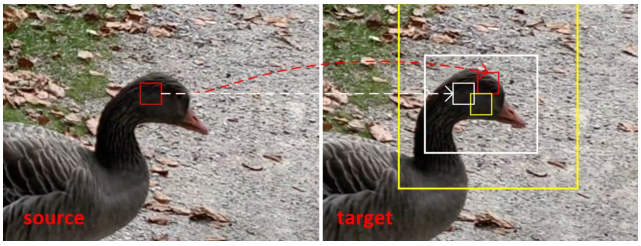
Supplementary Material

1. Patchmatch in Flow

The traditional Patchmatch methods [1] consists of three components: Random Initialization, Propagation and Random Search.

(a) Random Initialization

(b) Propagation



(c) Random Search

(d) Local Search

Figure 1. A toy example for the Patchmatch in flow

In the initialization stage, the flow is initialized either randomly or based on some prior information. A toy example for this stage is shown in Fig. 1a, the flow is initialized randomly. So for a patch represented by the red box with its 4 neighbors represented by the white, blue, yellow and green box respectively in the source image, the random flow relation can be represented as the dotted arrows to the target patches. That is to say, the red box in the source image moves to the red box in the target image with a random flow. In DIP, the flow is initialized randomly at the begining and after getting the flow at a 1/16 resolution, we use it as an initial flow at the 1/4 stage.

In the propagation stage, every patch compares the costs of its own flow with that of its neighbors and updates them if the flow of its neighbors lead to a lower cost. As the Fig. 1b shows, after the initialization, for the red box, the flows from itself and its neighbors will be used to compute 5 correlation volume, and it is obvious that the flow candidate from the yellow box results in the maxmium correlation. So the flow of the red box will be update to the flow from the yellow box. In order to make the propagation stage friendly to the end-to-end pipeline, we shift the flow map toward the 4 neighbors(top-left, top-right, bottom-left, botton-right) so that we can use the flow from the 4 neighbors to compute the corresponding correlation by a vectorization operator. For example, when shifting the flow to the down-right, the point(1,1) will get the flow of point(0,0), the correlation at point(1,1) actually is computed by the flow at point(0,0). After shifting 4 times, we can get 5 correlation coefficients for point(1, 1) based on the flow from point(1, 1), (0,0), (0,2), (2,0), (2,2). Then we can choose the best flow for point(1, 1) according to correlation volume.

The random search step is an essential step to make Patchmatch work. Propagation can converge very quickly but often end up in a local minimum. So it is necessary to introduce new information into the pipeline. In the random search stage, it is achieved by selecting a flow candidate randomly from an interval, whose length decreases exponentially with respect to the number of searches. Just like the Fig. 1b shows, the flow of the red box is updated and is closer to the good match, but it is not the best match. So it is necessary to add the random search stage to get more flow candidates further. As the Fig. 1c shows, the candidates can be searched in the target image by a binary random search method. Centered on the red box, the first random search will be done within the big yellow box whose radius is min(imagewidth/2, imageheight/2), and the better

match can be found at the small yellow box(if the small yellow box gets a worse match, the flow won't be updated). So the next random search will be done centered with the small yellow box within the big white box, and luckily the random search gets the small white box which is much better than the small yellow box and is extremely close to the best match. So after this stage, the flow for the red box is updated to the motion with the small white box which is represented by the white dotted arrows. However, random search is not friendly to the deep learning pipeline. So we replace this stage with a local search method, which aggregates the flow candidates from a 5x5 windows on the 1/16 resolution coarsely and the 1/4 resolution finely. It can be also represented by a toy example shown as the Fig. 1d, the good match can be found by aggregrating within the yellow box. And experiments also confirm that this alternative works well.

It is recommend to refer the work [9], they make a good summary of Patchmatch and application to stereo task.

2. Domain-invariance in Stereo Matching

In this supplementary document, we first applied DIP to Stereo to demonstrate the portability. The core of the stereo matching algorithm is to obtain a dense disparity map of a pair of rectified stereo images, where disparity refers to the horizontal relationship between a pair of corresponding pixels on the left and right images. Optical flow and stereo are closely related problems. The difference is that optical flow predicts the displacement of the pixel in the plane, while stereo only needs to estimate the displacement of the pixel in a horizontal line. Therefore, we improved the local search block in DIP to make it more relevant to stereo task. Specifically, we reduced the search range of local search block from 2D search to 1D search. The entire local search block for Stereo is shown in Fig. 2.

In the main paper we have proved that inverse patchmatch and local search in optical flow not only obtain high-precision results but also have strong domain-invariance. In the stereo matching experiments, we follow the training strategy of DSMNet [17], which is to train only on the Sceneflow dataset [10], and other real datasets (such as Kitti [5, 11], Middlebury [12], and ETH3D [13]) are used to evaluate the cross-domain generalization ability of the network. Before training, the input images are randomly cropped to 384 × 768, and the pixel intensity is normalized to -1 and 1. We train the model on the Sceneflow dataset for 160K steps with a OneCycle learning rate schedule of initial learning rate is 0.0004.

Domain-invariance ability The domain-invariance is an ability that generalizes to unseen data without training. In Tab. 1, we compare our DIP with other state-of-the-art deep neural network models on the four unseen real-world datasets. All the models are trained on SceneFlow data. On the KITTI and ETH3D dataset our result far outperforms the previous methods. In the Middlebury dataset, our results only lag behind DSMNet better than all the other methods. Compared to DIP-Flow, DIP-Stereo has more domain-invariance capability, which indicates that our proposed local search block for Stereo is effective in handling Stereo tasks.

Models	KITTI		Middlebury		ETH3D
	2012	2015	half	quarter	
CostFilter [8]	21.7	18.9	40.5	17.6	31.1
PatchMatch [2]	20.1	17.2	38.6	16.1	24.1
SGM [7]	7.1	7.6	25.2	10.7	12.9
Training set	SceneFlow				
HD3 [15]	23.6	26.5	37.9	20.3	54.2
PSMNet [4]	15.1	16.3	25.1	14.2	23.8
Gwcnet [6]	12.5	12.6	34.2	18.1	30.1
GANet [16]	10.1	11.7	20.3	11.2	14.1
DSMNet [17]	6.2	6.5	**13.8**	**8.1**	6.2
CFNet [14]	**4.7**	5.8	21.2	13.1	5.8
Ours-Flow	5.6	5.7	17.2	10.6	5.5
Ours-Stereo	4.9	**4.9**	14.9	8.8	**3.3**

Table 1. Comparing with other advanced methods on KITTI, Middlebury and ETH3D training sets. All methods were trained on SceneFlow. 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.

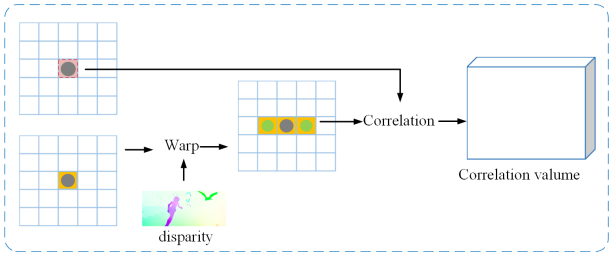


Figure 2. Local Search block for Stereo.

3. Adaptive Layers

Because DIP uses the same process and parameters for each pyramid, we can define any pyramid layers to make predictions, instead of using only two layers pyramid as we trained. Experiments show that when multilayer pyramid prediction is used, a more accurate optical flow can be ob-

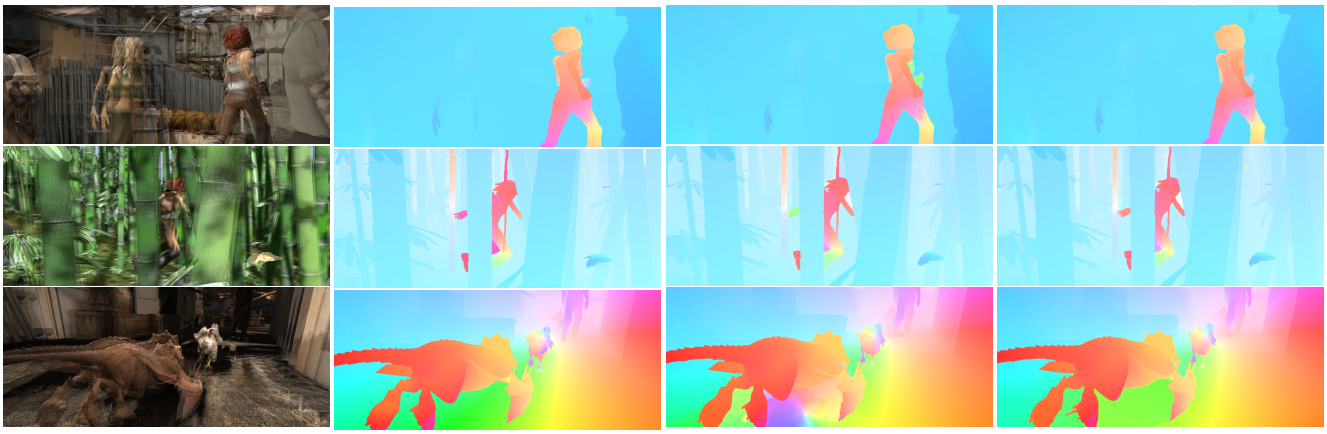


Figure 3. Results compare between fixed two layers and adaptive layers. The two-level pyramid adopts a strategy from 1/16 to 1/4 resolution. The adaptive way adaptively selects the initial resolution according to the initial optical flow, such as 1/16, 1/8, or 1/4 initial resolution.

Figure 4. Comparison of results between normal scenes and motion blur scenes. Motion blur causes incorrect optical flow estimation.

tained. Especially for continuous optical flow prediction, the adaptive pyramid layers can be used to obtain better results.

DIP supports initializing optical flow input. In the optical flow prediction of consecutive frames of video, we can take the forward interpolation of the previous result as the initialization input of the current frame. If the maximum displacement of the initialized optical flow is large, the motion of the current frame may also be large, at which point we need to start from a low-resolution layer. And to ensure accuracy, the sampling rate of the pyramid is 2 instead of 4. If previous displacement is very small, the motion of the current frame may also be small, at which point we need only one layer of pyramid prediction. Fig. 3 shows the comparison between the two-layers pyramid and the adaptive layers pyramid, and both initialize using the "warm-start" strategy.

4. More Results on High-Resolution

To verify the robustness of optical flow in different high-resolution real-world scenes, we first tested DIP on the free used public dataset[1] with the resolution of 1080×1920 and showed results in Fig. 5. Then, we further used our mobile phone to collect images with a larger resolution(1536×2048) for testing and showed results in Fig. 6. Experiments show that even if only virtual data is used for training, DIP still shows strong detail retention ability in high-resolution real-world scenes, which further confirms the strong cross-dataset generalization ability of DIP.

[1] https://www.pexels.com/videos/

5. Limitations

In the main paper, we observe that DIP is very friendly to the situations on fine-structure motions in the Sintel [3] clean dataset (such as the person in the palace). However, a special weakness of our method is dealing with blurry regions, which is due to the limitations of neighborhood propagation of DIP. The entropy of the propagated information is greatly reduced when the features of the neighborhood are blurred, which leads to a weakening of the overall optical flow quality. An incorrect case is shown in Fig. 4. In the Sintel Clean images, DIP is able to estimate the optical flow that takes into account details and large displacement. However, in strong motion blur scenes of Sintel Final data, the propagation of incorrectly matched information in the neighborhood leads to incorrect predictions. In order to solve such problems, a non-local attention mechanism will be introduced in the further works.

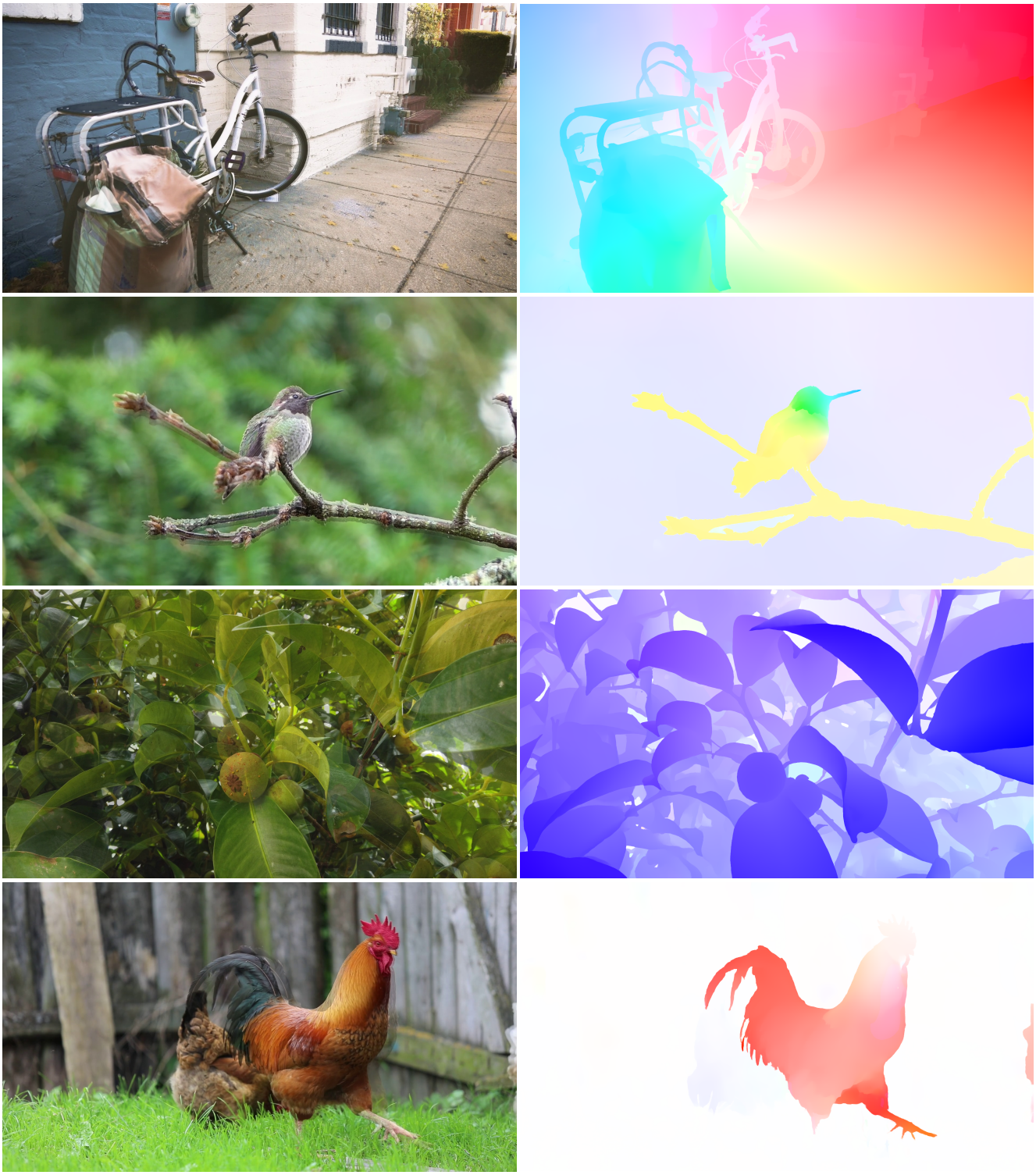


Image Overlay Optical Flow

Figure 5. High-resolution optical flow results on public real-world images. The test resolution is 1080×1920

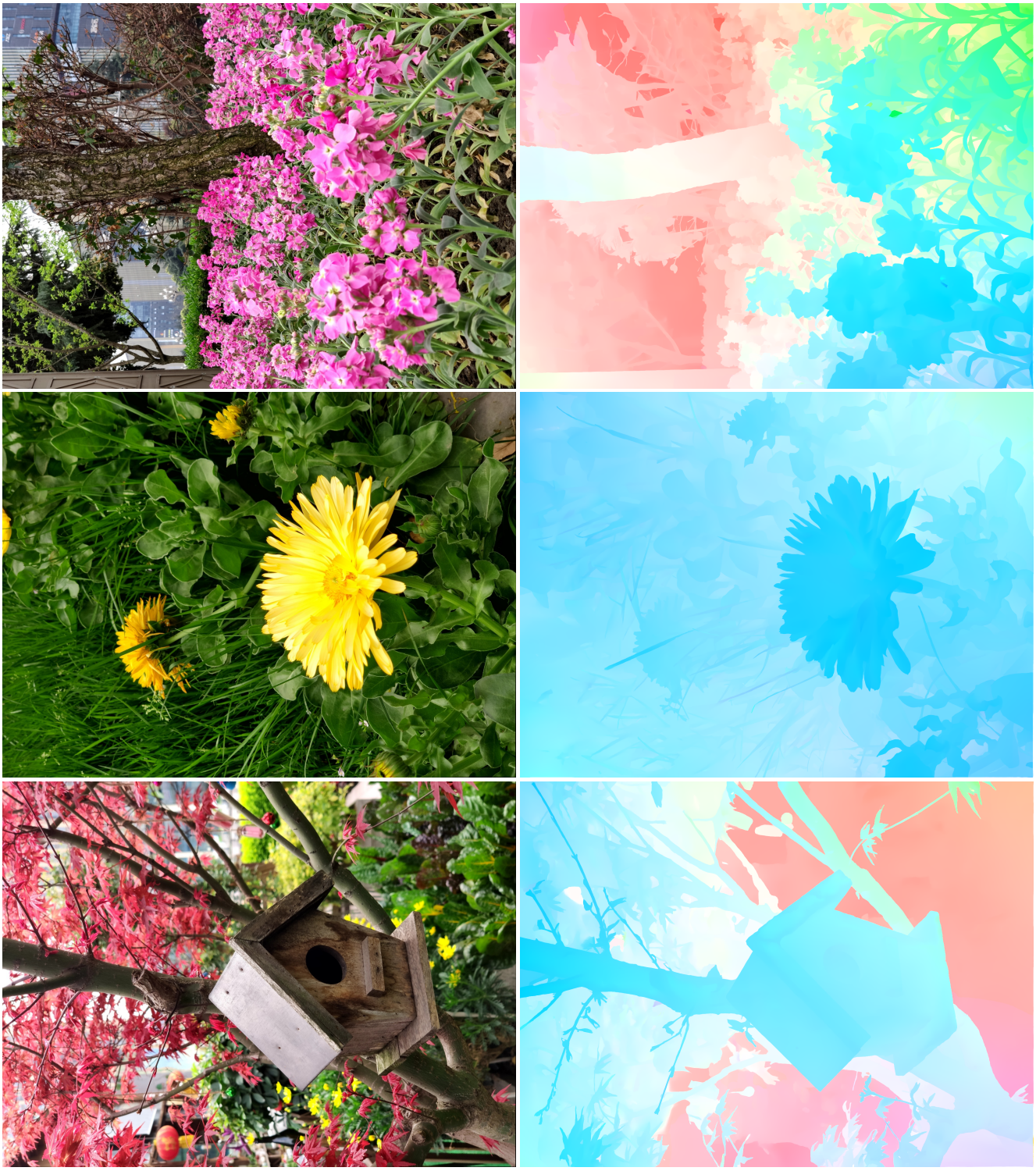


Image1 Optical Flow

Figure 6. High-resolution optical flow results on self-captured images. The test resolution is 1536×2048

References

[1] 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. 1

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011. 2

[3] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 3

[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. 2

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2

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

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

[8] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2012. 2

[9] Fangjun Kuang. Patchmatch algorithms for motion estimation and stereo reconstruction. Master's thesis, 2017. 2

[10] 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. 2

[11] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 2

[12] 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. 2

[13] Thomas Schops, Johannes L Schonberger, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017. 2

[14] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13906–13915, 2021. 2

[15] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6044–6053, 2019. 2

[16] 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

[17] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. In *European Conference on Computer Vision*, pages 420–439. Springer, 2020. 2