Preparation of Papers for IEEE Sponsored Conferences & Symposia*

Florian Kuhnt[1] and J. Marius Zöllner[1]

Abstract— This electronic document is a ÒliveÓ template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document.

I. Introduction

This template provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

Example Citation: [1].

A. Problem specification

B. Why RL? What is our goal/motivation?

II. Related Work

A. "Human-level control through deep reinforcement learning" (2015)

B. "CARLA: An Open Urban Driving Simulator"

C. Our contribution to the field

III. Concept

A. CarRacing - Understand and select algorithms

1) Preprocessing:
2) DQN:
3) A3C:
   a) Model architecture:
   b) Training details:
4) DDPG:
   a) Model architecture:
   b) Training details:

[1]The authors are with FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany {kuhnt, zoellner}@fzi.de

B. CARLA

1) State representation: Dealing with the high-dimensional environment in CARLA is one of the central challenges in implementing a well functioning RL algorithm. In consideration of an abundance of possible sensor types available in CARLA we decided to pursue increasing levels of realism which generally also correspond to increasing levels of difficulty. These aforementioned levels of realism involve the following:

- Ground truth segmented bird's eye view
- Ground truth segmented front view
- Latent space generated from ground truth segmented images
- Latent space generated from rgb images

Notably, these input representations differ in dimensions and thus lead to different network architectures in the appended network architectures of the RL agent.

a) Ground truth segmented bird's eye view: The Ground truth segmented bird's eye view is a rather unrealistic scenario, in which we assume availability of a camera positioned 20m above the performing agent. It is merely imaginable in a fully observed city where autonomous driving is part of a high-level traffic control system. Albeit rather unrealistic, we decided to implement a 13-class ground truth segmented bird's eye view due to its similarity to the CarRacing environment and its obvious advantages in containing information central to navigational tasks. The full list of the included classes are described by dosovitskiy et al.[2]. We deemed a 1-channel grayscale image with size 64x64 large enough to contain key information for the algorithm (fig. x l.)

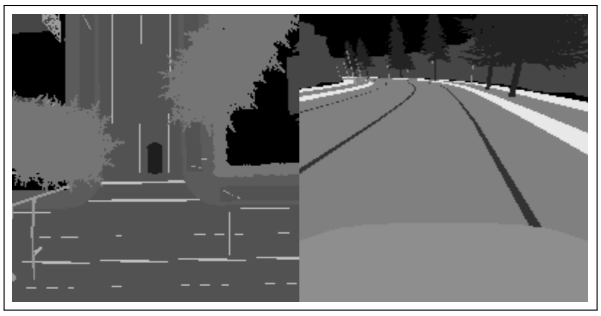

Fig. 1: left: Ground truth segmented bird's eye view image, 64x64 in grayscale, 9 classes right: Ground truth segmented front view image, 80x80 in grayscale, 9 classes

b) Ground truth segmented frontview: Much like the Ground truth segmented bird's eye view, the corresponding front view input representation assumes the

availability of a perfect segmentation camera. Nonetheless, we increase realism with this representation as the camera is placed in front of the car. Again, we chose a 9-class, 1-channel grayscale image with a slightly increased size of 80x80 to account for a larger view angle of the camera.

c) Latent space generated from ground truth segmented images: In this section we will further discuss a modified version of the integrated encoder-decoder network that is based on dziubinski's [3] implementation. In its original architecture, the network comprises 5 input tensors and 7 output tensors with the inputs representing images of a front-, rear-, left-, right, and top-view camera. The model is built with the Keras functional API and generally serves two purposes. On the one hand, each input tensor is encoded into a 64-sized feature vector and decoded into it's original shape with a cross entropy loss with regard to the original input. This is achieved with separate branches of 3-5 layered Convolutional Networks which can be considered autoencoders of their own. On the other hand, a separate generative branch creates a bird's eye view reconstruction based on the concatenated feature vectors of the vehicle-based cameras.

2) Used Sensors:

3) Implementing the reward function:

IV. Evaluation

A. Results

B. Comparison algorithms/reward functions

V. Conclusions

References

[1] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" 2008 IEEE Intelligent Vehicles Symposium, pp. 1068–1073, June 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4621210

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," arXiv preprint arXiv:1711.03938, 2017.

[3] M. Dziubiński, "From semantic segmentation to semantic bird's-eye view in the CARLA simulator," https://medium.com/asap-report/from-semantic-segmentation-to-semantic-birds-eye-view-in-the-carla-simulator-1e636741af3f, May 2019.