Supplemental Materials for PatchFlow

Ahmed Alhawwary, Janne Mustaniemi, and Janne Heikkilä

Center for Machine Vision and Signal Analysis, University of Oulu, Finland
{ahmed.alhawwary,janne.mustaniemi,janne.heikkila}@oulu.fi

1 Additional Training Details

Similar to the training process with the FlyingChairs dataset [1], in the beginning, we finetune the first stage alone with the FlyingThings3D dataset [4], and then we initialize the weights of the second stage with the first. After that, the whole pipeline is finetuned while freezing the first stage's weights. This is different from the training with FlyingChairs where the first stage's weights are not frozen when the two stages are trained in an end-to-end manner. Following [2], we used the FlyingThing3D subset where some extremely hard samples from the FlyingThings3D dataset are omitted. The images are cropped to a size of 512×768. The learning rate is set to $1e-5$ and decreased by 0.5 at 19, 30 and 40 epochs. We finetune for 50 epochs with a batch of size 32 samples on four 32GB V100 GPUs. We used the PyTorch framework [5] for our implementation.

2 FLOPs Computations

The number of floating-point operations (FLOPs) considers only convolution, multiplication and addition operations. We use the `profile` function from PyTorch [5] for this purpose.

3 RealP40 Dataset

The 2K videos captured by the Huawei P40 Pro phone are available in the following anonymous Google Drive link https://drive.google.com/drive/folders/1ypqfPebN1I7Pwd9TFo5U0Fu1oozIxnvX?usp=sharing.

In Addition, we provide a demonstration video that includes a qualitative comparison between our pipeline (PatchFlow) and FastFlowNet (FFN) [3] based on two 4K videos captured with the same phone. For visualizing the alignment or the motion between two frames, we overlay the first frame over the second frame by replacing the red colour channel of the latter with its counterpart from the first frame. For each video, we first visualize the alignment of the video frames using both methods. We refer to the overlay of the two frames before they are aligned using the predicted optical flows with the word 'Before' in the video. After that, we visualize the optical flow following [2]. The predicted optical flow of the FFN method for an image pair is normalized using the maximum 2D pixel displacement from the corresponding optical flow of our model to visualize the

results. It is noticed that FFN produces some alignment artefacts, especially near the smooth regions (such as walls), which result from severe wrong predictions in those regions.

References

1. 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: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
2. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)
3. Kong, L., Shen, C., Yang, J.: Fastflownet: A lightweight network for fast optical flow estimation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 10310–10316. IEEE (2021)
4. Mayer, N., Ilg, E., Hausser, 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: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
5. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)