Supplementary Materials for Separable Flow

1. Architecture Details

1.1. Feature & Context Net

Features are extracted from the input images using a convolutional network. We use the feature encoder network from RAFT [7]. The network is applied to both I_1 and I_2 and maps the input images to dense feature maps at a lower resolution. As detailed in Table 1, our encoder outputs features at 1/8 resolution, $F \in \mathbb{R}^{H\times W\times D}$ where H and W are one eighth the height and width of the input image, and we set $D = 256$. The feature encoder consists of 6 residual blocks, 2 at 1/2 resolution, 2 at 1/4 resolution, and 2 at 1/8 resolution.

We additionally use a context network, again identical to the one used in RAFT [7]. The context network extracts features only from the first view I_1. The architecture of the context network is similar to the feature extraction network. The only differences are that the feature encoder uses instance normalization while the context encoder uses batch normalization, and the context network also outputs the weight $w_1, ..., w_4$ for the cost aggregation net.

1.2. Cost Aggregation Net

Table 1 presents the details of the parameters of the matching cost aggregation net that is used in experiments to produce state-of-the-art accuracy on Sintel [1] and KITTI [5] benchmarks. It has four SGA layers and eight 3D convolutional layers for cost aggregation. Two cost aggregation networks are used, one for horizontal motion u, and one for vertical motion v, with no weight sharing.

1.3. Refinement & Upsampling Net

We use the recurrent transformer proposed in RAFT [7] as the the upsampling & refinement block. It takes in context features, cost/correlation features (lookup from the cost volume or correlation volume), and the current estimate of flow (initially computed by the motion regression) to update the latent hidden state. The updated hidden state is used to predict the flow update.

We made two improvements to the refinement block of RAFT [7]. First, we initialize the flow with our motion regression results; the original RAFT [7] initializes flow with zero motion. The initialization with motion regression can improve both the convergence speed and the prediction accuracy. Second, for the correlation/cost feature, we lookup the cost values from the two aggregated cost volumes, C_u and C_v, output by the cost aggregation block. The features are then concatenated along with the original C (without aggregation) to construct the cost/correlation features $[C_u(u), C_v(v), C(u,v)]$ (where (u,v) are the candidate lookup set in each refinement iteration [7].)

2. Training Details

Table 2 presents the details of the training schedule and settings used with each dataset. We follow RAFT [7] for the data augmentation.

Photometric Augmentation: We randomly perturb brightness, contrast, saturation, and hue. We use the Torchvision `ColorJitter` with brightness 0.4, contrast 0.4, saturation 0.4, and hue $0.5/\pi$. On KITTI, we reduce the degree of augmentation to brightness 0.3, contrast 0.3, saturation 0.3, and hue $0.3/\pi$. Photometric augmentation is performed on each image randomly with probability 0.2.

Spatial Augmentation: We randomly rescale and stretch the images. The degree of random scaling depends on the dataset. For FlyingChairs, we perform spatial augmentation in the range $2^{[-0.2,1.0]}$, FlyingThings $2^{[-0.4,0.8]}$, Sintel $2^{[-0.2,0.6]}$, and KITTI $2^{[-0.2,0.4]}$. Spatial augmentation is performed with probability 0.8.

No.	Layer Description	Output Tensor
	Feature & Context Net	
input	image view	H×W×3
1	7×7 conv, stride 2	$1/2$H×$1/2$W×64
2	Residual Block	$1/2$H×$1/2$W×64
3	Residual Block, stride 2	$1/4$H×$1/4$W×96
4	Residual Block, stride 2	$1/8$H×$1/8$W×128
output	3×3 conv	$1/8$H×$1/8$W×256
weight w_1	from 4: 3×3 conv, 1×1 conv, reshape	$1/8$H×$1/8$W×10×20
weight w_2	from 4: 3×3, stride 2, conv, 1×1 conv, reshape	$1/16$H×$1/16$W×20×20
weight w_3	from 4: 3×3, stride 2, conv, 1×1 conv, reshape	$1/16$H×$1/16$W×20×20
weight w_4	from 4: 3×3 conv,1×1 conv, reshape	$1/8$H×$1/8$W×10×20
	Cost Aggregation	
input	cost volume	$1/8$H×$1/8$W×$1/8|U|$×4
[1]	3×3×3, 3D conv	$1/8$H×$1/8$W×$1/8|U|$×10
[2]	SGA layer [10]: weight from w_1	$1/8$H×$1/8$W×$1/8|U|$×10
[3]	concatenate [1,2], 3×3×3, 3D conv	$1/8$H×$1/8$W×$1/8|U|$×10
[4]	3×3×3, 3D conv, stride 2	$1/16$H×$1/16$W×$1/16|U|$×20
[5]	SGA layer [10]: weight from w_2	$1/16$H×$1/16$W×$1/16|U|$×20
[6]	concatenate [4,5], 3×3×3, 3D conv	$1/16$H×$1/16$W×$1/16|U|$×20
[7]	SGA layer [10]: weight from w_3	$1/16$H×$1/16$W×$1/16|U|$×20
[8]	concatenate [6,7], 3×3×3, 3D conv	$1/16$H×$1/16$W×$1/16|U|$×20
[9]	3×3×3, 3D deconv, stride 2	$1/8$H×$1/8$W×$1/8|U|$×10
[10]	SGA layer [10]: weight from w_4	$1/8$H×$1/8$W×$1/8|U|$×10
[11]	concatenate [3,9,10], 3×3×3, 3D conv	$1/8$H×$1/8$W×$1/8|U|$×10
[12]	3×3×3, 3D to 2D conv, output cost volume	$1/8$H×$1/8$W×$1/8|U|$
regression	softmax, motion regression	$1/8$H×$1/8$W×1

Table 1: Parameters of the feature extraction and cost aggregation blocks

Stage	Weights	Training Data	Learning Rate	Batch Size	Weight Decay	Crop Size	Iterations
Chairs	-	C	4e-4	12	1e-4	[368, 496]	100k
Things	Chairs	T	1.25e-4	6	1e-4	[400, 720]	100k
Sintel	Things	S+T+K+H	1.25e-4	6	1e-5	[368, 768]	100k
KITTI	Sintel	K	1e-4	6	1e-5	[288, 960]	50k

Table 2: Details of the training schedule. Dataset abbreviations: C: FlyingChairs, T: FlyingThings, S: Sintel, K: KITTI-2015, H: HD1K. During the Sintel Finetuning phase, the dataset distribution is S(.71), T(.135), K(.135), H(.02).

Occlusion Augmentation: Following HSM-Net [9], we also randomly erase rectangular regions in I_2 with probability 0.5 to simulate occlusions.

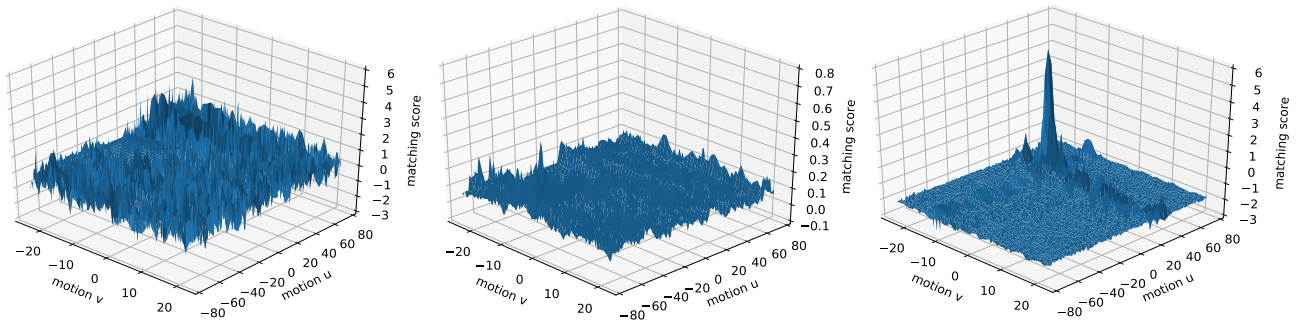
3. Computational Complexity

We measure the computational complexity of our model in terms of FLOPs (floating-point operations of multiply-adds). Using KITTI resolution (1242×375) as input, the state-of-the-art RAFT has a complexity of ***4.5T*** (Trillion) FLOPs. As a comparison, the number of FLOPs in our model is ***4.6T*** which is only ***2%*** larger than the RAFT baseline.

4. More Results

4.1. Challenging Regions

Table 3 evaluates the performance of the models in challenging regions/objects (using occlusion mask and car mask in KITTI dataset for evaluations). In the occluded regions, the improvement (2.3%) is 9.5× as much per pixel as that in the



(a) Cost Volume of RAFT [7] (b) Cost Volume of SGM [8] (c) Cost Volume of Ours

Figure 1: Cost volume visualization with respect to displacements (u, v), for a pixel selected from reflections in a car window. The matching cost distribution output by state-of-the-art RAFT [7] (a), has many false peaks due to the influence of the ambiguities (reflections) and lack of non-local aggregation. The wrong matches can easily have higher matching score than the correct one. We also compare with the cost volume after direct SGM processing [8] (b). We find that our Separable Flow (c) can learn a better cost volume, with sharp peaks at the ground truth motions.

Models	Non-occ bg (83%)	Occ (12%)	Cars (5%)	All
RAFT	3.07	20.1	7.07	5.36
Ours	2.83 (0.24↑)	17.8 (2.3↑)	6.34 (0.73↑)	4.85 (0.51↑)

Table 3: Evaluations in challenging occlusions and foreground objects (cars).

non-occluded background (0.24%). And in the challenging objects (*e.g.* cars with reflections and large smoothness regions), the improvement is 3× that of the background.

4.2. Synthetic to Real

We train our model on the synthetic data (FlyingChairs [4], FlyingThings [6] and VKITTI [2]), and test on the real KITTI scenes. Examples of results are presented in Fig. 2.

4.3. High-resolution Predictions

To demonstrate that our method scales well to scenes of very high resolution, we apply our network to HD video from the Cityscapes [3] dataset. With the model trained on synthetic data, we test the performance on 1024×2048 resolution Cityscapes images/videos. Fig. 3 visualizes example results on Cityscapes.

References

[1] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 611–625. Springer, 2012. 1

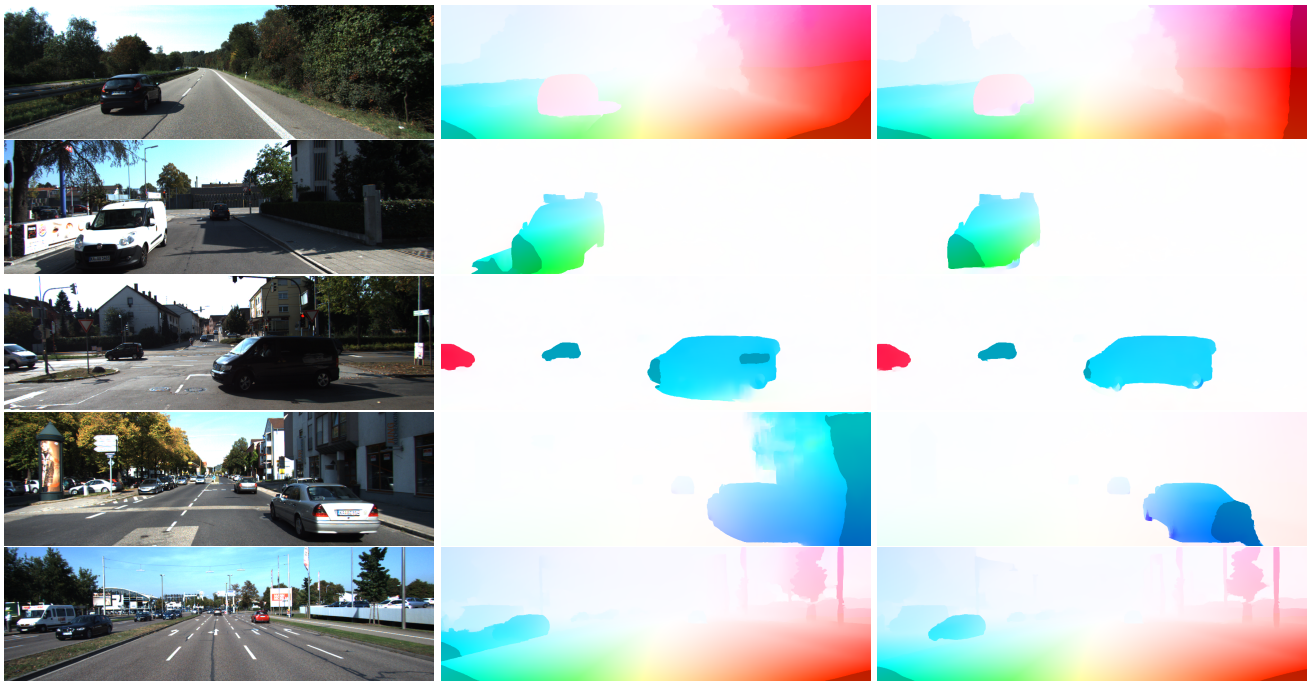
[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 3

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 3

[4] 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. 3

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[6] Nikolaus Mayer, Eddy Ilg, Philip Hausser, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 3



(a) Input views (b) RAFT [7] (c) Ours

Figure 2: Qualitative comparisons. The models are trained on the synthetic data and tested on the real KITTI scenes. (b) Results of the state-of-the-art RAFT [7]. (c) Results of our Separable Flow. Our Separable Flow performs better in synthetic-to-real generalization.

[7] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020. 1, 3, 4

[8] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1289–1297, 2017. 3

[9] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019. 2

[10] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 2

(a) Input views (b) Optical Flow Predictions

Figure 3: Results on high-resolution videos. The model is trained on the synthetic data and tested on the high-resolution Cityscapes scenes.