CMSC733 Project 4: Learning SfM, an Unsupervised Way

Xiangyu Liu
Department of Computer Science
University of Maryland, College Park
Email: xyliu999@umd.edu

Param Dave
Masters in Robotics
University of Maryland, College Park
Email: pdave1@umd.edu

***Abstract*—In this project we explored an unsupervised deep learning approach SFMLearner to retrieve depth and Ego-Motion from motion to compute the flow. We started with the code published with [1] and tried to improve its output.**

I. Introduction

The SfMLearner Network as described in [1] tries to predict the likely camera motion (Ego-motion) and scene structure and trains itself without supervision (Labelled data) by minimizing the loss function so as to get the predictions as close as possible to the ground truth. The network can be trained using sequence of images with no labeling or camera motion information.

The network consists of two different networks, each dedicated to training Depth and Pose respectively. The Depth network is realized using DispNet [3] which is an Encoder - Decoder based architecture with ReLU as activation function after each convolution network and sigmoid as prediction layer. The Depth Network takes one image frame as input and generates a depth map as output. On the other hand, Pose Network takes target view as an input which is concatenated with source views. The network synthesizes target image from multiple source images and outputs the relative camera poses. The output of both the networks are then used to inverse warp the source views to reconstruct target views and the photometric reconstruction loss is used for training the CNNs. Figure 1 shows the block diagram of SfMLearner network architecture.

The SfMLearner makes certain assumptions about the video feed.

- Scenes are mostly rigid i.e. scene appearance across different frames is dominated by the camera motion.
- There are no occlusion/dis-occlusions in the scene.
- The surfaces in the scene are considered to be non-Lambertian surfaces

To improve the robustness of the learning pipeline to these factors, an additional network named 'explainability prediction network' is trained that outputs a per-pixel soft mask Es for each target source pair. Based on this predicted Es the view synthesis objective is weighted correspondingly. This in a nutshell, summarizes the structure of the SfMLearner network. As cited by [1] and verified by running the code on our systems, the network tends to converge after about 180K iterations. For this project, we restrict ourselves with training and testing on KITTI dataset.

In the following sections, we shall discuss some of the modifications we've made in an attempt to improve the prediction of the Network.

II. Our Approach

Our improvement mainly lies in two aspects: the new SSIM loss and random color augmentation.

A. SSIM Loss

SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. The difference with other techniques such as MSE or PSNR is that these approaches estimate absolute errors. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. We implement the following SSIM loss

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}}$$

$$\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$\sigma_y = \left(\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2\right)^{\frac{1}{2}}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

B. Random Color Augmentation

For data augmentation, we randomly shift the brightness and color of the given image. The augmentation is applied to each

```python
def random_coloring(im):
    batch_size, in_h, in_w, in_c = im.get_shape().as_list()
    im_aug = tf.image.convert_image_dtype(im, tf.float32) ** tf.random_uniform([], 0.8, 1.2)
    im_aug *= tf.random_uniform([], 0.5, 2.0)
    im_aug *= tf.stack(
        [tf.ones([batch_size, in_h, in_w]) * (tf.random_uniform([in_c], 0.8, 1.2))[i] for i in range(in_c)],
        axis=3)
    im_aug = tf.clip_by_value(im_aug, 0, 1)
    im_aug = tf.image.convert_image_dtype(im_aug, tf.uint8)
    return im_aug
```

Fig. 1. Random color data augmentation

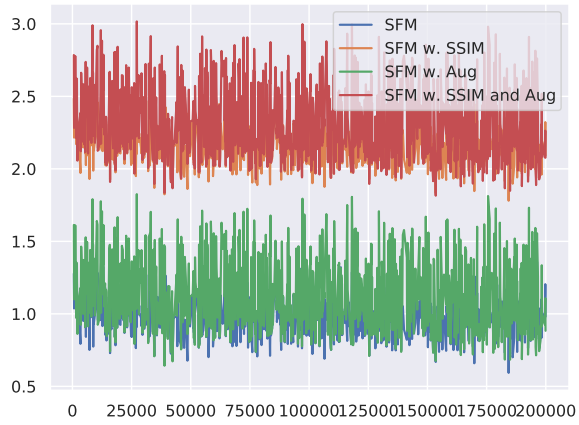

Fig. 2. Loss during training of four models

image randomly with probability 0.5. The implementation goes in Figure 1.

III. Results and Comparison

A. Depth Prediction Results

The loss of models is given in Fig 2. Note since this is an unsupervised learning framework, the decrease in loss may not be obvious. The metrics on the test set is given in Table I

Sampled depth prediction map is given in Figures 3, 4, 5, 6

B. Pose Prediction Results

The pose predictions on Seq 9 is given in Table II The pose predictions on Seq 10 is given in Table III The ground truth and prediction trajectories for Seq 9 are given in Figure 7, 8, 9, 10, 11.

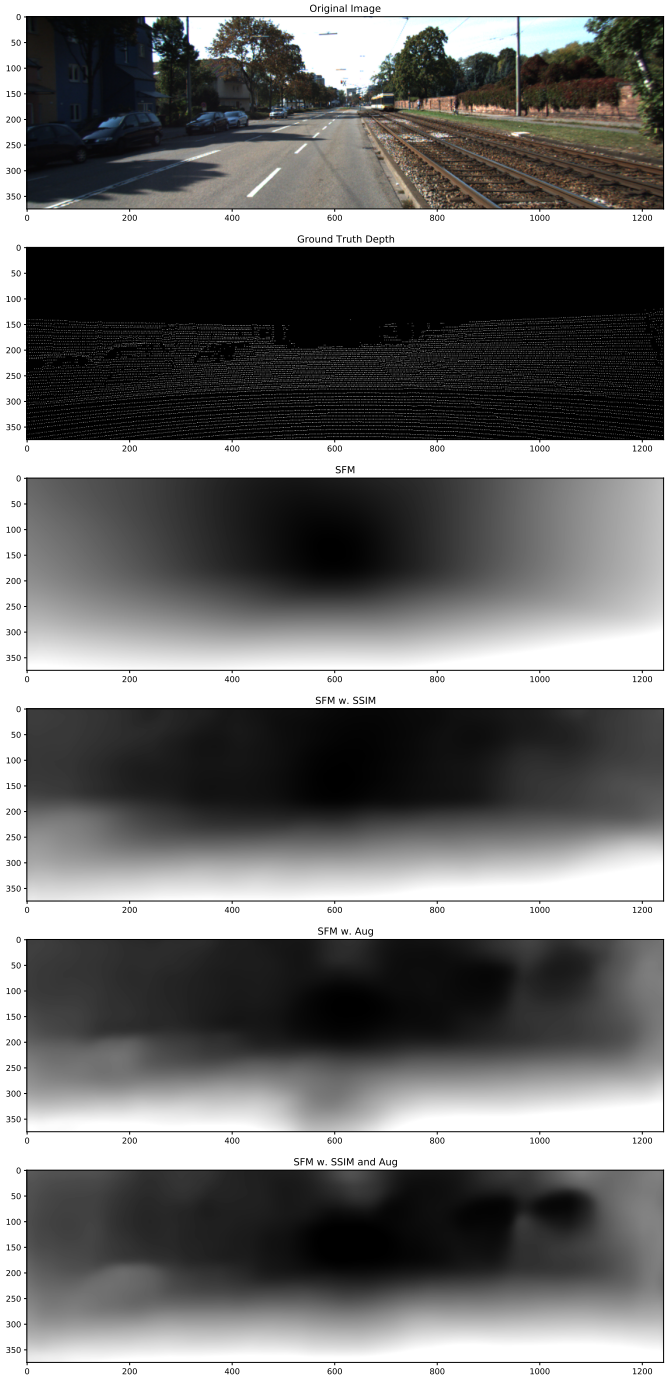

Fig. 3. Depth Map Comparison (Original Image, Ground Truth, SFM, SFM w. SSIM, SFM w. Aug, SFM w. SSIM and Aug)

	SFM	SFM w. SSIM	SFM w. Aug	SFM w. SSIM and Aug
abs_rel	0.2586,	0.1721,	0.1881,	0.1783,
sq_rel	3.2879,	1.5479,	3.2440.	2.3003,
rms	9.0565,	6.1315,	6.3793,	6.0506,
log_rms	0.3406.	0.2315,	0.2505	0.2372,
a1	0.5840,	0.7580,	0.8033	0.7871,
a2	0.8482,	0.9183,	0.9277	0.9324,
a3	0.9324	0.9700	0.9641	0.9699

TABLE I
Test Errors of Depth Prediction

	SFM	SFM w. SSIM	SFM w. Aug	SFM w. SSIM and Aug
ATE mean	0.0274	0.0219	0.0186	0.0185
Std	0.0145	0.0115	0.0082	0.0076

TABLE II
Test Errors of Pose Prediction on Seq 9

	SFM	SFM w. SSIM	SFM w. Aug	SFM w. SSIM and Aug
ATE mean	0.0205	0.0170	0.0152	0.0144
Std	0.0123	0.0101	0.0086	0.0086

TABLE III
TEST ERRORS OF POSE PREDICTION ON SEQ 10

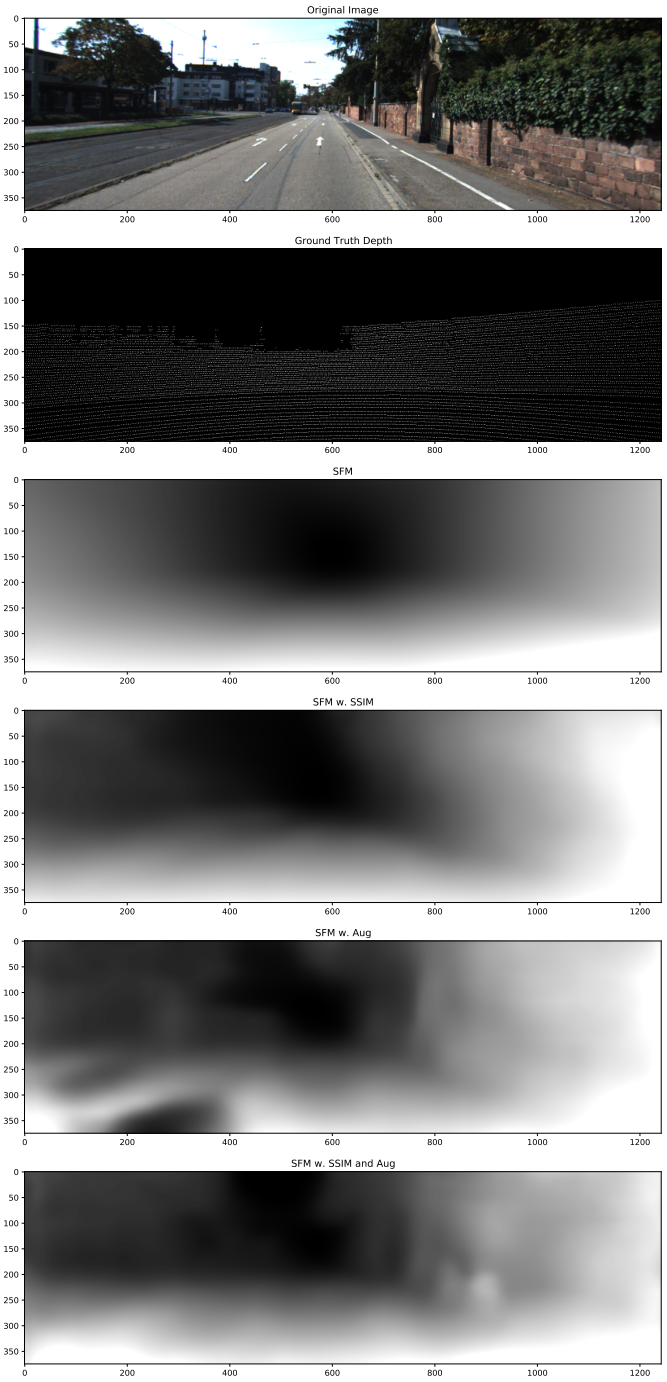

Fig. 4. Depth Map Comparison (Original Image, Ground Truth, SFM, SFM w. SSIM, SFM w. Aug, SFM w. SSIM and Aug)

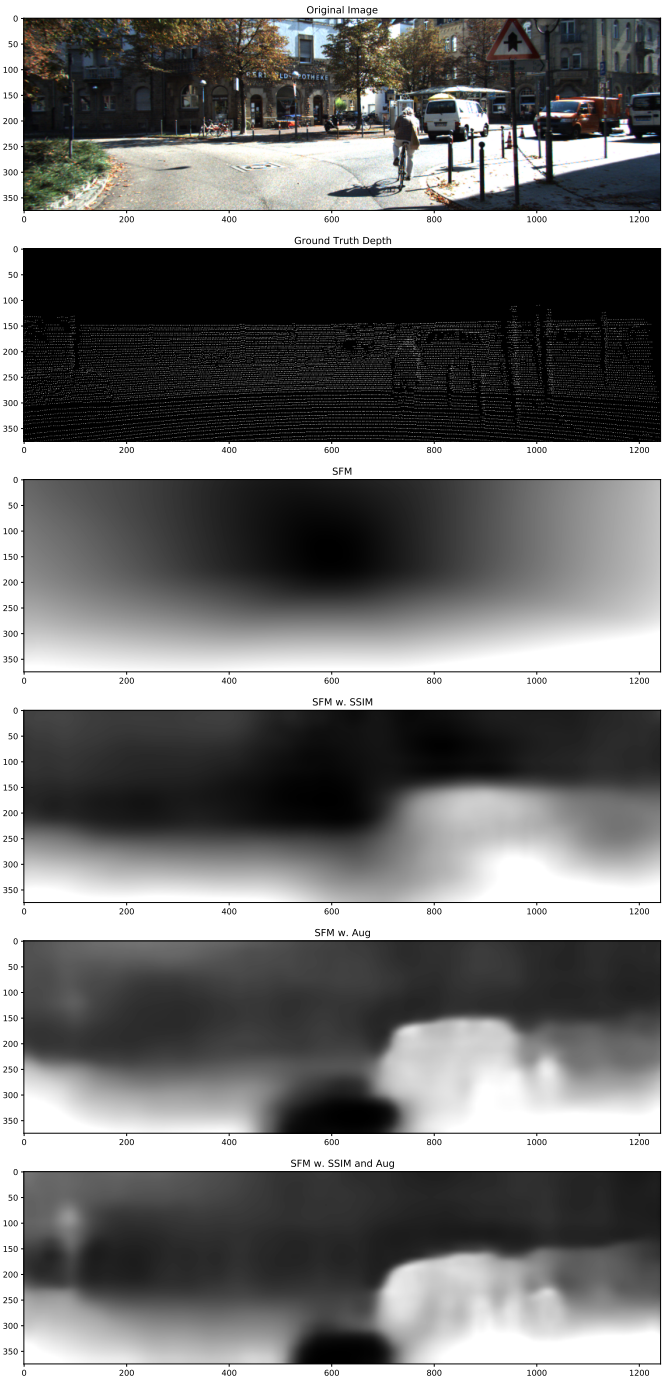

Fig. 5. Depth Map Comparison (Original Image, Ground Truth, SFM, SFM w. SSIM, SFM w. Aug, SFM w. SSIM and Aug)

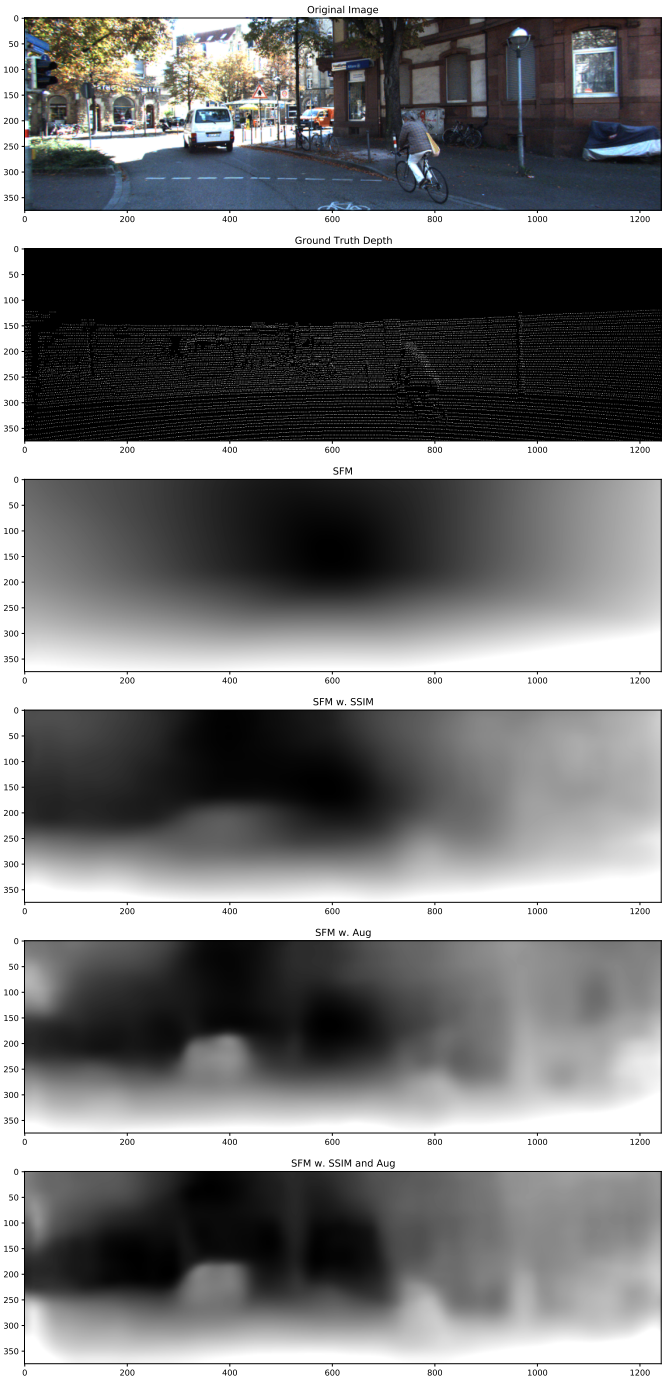

Fig. 6. Depth Map Comparison (Original Image, Ground Truth, SFM, SFM w. SSIM, SFM w. Aug, SFM w. SSIM and Aug)

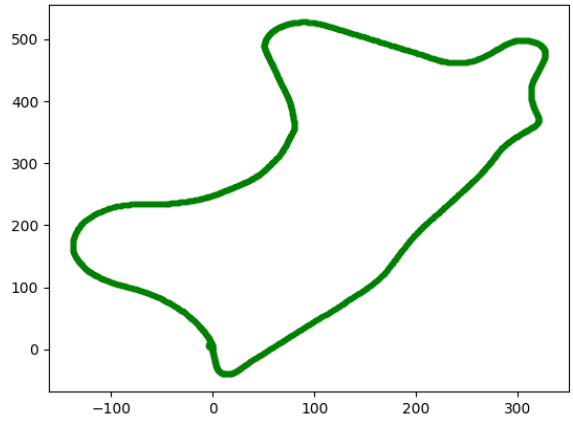

Fig. 7. Ground Truth Trajectory for Seq 9

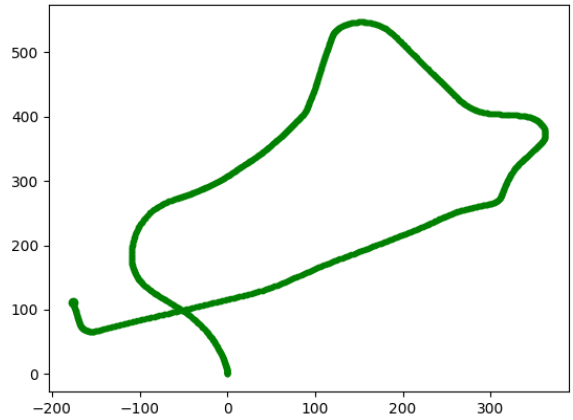

Fig. 8. SFM for Seq 9

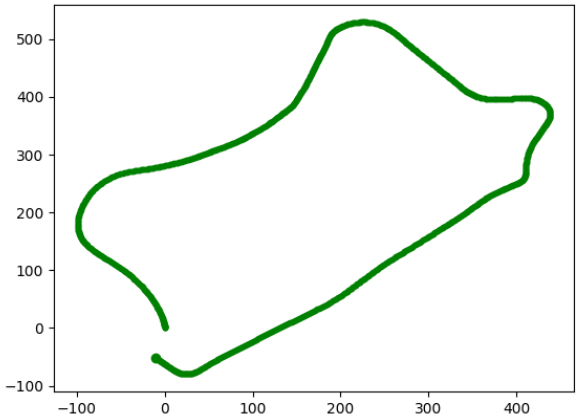

Fig. 9. SFM w. SSIM for Seq 9

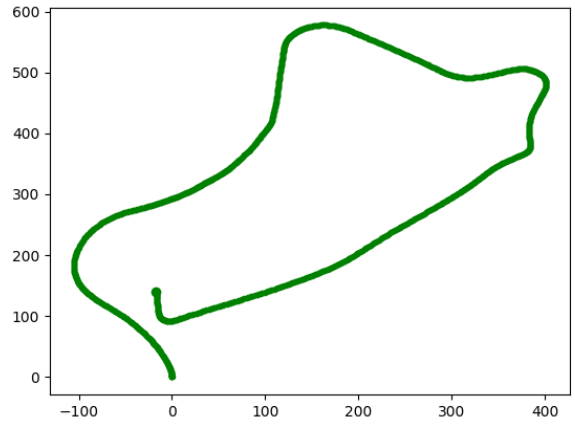

Fig. 10. SFM w. Aug for Seq 9

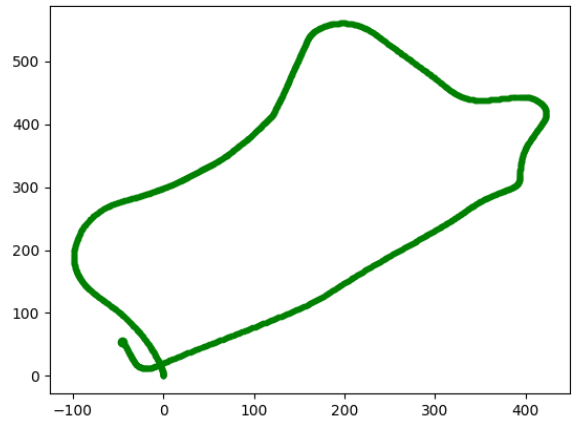

Fig. 11. SFM w. SSIM and Aug for Seq 9

The ground truth and prediction trajectories for Seq 9 are given in Figure 12, 13, 14, 15, 16.

IV. Conclusion

In this project, we mainly consider two improvements, ssim loss and random color augmentation. Both ssim loss and data augmentation improve the prediction accuracy. By our comparison, the random color augmentation is more useful. By combining ssim loss and random color augmentation, we could get the best depth prediction and pose prediction accuracy.

Acknowledgment

The authors would like to thank the instructor and TAs.

References

[1] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1851–1858, 2017

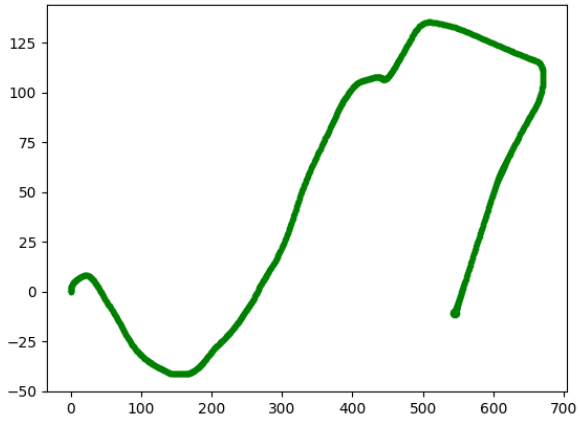

Fig. 12. Ground Truth Trajectory for Seq 10

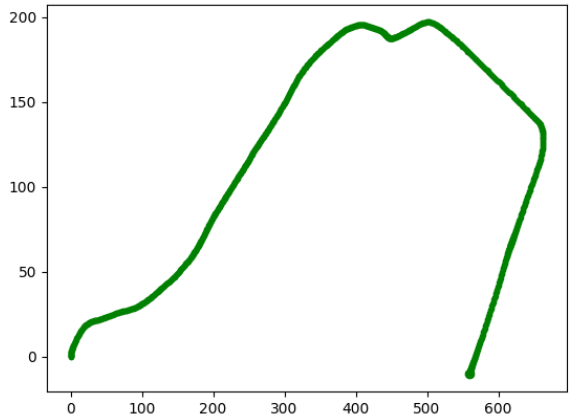

Fig. 13. SFM for Seq 10

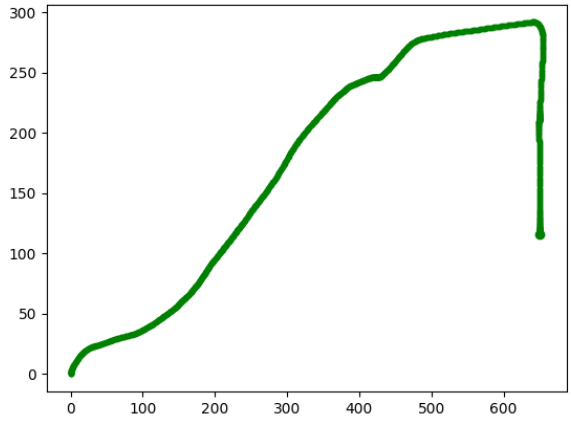

Fig. 14. SFM w. SSIM for Seq 10

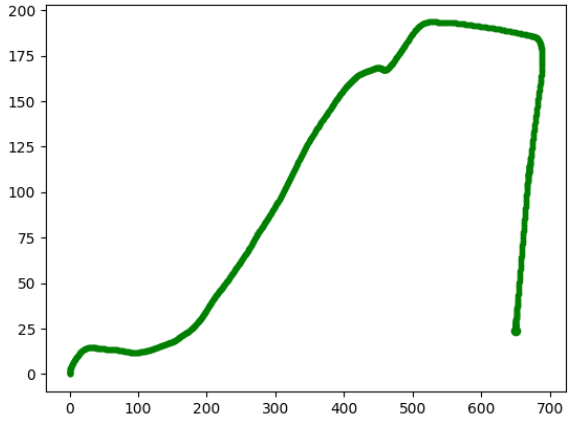

Fig. 15. SFM w. Aug for Seq 10

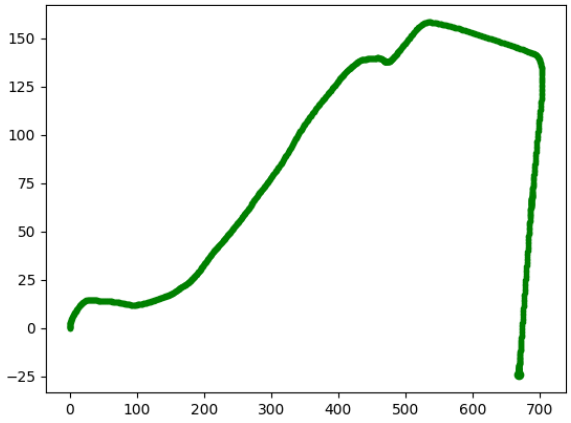

Fig. 16. SFM w. SSIM and Aug for Seq 10

[2] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1983–1992, 2018

[3] Nikolaus Mayer et al. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". In: CoRR abs/1512.02134 (2015). arXiv: 1512 . 02134. URL: http : / / arxiv. org / abs / 1512 . 02134.