the camera motion. A qualitative example is depicted in Figure 1(f).

4. Architecture and Training Schedule

In this section we present the networks composing \( \OmegaNet \) (highlighted in red in Figure 2), and delineate their training protocol. We set \( N = 3 \), using 3-frames sequences. The source code is available at https://github.com/CVLAB-Unibo/omeganet.

4.1. Network architectures

We highlight the key traits of each network, referring the reader to the supplementary material for exhaustive details.

**Depth and Semantic Network (DSNet).** We build a single model, since shared reasoning about the two tasks is beneficial to both [90, 16]. To achieve real-time performance, DSNet is inspired to PydNet [62], with several key modifications due to the different goals. We extract a pyramid of features down to \( \frac{1}{16} \) resolution, estimating a first depth map at the bottom. Then, it is upsampled and concatenated with higher level features in order to build a refined depth map. We repeat this procedure up to half resolution, where two estimators predict the final depth map \( D_t \) and semantic labels \( S_t \). These are bi-linearly upsampled to full resolution. Each conv layer is followed by batch normalization and ReLU, but the prediction layers, using reflection padding. DSNet counts 1.93M parameters.

**Camera Network (CamNet).** This network estimates both camera intrinsics and poses between a target \( I_t \) and some source views \( I_s (1 \leq s \leq 3, s \neq t) \). CamNet differs from previous work by extracting features from \( I_t \) and \( I_s \) independently with shared encoders. We extract a pyramid of features down to \( \frac{1}{16} \) resolution for each image and concatenate them to estimate the 3 Euler angles and the 3D translation for each \( I_s \). As in [30], we also estimate the camera intrinsics. Akin to DSNet, we use batch normalization and ReLU after each layer but for prediction layers. CamNet requires 1.77M parameters for pose estimation and 1.02K for the camera intrinsics.

**Optical Flow Network (OFNet).** To pursue real-time performance, we deploy a 3-frame PPC-Net [75] network as in [54], which counts 4.79M parameters. Thanks to our novel training protocol leveraging on semantics and self-distillation, our OFNet can outperform other multi-task frameworks [3] built on the same optical flow architecture.

4.2. Training Protocol

Similarly to [88], we employ a two stage learning process to facilitate the network optimisation process. At first, we train DSNet and CamNet simultaneously, then we train OFNet by the self-distillation paradigm described in 3.2. For both stages, we use a batch size of 4 and resize input images to \( 640 \times 192 \) for the KITTI dataset (and to \( 768 \times 384 \) for pre-training on Cityscapes), optimizing the output of the networks at the highest resolution only. We also report additional experimental results for different input resolutions where specified. We use the Adam optimizer [46] with \( \beta_1 = 0.9, \beta_2 = 0.999 \) and \( \epsilon = 10^{-8} \). As photometric loss \( \psi \), we employ the same function defined in [28]. When training our networks, we apply losses using as \( I_s \) both the previous and the next image of our 3-frame sequence. Finally, we set both \( \tau \) and \( \xi \) to be 0.5 in our experiments.

**Depth, Pose, Intrinsics and Semantic Segmentation.** In order to train DSNet and CamNet we employ sequences of 3 consecutive frames and semantic proxy labels yielded by a state-of-the-art architecture [12] trained on Cityscapes with ground-truth labels. We trained DSNet and CamNet for 300K iterations, setting the initial learning rate to \( 10^{-4} \), manually halved after 200K, 250K and 275K steps. We apply data augmentation to images as in [28]. Training takes...
follow the same pre-processing procedure as [95] to remove static images from the training split while using all the 697 images for testing. LiDAR points provided in [27] are re-projected on the left input image to obtain ground-truth labels for evaluation, up to 80 meters [25]. Since the predicted depth is defined up to a scale factor, we align the scale of our estimates by multiplying them by a scalar that matches the median of the ground-truth, as introduced in [95]. We adopt the standard performance metrics defined in [24]. Table 1 reports extensive comparison with respect to several monocular depth estimation methods. We outperform our main competitors such as [88, 96, 17, 3] that solve multi-task learning or other strategies that exploit additional information during the training/testing phase [9, 83]. Moreover, our best configuration, i.e., pre-training on CS and using 1024 \times 320 resolution, achieves better results in 5 out of 7 metrics with respect to the single-task, state-of-the-art proposal [29] (and is the second best and very close to it on the remaining 2) which, however, leverages on a larger ImageNet pre-trained model based on ResNet-18. It is also interesting to note how our proposal without pre-training obtains the best performance in 6 out of 7 measures on 640 \times 192 images (row 1 vs 15). These results validate our intuition about how the use of semantic information can guide geometric reasoning and make a compact network provide state-of-the-art performance even with respect to larger and highly specialized depth-from-mono methods.

Ablation study. Table 2 highlights how progressively adding the key innovations proposed in [30, 29, 77] contributes to strengthen \(\Omega\)Net, already comparable to other methodologies even in its baseline configuration (first row). Interestingly, a large improvement is achieved by deploying joint depth and semantic learning (rows 5 vs 7), which forces the network to simultaneously reason about geometry and content within the same shared features. By replacing DSNet within \(\Omega\)Net with a larger backbone [88] (rows 5 vs 6) we obtain worse performance, validating the design decisions behind our compact model. Finally, by pre-training on CS we achieve the best accuracy, which increases alongside with the input resolution (rows 8 to 10).

Depth Range Error Analysis. We dig into our depth evaluation to explain the effectiveness of \(\Omega\)Net with respect to much larger networks. Table 3 compares, at different depth ranges, our model with more complex ones [29, 88]. This experiment shows how \(\Omega\)Net superior performance comes from better estimation of large depths: \(\Omega\)Net outperforms both competitors when we include distances larger than 8 m in the evaluation, while it turns out less effective in the close range.

5.3. Semantic Segmentation

In Table 4, we report the performance of \(\Omega\)Net on semantic segmentation for the 19 evaluation classes of CS ac-
**Table 1.** Depth evaluation on the Eigen split [24] of KITTI [26]. We indicate additional features of each method. M: multi-task learning, A: additional information (e.g. object knowledge, semantic information), I: feature extractors pre-trained on ImageNet [20], CS: network pre-trained on Cityscapes [19].

<table>
<thead>
<tr>
<th>Method</th>
<th>M</th>
<th>A</th>
<th>I</th>
<th>CS (pre-trained on K)</th>
<th>Lower is better</th>
<th>Higher is better</th>
</tr>
</thead>
<tbody>
<tr>
<td>Godard et al. [29]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Godard et al. [29] (1024 × 320)</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.115</td>
<td>0.879</td>
</tr>
<tr>
<td>Zhou et al. [94]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.121</td>
<td>0.853</td>
</tr>
<tr>
<td>Mahjourian et al. [57]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.159</td>
<td>0.784</td>
</tr>
<tr>
<td>Wang et al. [77]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.151</td>
<td>0.810</td>
</tr>
<tr>
<td>Bian et al. [6]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.128</td>
<td>0.846</td>
</tr>
<tr>
<td>Yin et al. [58]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.153</td>
<td>0.802</td>
</tr>
<tr>
<td>Zou et al. [96]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.146</td>
<td>0.818</td>
</tr>
<tr>
<td>Chen et al. [17]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.135</td>
<td>0.841</td>
</tr>
<tr>
<td>Luo et al. [56]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.141</td>
<td>0.816</td>
</tr>
<tr>
<td>Ranjan et al. [3]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.139</td>
<td>0.827</td>
</tr>
<tr>
<td>Xu et al. [85]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.138</td>
<td>0.823</td>
</tr>
<tr>
<td>Casser et al. [9]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.141</td>
<td>0.816</td>
</tr>
<tr>
<td>Gordon et al. [30]</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>0.128</td>
<td>0.844</td>
</tr>
</tbody>
</table>

**Table 2.** Ablation study of our depth network on the Eigen split [24] of KITTI. †: our network is replaced by a ResNet50 backbone [88].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>640 × 192</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.126</td>
<td>0.835</td>
</tr>
<tr>
<td>640 × 192</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.126</td>
<td>0.836</td>
</tr>
<tr>
<td>640 × 192</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>0.120</td>
<td>0.792</td>
</tr>
<tr>
<td>640 × 192</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>0.118</td>
<td>0.748</td>
</tr>
</tbody>
</table>

**Table 3.** Depth errors by varying the range. †: our network is replaced by a ResNet50 backbone [88].

<table>
<thead>
<tr>
<th>Method</th>
<th>Cap (m)</th>
<th>Lower is better</th>
<th>Higher is better</th>
</tr>
</thead>
<tbody>
<tr>
<td>Godard et al. [29]</td>
<td>0-8</td>
<td>0.059</td>
<td>0.082</td>
</tr>
<tr>
<td>ΩNet</td>
<td>0-8</td>
<td>0.060</td>
<td>0.082</td>
</tr>
<tr>
<td>Godard et al. [29]</td>
<td>0.50</td>
<td>0.125</td>
<td>0.198</td>
</tr>
<tr>
<td>ΩNet</td>
<td>0.50</td>
<td>0.127</td>
<td>0.199</td>
</tr>
<tr>
<td>Godard et al. [29]</td>
<td>0.50</td>
<td>0.124</td>
<td>0.195</td>
</tr>
<tr>
<td>ΩNet</td>
<td>0.50</td>
<td>0.120</td>
<td>0.199</td>
</tr>
<tr>
<td>Godard et al. [29]</td>
<td>0-8</td>
<td>0.132</td>
<td>0.210</td>
</tr>
<tr>
<td>ΩNet</td>
<td>0-8</td>
<td>0.134</td>
<td>0.213</td>
</tr>
<tr>
<td>Godard et al. [29]</td>
<td>0.50</td>
<td>0.126</td>
<td>0.199</td>
</tr>
<tr>
<td>ΩNet</td>
<td>0.50</td>
<td>0.126</td>
<td>0.197</td>
</tr>
</tbody>
</table>

**Table 4.** Semantic segmentation on Cityscapes (CS) and KITTI (K). S: training on ground-truth, P: training on proxy labels.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test</th>
<th>Train</th>
<th>mIoU Class</th>
<th>mIoU Cat.</th>
<th>PxAcc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DABNet [50]</td>
<td>CS(S)</td>
<td>CS(S)</td>
<td>69.64</td>
<td>87.36</td>
<td>94.62</td>
</tr>
<tr>
<td>ΩNet</td>
<td>CS(P)</td>
<td>CS(P)</td>
<td>54.80</td>
<td>82.92</td>
<td>92.50</td>
</tr>
<tr>
<td>DABNet [50]</td>
<td>CS(K)</td>
<td>K</td>
<td>35.40</td>
<td>61.49</td>
<td>80.30</td>
</tr>
<tr>
<td>FCHardNet [11]</td>
<td>CS(K)</td>
<td>K</td>
<td>44.74</td>
<td>68.20</td>
<td>72.07</td>
</tr>
<tr>
<td>ΩNet</td>
<td>CS(P)</td>
<td>K</td>
<td>43.80</td>
<td>74.31</td>
<td>88.31</td>
</tr>
<tr>
<td>ΩNet</td>
<td>CS(P) + ΩNet</td>
<td>K</td>
<td>46.88</td>
<td>75.84</td>
<td>88.12</td>
</tr>
</tbody>
</table>

**Table 5.** Optical Flow

In Table 5, we compare the performance of our optical flow network with competing methods using the KITTI 2015 stereo/flow training set [26] as testing set, which con-
Table 5. Optical flow evaluation on the KITTI 2015 dataset. †: pre-trained on ImageNet, SYN: pre-trained on SYNTHIA [71], *: trained on stereo pairs, **: using stereo at testing time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Noc</th>
<th>All</th>
<th>F1</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meisterer al. [58] - C</td>
<td>SYN + K</td>
<td>8.30</td>
<td>28.94%</td>
<td>29.46%</td>
<td></td>
</tr>
<tr>
<td>Meisterer al. [58] - CSS</td>
<td>SYN + K</td>
<td>8.10</td>
<td>23.27%</td>
<td>23.30%</td>
<td></td>
</tr>
<tr>
<td>Zou al. [96]</td>
<td>SYN + K</td>
<td>8.98</td>
<td>26.0%</td>
<td>25.7%</td>
<td></td>
</tr>
<tr>
<td>Ranjan et al. [3]</td>
<td>SYN + K</td>
<td>5.66</td>
<td>20.93%</td>
<td>25.27%</td>
<td></td>
</tr>
<tr>
<td>Wang et al. [17] **</td>
<td>K</td>
<td>5.38</td>
<td>-</td>
<td>19.03%</td>
<td></td>
</tr>
<tr>
<td>Yin et al. [88]</td>
<td>K</td>
<td>8.05</td>
<td>10.81%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Chen et al. [17] †</td>
<td>K</td>
<td>5.40</td>
<td>8.95%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Chen et al. [17] (online) †</td>
<td>K</td>
<td>4.86</td>
<td>8.35%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Ranjan et al. [3]</td>
<td>K</td>
<td>6.21</td>
<td>26.41%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Luo et al. [56]</td>
<td>K</td>
<td>5.84</td>
<td>-</td>
<td>21.56%</td>
<td></td>
</tr>
<tr>
<td>Luo et al. [56] *</td>
<td>K</td>
<td>5.43</td>
<td>-</td>
<td>20.61%</td>
<td></td>
</tr>
<tr>
<td>tNet (Ego-motion)</td>
<td>K</td>
<td>11.72</td>
<td>13.50</td>
<td>51.22%</td>
<td></td>
</tr>
<tr>
<td>OFNet</td>
<td>K</td>
<td>3.48</td>
<td>11.61</td>
<td>25.78%</td>
<td></td>
</tr>
<tr>
<td>SD-OFNet</td>
<td>K</td>
<td>3.29</td>
<td>5.39</td>
<td>20.0%</td>
<td>19.47%</td>
</tr>
</tbody>
</table>

Table 6. Motion segmentation evaluation on the KITTI 2015 dataset. *: trained on stereo pairs, **: using stereo at testing time.

<table>
<thead>
<tr>
<th>Device</th>
<th>Watt</th>
<th>D</th>
<th>DS</th>
<th>OF</th>
<th>Cam</th>
<th>Ω</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jetson TX2</td>
<td>15</td>
<td>12.5</td>
<td>10.3</td>
<td>6.5</td>
<td>49.2</td>
<td>4.5</td>
</tr>
<tr>
<td>i7-7700K</td>
<td>91</td>
<td>5.0</td>
<td>4.2</td>
<td>4.9</td>
<td>31.4</td>
<td>2.4</td>
</tr>
<tr>
<td>Titan XP</td>
<td>250</td>
<td>170.2</td>
<td>134.1</td>
<td>94.1</td>
<td>446.7</td>
<td>57.4</td>
</tr>
</tbody>
</table>

Table 7. Runtime analysis on different devices. We report the power consumption in Watt and the FPS. D: Depth, S: Semantic, OF: Optical Flow, Cam: camera pose, Ω: Overall architecture.

6. Conclusions

In this paper, we have proposed the first real-time network for comprehensive scene understanding from monocular videos. Our framework reasons jointly about geometry, motion and semantics in order to estimate accurately depth, optical flow, semantic segmentation and motion masks at about 60 FPS on high-end GPU and 5FPS on embedded systems. To address the above multi-task problem we have proposed a novel learning procedure based on distillation of proxy semantic labels and semantic-aware self-distillation of optical-flow information. Thanks to this original paradigm, we have demonstrated state-of-the-art performance on standard benchmark datasets for depth and optical flow estimation as well as for motion segmentation.

As for future research, we find it intriguing to investigate on whether and how it would be possible to self-adapt ?Net on-line. Although some very recent works have explored this topic for depth-from-mono [9] and optical flow [17], the key issue with our framework would be to conceive a strategy to deal with semantics.

Acknowledgement. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References


GHz CPU. Timings averaged over 200 frames at 640 x 192 resolution. Moreover, as each component of ?Net may be used on its own, we report the runtime for each independent task. As summarized in Table 7, our network runs in real-time on the Titan Xp GPU and at about 2.5 FPS on a standard CPU. It also fits the low-power NVIDIA Jetson TX2, achieving 4.5 FPS to compute all the outputs. Additional experiments are available in the supplementary material.


4663

[56] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *PAMI*, 2019. 2, 7, 8


[71] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 8


