Supplementary Material StereoFlowGAN: Co-training for Stereo and Flow with Unsupervised Domain Adaptation

Zhexiao Xiong[1]
x.zhexiao@wustl.edu
Feng Qiao[2]
feng.qiao.ad@gmail.com
Yu Zhang[3]
yuzh03@gmail.com
Nathan Jacobs[1]
jacobsn@wustl.edu

[1] Department of Computer Science & Engineering,
Washington University in St. Louis
St. Louis, MO, USA

[2] Institute for Automotive Engineering,
RWTH Aachen University
Templergraben 55, Aachen, Germany

[3] Department of Computer Science,
University of Kentucky
Lexington, KY, USA

1 Implementation Details

For IGEV-Stereo [4] and DispNetC [2], we follow the setting of their original papers. For Unimatch-flow [5] network, we set the number of scales as 2 and do not use task-specific refinement. For IGEV-Stereo and Unimatch-flow network, we use AdamW optimizer with the momentum $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate $\alpha = 0.0001$. For DispNetC, we use Adam optimizer with the momentum $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate $\alpha = 0.0001$. We implement this method on *Pytorch*. For Driving & KITTI2015 datasets we train for 100k steps and for VKITTI2 & KITTI2015 datasets we train for 200k steps. During training, we optimize the domain translation network every 3 steps and optimize the stereo matching network and optical flow estimation network every step. We set batch size to 4 in the training process on 2 NVIDIA A6000 GPUs.

2 Further Comparison

We further compare our methods with UnDAF[1], following their experiment settings with the same stereo matching and optical flow estimation backbones respectively on VKITTI2 & KITTI15 datasets. We only report the F1 scores as UnDAF only provide F1 scores on two tasks and they do not release their code. The results are shown in Table 1. Our framework performs better.

Table 1: Results on datasets from VKITTI2 to KITTI2015.

	F1-disparity (%)	F1-flow (%)
UnDAF	2.85	10.23
Ours	**2.72**	**9.81**

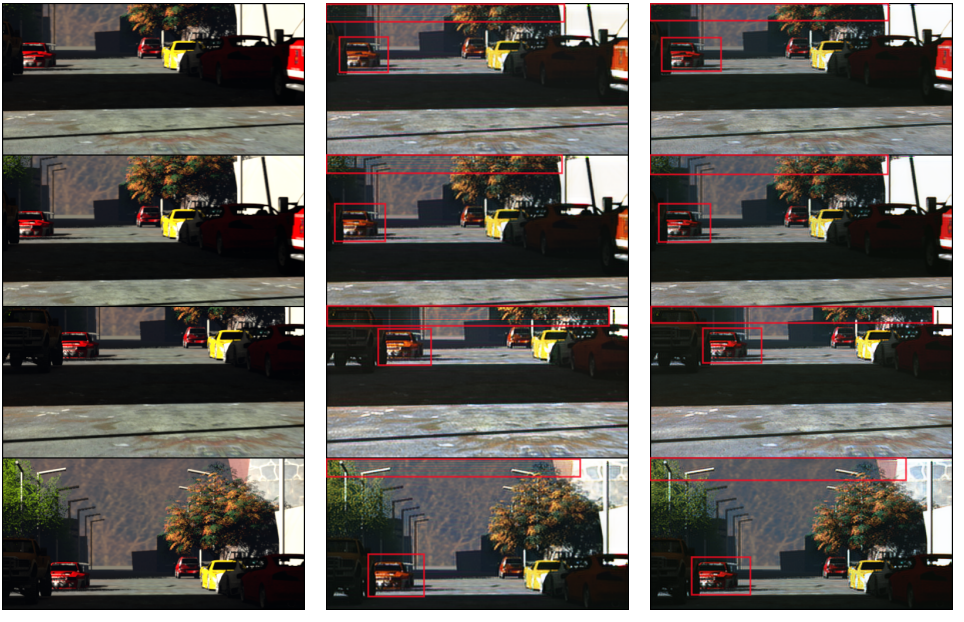

Figure 1: Synthetic-to-real translation from Driving to KITTI2015. Left: RGB leftA images; Middle: generated fake_leftB images by StereoGAN; Right: generated fake_leftB images of our proposed method. Our proposed method avoids the appearance of noisy strips and helps maintain more realistic color.

3 Visualizations

We provide qualitative visualizations that compare our method to previous approaches on domain translation, stereo matching, and optical flow estimation.

3.1 Visualization of Domain Translation

Here we visualize the results of domain translation and compare our results with StereoGAN. We visualize the results on both Driving and VKITTI2. Figure 1 shows the comparison in synthetic-to-real translation, Figure 2 shows the comparison in real-to-synthetic translation, and Figure 3 shows the comparison of real-synthetic-real cycle translation. It can be seen that compared with StereoGAN [1], under the supervision of perceptual loss $\mathcal{L}_{perceptual}$, cosine similarity loss $\mathcal{L}_{cos}$ and $\mathcal{L}_{flow\_warpx}$ in the same iteration, the generated images are of high quality with accurate color information and fewer visual artifacts, maintaining both global consistency and detailed information in edges. The accurate domain translation contributes to better performance in stereo matching and optical flow estimation tasks.

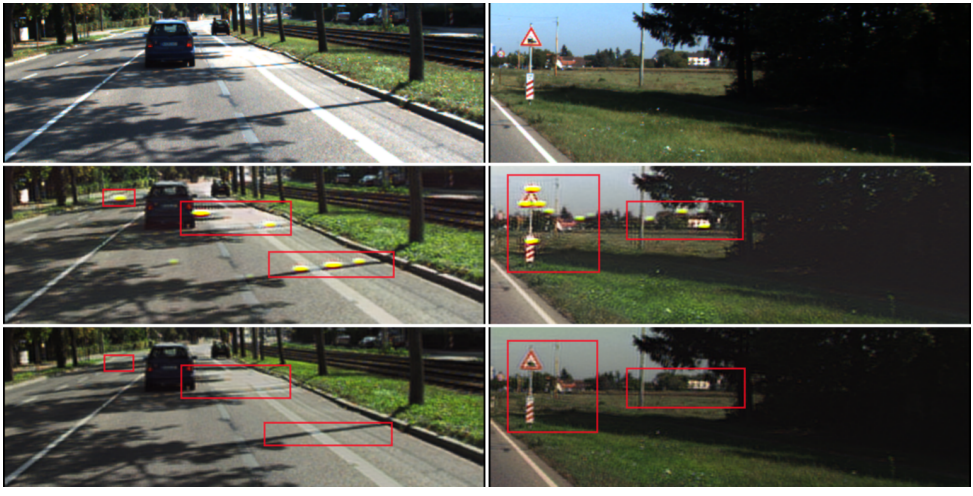

Figure 2: Real-to-synthetic translation from KITTI2015 to Driving. The first row: RGB leftB image. The second row: real-to-synthetic generated fake_leftA images by StereoGAN. The third row: real-to-synthetic generated fake_leftA images by our proposed method. Our proposed method avoids the appearance of noisy points and helps maintain global consistency.

3.2 Visualization on Stereo Matching Task

We visualize the results of our proposed framework and compare the results with the results of StereoGAN in Figure 4. Our method generates disparity maps that maintain more accurate details in boundary regions.

3.3 Visualization on Optical Flow Estimation Task

We visualize the results of our proposed framework and compare the results with the source-only results of Unimatch-flow in Figure 5. Our proposed method effectively handles occluded and out-of-boundary pixels.

References

[1] Rui Liu, Chengxi Yang, Wenxiu Sun, Xiaogang Wang, and Hongsheng Li. Stereogan: Bridging synthetic-to-real domain gap by joint optimization of domain translation and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Hengli Wang, Rui Fan, Peide Cai, Ming Liu, and Lujia Wang. Undaf: A general unsu-

Figure 3: Real-synthetic-real cycle translation between Driving and KITTI2015. Left: RGB leftB image; Middle: generated rec_leftB images by StereoGAN; Right: generated rec_leftB images by our proposed method. Our proposed method maintains accurate color in the sky and car region.

pervised domain adaptation framework for disparity or optical flow estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[4] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[5] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2023. doi: 10.1109/TPAMI.2023.3298645.

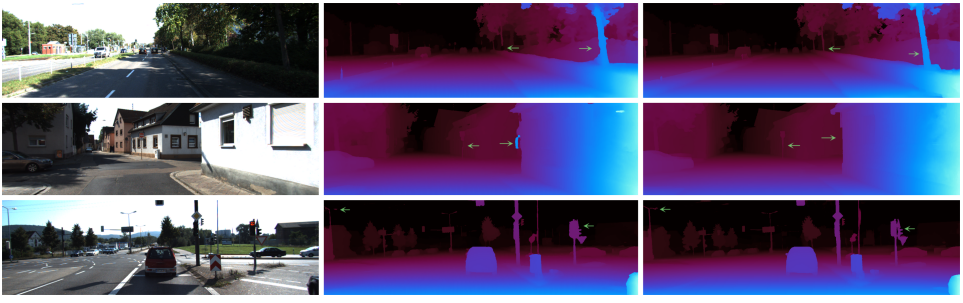

Figure 4: Stereo matching task comparison from VKITTI2 to KITTI2015. Left: RGB left image; Middle: StereoGAN based on IGEV backbone; Right: our method based on IGEV and Unimatch-flow backbone.

Figure 5: Optical flow estimation task comparison from VKITTI2 to KITTI2015. Left: RGB left image; Middle: Unimatch-flow source only; Right: our method based on IGEV and Unimatch-flow backbone.