Progress Update: Motion Segmentation

23/02/2022

Introduction

This week we finally finished our entire data-generation pipeline for the self-supervised architecture, including the static optical flow and additional tweaks in CARLA.

Next, we improved our Dataloader and expanded it to additionally load new sensor modalities and types of data.

Furthermore, we made a start of formulating our new loss more explicitly and plan to finalize this loss as our next goal.

Static Optical Flow Plotting/Verification

After last week's issues with the optical flow computation and visualization we realized that our transformations were from ego-frame to ego-frame. Upon realizing this, we added a mapping from sensor-frame to ego-frame before the transformation and back from ego-frame to sensor-frame. This then gives the final transformation between point clouds, shown in Equation 1.

$$_{sensor2}T_{sensor1} \;=\; _{ego}T^{-1}_{sensor}\; _{ego2}T_{ego1}\; _{ego}T_{sensor} \tag{1}$$

where

$$_{ego}T_{sensor} \;=\; RotY(90)RotZ(90) \tag{2}$$

We then wrote a script to generate our static optical flow data from the previously generated ego poses and depths in CARLA for all sequences at once and pickling this data. We now have all the data required to train our self-supervised network.

Unsupervised Network

Loss

We adapted the loss from [1] to now additionally include an error term which minimizes the geometric error between two images, see Equation 3.

$$E = \lambda_M E_M + \lambda_C E_C + \lambda_S E_S \tag{3}$$

The E_M term forces the network to become more confident about its prediction m_s about whether a pixel is static (1) or dynamic (0).

$$E_M = \sum_{\Omega} H(\mathbf{1}, m_s) \tag{4}$$

At the same time the E_C term penalizes the network when the prediction for a pixel is wrong.

$$E_C = \sum_{\Omega} H(\mathbb{I}_{\rho_R < \rho_F} \vee \mathbb{I}_{\|\nu(e_s,d) - u_s\| < \lambda_c}, m_s) \tag{5}$$

In both cases the H represents the cross-entropy loss.

Since for now our network will include the ground truth static optical flow and dynamic optical flow masks, the error terms E_R and E_F from the original paper would result in no loss. Thus these two loss terms were not be included in equation 3.

Expanding equation 12 from [1] to include the geometric component in the third indicator function leads to

$$E_C = \sum_{s\in\{+,-\}} \sum_{\Omega} H(\mathbb{I}_{\rho_R<\rho_F} \vee \mathbb{I}_{\|\nu(e_s,d)-u_s\|<\lambda_c} \vee \mathbb{I}_{\|p_s-\tilde{p_s}\|<\lambda_{geo}}, m_s) \tag{6}$$

where p_s is the position of the pixel in the first frame after reprojection into 3D space and $\tilde{p_s}$ the position of the associated pixel in the second frame projected into 3D space, using the respective depths.

The first indicator function checks for the photometric error, the second for the optical flow and the third ensures geometric consistency.

Since our final goal is to obtain the motion segmentation masks, the model would not iterate between a *collaboration* and a *co,petition* step but instead just train during the *competition* phase.

Dataloader

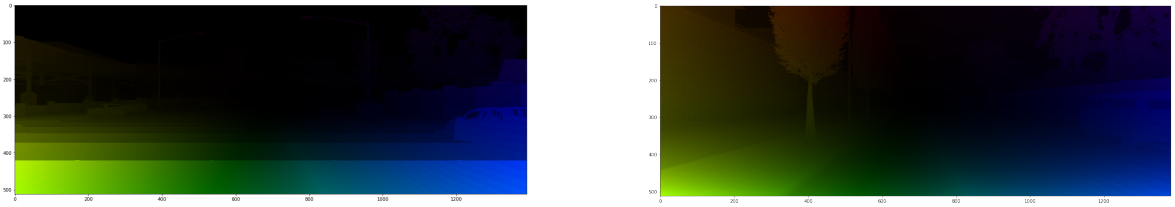
We needed to build a new data loader for the new modalities included in the unsupervised object segmentation task. Our new dataloader will load the following things:

- RGB images - for photometric error computation and general visualization
- Depth images - for geometric error computation
- Static flow image - ground truth computed by post processing RGB and depth maps
- Dynamic flow - obtained from CARLA optical flow sensor
- Motion segmentation masks - used for evaluation in inference but not during training

CARLA

Optical flow improvements through depth sensor

When converting the optical flow vectors v_x and v_y to the HSV color scheme where the magnitude represents the saturation and the angle of the vectors the hue, we noticed some strange artifacts in our results. After some debugging checks we saw that our ground truth depth maps also had these step-like artifacts, as shown in 1a. After consulting the CARLA documentation we changed our depth sensor to save the logarithmic depth which yielded a much smoother visualization. Below are the optical flows before and after the fix. These visualizations are not directly from the CARLA sensor visualization but through our own optical flow computations.



(a) Optical flow before fix

(b) Optical flow after the fix

Figure 1: Static Optical Flow Improvement - Depth vs. Logarithmic Depth

Questions/Discussion Points for Self-Supervised Approach

- Initialization for motion segmentation? - Gaussian in the image center ($\vec{\mu} = [\frac{w}{2}, \frac{h}{2}]$ and σ as a hyperparameter) since cars would most likely appear in the center of a frame

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