# Consistency Guided Scene Flow Estimation

Yuhua Chen[1,2], Luc Van Gool[2], Cordelia Schmid[1], and Cristian Sminchisescu[1]

[1] Google Research
[2] ETH Zurich

**Abstract.** Consistency Guided Scene Flow Estimation (*CGSF*) is a self-supervised framework for the joint reconstruction of 3D scene structure and motion from stereo video. The model takes two temporal stereo pairs as input, and predicts disparity and scene flow. The model self-adapts at test time by iteratively refining its predictions. The refinement process is guided by a consistency loss, which combines stereo and temporal photo-consistency with a geometric term that couples disparity and 3D motion. To handle inherent modeling error in the consistency loss (e.g. Lambertian assumptions) and for better generalization, we further introduce a learned, output refinement network, which takes the initial predictions, the loss, and the gradient as input, and efficiently predicts a correlated output update. In multiple experiments, including ablation studies, we show that the proposed model can reliably predict disparity and scene flow in challenging imagery, achieves better generalization than the state-of-the-art, and adapts quickly and robustly to unseen domains.

**Keywords:** scene flow, disparity estimation, stereo video, geometric constraints. self-supervised learning.

## 1 Introduction

Scene flow is the task of jointly estimating the 3D structure and motion of a scene [34]. This is critical for a wide range of downstream applications, such as robotics, autonomous driving and augmented reality. Typically, scene flow is estimated from consecutive stereo pairs, and requires simultaneously solving two sub-problems: stereo matching and optical flow. Though closely related, the two tasks cover different aspects of scene flow: stereo matching seeks to estimate disparity from a static stereo pair, whereas the objective of optical flow is to find the correspondence between temporally adjacent frames.

Many end-to-end models powered by deep neural network have emerged for the scene flow estimation [1, 15] and its components [5, 13, 29], and they showed promise in benchmarks. However, training such models relies on the availability of labeled data. Due to difficulty in acquiring real-world data, synthetic data [23] is widely used as the major supervision source for deep models, with 3D structure or motion generated automatically using computer graphics techniques. Unfortunately, synthetic data still exhibits noticeable appearance dissimilarity leading to domain shift, which often prevents models trained this way from generalizing to the real-world. This has been observed previously [27, 31]. This issue can be partially addressed by finetuning on labeled real-world data. However, collecting ground-truth labels for scene flow can be extremely challenging, requiring the use of costly sensors and additional manual intervention [25, 26]. This, however, may not always be feasible in many real-world scenarios, greatly limiting the applicability of deep models.

To address this issue, we present Consistency Guided Scene Flow (*CGSF*), a framework that additionally models output consistency. The framework begins with producing scene flow prediction using feedforward deep network. The predictions include disparity, optical flow and disparity change. These predictions are then coupled by a consistency loss which captures the photometric and geometric relations valid in scene flow, irrespective of domain shift. Therefore, the consistency loss can be a powerful cue in guiding predictions for better generalization in unseen domains. However, the consistency loss is inherently noisy due to the complexity of natural data, such as non-Lambertian surfaces or occlusion. As a result, the provided guidance is not always precise, and can exhibit undesired artifacts. To further correct such issues that are difficult to model explicitly, we further introduce a learned refinement module, which takes the initial predictions, the loss, and the gradient as input, and predicts an update to recursively refine the prediction. The refinement module is implemented as a neural network, and thus can be trained jointly with the original feedforward module.

Our *CGSF* model can be trained either using synthetic data, which can generalize well to real imagery, or trained self-supervised, using unlabeled data, based on the proposed consistency loss. In diverse experiments, we show our *CGSF* can reliably predict disparity and scene flow in challenging scenarios. Moreover, we observe that the proposed, learned refinement module, can significantly improve the results of the more classical feedforward network, by ensuring consistent predictions. In particular, we demonstrate that the proposed model significantly improves generalization to unseen domains, thus better supporting real-world applications of scene flow, where a degree of domain shift is inevitable.

## 2 Related Work

**Scene Flow Estimation.** introduced by Vedula *et al.* [34], scene flow estimation aims to recover the 3D structure and motion of a scene. The task has been formulated as a variational inference problem by multiple authors [3, 12, 35, 38]. Recently, several deep models have emerged for this task. Ilg *et al.* [14] combine networks for stereo and optical flow using an additional network to predict disparity change. Ma *et al.* [22] stack three networks to predicting disparity, flow and segmentation, then use a Gaussian-Newton solver to estimate per-object 3D motion. To leverage correlation between tasks, Jiang *et al.* [15] propose an encoder architecture shared among the tasks of disparity, optical flow, and segmentation. Similarly, Aleotti *et al.* [1] propose a lightweight architecture to share information between tasks. Complementary to previous work, our main objective is to improve generalization of the scene flow by means of additional constraints, self-supervised losses, and learnt refinement schemes.

**Flow and Disparity Estimation.** As scene flow requires simultaneously reasoning about two tasks: disparity and flow estimation, it has been largely influenced by the techniques used in end-to-end stereo matching and optical flow.

For optical flow estimation, FlowNet [8] represents the first attempt in building end-to-end deep models. FlowNet 2.0 [13] further improved performance by dataset scheduling and refinement. A spatial pyramid design is adopted in PWC-Net [29], which includes cost volume processing at multiple levels. PWC-Net achieved highly competitive performance on several benchmarks for optical flow estimation.

In stereo matching, Mayer *et al.* [23] introduce an architecture similar to FlowNet, for disparity estimation from rectified image pairs. Several techniques have been introduced to improve the matching accuracy, including 3D convolution in GC-Net [16], pyramid pooling in PSM-Net [5], *etc.* More recently, GA-Net [40] integrated two aggregation layers with good results.

**Adaptation.** Due to the practical difficulty of acquiring ground-truth labels, the aforementioned models are often trained on synthetic data, although a performance drop is observed when applied in the real-world [27, 31].

To address this issue, several techniques [11, 28, 31–33, 41] have been proposed to adapt the model to new domains where labels aren't available. For the purpose, either offline adaptation [31] or online adaptation [33] is performed to refine the model parameters in the new domain. One key difference in our work is that our refinement module operates on prediction (outputs) directly instead of network parameters, and it thus relies on the output consistency among outputs to overcome domain shift.

**Self-supervision.** On the other hand, self-supervised learning has been used with unlabelled data [4, 9, 10, 18, 36, 39, 42]. Some methods require collecting unlabelled data for offline training, which might not be always possible. Even so, trained model suffer from domain shift in new environments, which still requires online adaptation. Part of our modeling is based on similar ideas, in that we can also train with self-supervision in an offline stage. However, we additionally integrate output consistency terms in our refinement module, which makes our framework easier to self-adapt to new environments without collecting unlabeled data upfront.

**Online Refinement.** Online optimization under self-supervised consistency losses has shown to be effective for depth prediction in video [4, 6]. In this work, we learn a refinement network to preserve the output consistency. Our refinement network shares ideas with learning-based optimization [2, 19], where one aims to integrate iterative refinement into deep networks. Different metalearning ideas have also been explored in different applications, such as learning depth and pose estimation [7], 3D rigid motion estimation [21], or monocular reconstruction of static scenes [30]. Here we focus on problem-domain updates that integrate gradient information in order to ensure progress, for robustness, as well as computational efficiency. In contrast, our model consists of several novel geometric constraints for scene flow estimation, designed for self-supervision and overcoming domain shift.

# 3 Consistency Guided Scene Flow

## 3.1 Overview

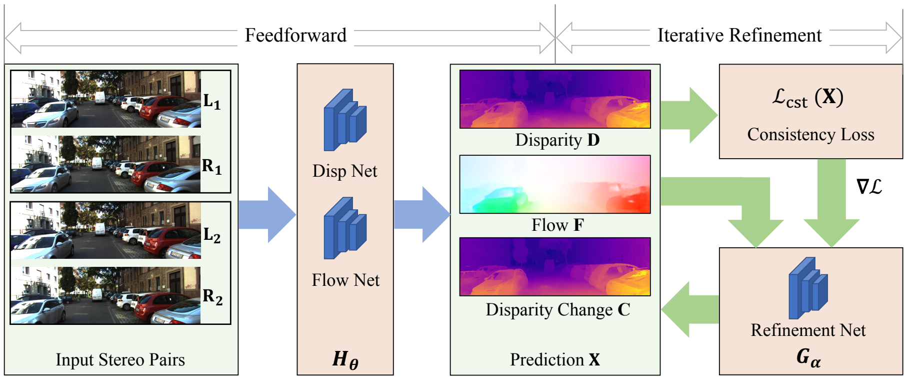

**Fig. 1. Illustration of our proposed Consistency Guided Scene Flow (*CGSF*).** The model can be obtained either using labeled data, or unlabeled images and self-supervision. At runtime, it self-adapts by producing a set of outputs using feedforward networks ($\mathbf{H}_\theta$, flow by blue arrows), then iteratively refining those estimates on two temporally adjacent stereo pairs, using a learned update (network $\mathbf{G}_\alpha$, flow by green arrows). The refinement process is guided by a consistency loss, which captures several intrinsic photometric and geometric constraints associated to scene flow.

In this section we present our Consistency Guided Scene Flow framework (*CGSF*). An overview is given in fig. 1. Given a couple of stereo images at subsequent timesteps $(\mathbf{L}_1, \mathbf{R}_1)$ and $(\mathbf{L}_2, \mathbf{R}_2)$, respectively, as input, the model predicts scene flow $\mathbf{X}$, *i.e.*, the 3D structure and motion of the environment. The 3D scene structure can be represented by the disparity in the first frame $\mathbf{D}_1$, as disparity is inversely proportional to depth. The 3D scene motion can be decomposed into the motion parallel to the image plane, represented by optical flow $\mathbf{F}_{1\to2}$, and the motion orthogonal to the image plane, represented as disparity change $\mathbf{C}_{1\to2}$.

At runtime, we formulate scene flow estimation as an iterative optimization problem. Our framework predicts scene flow in two stages: a more typical feedforward block and an iterative output refinement module (N.B. this is also feedforward, but has a recurrent structure). In the first stage, we rely on a feedforward module $\mathbf{H}_\theta$ to predict scene flow given input images. In the refinement stage, our geometric and photometric consistency losses are further optimized with respect to the outputs using a refinement network $\mathbf{G}_\alpha$. Both the feedforward module $\mathbf{H}_\theta$ and the refinement network $\mathbf{G}_\alpha$ are implemented as neural networks with learnt parameters $\theta$ and $\alpha$. Therefore the two components can be trained jointly in an end-to-end process.

In the sequel, we introduce the feedforward module in §3.2. We formulate the geometric and photometric consistency terms for scene flow estimation in §3.3. We present the learned refinement network in §3.4 and describe the training and evaluation protocol in §3.5.

### 3.2 Feedforward Scene Flow Module

In the feedforward stage, the prediction of scene flow is obtained by passing the input images through disparity and flow networks. The disparity network predicts the 3D structure given stereo pairs. Specifically, the module predicts $\mathbf{D}_1$ from $(\mathbf{L}_1, \mathbf{R}_1)$, and $\mathbf{D}_2$ from $(\mathbf{L}_2, \mathbf{R}_2)$. Given $(\mathbf{L}_1, \mathbf{L}_2)$ as input, the flow network predicts optical flow $\mathbf{F}_{1\to 2}$, and the disparity change $\mathbf{C}_{1\to 2}$ which encodes the motion component orthogonal to the image plane.

In this work, we build the feedforward module upon GA-Net [40] and PWC-Net [29], mainly due to both competitive performance and implementation compatibility. Here we briefly introduce the architectures.

**Disparity network**: We rely on GA-Net [40] to predict scene structure, *i.e.*, $\mathbf{D}_1$ and $\mathbf{D}_2$. The network extracts features from a stereo pair with a shared encoder. Then features are used to build a cost volume, which is fed into a cost aggregation block for regularization, refinement and disparity prediction.

**Flow network**: We modify PWC-Net [29] to predict the 3D motion, including optical flow $\mathbf{F}_{1\to 2}$ and disparity change $\mathbf{C}_{1\to 2}$. We follow the design in PWC-Net to construct a cost volume from the features of the first frame, and the warped features of the second frame. The cost volume is further processed to obtain the motion prediction. We increase the output channel from 2 to 3 to encode 3D motion.

In principle, our framework is agnostic to specific choices of model design– other options for the structure and motion networks are possible. This offers the flexibility of leveraging the latest advances in the field, and focus on our novel self-supervised geometric and photometric consistency losses, described next.

### 3.3 Scene Flow Consistency Loss

In this section we formulate the photometric and geometric constrains that are part of our proposed consistency loss.

**Stereo Photometric Consistency.** Photometric consistency between stereo pairs is widely used as a self-supervised loss in deep models [9, 17]. The idea is to warp the source view via a differentiable bilinear map produced by the network prediction. The photometric difference between the warped source image and the target image can be used as a self-supervision signal.

Given a stereo pair $(\mathbf{L}, \mathbf{R})$, we use the left view $\mathbf{L}$ as the target image, and the right view $\mathbf{R}$ as the source image. Considering a pixel $\mathbf{p}$ in the left view, its correspondence in the right view can be obtained by subtracting its disparity value from its x coordinate. With a slight abuse of notation, we denote it as $\mathbf{p} - \mathbf{D}(\mathbf{p})$. If $\mathbf{p}$ corresponds to a 3D scene point visible in both views, its correspondence should be visually consistent. Following [9], we use a weighted sum of L1 loss and SSIM [37] loss as the photometric error, denoted as $pe$. Therefore the photometric stereo consistency model can be written as

$$\mathcal{L}_{pd}(\mathbf{p}) = pe(\mathbf{L}(\mathbf{p}), \mathbf{R}(\mathbf{p} - \mathbf{D}(\mathbf{p}))) \tag{1}$$

The stereo consistency applies to both $\mathbf{D}_1$ and $\mathbf{D}_2$.

**Flow Photometric Consistency.** We also rely on photometric consistency for temporal frames. Given a pixel $\mathbf{p}$ in the first frame $\mathbf{L}_1$, its correspondence in the second frame $\mathbf{L}_2$ can be obtained by displacing $\mathbf{p}$ using optical flow. We can write the flow photometric consistency as

$$\mathcal{L}_{pf}(\mathbf{p}) = \mathbf{O}_f(\mathbf{p}) \cdot pe(\mathbf{L}(\mathbf{p}), \mathbf{L}(\mathbf{p} + \mathbf{F}_{1\to2}(\mathbf{p}))) \tag{2}$$

$\mathbf{O}_f$ represents the occlusion map caused by optical flow. Similarly with previous work [20, 24], we infer the occlusion map using a forward-backward flow consistency check. In more detail, an additional backward flow is computed and then warped by the the forward flow $\mathbf{F}_{1\to2}$. We denote the computed warped backward flow map by $\hat{\mathbf{F}}_{2\to1}$. A forward-backward consistency check is performed to estimate the occlusion map

$$\mathbf{O}_f = [\![ |\mathbf{F}_{1\to2} + \hat{\mathbf{F}}_{2\to1}|^2 < w_1(|\mathbf{F}_{1\to2}|^2 + |\hat{\mathbf{F}}_{2\to1}|^2) + w_2 ]\!] \tag{3}$$

where $[\![\cdot]\!]$ is the Iverson bracket, and we set $w_1 = 0.01, w_2 = 0.05$ in inferring the occlusion mask.

**Disparity-Flow Consistency.** We formulate a new consistency term to connect disparity and 3D motion. Given a pixel $\mathbf{p}$ in the left view of the first frame $\mathbf{L}_1$, its correspondence in $\mathbf{L}_2$ is associated by optical flow $\mathbf{F}_{1\to2}$ can be written as $\mathbf{p} + \mathbf{F}_{1\to2}(\mathbf{p})$. Thus its disparity value in the second frame should be $\mathbf{D}_2(\mathbf{p} + \mathbf{F}_{1\to2}(\mathbf{p}))$. On the other hand, $\mathbf{p}$'s disparity value in the second frame can also be obtained by adding the disparity change $\mathbf{C}_{1\to2}$ to the disparity of the first frame $\mathbf{D}_1$. Therefore the disparity flow consistency can be written as

$$\mathcal{L}_{df}(\mathbf{p}) = \mathbf{O}_f(\mathbf{p}) ||\mathbf{D}_1(\mathbf{p}) + \mathbf{C}_{1\to2}(\mathbf{p}) - \mathbf{D}_2(\mathbf{p} + \mathbf{F}_{1\to2}(\mathbf{p}))|| \tag{4}$$

We use $\mathbf{O}_f$ to mask out occluded pixels as flow warping is used in the process.

**Overall Consistency Loss.** By integrating all previously derived terms, the consistency loss can be written as

$$\mathcal{L}_{cst} = \mathcal{L}_{pd} + \mathcal{L}_{pf} + \mathcal{L}_{df} \tag{5}$$

The consistency loss regularizes predictions resulting from the individual structure and flow networks $\mathbf{X}$, and captures several intrinsic geometric and photometric constraints that should always hold. To provide additional regularization of predictions, we use an edge-aware smoothness term as in [9]. This is applied to the disparity map $\mathbf{D}_1, \mathbf{D}_2$, optical flow $\mathbf{F}_{1\to2}$, and disparity change $\mathbf{C}_{1\to2}$.

### 3.4 Consistency Guided Refinement

In the refinement stage, the goal is to recursively improve scene flow prediction. We denote as $\mathbf{X}^t$ the scene flow prediction after $t$ steps of refinement. The initial prediction from the classical feedforward module can be denoted as $\mathbf{X}^0$.

As the consistency loss $\mathcal{L}_{cst}$ reflects the intrinsic structure of scene flow, it can be used as an indicator of how good the prediction is. Therefore we aim to leverage the signal in consistency loss to guide online finetuning. To this end, we build a refinement network $\mathbf{G}_\alpha$ parameterized by $\alpha$, which takes as input the previous prediction $\mathbf{X}^t$, the consistency loss over the previous prediction $\mathcal{L}_{cst}(\mathbf{X}^t)$, and the gradient on predictions $\nabla_{\mathbf{X}}\mathcal{L}_{cst}(\mathbf{X}^t)$. The reason for using the gradient is to provide an exact signal for the descent direction. The refinement network then predicts an update to refine the prediction to $\mathbf{X}^{t+1}$.

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \mathbf{G}_\alpha(\mathbf{X}, \mathcal{L}_{cst}, \nabla_{\mathbf{X}}\mathcal{L}_{cst}) \tag{6}$$

For implementation, we first concatenate all predictions, including $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{C}_{1\to2}$ each with size $W \times H \times 1$, and optical flow $\mathbf{F}_{1\to2}$ with size $W \times H \times 2$. The concatenated prediction map is of dimension $W \times H \times 5$, with $W$ and $H$ being the width and height of the image, respectively. The gradient is of the same dimensionality as the prediction, and the loss map is of 1 channel. Therefore, all inputs can be concatenated as a $W \times H \times 11$ map. The benefit of using a concatenated map as input is that cross-channel correlations can be better captured. For instance, disparity information might be helpful to improve optical flow.

We implement the refinement module as a small fully convolutional network(FCN) operating on the input map, to provide dense (per pixel) output for the updates. The advantage is that the FCN architecture can learn to leverage the local inter-pixel context, which is helpful in reducing noise in the consistency loss, and in propagating information to and from neighbouring pixels. We use a lightweight design of 3 convolution layers, each with 512 hidden units and kernel size of $3\times3$. ReLU is used between convolution layers as the activation function. The light-weight design is motivated by the recurrent use of the module.

### 3.5 Training

To learn the parameters of the feedforward scene flow $\mathbf{H}_\theta$ and refinement network $\mathbf{G}_\alpha$, the framework can be initialized based on pre-training using synthetic data with ground-truth labels (whenever available), and then/or further trained using the self-supervised loss $\mathcal{L}_{cst}$, in order to better generalize to real imagery. When ground-truth labels are available, a standard supervised loss can be used

$$\mathcal{L}_{sup}(\mathbf{X}) = ||\mathbf{F}_{1\to 2} - \mathbf{F}^*_{1\to 2}|| + ||\mathbf{D}_1 - \mathbf{D}^*_1|| + ||\mathbf{C}_{1\to 2} - \mathbf{C}^*_{1\to 2}|| \tag{7}$$

where $*$ indicates ground-truth. In either case, the network parameters $\theta, \alpha$ can be learned jointly by minimizing the loss using stochastic gradient descent. The optimization objective writes

$$\arg\min_{\theta,\alpha} \sum_{t\in\{0,\ldots,T\}} \mathcal{L}(\mathbf{X}^t) \tag{8}$$

where the loss $\mathcal{L}$ can be either $\mathcal{L}_{sup}$(when the label is available) or $\mathcal{L}_{cst}$(in self-supervised training). The loss is applied to the initial prediction, as well as to the prediction after each refinement step (i.e. we supervise the augmented step update at each iteration). Essentially after each step we minimize the loss w.r.t. $\boldsymbol{\alpha}$ so that the learnt descent direction produces as large of a loss decrease as possible, and aggregate in parallel over $T = 5$ refinement steps.

## 4 Experiments

In this section, we experimentally validate the proposed *CGSF*. First, we test our model on synthetic data on FlyingThings3D in §4.1. Then, in order to verify real-image generalization, we test the model on KITTI in §4.2. In §4.3 we provide ablation studies to provide further insight into our design choices. Finally, in §4.4 we show the performance of *CGSF* on different scenes captured by a stereo camera, in order to demonstrate the generalization to environments not seen in training.

### 4.1 Evaluation on Synthetic Data

First we evaluate the model on synthetic data. We use the FlyingThings3D [23] which is a synthetic dataset for scene flow estimation, containing $22,390$ stereo training pairs. Dense ground-truth labels are available in all images for disparity, optical flow and disparity change. We use the ground-truth to supervise our model, including the feedforward module and the refinement network.

In the refinement module, we use a set of trade-off weights to balance different consistency terms. We use $\omega_{pd}, \omega_{pf}, \omega_{df}$ to denote, respectively, the weight of stereo photometric consistency $\mathcal{L}_{pd}$, flow disparity consistency $\mathcal{L}_{pf}$, and disparity disparity-flow consistency $\mathcal{L}_{df}$. The weight of the smoothness term is denoted as $\omega_s$. We set $\omega_{pd} = 1, \omega_{pf} = 1, \omega_{df} = 1, \omega_s = 0.1$. A stage-wise strategy is used in

| | Disparity EPE | Flow EPE | Disparity Change EPE * |
|---|---|---|---|
| DWARF [1] | 2.00 | 6.52 | 2.23 |
| Feedforward | 0.83 | 7.02 | 2.02 |
| *CGSF*(ours) | **0.79** | **5.98** | **1.33** |

**Table 1. Scene flow results on the FlyingThings3D test set**. Results are evaluated as end point error (EPE). All models are trained on the training set of FlyingThings3D. Disparity change EPE represents the error of the motion orthogonal to the image plane.

order to improve the stability of the training process. In more detail, we first train the disparity network and the flow network separately. For this part, we follow [40] and [29] in setting hyperparameters. These are then used as initialization for additional, joint fine-tuning. Specifically, we randomly initialize the weights of the refinement module, and jointly fine-tune the whole network including the pre-trained feedforward networks. To fine-tune the entire network, we use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to $10^{-4}$. The network is trained on FlyingThings 3D for 10 epochs with a batch size of 4.

The models are evaluated on the test set of FlyingThings3D, which contains $4,370$ images. As scene flow model requires 4 images as input, we drop the last frame of every sequence, resulting in a test set with $3,933$ images. The performance is measured by end point error(EPE) for the three key outputs: optical flow, disparity, and disparity change. We use the results produced by the feedforward module without a refinement network as baseline. To facilitate the comparison with the state-of-the-art, we test against a recent model DWARF [1] in the same experimental setting. As shown in table 1, the refinement model improves over the feedforward baseline in all three metrics, thus supporting the effectiveness of the consistency guided refinement process. Our model also outperforms DWARF [1] by a large margin. Again, this shows the benefit of geometrically modelling the consistency of scene flow outputs.

### 4.2 Generalization to Real Images

To verify the performance in real imagery where no labeled data is available, we evaluate on the KITTI scene flow dataset [25, 26], which contains 200 training images with sparse ground-truth disparity and flow acquired by a lidar sensor. The ground-truth is only used in evaluation. To comply with the KITTI scene flow benchmark, the percentage of outliers in the following 4 categories is used as evaluation metrics, D1: outliers in the disparity of first frame (*i.e.*, $\mathbf{D}_1$), D2: outliers in second frame warped to the first frame (*i.e.*, $\mathbf{D}_1 + \mathbf{C}_{1\to2}$), F1: outliers in optical flow (*i.e.*, $\mathbf{F}_{1\to2}$), SF: outliers in either D1,D2 or F1. A pixel is considered correct if the prediction end-point error is smaller than $3px$ or 5%.

As discussed in §3.3, our *CGSF* model can be trained with round-truth labels from synthetic data, or using self-supervision. We evaluate both strategies. For

| Method | Training Data | D1 | D2 | F1 | SF |
|---|---|---|---|---|---|
| Godard *et al* [9] | K(u) | 9.19 | - | - | - |
| BridgeDepthFlow [18] | K(u) | 8.62 | - | 26.72 | - |
| UnOS [36] | K(u) | 5.94 | - | 16.30 | - |
| MADNet [33] | F(s)+ K(u) | 8.41 | - | - | - |
| DWARF [1] | F(s) | 11.60 | 29.64 | 38.32 | 45.58 |
| Feedforward | F(s) | 11.53 | 27.83 | 33.09 | 43.55 |
| *CGSF*(ours) | F(s) | 7.10 | 19.68 | 27.35 | 35.40 |
| *CGSF*(ours) | K(u) | 6.65 | 17.69 | 23.05 | 31.52 |
| *CGSF*(ours) | F(s)+ K(u) | 5.75 | 15.53 | 20.45 | 28.14 |

**Table 2. Evaluation on the KITTI 2015 scene flow dataset.** The percentage of outliers is used as evaluation metric. A pixel is considered as correct if the prediction end-point error is smaller than $3px$ or 5%. For training data, 'K(u)' stands for unsupervised training on KITTI, and 'F(s)' represents supervised training on FlyingThings3D.

training using ground-truth on synthetic data, we already illustrated FlyingThings3D. For self-supervised training on KITTI, we use KITTI raw as a training set, with all test scenes excluded.

We compare with the supervised feedforward network without refinement as a baseline. Additionally, we compare with several recent competing methods, including self-supervised models based on stereo images [9] or stereo video [18, 36], scene flow supervised on synthetic data [1], and an adaptation model for stereo estimation [33]. We initialize [33] with the pre-trained weights from synthetic data, and do unsupervised adaptation on KITTI raw (the same set used in our model's unsupervised training).

The results are summarized in table 2. With the feedforward module only, the baseline achieves an error rate of 43.55%. The error can be significantly reduced to 35.40% by the proposed refinement network. Notably, the performance gain is much larger compared to the one in FlyingThings3D, which demonstrates that consistency plays a central and positive role in cross-domain scenarios. This indicates that consistency can be used as an effective model to overcome domain gaps and improve generalization.

Whenever collecting an unlabeled training set beforehand is possible, the model can also be trained under self-supervision. This results in an error rate of 31.52%, which demonstrates both the effectiveness of a self-supervised consistency loss, and good compatibility when supervision is available. Performance can be further improved by combining pre-training on synthetic data, and self-supervised finetuning. This combination results in the lowest scene flow error rate of 28.14%.

Compared to other component methods, our model outperforms the scene flow approach DWARF [1]. The performance of disparity estimation is also considerably better than the self-adaptive stereo model MADNet [33]. Compared to other self-supervised models relying on stereo data [9, 18, 36], our model achieves

| Refinement | $\mathcal{L}_{pd}$ | $\mathcal{L}_{pf}$ | $\mathcal{L}_{df}$ | D1 | D2 | F1 | SF |
|---|---|---|---|---|---|---|---|
| | | | | 11.29 | 26.51 | 32.71 | 43.01 |
| ✓ | | | | 10.40 | 25.15 | 32.05 | 42.43 |
| ✓ | ✓ | | | 7.95 | 22.48 | 31.38 | 39.80 |
| ✓ | | ✓ | | 9.84 | 23.92 | 29.06 | 39.64 |
| ✓ | | | ✓ | 9.11 | 21.14 | 30.95 | 39.59 |
| ✓ | ✓ | ✓ | | 7.60 | 22.73 | 29.07 | 38.23 |
| ✓ | ✓ | ✓ | ✓ | 7.10 | 19.68 | 27.35 | 35.40 |

**Table 3. Ablation study for the consistency terms.** Different consistency terms are used in the refinement module. All models are trained on FlyingThings3D and evaluated on KITTI.

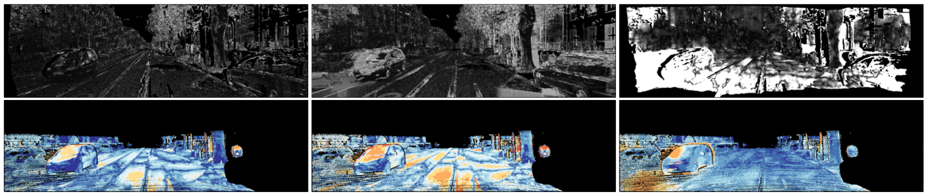

**Fig. 2. Illustration of different consistency terms.** From left to right we show results produced using $\mathcal{L}_{pd}, \mathcal{L}_{pf}, \mathcal{L}_{df}$. The first row presents the pixel-wise loss map of the feedforward prediction, where brighter is larger loss. The second row shows the error map for disparity estimation, when compared with the ground-truth label. Red indicates higher error, whereas blue indicates lower error.

competitive results on disparity and flow estimation, while being additionally capable of predicting 3D motion.

### 4.3 Ablation Studies

We provide further experimental analysis on ablated versions of our model, for further insight into the design choices. To ensure a fair comparison, all models used in this section are trained on FlyingThings3D, and results are reported on KITTI2015 without self-supervised finetuning.

**Consistency in Refinement Network** In the refinement network, the consistency loss is used as a guidance to refine the network outputs. The consistency loss mainly consists of three terms: a stereo photometric loss ($\mathcal{L}_{pd}$), a flow photometric loss ($\mathcal{L}_{pf}$) and a disparity flow consistency loss ($\mathcal{L}_{df}$). To understand the role of each loss, we conduct an ablation study where different combinations are used in the refinement network.

We summarize the results in table 3. When no consistency is used, the refinement network only takes the prediction **X** as input. This baseline results in

| $\mathbf{X}$ | $\mathcal{L}$ | $\nabla\mathcal{L}$ | D1 | D2 | F1 | SF |
|---|---|---|---|---|---|---|
| ✓ | | | 10.40 | 25.15 | 32.05 | 42.43 |
| ✓ | ✓ | | 9.77 | 22.80 | 30.95 | 40.47 |
| ✓ | | ✓ | 7.44 | 20.01 | 28.20 | 36.31 |
| ✓ | ✓ | ✓ | 7.10 | 19.68 | 27.35 | 35.40 |

**Table 4. Ablation study on the input to the refinement network.** All models are trained on FlyingThings3D, and evaluated on KITTI.

a scene flow error of of 42.43, slightly better than the feedforward result which is 43.01. This suggests that only $\mathbf{X}$ is insufficient in effectively guiding refinement. The error can be reduced by additionally including consistency terms, which supports the efficacy of our guided refinement strategy. The contributions of each loss term to the scene flow error are similar. However, each loss term exhibits different improvements over the three sub-measurements. Notably, $\mathcal{L}_{pd}$ improves $D1$ measure the most, $\mathcal{L}_{pf}$ improves $F1$ the most, whereas $\mathcal{L}_{df}$ reduces the $D2$ error, which is related to both disparity and optical flow. The results are understandable, as each loss intrinsically constraints different outputs. For example, the stereo photometric loss $\mathcal{L}_{pd}$ relates stronger to disparity, whereas the flow photometric loss $\mathcal{L}_{pf}$ better constrains flow.

To further understand the impact of different losses, in fig. 2 we visualize refinement results by using only $\mathcal{L}_{pd}$, $\mathcal{L}_{pf}$ or $\mathcal{L}_{df}$. We observe that the photometric loss (the left and middle) is sparse and cannot provide sufficient signal in the textureless regions. As a result, the error in flat regions (such as road) remains large after refinement. On the other hand, the disparity flow consistency term $\mathcal{L}_{df}$ provides a denser signal, especially in textureless regions, and can reduce errors there. However, it still produces errors on the motion boundaries and in occluded areas, due to the output inconsistency in such regions.

**Input to Refinement Network** In the refinement network, consistency is used to guide the change of scene flow predictions. In doing so, we use the current scene flow prediction $\mathbf{X}^t$, the consistency computed on the scene flow $\mathcal{L}_{cst}(\mathbf{X})$, and its gradient as input $\mathcal{L}_{cst}(\mathbf{X})$.

In this study, we study the influence of different inputs on the final performance. For this purpose, we train several versions of the refinement network, with various combinations of input features. As shown in table 4, with only the variable as input, the refinement network achieved 42.43. By additionally including the loss and gradient as inputs, a further decrease of 0.96 and 6.12 are observed, respectively, which suggests that the gradient is a stronger signal for refinement, compared to the loss. Nevertheless, combining the two results in the best performance in scene flow, of 35.40 error rate. The results highlight the effectiveness of the gradient feature in the learned refinement network.

| Refinement Method | D1 | D2 | F1 | SF | Time |
|---|---|---|---|---|---|
| Outputs | 8.53 | 22.82 | 30.61 | 39.95 | 0.5s |
| Parameters | 6.88 | 19.47 | 27.69 | 35.49 | 120s |
| Learned (ours) | 7.10 | 19.68 | 27.35 | 35.40 | 0.8s |

**Table 5. Ablation study of online refinement strategies.** We use different online refinement strategies for scene flow estimation. All models are trained on FlyingThings3D and results are reported on KITTI.

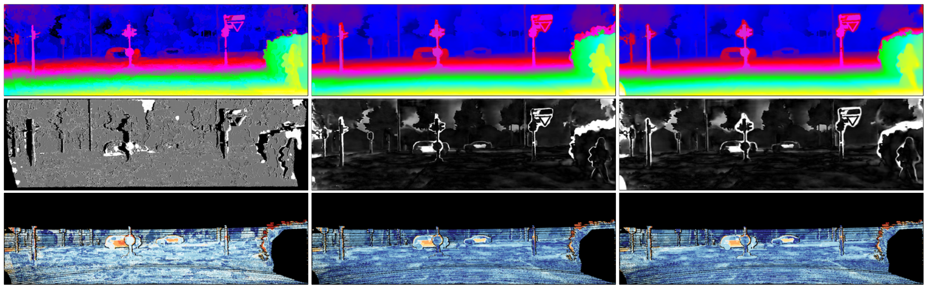

**Fig. 3. Qualitative results for different online refinement strategies.** From left to right, we show the results refined with output finetuning, parameter finetuning and our proposed learning-based refinement. The first row illustrates disparity estimation, the second the accumulated updates, where brighter means larger change in disparity. The last row shows the error map evaluated against ground-truth. Red represents higher error, and blue lower error.

**Online Refinement Strategies** In this experiment we compare our learned refinement module against alternative design choices. In this work we use a refinement network to predict an update from the loss signal, and to refine the scene flow prediction without updating the model parameters. Alternatively, one can perform online finetuning to update the model parameters, based on the same consistency loss. We refer to this approach as parameter finetuning. Another choice is to change the scene flow prediction directly, without passing through the network, which we refer to as output finetuning [6].

We test the two alternatives with the same consistency loss. The results are summarized in table 5. As seen from the table, our learned refinement achieves similar performance with parameter finetuning. However, as multiple iterations of forward and backward passes are needed for finetuning network parameters, this approach results in a large run time of 120 seconds per image. Compared to parameter finetuning, our method is much faster.

To further understand the difference among the three refinement strategies, we visualize the disparity outputs in fig. 3. We can observe from the figure, that output finetuning is very noisy and sparse – this is due to properties of each pixel being refined independently. As shown previously, the loss signal is sparse

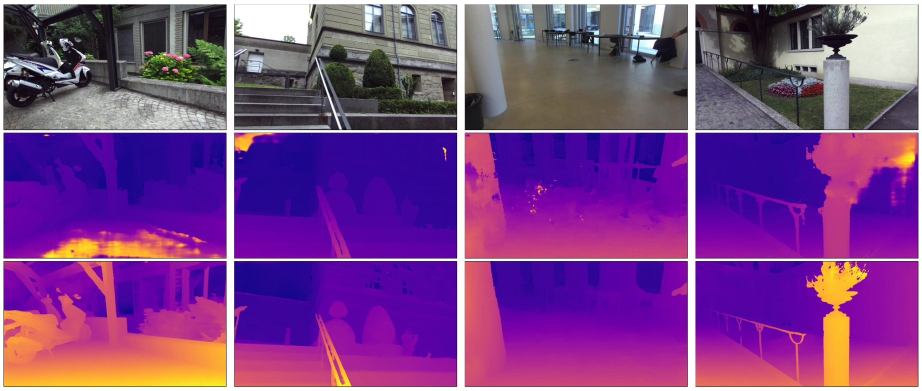

**Fig. 4. Disparity estimation results in real scenes different from training.** Top: input left image, Middle: feedforward results, Bottom: results with consistency guided refinement. Note that refinement can significantly improve drastically incorrect predictions.

and noisy, and is often invalid in occluded regions, which results in inaccurate updates. In contrast, our learned refinement can take advantage of this sparse signal and of contextual information. As a result, a much smoother refinement–similar to the one produced by parameter finetuning, but *two orders of magnitude faster*–is produced by the proposed learned refinement module.

### 4.4 Performance for Unseen Real Visual Data

To test our method's generalization in other environments, we apply the model to real scenes captured by a ZED stereo camera and compare against results from the feedforward module. As shown in fig. 4, our consistency guided refinement can notably improve the performance of disparity estimation in such cases. This again confirms that scene flow consistency can be a reliable model in overcoming domain shift, and support more robust performance in unseen scenarios.

## 5 Conclusions

We have presented Consistency Guided Scene Flow (*CGSF*), a framework for the joint estimation of disparity and scene flow from stereo video. The consistency loss combines stereo and temporal photo-consistency with a novel geometric model which couples the depth and flow variables. Besides the usual online parameter and output finetuning strategies typical of self-supervised test-time domain adaptation, we introduce new, learned, output finetuning descent models with explicit gradient features, that produce good quality results, on par with parameter finetuning, but two orders of magnitude faster. Extensive experimental validation on benchmarks indicates that *CGSF* can reliably predict disparity and scene flow, with good generalization in cross-domain settings.

## References

1. Aleotti, F., Poggi, M., Tosi, F., Mattoccia, S.: Learning end-to-end scene flow by distilling single tasks knowledge. In: AAAI (2020)
2. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. In: Advances in neural information processing systems. pp. 3981–3989 (2016)
3. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: A view centered variational approach. International journal of computer vision **101**(1), 6–21 (2013)
4. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI (2019)
5. Chang, J., Chen, Y.: Pyramid stereo matching network. In: CVPR (2018)
6. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: ICCV (2019)
7. Clark, R., Bloesch, M., Czarnowski, J., Leutenegger, S., Davison, A.J.: Ls-net: Learning to solve nonlinear least squares for monocular stereo. arXiv preprint arXiv:1809.02966 (2018)
8. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: ICCV (2015)
9. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR (2017)
10. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019)
11. Guo, X., Li, H., Yi, S., Ren, J., Wang, X.: Learning monocular depth by distilling cross-domain stereo networks. In: ECCV (2018)
12. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: ICCV (2007)
13. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
14. Ilg, E., Saikia, T., Keuper, M., Brox, T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In: ECCV (2018)
15. Jiang, H., Sun, D., Jampani, V., Lv, Z., Learned-Miller, E., Kautz, J.: Sense: A shared encoder network for scene-flow estimation. In: ICCV (2019)
16. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: ICCV (2017)
17. Kuznietsov, Y., Stuckler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: CVPR (2017)
18. Lai, H.Y., Tsai, Y.H., Chiu, W.C.: Bridging stereo matching and optical flow via spatiotemporal correspondence. In: CVPR (2019)
19. Li, K., Malik, J.: Learning to optimize. arXiv preprint arXiv:1606.01885 (2016)
20. Liu, P., Lyu, M., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: CVPR (2019)

21. Lv, Z., Dellaert, F., Rehg, J.M., Geiger, A.: Taking a deeper look at the inverse compositional algorithm. In: CVPR (2019)
22. Ma, W.C., Wang, S., Hu, R., Xiong, Y., Urtasun, R.: Deep rigid instance scene flow. In: CVPR (2019)
23. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR (2016)
24. Meister, S., Hur, J., Roth, S.: UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In: AAAI (2018)
25. Menze, M., Heipke, C., Geiger, A.: Joint 3d estimation of vehicles and scene flow. In: ISPRS Workshop on Image Sequence Analysis (ISA) (2015)
26. Menze, M., Heipke, C., Geiger, A.: Object scene flow. ISPRS Journal of Photogrammetry and Remote Sensing (JPRS) (2018)
27. Pang, J., Sun, W., Yang, C., Ren, J., Xiao, R., Zeng, J., Lin, L.: Zoom and learn: Generalizing deep stereo matching to novel domains. In: CVPR (2018)
28. Poggi, M., Pallotti, D., Tosi, F., Mattoccia, S.: Guided stereo matching. In: CVPR (2019)
29. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-net: CNNs for optical flow using pyramid, warping, and cost volume. In: CVPR (2018)
30. Tang, C., Tan, P.: Ba-net: Dense bundle adjustment network. arXiv preprint arXiv:1806.04807 (2018)
31. Tonioni, A., Poggi, M., Mattoccia, S., Di Stefano, L.: Unsupervised adaptation for deep stereo. In: ICCV (2017)
32. Tonioni, A., Rahnama, O., Joy, T., Stefano, L.D., Ajanthan, T., Torr, P.H.: Learning to adapt for stereo. In: CVPR (2019)
33. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: CVPR (2019)
34. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: ICCV (1999)
35. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: ICCV (2013)
36. Wang, Y., Wang, P., Yang, Z., Luo, C., Yang, Y., Xu, W.: Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In: CVPR (2019)
37. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
38. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: ECCV (2008)
39. Yin, Z., Shi, J.: GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In: CVPR (2018)
40. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: CVPR (2019)
41. Zhong, Y., Li, H., Dai, Y.: Open-world stereo video matching with deep rnn. In: ECCV (2018)
42. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017)