SENSE: a Shared Encoder Network for Scene-flow Estimation

Huaizu Jiang† Deqing Sun∗ Varun Jampani∗
Zhaozang Lv∥ Erik Learned-Miller Jan Kautz
1UMass Amherst 2NVIDIA 3Georgia Tech

Abstract

We introduce a compact network for holistic scene flow estimation, called SENSE, which shares common encoder features among four closely-related tasks: optical flow estimation, disparity estimation from stereo, occlusion estimation, and semantic segmentation. Our key insight is that sharing features makes the network more compact, induces better feature representations, and can better exploit interactions among these tasks to handle partially labeled data. With a shared encoder, we can flexibly add decoders for different tasks during training. This modular design leads to a compact and efficient model at inference time. Exploiting the interactions among these tasks allows us to introduce distillation and self-supervised losses in addition to supervised losses, which can better handle partially labeled real-world data. SENSE achieves state-of-the-art results on several optical flow benchmarks and runs as fast as networks specifically designed for optical flow. It also compares favorably against the state of the art on stereo and scene flow, while consuming much less memory.

1. Introduction

Scene flow estimation aims at recovering the 3D structure (disparity) and motion of a scene from image sequences captured by two or more cameras [52]. It generalizes the classical problems of optical flow estimation for monocular image sequences and disparity prediction for stereo image pairs. There has been steady and impressive progress on scene flow estimation, as evidenced by results on the KITTI benchmark [39]. State-of-the-art scene flow methods outperform the best disparity (stereo) and optical flow methods by a significant margin, demonstrating the benefit of additional information in the stereo video sequences. However, the top-performing scene flow methods [5, 54] are based on the energy minimization framework [18] and are thus computationally expensive for real-time applications, such as 3D motion capture [11] and autonomous driving [27].

Recently, a flurry of convolutional neural network (CNN)-based methods have been developed for the sub-problems of stereo and optical flow. These methods achieve state-of-the-art performance and run in real-time. However, while stereo and flow are closely-related, the top-performing networks for stereo and flow adopt significantly different architectures. Further, existing networks for scene flow stack sub-networks for stereo and optical flow together [37, 25], which does not fully exploit the structure of the two tightly-coupled problems.

As both stereo and flow rely on pixel features to establish correspondences, will the same features work for these two or more related tasks? To answer this question, we take a modular approach and build a Shared Encoder Network for Scene-flow Estimation (SENSE). Specifically, we share a feature encoder among four closely-related tasks: optical flow, stereo, occlusion, and semantic segmentation. Sharing features makes the network compact and also leads to better feature representation via multi-task learning.

The interactions among closely-related tasks further con-
strain the network training, ameliorating the issue of sparse ground-truth annotations for scene flow estimation. Unlike many other vision tasks, it is inherently difficult to collect ground-truth optical flow and stereo for real-world data. Training data-hungry deep CNNs often relies on synthetic data [7, 10, 37], which lacks the fine details and diversity ubiquitous in the real world. To narrow the domain gap, fine-tuning on real-world data is necessary, but the scarcity of annotated real-world data has been a serious bottleneck for learning CNN models for scene flow.

To address the data scarcity issue, we introduce a semi-supervised loss for SENSE by adding distillation and self-supervised loss terms to the supervised losses. First, no existing dataset provides ground truth annotations for all the four tasks we address. For example, the KITTI benchmark has no ground truth annotations for occlusion and semantic segmentation. Thus, we train separate models for tasks with missing ground truth annotations using other annotated data, and use the pre-trained models to “supervise” our network on the real data via a distillation loss [17]. Second, we use self-supervision loss terms that encourage corresponding visible pixels to have similar pixel values and semantic classes, according to either optical flow or stereo. The self-supervision loss terms tightly couple the four tasks together and are critical for improvement in regions without ground truth, such as sky regions.

Experiments on both synthetic and real-world benchmark datasets demonstrate that SENSE achieves state-of-the-art results for optical flow, while maintaining the same run-time efficiency as specialized networks for flow. It also compares favorably against state of the art on disparity and scene flow estimation, while having a much smaller memory footprint. Ablation studies confirm the utility of our design choices, and show that our proposed distillation and self-supervised loss terms help mitigate issues with partially labeled data.

To summarize, we make the following contributions:

- We introduce a modular network design for holistic scene understanding, called SENSE, to integrate optical flow, stereo, occlusion, and semantic segmentation.
- SENSE shares an encoder among these four tasks, which makes networks compact and also induces better feature representation via multi-task learning.
- SENSE can better handle partially labeled data by exploiting interactions among tasks in a semi-supervised approach; it leads to qualitatively better results in regions without ground-truth annotations.
- SENSE achieves state-of-the-art flow results while running as fast as specialized flow networks. It compares favorably against state of the art on stereo and scene flow, while consuming much less memory.

1 Segmentation is only available for left images of KITTI 2015 [1].

2. Related Work

A comprehensive survey of holistic scene understanding is beyond our scope and we review the most relevant work. Energy minimization for scene flow estimation. Scene flow was first introduced by Vedula et al. [52] as the dense 3D motion of all points in an observed scene from several calibrated cameras. Several classical methods adopt energy minimization approaches, such as joint recovery of flow and stereo [20] and decoupled inference of stereo and flow for efficiency [56]. Compared with optical flow and stereo, the solution space of scene flow is of higher dimension and thus more challenging. Vogel et al. [53] reduce the solution space by assuming a scene flow of piecewise rigid moving planes over superpixels. Their work first tackles scene flow from a holistic perspective and outperforms contemporary stereo and optical flow methods by a large margin on the KITTI benchmark [12].

Joint scene understanding. Motion and segmentation are chicken-and-egg problems: knowing one simplifies the other. While the layered approach has long been regarded as an elegant solution to these two problems [55], existing solutions tend to get stuck in local minima [47]. In the motion segmentation literature, most methods start from an estimate of optical flow as input, and segment the scene by jointly estimating (either implicitly or explicitly) camera motion, object motion, and scene appearance, e.g. [6, 51]. Lv et al. [35] show that motion can be segmented directly from two images, without first calculating optical flow. Taylor et al. [50] demonstrate that occlusion can also be a useful cue.

Exploiting advances in semantic segmentation, Sevilla et al. [46] show that semantic information is good enough to initialize the layered segmentation and thereby improves optical flow. Bai et al. [2] use instance-level segmentation to deal with a small number of traffic participants. Hur and Roth [22] jointly estimate optical flow and temporally consistent semantic segmentation and obtain gains on both tasks. The object scene flow algorithm [39] segments a scene into independently moving regions and enforces superpixels within each region to have similar 3D motion. The “objects” in their model are assumed to be planar and initialized via bottom-up motion estimation. Behl et al. [5], Ren et al. [42], and Ma et al. [36] all show that instance segmentation helps scene flow estimation in the autonomous setting. While assuming a rigid motion for each individual instance works well for cars, this assumption tends to fail in general scenes, such as Sintel, on which our holistic approach achieves state-of-the-art performance.

The top-performing energy-based approaches are too computationally expensive for real-time applications. Here we present a compact CNN model to holistically reason about geometry (disparity), motion (flow), and semantics, which runs much faster than energy-based approaches.
End-to-end learning of optical flow and disparity. Recently, CNN based methods have made significant progress on optical flow and disparity, two sub-problems of scene flow estimation. Dosovitskiy et al. [10] first introduce two CNN models, FlowNetS and FlowNetC, for optical flow and bring about a paradigm shift to optical flow and disparity estimation. Ilg et al. [24] propose several technical improvements, such as dataset scheduling and stacking basic models into a big one, i.e., FlowNet2. FlowNet2 has near real-time performance and obtains competitive results against hand-designed methods. Ilg et al. [25] stack networks for flow, disparity together for the joint task of scene flow estimation. However, there is no information sharing between the networks for flow and disparity. Ranjan and Black [41] introduce a spatial pyramid network that performs on par with FlowNetC but has more than 100 times fewer parameters, due to the use of two classical principles: pyramids and warping. Sun et al. [48] develop a compact yet effective network, called PWC-Net, which makes frequent use of three principles to construct the network: pyramids of learnable features, warping operations, and cost volume processing. PWC-Net obtains state-of-the-art performance on two major optical flow benchmarks.

The FlowNet work also inspired new CNN models for stereo estimation [30, 8, 60]. Kendall et al. [30] concatenate features to construct the cost volume, followed by 3D convolutions. The 3D convolution becomes commonly used for stereo but is computationally expensive in speed and memory. Chang and Chen [8] introduce a pyramid pooling module to exploit context information for establishing correspondences in ambiguous regions. Yang et al. [60] incorporate semantic cues to tackle textureless regions. Yin et al. cast optical flow and disparity estimations as probabilistic distribution matching problems [61] to provide uncertainty estimation. They do not exploit the shared encoder of the two tasks as we do.

Existing scene flow networks [25, 36, 38] stack independent networks for disparity and flow together. We are interested in exploiting the interactions among multiple related tasks to design a compact and effective network for holistic scene understanding. Our holistic scene flow network performs favorably against state of the art while being faster for inference and consuming less memory. In particular, we show the benefit of sharing the feature encoder between different tasks, such as flow and disparity.

**Self-supervised learning from videos.** Supervised learning often uses synthetic data, as it is hard to obtain ground truth optical flow and disparity for real-world videos. Recently self-supervised learning methods have been proposed to learn scene flow by minimizing the data matching cost [65] or interpolation errors [29, 32]. However, the self-supervised methods have not yet achieved the performance of their supervised counterparts.

### 3. Semi-Supervised Scene Flow Estimation

We follow the problem setup of the KITTI scene flow benchmark [39], as illustrated in Fig. 2. The inputs are two stereo image pairs over time \((I_{1}, l, I_{1}, r, I_{2}, l, I_{2}, r)\), where the first number in the superscript indicates the time step and the second symbol denotes the left or right camera. To save space, we will omit the superscript if the context is clear. We want to estimate optical flow \(F_{1, l}\) from the first left image to the second left image and disparity \(D_{1, l}\) and \(D_{2, l}\) from the left image to the right image at the first and second frames, respectively. We also consider occlusion between two consecutive frames \(O_{E, l}^{1}\) and between the two sets of stereo images \(O_{D, l}^{1}\) and \(O_{D, l}^{2}\), as well as semantic segmentation for the reference (first left) image, i.e., \(S_{1, l}\). These extra outputs introduce interactions between different tasks to impose more constraints in the network training. Further, we hypothesize that sharing features among these closely-related tasks induces better feature representations.

We will first introduce our modular network design in Section 3.1, which shares an encoder among different tasks and supports flexible configurations during training. We will then explain our semi-supervised loss function in Section 3.2, which enables learning with partially labeled data.

#### 3.1. Modular Network Design

To enable feature sharing among different tasks and allow flexible configurations during training, we design the network in a modular way. Specifically, we build our network on top of PWC-Net [48], a compact network for optical flow estimation. PWC-Net consists of an encoder and a decoder, where the encoder takes the input images and extracts features at different hierarchies of the network. The decoder is specially designed with domain knowledge of optical flow. The encoder-decoder structure allows us to design a network in a modular way, with a single shared encoder and several decoders for different tasks.

**Shared encoder.** The original encoder of PWC-Net, however, is not well-suited to multiple tasks because of its small capacity. More than 80% of the parameters of PWC-Net are concentrated in the decoder, which uses DenseNet [19] blocks at each pyramid level. The encoder consists of plain convolutional layers and uses fewer than 20% of the parameters. While sufficient for optical flow, the encoder does not work well enough for disparity estimation. To make the encoder versatile for different tasks, we make the following modifications. First, we reduce the number of feature pyramid levels from 5 to 3, which reduces the number of parameters by nearly 50%. It also allows us to borrow the widely-used 5-level ResNet-like encoder architecture [8, 16], which has been proven to be effective in a variety of vision tasks. Specifically, we replace plain CNN layers with residual blocks [16] and add Batch Normaliza-
tion layers [26] in both encoder and decoder. With these modifications, the new model has slightly fewer parameters but gives better disparity estimation results (Table ??) and also better flow (Table 1).

**Decoder for disparity.** Next we explain how to adapt PWC-Net to disparity estimation between two stereo images. Disparity is a special case of optical flow computation, with correspondences lying on a horizontal line. As a result, we need only to build a 1D cost volume for disparity, while the decoder of the original PWC-Net constructs a 2D cost volume for optical flow. Specifically, for optical flow, a feature at \( p = (x, y) \) in the first feature map is compared to features at \( q \in [x-k, x+k] \times [y-k, y+k] \) in the warped second feature map. For disparity, we need only to search for correspondences by comparing \( p \) in the left feature map to \( q \in [x-k, x+k] \times y \) in the warped right feature map. We use \( k = 4 \) for both optical flow and disparity estimations. Across the feature pyramids, our decoder for disparity adopts the same warping and refinement process as PWC-Net.

To further improve disparity estimation accuracy, we investigate more design choices. First, we use the Pyramid Pooling Module (PPM) [64] to aggregate the learned features of input images across multiple levels. Second, the decoder outputs a disparity map one fourth the size of the input resolution, which tends to have blurred disparity boundaries. As a remedy, we add a simple hourglass module widely used in disparity estimation [8]. It takes a twice upsampled disparity, a feature map of the first image, and a warped feature map of the second image to predict a residual disparity that is added to the upsampled disparity. Both the PPM and hourglass modifications lead to significant improvements in disparity estimation. They are not helpful for optical flow estimation though, indicating that the original PWC-Net is well designed for optical flow. The modular design allows us to flexibly configure networks that work for different tasks, as shown in Fig. 2. Table ?? summarizes the effects of our design choices on disparity estimation.

**Decoder for segmentation.** To introduce more constraints to network training, we also consider semantic segmentation. It encourages the encoder to learn some semantic information, which may help optical flow and disparity estimations. For semantic segmentation decoder, we use the UPerNet [58] for its simplicity.

**Occlusion estimation.** For occlusion predictions, we add sibling branches to optical flow or disparity decoders to perform pixel-wise binary classification, where 1 means fully occluded. Adding such extra modules enables holistic scene understanding that helps us to induce better feature representations in the shared encoder and use extra supervision signals for network training to deal with partially labeled data, which is discussed in Section 3.2. Critically, for scene flow estimation, the shared encoder results in a more compact and efficient model. For optical flow and disparity estimations, we can combine modules as needed during training, with no influence on the inference time. For scene flow estimation, extra modules can be used optionally, depending on configuration. See explanations in Section 4.2.

### 3.2. Semi-Supervised Loss

No fully labeled datasets are available to directly train our holistic scene flow network. For example, KITTI has no ground-truth occlusion masks. Even for optical flow and disparity ground-truths, only around 19% of pixels of the KITTI data have annotations due to the difficulty in...
data capturing. The synthetic SceneFlow dataset [38] has no ground truth for semantic segmentation. To address these issues, we introduce our semi-supervised loss functions, which consist of supervised, distillation, and self-supervised loss terms.

**Supervised loss.** When corresponding ground-truth annotations are available, we define our supervised loss as

\[
L_{sp} = (L_F + L_{OF}) + (L_D + L_{OD}),
\]

where \(L_F\) and \(L_{OF}\) are loss terms for estimating optical flow and its corresponding occlusion. \(L_D\) and \(L_{OD}\) are the loss terms for estimating disparity and its corresponding occlusion. \(L_F\) is defined across multiple pyramid levels as

\[
L_F = \sum_{i=1}^{N_F} \omega_i \sum_p \rho \left( F_i(p), \hat{F}_i(p) \right),
\]

where \(\omega_i\) denotes optical flow and disparity weights at pyramid level \(i\), \(N_F\) is the number of pyramid levels, and \(\rho(\cdot, \cdot)\) is a loss function measuring the similarity between the ground-truth \(F_i(p)\) and estimated optical flow \(\hat{F}_i(p)\) at pixel \(p\). Disparity and occlusion loss functions, \(L_D, L_{OF}, L_D,\) and \(L_{OD}\) are defined in a similar way. We use \(L_2\) and smooth\_ll [13, 8] loss for optical flow and disparity estimations, respectively. For the occlusions, we use binary cross entropy loss when ground-truth annotations are available (e.g., on FlyingThings3D [37]). For semantic segmentation, only ground-truth annotations of the left images are available for KITTI2015. We empirically found using distillation loss only introduced below yields better accuracy.

**Distillation loss.** For occlusion estimation and semantic segmentation tasks, ground-truth annotations are not always available. They are important, however, during network training. For instance, on KITTI, supervised loss can only be computed on sparsely annotated pixels. Adding extra supervision for occlusion estimation is helpful for the network to extrapolate optical flow and disparity estimations to regions where ground-truth annotations are missing, yielding visually appealing results.

We find the occlusion estimations provided by a pre-trained model on synthetic data are reasonably good, as shown in Fig. 3. As a soft supervision, we encourage the occlusion estimations of the network during training not to deviate much from what it learned in the pre-training stage. Therefore, we simply use the estimations of a pre-trained network as pseudo ground-truth and smooth\_ll loss function during training, computed in multiple pyramid levels as \(L_F\) and \(L_D\). Adding extra supervision using distillation loss for occlusion is helpful for reducing artifacts in disparity estimation, as shown in Fig. 3.

For semantic segmentation, we use the distillation loss formulation proposed in [17]. Specifically, semantic segmentation distillation loss \(L_{Sd}\) for a single pixel \(p\) (omitted here for simplicity) is defined as

\[
L_{Sd} = T \sum_{i=1}^{C} \tilde{y}_i \log \hat{y}_i, \quad \hat{y}_i = \frac{\exp^{-z_i/T}}{\sum_k \exp^{-z_k/T}},
\]

where \(C\) is the number of segmentation categories, \(z_i\) and \(\tilde{y}_i\) come from a more powerful teacher segmentation model, where \(z_i\) is the output for the \(i\)-th category right before the softmax layer, also known as logit. \(\tilde{y}_i\) is “softened” posterior probability for the \(i\)-th category, controlled by the hyper-parameter \(T\) [17]. We empirically found \(T=1\) works well on a validation set. \(\hat{y}_i\) is the estimated posterior probability of our model. The distillation is aggregated over all pixels in training images.
**Self-supervised loss.** To further constrain the network training, we also define self-supervised loss. Optical flow and disparity are defined as correspondence between two input images. We can therefore compare two corresponding pixels defined by either optical flow or disparity as supervision for network training.

The most straightforward metric is to compare values between two corresponding pixels that are visible in both frames, known as photometric consistency. In a single pyramid level, it is defined as $L_{PC} =$

$$
||I^1 - g(I^1, D^1)||_1 \odot O_D + ||I^2 - g(I^2, F^1)||_1 \odot O_F,
$$

where $g(\cdot, \cdot)$ is the differentiable warping function, $\tilde{O} = 1 - O$, $\odot$ denotes element-wise multiplication followed by summation, and we omit some superscripts when the context is clear. This loss term reasons about occlusion by modulating the consistency loss using the occlusion map and tightly couples occlusion with optical flow and stereo.

As photometric consistency is not robust to lighting changes, we further introduce semantic consistency, encouraging two corresponding pixels to have similar semantic segmentation posterior probability. Specifically, this semantic consistency is defined as $L_{SC} =$

$$
||\tilde{y}^1 - g(\tilde{y}^1, D^1)||_1 \odot \tilde{O}_D + ||\tilde{y}^2 - g(\tilde{y}^2, F^1)||_1 \odot \tilde{O}_F,
$$

where $\tilde{y}$ denotes a posterior probability image coming from the teacher segmentation network used in Eq.(3). Unlike raw pixel values, the segmentation posterior probability is more robust to lighting changes.

Finally, we consider the structural similarity loss

$$
L_{SS} = \gamma_D (1 - SS(I^1, I^1 \odot O_D + g(I^1, D^1) \odot O_D)) + \gamma_F (1 - SS(I^2, I^1 \odot O_F + g(I^2, F^1) \odot O_F)),
$$

where $\odot$ indicates element-wise multiplications only. $SS(\cdot, \cdot)$ is a differentiable function that outputs a single scalar value to measure the structural similarity between two input images [63]. Note that for occluded pixels in the warped image, their values are replaced with values of pixels at the same position in the left/first image.

Table 1. Average EPE results on MPI Sintel optical flow dataset. “-ft” means fine-tuning on the MPI Sintel training set and the numbers in parentheses are results on the data the methods have been fine-tuned on.

| Methods | Training | Test | Time (s) |
|---------|----------|------|----------|
| FlowFields [3] | - | - | 3.75 | 5.81 | 28.0 |
| MRFlow [57] | 1.83 | 3.59 | 2.53 | 5.38 | 480 |
| FlowFieldsCNN [4] | - | - | 3.78 | 5.36 | 23.0 |
| DCFlex [59] | - | - | 3.54 | 5.12 | 6.00 |
| SpyNet-ft [41] | (1.37) | (4.32) | 6.44 | 8.36 | 0.16 |
| FlowNet2 [24] | 2.02 | 3.14 | 3.96 | 6.02 | 0.12 |
| FlowNet2-ft [24] | (1.45) | (2.01) | 4.16 | 5.74 | 0.12 |
| LiteFlowNet [21] | (1.64) | (2.23) | 4.86 | 6.09 | 0.09 |
| PWC-Net [48] | 2.55 | 3.93 | - | - | 0.03 |
| PWC-Net-ft [48] | (1.70) | (2.21) | 3.86 | 5.13 | 0.03 |
| FlowNet3 [25] | 2.08 | 3.94 | 3.61 | 6.03 | 0.07 |
| FlowNet3-ft [25] | (1.47) | (2.12) | 4.35 | 5.67 | 0.07 |
| SENSE-ft | 1.91 | 3.78 | - | - | 0.03 |
| SENSE-ft | (1.54) | (2.05) | 3.60 | 4.86 | 0.03 |

estimation as well. Thus, our networks tightly couple these four closely-related tasks together.

Our final semi-supervised loss consists of supervised, distillation, and self-supervised loss terms. More details can be found in the supplementary material.

4. Experiments

4.1. Implementation Details

**Pre-training of stereo and optical flow.** We use the synthetic SceneFlow dataset [37], including FlyingThings3D, Monkaa, and Driving, for pre-training. All three datasets contain optical flow and disparity ground-truth. Occlusion labels are only available in FlyingThings3D. During training, we uniformly sample images from all three datasets and compute occlusion loss when the ground-truths are available. During training, we use color jittering for both optical flow and disparity training. Additionally, we use random crops and vertical flips for stereo training images. The crop size is 256 × 512. For optical flow training images, we perform extensive data augmentations including random crop, translation, rotation, zooming, squeezing, and horizontal and vertical flip, where the crop size is 384 × 640. The network is trained for 100 epochs with a batch size of 8 using the Adam optimizer [31]. We use synchronized Batch Normalization [58] to ensure there are enough training samples for estimating Batch Normalization layers’ statistics when using multiple GPUs. The initial learning rate is 0.001 and decreased by factor of 10 after 70 epochs.

**Fine-tuning.** For Sintel, we use a similar learning rate schedule as used in [48]. On KITTI 2012 [12] and KITTI 2015 [40], we use longer learning rate schedule, where the model is trained for 1.5K epochs with an initial learning rate of 0.001. We perform another 1K-epoch training with an ini-
Table 2. Results on the KITTI optical flow dataset. “-ft” means fine-tuning on the KITTI training set and the numbers in the parenthesis are results on the data that have been fine-tuned on.

| Methods              | KITTI 2012 | KITTI 2015 |
|----------------------|------------|------------|
|                      | AEPE       | AEPE       | AEPE       | AEPE       |
|                      | train     | test      | train     | test      |
| FlowFields [3]       | -         | -         | -         | -         | 19.80%     |
| MRFLOW [57]          | -         | -         | -         | -         | 14.09 %    |
| DCflow [59]          | -         | -         | -         | -         | 15.09 %    |
| SDF [2]              | -         | 2.3       | 3.80%     | -         | 11.01%     |
| MirrorFlow [23]      | -         | 2.6       | 4.38%     | -         | 9.93%      |
| SpyNet-ft [41]       | (4.13)    | 4.7       | 12.31%    | -         | 35.07%     |
| FlowNet2-ft [24]     | 4.09      | -         | -         | 10.06     | 30.37%     |
| FlowNet2-ft [24]     | (1.28)    | 1.8       | 4.82%     | (2.30)    | 8.61%      |
| LightFlowNet [21]    | (1.26)    | 1.7       | -         | (2.16)    | 8.16%      |
| PW2-Net [48]         | 4.14      | -         | -         | 10.35     | 33.67%     |
| PW-CNet-ft [48]      | (1.45)    | 1.7       | 4.22%     | (2.16)    | 9.80%      |
| FlowNet [25]         | 3.69      | -         | -         | 9.33      |            |
| FlowNet-ft [25]      | (1.19)    | 3.45%     | (1.79)    | -         | 8.60%      |
| SENSE                 | 2.55      | -         | -         | 6.23      | 23.29%     |
| SENSE-ft              | (1.14)    | 1.5       | 3.00%     | (2.01)    | 9.20%      |
| SENSE+semi            | (1.18)    | 1.5       | 3.03%     | (2.05)    | 9.69%      |

Table 3. Results on KITTI stereo datasets (test set).

| Methods              | KITTI 2012 | KITTI 2015 |
|----------------------|------------|------------|
|                      | AEPE       | AEPE       | AEPE       | AEPE       |
|                      | All       | Non-Oc     | All       | Non-Oc     | Time           |
|                      | train     | test      | train     | test      |               |
| Content-CNN [33]     | 3.07      | 4.29      | 8.58      | 4.54      | 7.44          | 4.00           | 1.0            |
| DispNetC [37]        | -         | -         | 4.41      | 4.34      | 3.72          | 4.05           | 0.06           |
| MC-CNN [62]          | 2.43      | 3.63      | 8.88      | 3.89      | 7.64          | 3.33          | 67             |
| PBCP [45]            | 2.36      | 3.45      | 8.74      | 3.61      | 7.71          | 3.17          | 68             |
| Displots v2 [15]     | 2.37      | 3.09      | 5.56      | 3.43      | 4.93          | 3.09          | 265            |
| GC-Net [30]          | 1.77      | 2.30      | 6.16      | 2.87      | 5.58          | 2.61          | 0.9            |
| PSMNet [8]           | 1.49      | 1.89      | 4.62      | 2.33      | 4.31          | 2.14          | 0.41           |
| SegStereo [60]       | 1.68      | 2.03      | 3.70      | 2.08      | 4.07          | 2.25          | 0.6            |
| FlowNet3 [25]        | 1.82      | -         | 2.19      | -         | 2.19          | -             | 0.07           |
| SENSE                 | 1.77      | 2.18      | 3.13      | 2.33      | 2.79          | 2.13          | 0.06           |
| SENSE+semi            | 1.73      | 2.16      | 3.01      | 2.22      | 2.76          | 2.05          | 0.06           |

4.2. Main Results

Optical flow results. Table 1 shows the results for optical flow estimation on the MPI Sintel benchmark dataset. Our approach outperforms CNN-based approaches without or with fine-tuning. On the more photorealistic (final) pass of the test set, which involves more rendering details such as lighting change, shadow, motion blur, etc, our approach outperforms both CNN-based and traditional hand-designed approaches by a large margin.

Disparity results. For disparity estimation, SENSE significantly outperforms previous CNN-based approaches including DispNetC [37] and GC-Net [30] and achieves comparable accuracy with state-of-the-art approaches like PSMNet [8], SegStereo [60], and FlowNet3 [25]. Notably, our approach performs the best on the foreground region in both all and non-occluded regions on KITTI2015.

Scene flow results. Table 4 shows Scene flow results on KITTI 2015. SENSE performs the best in general CNN-based scene flow methods, compared to FlowNet3 [25]. Compared to ISF [5], SENSE is 2K times faster and can handle general nonrigid scene motions.

To remove artifacts introduced by the second frame disparity warping operation, we use a refinement network of a encoder-decoder structure with skip connections. It takes \( I^{1,F}, O^{2,F}, D^{1,F}, \) and \( g(D^{2,F}, F^{1,F}) \) to generate a residual that is added to the warped disparity. From our holistic outputs, it manages to handle the challenging cases of nonrigid motion.
we can refine the background scene flow using a rigidity refinement step. We first determine the static rigid areas according to semantic segmentation outputs. We then calculate the ego-motion flow by minimizing the geometry consistency between optical flow and disparity images using the Gauss-Newton algorithm. Finally, we compute the warped scene flow using the disparity of the reference frame and the ego-motion to substitute the raw scene flow only in the rigid background region. This step additionally produces camera motion and better scene flow with minimal costs. Details of refinement steps are provided in supplementary material.

Running time. SENSE is an efficient model. SENSE takes 0.03s to compute optical flow between two images of size 436 \times 1024. For disparity, SENSE is an order of magnitude faster than PSMNet and SegStereo, and slightly faster than FlowNet3. For scene flow using KITTI images, SENSE takes 0.15s to generate one optical flow and two disparity maps. The additional warping refinement network takes 0.01s and the rigidity refinement takes 0.15s to generate one optical flow and two disparities.

Model size and memory. SENSE is small in size. It has only 8.8M parameters for the optical flow model, and 8.3M for the disparity model. The scene flow model with shared encoder has 13.4M parameters. In contrast, FlowNet3 has a flow model (117M) and a disparity model (117M), which is 20 times larger. SENSE also has a low GPU memory footprint. FlowNet3 costs 7.4GB while SENSE needs 1.5GB RAM only. Although PSMNet has fewer parameters (5.1M), it costs 4.2GB memory due to 3D convolutions.

4.3. Ablation Studies

Performance of different tasks. We report results of different tasks using different combinations of encoder and decoders. Our models are trained using 160 images of KITTI 2015 with a half of the aforementioned learning rate schedule. Results are reported on the rest 40 images in Table 5. We can see that the shared encoder model performs better than models trained separately.

Semi-supervised loss. To study the effects of distillation and self-supervised loss terms, we perform ablation studies using all images of KITTI 2012 and 160 images of KITTI 2015 for training with a half of full learning rate schedule. The rest 40 ones of KITTI 2015 are used for testing. We finetune the baseline model using sparse flow and disparity annotations only. Table 6 shows the quantitative comparisons and Fig. 4 highlights the effects qualitatively.

Regarding distillation loss, both segmentation and occlusion distillation loss terms are useful for disparity and optical flow estimation. However, distillation loss is not helpful for reducing the artifacts in sky regions. Thus, the self-supervised loss is essential, as shown in Fig. 4, though quantitatively self-supervised loss is not as effective as the distillation loss. Finally, combining all loss terms yields the best optical flow and disparity accuracies. We also test SENSE trained using semi-supervised loss on KITTI, as summarized in Tables 2, 3, and 4. We can see it improves disparity and optical flow accuracy on KITTI 2015 and also leads to better disparity on KITTI 2012.

5. Conclusion

We have presented a compact network for four closely-related tasks in holistic scene understanding: Sharing an encoder among these tasks not only makes the network compact but also improves performance by exploiting the interactions among these tasks. It also allows us to introduce distillation and self-supervision losses to deal with partially labeled data. Our holistic network has similar accuracy and running time as specialized networks for optical flow. It performs favorably against state-of-the-art disparity and scene flow methods while being much faster and memory efficient. Our work shows the benefits of synergizing closely-related tasks for holistic scene understanding and we hope the insights will aid new research in this direction.

Acknowledgement

Huaizu Jiang and Erik Learned-Miller acknowledge support from AFRL and DARPA (#FA8750-18-2-0126) and the MassTech Collaborative grant for funding the UMass GPU cluster. The U.S. Gov. is authorized to reproduce and distribute reprints for Gov. purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFRL and DARPA or the U.S. Gov.
approach for efficient and accurate scene flow. In Proc. ECCV, 2016. 7
[35] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In Proc. ECCV, 2018. 2
[36] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In Proc. CVPR, 2019. 2, 3
[37] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proc. CVPR, 2016. 1, 2, 5, 6, 7
[38] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proc. CVPR, 2016. 3, 5
[39] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In Proc. CVPR, pages 3061–3070, 2015. 1, 2, 3
[40] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In ISPRS Workshop on Image Sequence Analysis (ISA), 2015. 6
[41] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In Proc. CVPR, 2017. 3, 6, 7
[42] Zhile Ren, Deqing Sun, Jan Kautz, and Erik Sudderth. Cascaded scene flow prediction using semantic segmentation. In 3DV, 2017. 2
[43] René Schuster, Christian Bailier, Oliver Wasenmüller, and Didier Stricker. Combining stereo disparity and optical flow for basic scene flow. In CVTS, 2018. 7
[44] René Schuster, Oliver Wasenmuller, Georg Kuschk, Christian Bailier, and Didier Stricker. Sceneflowfields: Dense interpolation of sparse scene flow correspondences. In WACV, 2018. 7
[45] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In BMVC, 2016. 7
[46] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J. Black. Optical flow with semantic segmentation and localized layers. In Proc. CVPR, 2016. 2
[47] Deqing Sun, Erik B Sudderth, and Michael J Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In Advances in Neural Information Processing Systems, pages 2226–2234, 2010. 2
[48] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In CVPR, June 2018. 3, 6, 7
[49] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE TPAMI, 2019. 7
[50] Brian Taylor, Vasiliiy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In Proc. CVPR, pages 4268–4276, 2015. 2
[51] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In Proc. CVPR, 2017. 2
[52] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In Proc. ICCV, 1999. 1, 2
[53] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In Proc. ICCV, pages 1377–1384, 2013. 2
[54] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. IJCV, 115(1), 2015. 1
[55] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. IEEE Transactions on Image Processing, 3(5):625–638, Sept. 1994. 2
[56] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In Proc. ECCV, pages 739–751. Springer, 2008. 2
[57] Jonas Wulf, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In Proc. CVPR, 2017. 6, 7
[58] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In Proc. ECCV, 2018. 4, 6, 7
[59] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In Proc. CVPR, 2017. 6, 7
[60] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. SegStereo: Exploiting semantic information for disparity estimation. In Proc. ECCV, 2018. 3, 7
[61] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 3
[62] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. JMLR, 2016. 7
[63] Hang Zhao, Orazio Gallo, Iuri Froisio, and Jan Kautz. Loss functions for image restoration with neural networks. IEEE Trans. Computational Imaging, 3(1):47–57, 2017. 6
[64] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In Proc. CVPR, 2017. 4
[65] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised learning of depth and flow using cross-task consistency. In Proc. ECCV, 2018. 3