Progress Update: Motion Segmentation

09/02/2022

Introduction

The focus of the past week was on training on our new Carla dataset and on the previously existing KITTI dataset and to evaluate/interpret the obtained results. We have now achieved sufficiently good results on both datasets to justify to move on to the self-supervised learning approach.

Our secondary focus was to extend our CARLA data collection script to include a depth sensor and also the pose changes from one frame to the next which is saved as a json file. This can be read out later to be used in the training step. We noticed some rendering issues in CARLA as well as some false ground truth in one sequence which is discussed in more detail later.

Supervised Quantitative Result

We have now finished full training on both the KITTI data and our generated CARLA data. Our network is a U-Net and together with focal loss was trained with a learning rate of 5e-5 and a learning rate scheduler with patience set to six epochs.

The performance evaluated with IoU can be seen in Table 1. As is evident, if the model is trained on one dataset, it is not able to generalize well to the other dataset. This is not all too surprising as there is likely a big domain gap between the two datasets; also because of problems such as inconsistent and poor labeling for KITTI and for CARLA the possibly too perfect annotation since there are ground truth masks for cars very far away, discussed in more detail later.

	training KITTI	training CARLA
inference KITTI	0.759	0.090
inference CARLA	0.199	0.767

Table 1: Quantitative Performance of Motion Segmentation Network(Using IoU as a Performance Metric)

What is perhaps most surprising is that a model trained on CARLA does not significantly perform better on CARLA data than a model trained on KITTI performs on KITTI data. We have identified a few reasons for this:

- Labeling errors in CARLA dataset which are caused by bugs in the CARLA simulator's instance segmentation class
- Few dynamic pixels corresponding to distant cars in KITTI and CARLA

More detail is provided in the following sections.

CARLA Data Observations

Rendering Inconsistencies in CARLA

When checking the ground truth and predicted images on Tensorboard we noticed, that CARLA's motion segmentation mask were very precise, but the corresponding RGB images

had some rendering problems. After checking some sequences we noticed that the first frame was always rendered nicely. In contrast, the second frame and the ones afterwards always had worse quality, see Figure 1 and Figure 2. We noticed the following artifacts:

- rubber of the tire seems to be staying in the initial position
- the texture of the white wall on the right has lost the black outlines
- the car roof seems to be missing and the taillights are not red but clear

Figure 1: First frame

Figure 2: Second frame with artifacts

We have tried to resolve the issue by decreasing our simulation time (which was low anyway at around 0.5 fps). Additionally, the physics rendering of the actors in the CARLA world have been disabled outside a circle with radius of 75m centered at the ego vehicle circle with the hope of improved rendering of the scene. In CARLA there is the possibility of off-screen rendering which starts the simulation invisible for the user but still renders the images for all the sensors. This is not possible in our case with the CARLA environment built from source.

Distant Dynamic Cars in Ground Truth

RGB images containing one or two very distant cars only, will almost always not be detected in our motion segmentation. This leads to an IoU score of 0.0 for that particular example which significantly decreases our average IoU. This is a problem in both KITTI and CARLA, however it is worse in CARLA because the labels are “pixel perfect” so even the tiniest parts of cars are picked up in the ground truth.

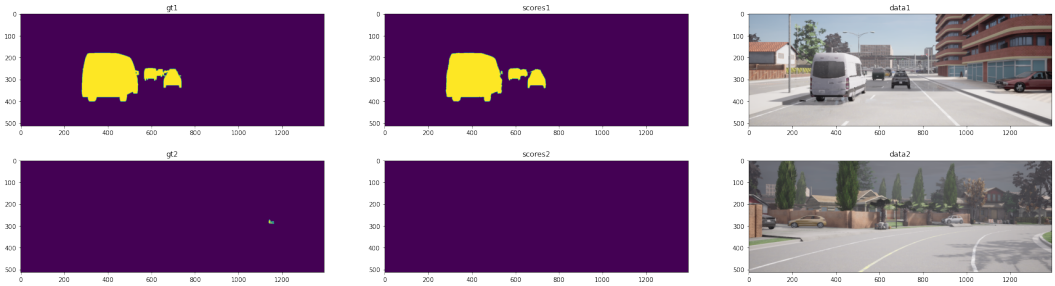


Figure 3: Few Dynamic Pixels in Mask (IoU top = 0.973, IoU bottom = 0.00)

Possible solutions include:

- Weighted IoU depending on number of dynamic pixels or other metrics → in [1] they use the average IoU of two classes: {moving cars, static cars}
- Discard images from training completely by filtering with some threshold for the minimum number of dynamic pixels
- Change the frame to be completely static in the modified ground truth

Dynamic Motion Detection of Non-Lambertian Surfaces

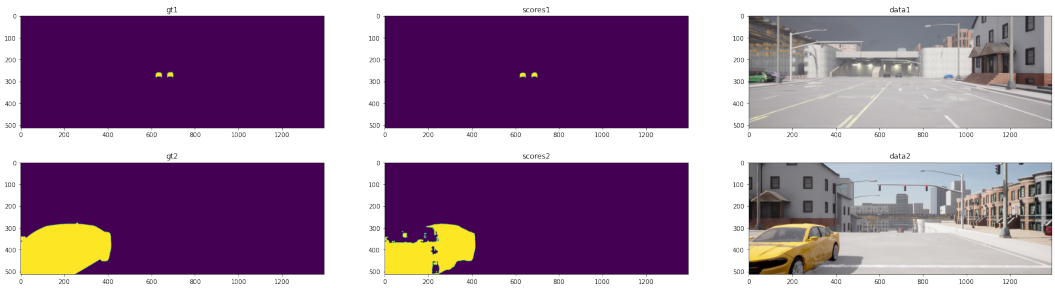


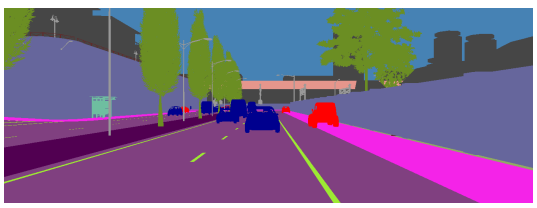
Figure 4: Motion Segmentation Issues with Non-Lambertian Surfaces (IoU both ~ 0.8)

CARLA Data Generation

We generated more Carla sequences from Town_03 and included ground truth depth images as well as ground truth transformations T_{i+1}^{i}.

Ground truth artifacts

After inspecting Tensorboard we noticed that one of our ground truth motion segmentation masks contains false positives. Checking the corresponding folder, all images containing this wall had the "moving" annotation, not just one random frame, see Figure 5b. Luckily, this was the only sequence with this irregular annotation sporadically caused by the CARLA simulator caused by some unknown bug.



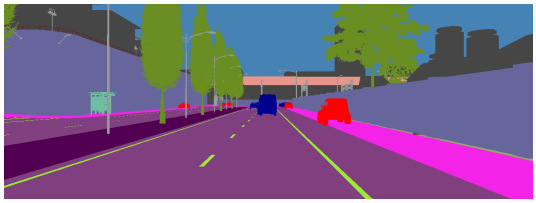
(a) Semantic segmentation

(b) Motion segmentation

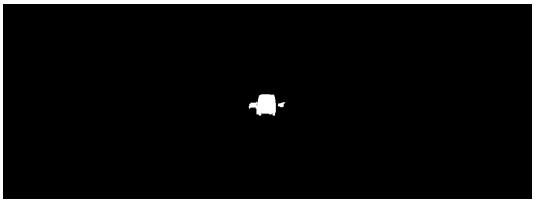
Figure 5: Image with artifact

Figure 6: RGB Image with artifact

In the new dataset that includes depth and the relative poses, the ego vehicle also passed by this location but does not segment this wall portion incorrectly as "moving", see Figure 7b.



(a) Semantic segmentation



(b) Motion segmentation

Figure 7: Image of the same corner without the artifact

Figure 8: RGB Image of the same location

Figure 9: Depth image sensor

Questions/Discussion Points for Self-Supervised Approach

Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation

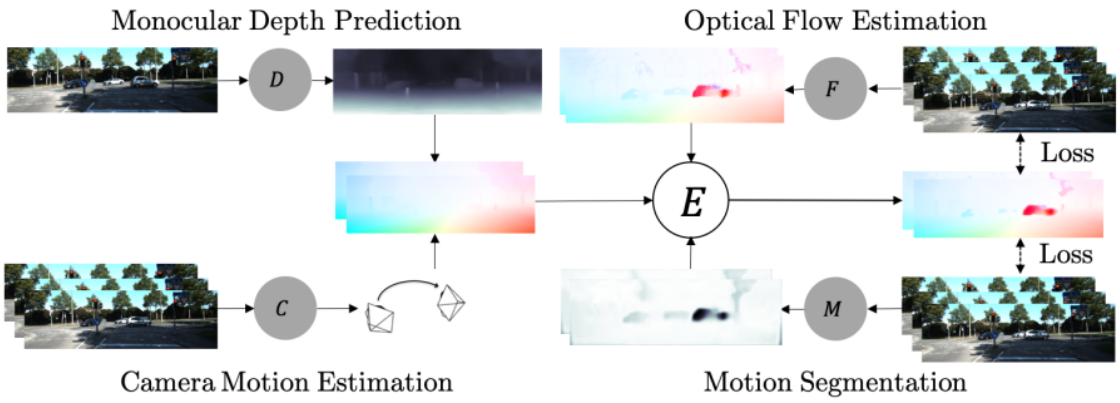


Figure 10: Figure 2 from [1]

- Static optical scene flow from depth and camera motion in Figure 10 from [1]. Why are there no holes in static scene flow masks where dynamic objects are, since the moderator should only pass what it thinks to be static pixels to (D,C)?
- On page 5 of [1] in the experiments section, the authors outline independent training of [(D,C), F, (D,C,F,M)] before the for-loop. Is this training supervised? Until which point do the authors train the networks?
- Competitive Collaboration uses 5 images per pass. This is not a problem in our fully convolutional network since we can directly concatenate 5 images for a total of 15 channels which is also done in the paper, see Figure 6 in the paper [1] on page 15.
- Since the paper predicts motion segmentation, depth, camera motion, optical flow etc. using frames $i_{--}, i_{-}, i, i_{+}, i_{++}$ it would not be possible to use the network for inference in real time. During inference still use frames which are ahead in time.
- Masking out dynamic pixels to create the static scene flow and feeding this directly into the architecture leaves the M network to assign the remaining (dynamic) pixels to the F network. So we implicitly provide the network with the perfect ground truth mask. Can we add noise so this does not happen BUT we don't *learn* the static flow since we deterministically provide ground truth for it therefore if we add noise it cannot improve.

References

[1] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 12 240–12 249. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Ranjan_Competitive_Collaboration_Joint_Unsupervised_Learning_of_Depth_Camera_Motion_Optical_CVPR_2019_paper.html