RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching

Lahav Lipson
Princeton University

Zachary Teed
Princeton University

Jia Deng
Princeton University

Abstract

We introduce RAFT-Stereo, a new deep architecture for rectified stereo based on the optical flow network RAFT [35]. We introduce multi-level convolutional GRUs, which more efficiently propagate information across the image. A modified version of RAFT-Stereo can perform accurate real-time inference. RAFT-stereo ranks first on the Middlebury leaderboard, outperforming the next best method on 1px error by 29% and outperforms all published work on the ETH3D two-view stereo benchmark. Code is available at https://github.com/princeton-vl/RAFT-Stereo.

1. Introduction

Stereo depth estimation is a fundamental vision problem with direct applications in robotics, augmented reality, photogrammetry, and video understanding problems. In the standard setup, two frames—a left frame and a right frame—are provided as input. The task is to estimate a pixelwise displacement map between the input images. In rectified stereo, the displacement of each pixel is constrained to a horizontal line. This displacement map, termed disparity, can be used alongside camera calibration parameters to recover depth, a 3D point cloud, or other 3D representations suitable for the target downstream application.

Early work has focused on two key parts of the problem: (1) feature matching and (2) regularization. Given two images, feature matching aims to compute a matching cost between a pair of image patches. Commonly used methods include mutual information [15], normalized cross-correlation [14], and the census transform followed by Hamming distance [11]. Given a set of noisy matches, regularization aims to recover a consistent depth map subject to priors such as smoothness and planarity. These two objectives can be naturally formulated as an optimization problem, maximizing some measure of visual similarity subject to priors over 3D geometry.

Optical flow and rectified stereo are closely related problems. In optical flow, the task is to predict a pixelwise displacement field, such that for every pixel in the first frame, we can estimate its correspondence in the second frame. In rectified stereo, the task is the same, except that we have the additional constraints that the x-displacement is always positive and the corresponding points lie on a horizontal line—hence, the y-displacement is always 0.

Despite the similarities between stereo and flow, neural network architectures for the two tasks are vastly different. In stereo, the predominant approach has been the use of 3D convolutional neural networks. First a 3D cost volume is built by enumerating integer disparities, then use a 3D convolutional network to filter the cost volume [43, 4, 19, 12, 47, 48]. This formulation leverages stereo geometry as an inductive prior in network design. However, using 3D convolutions to process the cost volume comes at a high computational cost and limits the possible operating resolution. Specialized approaches are required to operate at high resolutions [42] such as the mega-pixel images from the Middlebury dataset [28].

On the other hand, optical flow is typically approached using iterative refinement. RAFT [35] showed that iterative refinement can be performed entirely at high resolution, proposing a simple architecture that performed well on standard flow benchmarks. RAFT first extracts features from the input images, then builds a 4D cost volume by computing the correlation between all pairs of pixels. Finally, a GRU-based update operator iteratively updates the flow field using features retrieved from the correlation volume.

We introduce RAFT-Stereo, a new architecture for two-view stereo. An overview of our approach is shown in Fig. 1. The overall design is based on RAFT [35]. First, we replace the all-pairs 4D correlation volume with a 3D volume by only computing the visual similarly between pixels of the same height. Additionally, we introduce multi-level GRU units that maintain hidden states at multiple resolutions with cross-connections but still generate a single high-resolution disparity update. This improves the ability of the update operator to propagate information across the image, improving the global consistency of the disparity field.

RAFT-Stereo is substantially different from previous stereo networks. Existing work has commonly relied on 3D

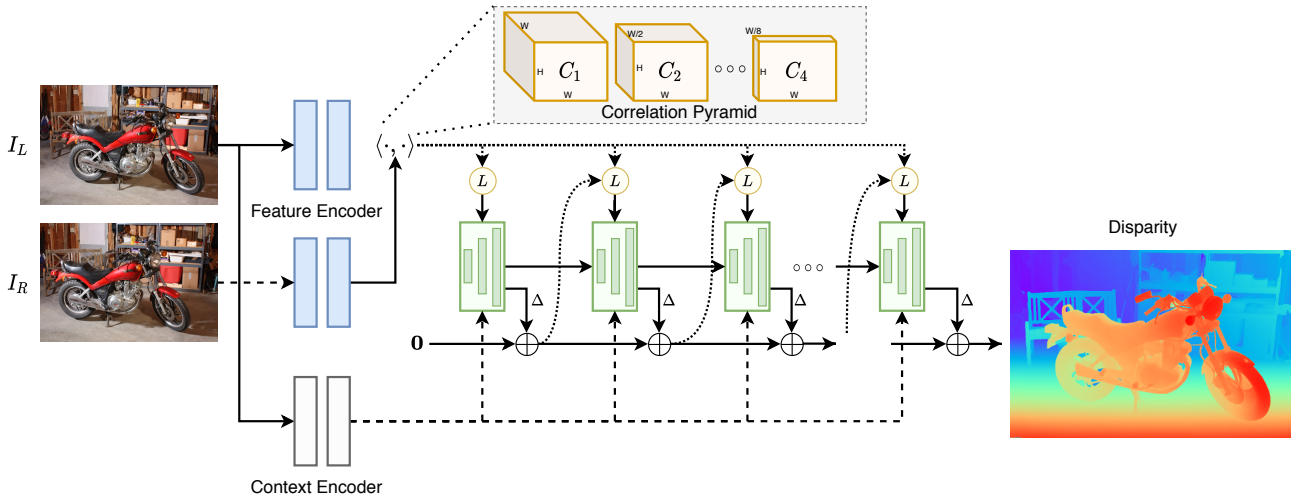

Figure 1. Correlation features (blue) are extracted from each of the images and are used to construct the correlation pyramid. "Context" image features (white) and an initial hidden state are also extracted from the context encoder. The disparity field is initialized to zero. Every iteration, the GRU(s) (green) use the current disparity estimate to sample from the correlation pyramid. The resulting correlation features, initial image features and current hidden state(s) are used by the GRU(s) to produce a new hidden state and an update to the disparity.

convolution networks to process stereo cost volumes [43, 4, 19, 12, 47, 48]. In contrast, RAFT-Stereo uses only 2D convolutions and a lightweight cost volume constructed using a single matrix multiplication. By avoiding the high computation and memory cost of 3D convolutions, RAFT-Stereo can be directly applied megapixel images without the need for resizing or processing the image in patches. Furthermore, by using an iterative network, we can easily trade accuracy for efficiency with early stopping. RAFT-Stereo also doesn't require additional complex loss terms, making it easy to train.

Our main contribution is a new stereo network which unifies stereo and optical flow approaches. RAFT-Stereo shows much better cross-dataset generalization than existing neural networks. When trained only on synthetic data, our network performs very well on real datasets such as KITTI [24], ETH3D [29], and Middlebury [28], outperforming all other works evaluated in the same setting. Additionally, RAFT-stereo is accurate. It ranks first on the Middlebury leaderboard [28] and outperforms all published work on the ETH3D leaderboard [29]. Due to its high accuracy and good generalization, we believe RAFT-Stereo will be useful as an off-the-self stereo algorithm.

2. Related Work

The task of predicting disparity between rectified stereo images is a longstanding problem in computer vision. Early work focused on designing better matching costs [13, 45] and efficient inference algorithms [20, 16, 2]. Traditional stereo pipelines generally consisted of a matching stage and a filtering stage. In the matching stage, pairwise costs were computed between images patches. In the optimization and filtering stages, priors could be imposed to correct erroneous matching and recover a consistent disparity map.

Deep learning was first applied to improve matching costs in the stereo pipeline. Žbontar and LeCun [46] proposed a network for evaluating a matching score between a pair of image matches. The matching costs were then processed using semiglobal matching, consistency checking, and filtering. Mayer et al. [23] proposed the first end-to-end trainable stereo matching network, based on the Flownet architecture [7], in addition to a large synthetic dataset which made training convolutional networks for stereo possible.

Inspired by the classical pipeline, many works have adopted a 3D neural network architecture for end-to-end stereo matching[43, 4, 19, 12, 47, 48]. GCNet[19] was one of the first papers to propose this approach. In this framework, images are first mapped through a 2D convolutional network to obtain a dense feature representation. Next, a 3D cost volume is constructed over the 2D feature maps, either through concatenation[19] or correlation operator[12]. The cost volume is then filtered through a series of 3D convolutional layers, before being mapped to a pointwise depth estimate through a differentiable arg-min operator. Many variations on this design have been proposed, such as using a stacked 3D hourglass to process the cost volume[4], or designing new aggregation layers to better propagate information[47]. The 3D convolutions aim to act as a differentiable approximation to classical filtering algorithms such as SGM[16].

While this approach has outperformed traditional methods such on datasets such as KITTI[24] and FlyingThings3D[23] the 3D convolutions come at a high computational cost and often fail to generalize outside the

domain they were trained, meaning that they cannot be readily used on datasets which don't have ground truth training data. There have been several efforts to improve the generalization ability of deep stereo networks such as the addition of new network components[48] or generating additional training data [40]. DSMNet [48] tries to improve the generalization ability of the GA-Net architecture by normalizing the features used to construct the cost volume and by utilizing a non-local graph-based filtering approach which reduces GA-Net's dependence on local patterns. DSMNet achieves better generalization than prior works, but still uses 3D convolutions in their architecture design. This results in a high computational cost and limits the operating resolution of DSMNet. These works have focused on zero-shot cross dataset generalization. In this paper, we also evaluate cross dataset generalization on the ETH3D[29], KITTI [24], and Middlebury [28] datasets.

Another line of work has looked replacing the more costly components of the 3D networks with more lightweight modules. Liang et al. [21] first proposed a 2 stage refinement network for stereo. Bi3D [1] proposed estimating depth with a series of classification stages. Recently HITNet [34] leveraged the planar geometry of the scene as an inductive prior in the network design by guiding the stereo predictions using predicted tiles. In the forward pass, HITNet's tile-based method must decide if each pixel lies on a plane. To learn this behavior, they must impose several additional loss terms on the angle of the tiles and the decision weights, as opposed to RAFT-Stereo which solely uses a standard L1 loss. HITNet also maintains a running stereo prediction at full resolution, while RAFT-Stereo only upsamples the stereo prediction at the very end. This makes RAFT-Stereo more memory efficient, enabling us to predict full-resolution stereo on megapixel images.

3. Approach

Given a pair of rectified images (I_L, I_R), we aim to estimate a disparity field $\mathbf{d}$ giving the horizontal displacement for every pixel in I_L. Similar to RAFT [35] our approach is composed of three main components: a feature extractor, a correlation pyramid, and a GRU-based update operator as shown in Fig. 1. The update operator iteratively retrieves features from the correlation pyramid and performs updates on the disparity field.

3.1. Feature Extraction

We use two separate feature extractors termed the *feature encoder* and the *context encoder*. The feature encoder is applied to both the left and right images and maps each image to a dense feature map, which is then used to construct the correlation volume. The network consists of a series of residual blocks and downsampling layers, producing feature maps at 1/4 or 1/8 the input image resolution with 256 channels, depending on the number of downsampling layers used in our experiments. We use instance normalization [37] in the feature encoder.

The context encoder has identical architecture to the feature encoder except we replace instance normalization with batch normalization [17] and only apply the context encoder on the left image. The context features are used to initialize the hidden state of the update operator and also injected into the GRU during each iteration of the update operator.

3.2. Correlation Pyramid

Correlation Volume: We use the dot product between feature vectors as a measure of visual similarity. Similar to how RAFT [35] constructs a 4D correlation volume by computing the visual similarity between all pairs of pixels, we restrict computation of the correlation volume to pixels which share the same y-coordinate. Given feature maps $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{H \times W \times D}$ extracted from I_L and I_R respectively, the 3D correlation volume can be computed using a modification of the 4D volume construction by restricting computation of the inner product to feature vectors which share the same first index:

$$\mathbf{C}_{ijk} = \sum_h \mathbf{f}_{ijh} \cdot \mathbf{g}_{ikh}, \quad \mathbf{C} \in \mathbb{R}^{H \times W \times W} \tag{1}$$

Like the 4D volume, computation of the 3D volume can be efficiently implemented using a single matrix multiplication, which can be easily computed on the GPU and takes up only a small fraction of total runtime.

In rectified stereo, we can typically assume that all disparities are positive; thus, the correlation volume really only needs to be computed for positive disparities. However, the advantage of computing the full volume is that the operation can be implemented using matrix multiplication which is highly optimized. This simplifies the overall architecture, allowing us to use common operations instead of requiring custom GPU kernels.

Correlation Pyramid: We construct a 4 level pyramid of correlation volumes through repeated average pooling of the last dimension. The k^{th} level of the pyramid is constructed from the volume at level k using 1D average pooling with a kernel size of 2 and a stride of 2 producing a new volume $\mathbf{C}^{k+1}$ with dimension $H \times W \times W/2^k$. Each level of the pyramid has an increased receptive field, but by only pooling the last dimension, we maintain the high resolution information present in the original image, which allows us to recover very fine structures.

Correlation Lookup: To index into the correlation pyramid, we define a lookup operator $L_{\mathbb{C}}$ analogous to the one defined in RAFT. Given a current estimate of disparity d, we construct a 1D grid with integer offsets around the current disparity estimate as shown in Fig. 2. The grid is used to index from each level in the correlation pyramid. Since grid

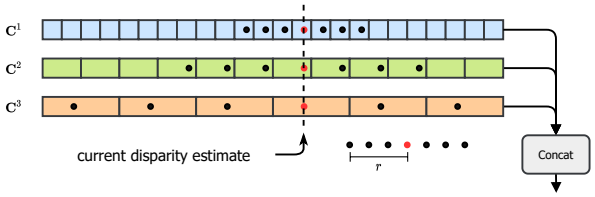

Figure 2. Lookup from the correlation pyramid. We use the current estimate of disparity to retrieve values from the each level of the correlation pyramid. We index from each level in the pyramid by linear interpolating at the current disparity estimate and at integer offsets, whose size depends on the correlation pyramid level.

values are real numbers, we use linear interpolation when indexing each volume. The retrieved values are then concatenated into a single feature map.

3.3. Multi-Level Update Operator

We predict a series of disparity fields $\{\mathbf{d}_1, ..., \mathbf{d}_N\}$ from an initial starting point $\mathbf{d}_0 = \mathbf{0}$. During each iteration, we use the current estimate of disparity to index the correlation volume, producing a set of correlation features. These features are passed through 2 convolutional layers. Similarly, the current disparity estimate is also passed through 2 convolutional layers. The correlation, disparity, and context features and then concatenated and injected into the GRU. The GRU updates the hidden state. The new hidden state is then used to predict the disparity update.

Multiple Hidden States: The original RAFT performs updates entirely at a fixed, high resolution. An issue with this approach is that the receptive field increases very slowly with the number of GRU updates. This can be problematic for scenes with large textureless regions with little local information. We combat this issue by proposing a multi-resolution update operator which operates on feature maps at 1/8, 1/16, and 1/32 resolutions simultaneously. In our experiments, we show that our use of a multi-resolution update operator results in better generalization performance.

The GRUs are *cross-connected* by using each other's hidden states as input as shown in Fig. 3. Correlation lookup and the final disparity update is performed by the GRU at the highest resolution. We also experiment with a higher resolution model, with GRU updates at 1/4, 1/8, and 1/16 the resolution of the input image.

Upsampling: The predicted disparity field is at 1/4 or 1/8 the input image resolution. To output full resolution disparity maps, we use the same convex upsampling method as RAFT. RAFT-Stereo takes the full resolution disparity values to be the convex combination of the 3x3 grid of their coarse resolution neighbors. The convex combination weights are predicted by the highest resolution GRU.

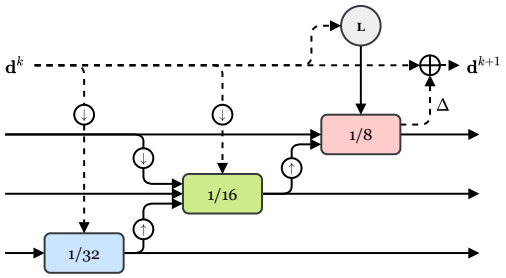

Figure 3. Multilevel GRU. We use a 3-level convolutional GRU which acts on feature maps at 1/32, 1/16, and 1/8 the input image resolution. Information is passed between GRUs at adjacent resolutions using upsampling and downsampling operations. The GRU at the highest resolution (red) performs lookups from the correlation pyramid and updates the disparity estimate.

3.4. Slow-Fast GRU

A GRU-update to a 1/8 resolution hidden state takes approximately 4x as many FLOPs compared to updating a 1/16 resolution hidden state. In order to leverage this fact for faster inference, we train a version of RAFT-Stereo in which we update the 1/16 and 1/32 resolution hidden states several times for every single update to the 1/8 resolution hidden state. On KITTI resolution images with 32 GRU updates, this simple change reduces the runtime of RAFT-Stereo from 0.132s to 0.05s, a 52% decrease. See table 6.

This modification allow us to achieve performance competitive with state-of-the-art approaches for stereo vision in real-time with RAFT-Stereo (See section 4.7), with a method that runs an order of magnitude faster.

3.5. Supervision

We supervised on the l_1 distance between the predicted and ground truth disparity over the full sequence of predictions, $\{\mathbf{d}_1, ..., \mathbf{d}_N\}$, with exponentially increasing weights. Given ground truth disparity $\mathbf{d}_{gt}$, the loss is defined as

$$\mathcal{L} = \sum_{i=1}^{N} \gamma^{N-i} ||\mathbf{d}_{gt} - \mathbf{d}_i||_1, \qquad \text{where } \gamma = 0.9. \tag{2}$$

4. Experiments

We evaluate RAFT-Stereo on ETH3D [29], Middlebury [28] and KITTI-2015 [24]. Following previous works, we pretrain our model on the synthetic Sceneflow datasets [23]. Our method achieves state-of-the-art performance on the ETH3D and Middlebury leaderboards and we outperform existing methods in the zero-shot generalization setting on ETH3D, KITTI and Middlebury.

Implementation Details: RAFT-Stereo is implemented in Pytorch [26] and is trained using two RTX 6000 GPUs. All modules are initialized from scratch with random weights. During training, we use the AdamW [22] optimizer. We evaluate RAFT-Stereo after 32 disparity-field updates in our

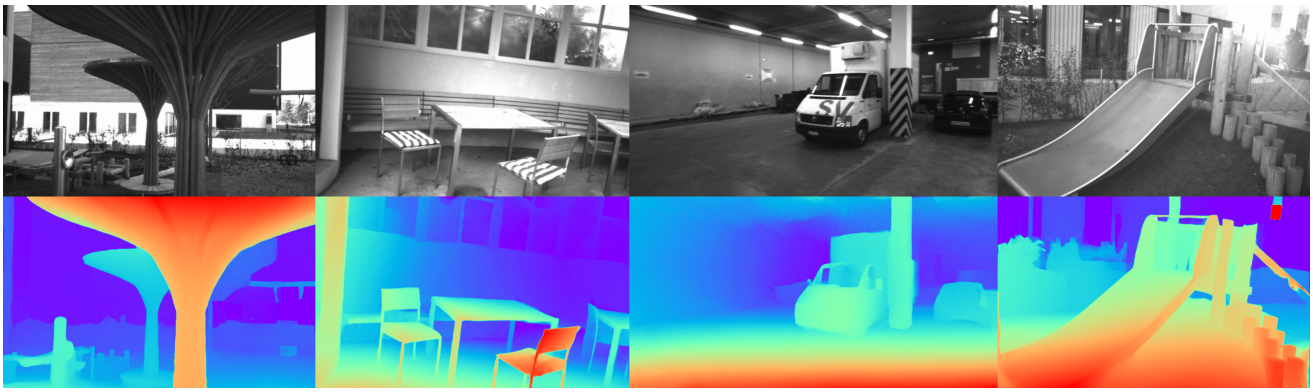

Figure 4. Results on the ETH3D stereo dataset. RAFT-Stereo is robust to difficulties like textureless surfaces and overexposure.

Method	KITTI-15	Middlebury			ETH3D
		full	half	quarter	
HD^3 [44]	26.5	50.3	37.9	20.3	54.2
gwcnet [12]	22.7	47.1	34.2	18.1	30.1
PSMNet [4]	16.3	39.5	25.1	14.2	23.8
GANet [47]	11.7	32.2	20.3	11.2	14.1
DSMNet [48]	6.5	21.8	13.8	**8.1**	6.2
Ours	**5.74**	**18.33**	**12.59**	9.36	**3.28**

Table 1. Synthetic to real generalization experiments. All methods were trained on SceneFlow[23] and tested on the KITTI-2015, Middlebury, and ETH3D validation datasets. We report average results across six independent training runs evaluated after 200k steps. Errors are the percent of pixels with end-point-error greater than the specified threshold. We use the standard evaluation thresholds: 3px for KITTI, 2px for Middlebury, 1px for ETH3D.

Method	all	foregr.	backgr.
AcfNet [49]	1.89	3.80	1.51
AMNet [10]	1.84	3.43	1.53
OptStereo [38]	1.82	3.43	1.50
GANet-deep [47]	1.81	3.46	1.48
SUW-Stereo [27]	1.80	3.45	1.47
GANet + DSMNet [48]	1.77	3.23	1.48
CSPN [5]	1.74	**2.88**	1.51
LEAStereo [6]	**1.65**	2.91	**1.40**
Ours	1.96	2.89	1.75

Table 2. Results on the KITTI-2015 [24] leaderboard. Only published results are included. Best results for each evaluation metric are bolded, second best are underlined. At the time of submission, RAFT-Stereo ranks second on the percentage of erroneous (EPE $>$ 3.0 px) foreground pixels among published methods.

ablation experiments and after 80 updates in table 1.

Training Schedule: Final models are trained on synthetic data for 200k steps with a batch size of 8, while ablation experiments are trained with a batch size of 6 for 100k steps. Ablation experiments (see table 6) are run with 16 disparity-field updates during training, and final results were trained with 22 updates. We use a one-cycle learning rate schedule [30] with a minimum learning rate of $1e^{-4}$. All RAFT-Stereo experiments were trained on random 360x720 crops (excluding benchmark submissions) and all experiments, excluding ablation experiments, were trained using data augmentation. Specifically: the image saturation was adjusted between 0 (greyscale) and 1.4; the right image was perturbed to simulate imperfect rectification that is common in datasets such as ETH3D and Middlebury; we stretch the images and disparity by random factors in the range $[2^{-0.2}, 2^{0.4}]$ in order to simulate a range of possible disparity distributions.

4.1. Zero-Shot Generalization

We evaluate RAFT-Stereo's ability to generalize from synthetic training data to unseen real-world datasets. This ability is critical as there exist no large-scale real-world datasets for training. In table 1, we report RAFT-Stereo's generalization from Sceneflow [23] directly to the KITTI-15, ETH3D and Middlebury validation sets, and compare to other methods in the same zero-shot setting.

Across all three validation datasets, RAFT-Stereo exhibits state-of-the-art performance in the zero-shot synthetic-to-real setting. RAFT-Stereo is trained for 200k iterations using data augmentation.

4.2. KITTI

We submit RAFT-Stereo to the KITTI-2015 stereo benchmark [24]. At the time of writing this paper, RAFT-Stereo ranks second on the percentage of erroneous foreground pixels on the KITTI-2015 Stereo leaderboard (See table 2) among published methods. For the KITTI leaderboard, we fine-tuned our method for 5k iterations on the KITTI training set using 320x1000 random crops, a minimum learning rate of $1e^{-5}$, and data augmentation.

4.3. ETH3D

The ETH3D dataset is too small for training, so we directly evaluate our model trained on the SceneFlow dataset. To generalize from Sceneflow to ETH3D, we simulate ETH3D's image distribution by fine tuning the network on additional greyscale Sceneflow images with gamma adjustment to simulate the often-overexposed black-and-white images in ETH3D. On the validation set, we note the accuracy increase from applying a large number of GRU iterations which can be performed without additional memory cost. To obtain our final validation results in table 1 and in table 3, we run RAFT-Stereo for 80 iterations. We show qualitative results on ETH3D in Fig. 4. Using only synthetic training data, RAFT-Stereo ranks 1st on the ETH3D two-view stereo leaderboard[29] among published methods, achieving a bad 1-pixel error (% of pixels with end-point-errors greater than 1px) of 2.44, outperforming the next best result of 2.69 by 9.3%.

4.4. Middlebury

RAFT-Stereo ranks first on the Middlebury Test set leaderboard, with a bad 2px error of 4.74%, a 26% reduction in error over the next best end-to-end deep learning method. See table 4. The Middlebury dataset provides 23 high resolution image pairs for training and/or validation, as well as versions with alternate lighting. After pre-training on Sceneflow [23], we fine-tune on 384x1000 random crops of the 23 Middlebury traning images for 4000 steps with a batch size of 2, using 22 update iterations during training, and 32 at inference.

RAFT-Stereo is extremely memory efficient, and is therefore able to output full-resolution (1900x3000) dense optical flow. This is in contrast to 33 of the remaining 34 best methods on the leaderboard, which require upsampling their output from half-resolution. To further reduce memory, we also adapt RAFT's memory efficient correlation implementation to 3D, where correlation features are computed on-the-fly. We refer the reader to section 3.2 in RAFT [35] for more information. Beyond the aforementioned horizontal image stretching, saturation adjustment and vertical perturbation of the right image, we do no additional data augmentation to adapt to the Middlebury dataset. Fig. 5 shows qualitative results of RAFT-Stereo on Middlebury.

Method	bad 0.5 (%)	bad 1.0 (%)	bad 2.0 (%)	AvgErr
HSM[42]	10.88	4.00	1.36	0.28
NOSS-ROB [18]	10.99	3.30	1.29	0.31
iResNet[25]	10.26	3.68	1.00	0.24
AdaStereo [31]	10.22	3.09	0.65	0.24
HIT-Net [34]	7.83	2.79	0.80	0.20
Ours	**7.04**	**2.44**	**0.44**	**0.18**

Table 3. Results on the ETH3D test set leaderboard. At the time of submission, RAFT-Stereo ranks first across every evaluation metric among all published methods. For all metrics, lower is better.

4.5. Synthetic Datasets

In order to improve zero-shot generalization performance, we train additional versions of RAFT-Stereo using additional synthetic data. As real-world stereo correspondence training data is difficult to obtain en masse, most stereo correspondence works such as PSMNet [4] and DSMNet [48] leverage synthetic training data, specifically only the Sceneflow dataset, for training.

The overall structure of the 3D scenes in Sceneflow, however, is not representative of other real-world datasets to which we hope to generalize. To remedy this, we investigate three additional publicly available synthetic datasets and demonstrate that combining them with Sceneflow can improve zero-shot generalization performance. We show in table 5 that certain combinations of datasets benefit generalization to specific validation datasets.

Falling Things: Falling things [36] is a photo-realistic synthetic dataset of miscellaneous objects placed sporadically around a scene. Originally intended as an object detection and 3D pose estimation dataset, Falling Things provides 61.5K image pairs for training stereo correspondence methods. We demonstrate that the use of this dataset improves generalization performance, specifically to the KITTI and Middlebury datasets.

Tartan Air: Tartan Air [39] is a publicly available photo-realistic synthetic dataset of simulation environments mod-

Methods	AvgErr	MedErr	bad 0.5 (%)	bad 1.0 (%)	bad 2.0 (%)	bad 4.0 (%)
EdgeStereo [32]	2.68	0.72	55.6	32.4	18.7	10.8
HSM-Net [8]	2.07	0.56	50.7	24.6	10.2	4.83
LEAStereo [6]	1.43	0.53	49.5	20.8	7.15	**2.75**
MC-CNN [9]	2.63	0.44	40.1	16.1	6.35	3.81
LocalExp [33]	2.24	0.43	38.7	13.9	5.43	3.69
CRLE [41]	2.25	0.42	38.1	13.4	5.75	3.90
HITNet [34]	1.71	0.40	34.2	13.3	6.46	3.81
NOSS-ROB [18]	2.08	0.42	38.2	13.2	5.01	3.46
Ours	**1.27**	**0.26**	**27.7**	**9.37**	**4.74**	**2.75**

Table 4. Results on the Middlebury test set leaderboard compared to the top performing methods. Lower is better for all metrics.

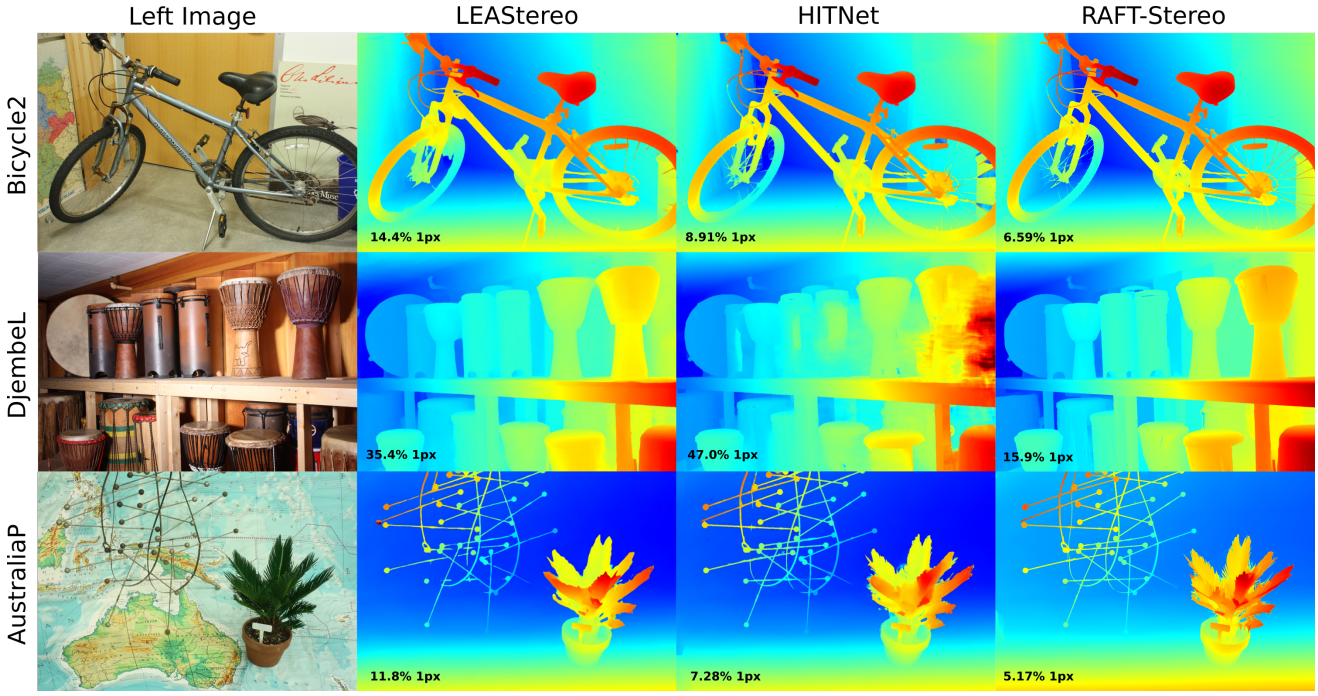

Figure 5. Results on the Middlebury [28] test set compared to the top end-to-end deep learning approaches. We also report the 1px error of each output in the corner. RAFT-Stereo is able to recover extremely fine details that other approaches cannot, such as the spokes of the bike wheel, the individual leaves of the plant, and sharp object boundaries.

Sceneflow [23]	Falling Things [36]	Tartan Air [39]	Sintel Stereo [3]	ETH3D	KITTI-15	Middlebury (Full)
✓	-	-	-	**4.44**	6.37	23.40
-	✓	-	-	26.93	6.13	25.39
-	-	✓	-	4.62	5.87	26.28
✓	✓	-	-	25.2	5.88	20.95
✓	✓	✓	-	7.65	5.76	**20.65**
✓	✓	✓	✓	5.65	**5.62**	21.99

Table 5. Synthetic data generalization experiments. All experiments were run twice with different weight initializations and the validation performances were averaged. Data were balanced so that each dataset represents an equal proportion of the training data. Experiments were done using RAFT-Stereo with a single hidden-state with random cropping and vertical perturbation of the right image.

eled after real-world settings. This dataset was intended primarily as a SLAM dataset, but also provides 296K image pairs for training stereo correspondence methods. In our experiments, we show that Tartan Air generalizes well to KITTI and to ETH3D.

Sintel-Stereo: The Sintel dataset [3] exists primarily as a synthetic dataset for training optical flow methods. In addition to optical flow training data, they also provide 2.1K image pairs for training stereo correspondence methods. While training a RAFT-Stereo exclusively on this dataset caused it to overfit, we found that leveraging Sintel-Stereo together with all three other synthetic datasets gave excellent generalization performance, specifically in that it improves generalization to the ETH3D dataset.

4.6. Ablations

GRU Levels: RAFT-Stereo maintains and updates multiple hidden states at multiple resolutions, typically 1/8, 1/16 and 1/32 resolutions as show in figure 3. Each hidden state is updated using a dedicated GRU which uses the adjacent hidden states as context in addition to specific context features for that resolution. Using multiple hidden states increases the runtime but results in better performance overall.

Backbone: RAFT-Stereo uses separate backbones in order to extract correlation features and context features for the GRU updates. We show that using a single backbone to produce both the correlation features and context features leads to faster inference without incurring any decrease in performance. We use a single-backbone architecture in the real-time version of RAFT (See section 4.7 and Fig. 6).

Resolution: RAFT-Stereo updates its running estimate of

Experiment	Method	FlyingThings3D	Runtime (s)	Parameters
# GRU Levels.	3 Levels	**9.40**	0.132	11.23M
	1 Level	9.64	**0.091**	9.46M
Backbone	Single Backbone	**9.37**	**0.121**	10.75M
	Sep. Backbones	9.40	0.132	11.23M
Resolution	1/4th	**7.92**	0.338	11.12M
	1/8th	9.40	**0.132**	11.23M
Slow-Fast GRU	Regular	**9.40**	0.132	11.23M
	Slow-Fast	9.98	**0.063**	11.23M
Collapsed Cost Volume	RAFT	-	0.224	5.26M
	RAFT-Stereo	-	**0.132**	11.23M

Table 6. Ablation experiments. Settings used in our final model are underlined. See section 4.6 for details. All experiments are run for 100k steps on random 320x720 crops of Sceneflow with vertical perturbations to the right image as the only augmentation. Methods are evaluated on the held-out FlyingThings3D [23] test set which we used to make all design decisions for our zero-shot generalization / real-time experiments. We report the 1px error evaluated after 100k steps, averaged across two independent training runs with random weight initialization. The reported the runtime comparisons were made on 1248x384 resolution images (i.e. KITTI resolution)

the disparity at 1/8 or 1/4 resolution. Maintaining the running disparity estimate at 1/4 resolution yields significantly better generalization, but results in slower runtimes and uses approximately 4x as much GPU memory. This is done by shrinking the stride in the feature extractors and by predicting a proportionally smaller mask for convex upsampling.

Collapsed Cost Volume: Rather than training a separate network for estimating stereo correspondence, one option is to apply an existing optical flow method and project the predicted flow onto the epipolar line. We show that specializing RAFT for stereo by simply collapsing the cost volume gives significantly faster runtime relative to RAFT.

Slow-Fast: We observe a significant decrease in runtime of RAFT-Stereo by iterating the lower resolution GRUs more often and the higher resolution GRUs less often, with a limited penalty to accuracy. In table 6, the "Slow-Fast" version of RAFT-Stereo updates the lowest, middle and highest resolution hidden states 30, 20 and 10 times, respectively, while "Regular" updates each hidden state 32 times. Both "Slow-Fast" and "Regular" use the same model weights

4.7. Real-time Inference

We demonstrate that RAFT-Stereo can be configured to achieve real-time inference on KITTI-resolution (1248x384) images with competitive performance. By leveraging Slow-Fast bi-level (1/8 and 1/16 resolution) GRUs and a single backbone, RAFT-Stereo runs at 26 FPS. Our real-time implementation of RAFT-Stereo's performance (5.91 D1 error) is competitive with DSMNet [48] (6.5 D1 error). See Fig. 6. Additionally, we implement our own bilinear sampler in CUDA as Pytorch's default implementation proved to be a runtime bottleneck.

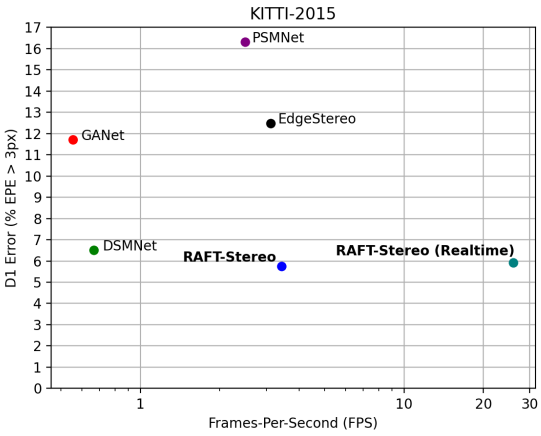

Figure 6. Plot comparing zero-shot generalization from synthetic data to KITTI-2015. All methods are trained only on Sceneflow [23], without any fine-tuning. RAFT-Stereo can be configured for real time inference and achieves competitive performance with the state-of-the-art stereo methods. Relative to our base model (blue), our realtime model (teal) uses a shared backbone, two hidden state resolutions and slow-fast GRUs updating the flow-field at 1/8th resolution (Sec. 4.7)

5. Conclusions

We have proposed RAFT-Stereo, a new deep architecture for two-view Stereo based on RAFT [35]. RAFT-Stereo extends RAFT by leveraging multi-level GRUs to efficiently pass information across the image. Our approach achieves state-of-the-art cross-dataset generalization and ranks first on the Middlebury benchmark and outperforms all published work on ETH3D.

Acknowledgements This work is partially supported by the National Science Foundation under Award IIS-1942981.

References

[1] Abhishek Badki, Alejandro Troccoli, Kihwan Kim, Jan Kautz, Pradeep Sen, and Orazio Gallo. Bi3d: Stereo depth estimation via binary classifications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1600–1608, 2020. 3

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 2

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 7

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 1, 2, 5, 6

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 5

[6] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33, 2020. 5, 6

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 2

[8] Sébastien Drouyer, Serge Beucher, Michel Bilodeau, Maxime Moreaud, and Loïc Sorbier. Sparse stereo disparity map densification using hierarchical image segmentation. In *International symposium on mathematical morphology and its applications to signal and image processing*, pages 172–184. Springer, 2017. 6

[9] Sébastien Drouyer, Serge Beucher, Michel Bilodeau, Maxime Moreaud, and Loïc Sorbier. Sparse stereo disparity map densification using hierarchical image segmentation. In *International symposium on mathematical morphology and its applications to signal and image processing*, pages 172–184. Springer, 2017. 6

[10] Xianzhi Du, Mostafa El-Khamy, and Jungwon Lee. Amnet: Deep atrous multiscale stereo disparity estimation networks. *arXiv preprint arXiv:1904.09099*, 2019. 5

[11] Wade S Fife and James K Archibald. Improved census transforms for resource-optimized stereo vision. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):60–73, 2012. 1

[12] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 1, 2, 5

[13] Marsha J Hannah. Computer matching of areas in stereo images. Technical report, Stanford Univ Ca Dept of Computer Science, 1974. 2

[14] Yong Seok Heo, Kyong Mu Lee, and Sang Uk Lee. Robust stereo matching using adaptive normalized cross-correlation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):807–822, 2010. 1

[15] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 1

[16] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 2

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 3

[18] Penglei Ji, Jie Li, Hanchao Li, and Xinguo Liu. Superpixel alpha-expansion and normal adjustment for stereo matching. *Journal of Visual Communication and Image Representation*, page 103238, 2021. 6

[19] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017. 1, 2

[20] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–1583, 2006. 2

[21] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2811–2820, 2018. 3

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[23] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. 2, 4, 5, 6, 7, 8

[24] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 2, 3, 4, 5

[25] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 887–895, 2017. 6

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[27] Haoyu Ren, Aman Raj, Mostafa El-Khamy, and Jungwon Lee. Suw-learn: Joint supervised, unsupervised, weakly supervised deep learning for monocular depth estimation. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 750–751, 2020. 5

[28] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 1, 2, 3, 4, 7

[29] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 4, 6

[30] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018. 5

[31] Xiao Song, Guorun Yang, Xinge Zhu, Hui Zhou, Zhe Wang, and Jianping Shi. Adastereo: A simple and efficient approach for adaptive stereo matching, 2020. 6

[32] Xiao Song, Xu Zhao, Liangji Fang, Hanwen Hu, and Yizhou Yu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *International Journal of Computer Vision*, 128(4):910–930, 2020. 6

[33] Tatsunori Taniai, Yasuyuki Matsushita, Yoichi Sato, and Takeshi Naemura. Continuous 3d label stereo matching using local expansion moves. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2725–2739, 2017. 6

[34] Vladimir Tankovich, Christian Häne, Sean Fanello, Yinda Zhang, Shahram Izadi, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. *arXiv preprint arXiv:2007.12140*, 2020. 3, 6

[35] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *arXiv preprint arXiv:2003.12039*, 2020. 1, 3, 6, 8

[36] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation. *CoRR*, abs/1804.06534, 2018. 6, 7

[37] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[38] Hengli Wang, Rui Fan, Peide Cai, and Ming Liu. Pvstereo: Pyramid voting module for end-to-end self-supervised stereo matching, 2021. 5

[39] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020. 6, 7

[40] Jamie Watson, Oisin Mac Aodha, Daniyar Turmukhambetov, Gabriel J Brostow, and Michael Firman. Learning stereo from single images. *arXiv preprint arXiv:2008.01484*, 2020. 3

[41] Huaiyuan Xu, Xiaodong Chen, Haitao Liang, Siyu Ren, Yi Wang, and Huaiyu Cai. Crosspatch-based rolling label expansion for dense stereo matching. *IEEE Access*, 8:63470–63481, 2020. 6

[42] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019. 1, 6

[43] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo, 2018. 1, 2

[44] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6044–6053, 2019. 5

[45] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994. 2

[46] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015. 2

[47] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 1, 2, 5

[48] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. *arXiv preprint arXiv:1911.13287*, 2019. 1, 2, 3, 5, 6, 8

[49] Youmin Zhang, Yimin Chen, Xiao Bai, Suihanjin Yu, Kun Yu, Zhiwei Li, and Kuiyuan Yang. Adaptive unimodal cost volume filtering for deep stereo matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12926–12934, 2020. 5