**libraries to use: CuDNN, TensorFlow**

**research domain: Machine Learning / AI**

**Programming Interfaces: Unity, Anaconda**

**Equipment requested: Nvidia Geforce Titan X, 1080Ti or Tesla GPU**

**Hardware Grant Request: Proposal**

**Project title:** Controlling Self-Driving Race Cars with Deep Neural Networks

**Team Leader:** Gordon Pipa, Osnabrück University

**Introduction & relevance**

State-of-the-art self-driving cars rely on carefully designed, handcrafted algorithms to control the car’s movement. Tactical decisions, such as what speed to aim for, what distance to keep from another car, and whether or not to overtake, are separated from the operational decisions. The latter concern the precise planning and eliciting of motor commands given to the car, i.e., accelerating, braking, and steering. Most of the anticipated advances of self-driving cars with respect to road safety so far rely on advances on the tactical side. The vehicles can make use of a variety of sensors to get a more complete understanding of their environment, and adjust their tactical commands in accordance with what is deemed safe on an operational level. More offensive tactical profiles thus pose a harder task on the operational systems and, in turn, more capable operative systems allow for more offensive tactical profiles at the same level of safety. Offensive tactical profiles can be deemed necessary in case of emergencies, e.g., in self-driving police cars or ambulances. Beyond this, motorsports take the notion of offensive tactical profiles to the extreme, requiring cars to be driven at their physical limit with safety being much less of a concern than in regular traffic.

[section on the advantages of self-learning instead of hard-coding (e.g. flexibility, number of scenarios possible to be covered, ever-learning algorithms) + advances in deep reinforcement learning]

**The current project**

In this research project, we aim to maximize a self-driving car’s operational capabilities by using deep neural networks as the controlling entity. As a learning environment, we implemented a racing simulator with the Unity game engine that uses a deep neural network (DNN), implemented with TensorFlow, as the operational command system in full control of accelerating, braking, and steering. To this end, suitable representations of ground-truth data on the car’s position and its driving dynamics are streamed to the DNN, and corresponding control commands are received from it in return. We use a combination of supervised and reinforcement learning to teach the car to complete the race tracks as fast as possible. Our approach also entails an adaptation to changing conditions, such as the tires’ grip level, without the necessity of learning from scratch.

The nature of this task holds high demands with respect to the used hardware, especially GPUs. During training, reinforcement learning requires vast amounts of training data and a very high number of training steps. Moreover, during inference, the incoming data from the simulation needs to be evaluated within milliseconds, before the resulting control commands are streamed back to the simulation for the next frame. We therefore rely on very fast GPUs running the most recent versions of CUDA, cuDNN and TensorFlow to minimize compute times both in training and during inference.

**Outlook**

Insights from this line of research are likely to be applicable to real-world applications, be it in the realm of AI-driven race cars, or to enhance the capabilities and thereby safety of operational control systems in self-driving road cars. We hope you will consider this grant application favorably, as it would allow us to use larger and more powerful DNNs than currently possible with our CPU-focused workstations, which is a critical aspect in this domain.

Sincerely,

Gordon Pipa