Designed for Flexibility: A New Highway Simulation Environment for Inverse Reinforcement Learning Algorithms

Rachel Gologorsky[1]

Self-Driving Cars: The Road to Inverse Reinforcement Learning

Developing autonomous highway driving policies is a challenging yet necessary task for the development of truly autonomous vehicles.

Rule-based and behavior cloning approaches are agnostic with respect to *why* humans take certain actions. As a result, they are unable to detect and adjust to unexpected situations.

To solve this issue, Abbeel and Ng introduce a new framework: Model human driving behavior as being governed by a cost function of the state features (e.g. assume that humans intuitively demonstrate an optimal balance between speed and clearance to other vehicles).

Apply Inverse Reinforcement Learning (IRL) algorithms to reverse-engineer the cost function from human driving demonstrations.

The Need for a Flexible Highway Simulator

Applying IRL algorithms requires the use of highway simulators to collect the human demonstration data and to simulate the driving policies resulting from the reconstructed cost function.

- Highway simulators for general research and gaming purposes are generally **complex to use** and **difficult to tailor** to specific research scenarios.
- IRL researchers construct their own "toy" highway environments **specifically tailored to their research**.

Inventing - and Reinventing - the Wheel: Many Similar Toy Highway Environments

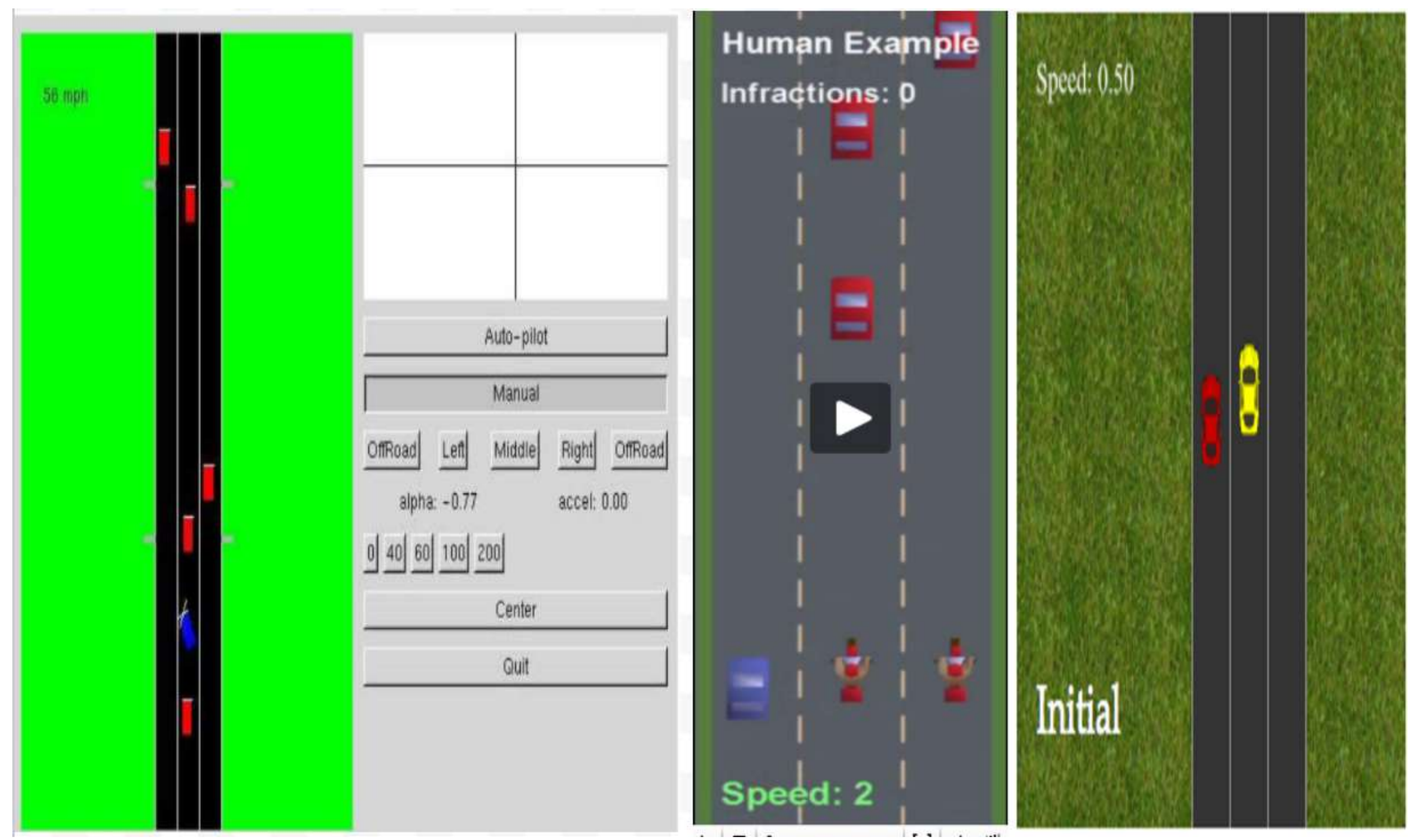
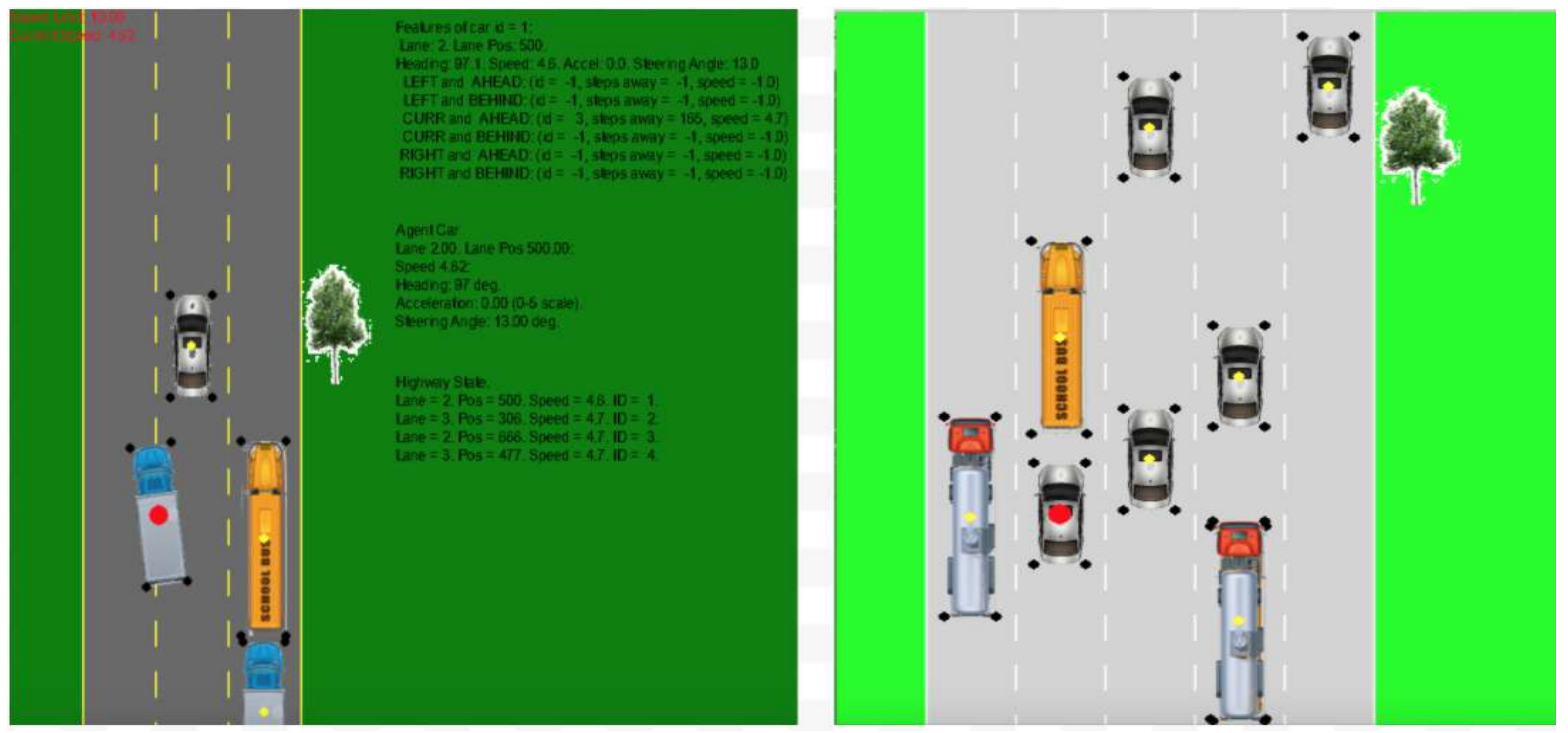


Figure 1. Many Specific Environments.
Left, Center, Right: Abbeel and Ng (2004), Levine et al. (2010) Sadigh et al. (2016)

Introducing a Flexible Highway Simulation Environment



```
Features of car id = 1:
 Lane: 2. Lane Pos: 500.
Heading: 97.1. Speed: 4.6. Accel: 0.0. Steering Angle: 13.0
  LEFT and  AHEAD: (id =  -1, steps away =  -1, speed = -1.0)
  LEFT and BEHIND: (id =  -1, steps away =  -1, speed = -1.0)
  CURR and  AHEAD: (id =   3, steps away = 165, speed = 4.7)
  CURR and BEHIND: (id =  -1, steps away =  -1, speed = -1.0)
 RIGHT and  AHEAD: (id =  -1, steps away =  -1, speed = -1.0)
 RIGHT and BEHIND: (id =  -1, steps away =  -1, speed = -1.0)

Agent Car:
Lane 2.00. Lane Pos 500.00;
Speed 4.62;
Heading: 97 deg.
Acceleration: 0.00 (0-5 scale).
Steering Angle: 13.00 deg.

Highway State:
Lane = 2. Pos = 500. Speed = 4.6. ID = 1.
Lane = 3. Pos = 306. Speed = 4.7. ID = 2.
Lane = 2. Pos = 666. Speed = 4.7. ID = 3.
Lane = 3. Pos = 477. Speed = 4.7. ID = 4.
```

Figure 2. One Universal Environment.
Left: agent = truck driving on a small highway, "night" effect. Features displayed.
Right: agent = sedan driving on busy "daytime" multi-lane highway. Features omitted.

Desired Flexibility	Resultant System Design: Modular, Functional
Custom Starting Scenario (Highway/Traffic Conditions)	**Car and Highway classes** have adjustable arguments, including: #highway lanes, highway lane width/length, #cars, car starting settings, car appearance.
Custom Car Behavior (enable "attentive" and "distracted" cars)	**AbstractCar class** encapsulates common car functionality/features so that individualized car behavior is implemented by specifying the move function and inheriting the rest of the functionality.
Custom Action Inputs (Compatibility with discrete/continuous, keyboard/mouse and wheel/pedal setups)	**Simulator class** interfaces between a car's state and action inputs. Action inputs can be easily customized/discretized by modifying the functions processing the raw input.

Additional Flexibilities

- Custom Data Recording
- Custom Drawing Options
- Custom Physics
- All Potential State Features Possible
- Unlimited Human Demonstration Time

Key mechanism: Total Information Sharing

1. A highway instance is required to initialize a car object.
2. At initialization, the car object adds itself to its highway instance's car_list field.

The highway object has a list of all car objects on it.

Every car object has, through its highway instance, access to all the other car objects on the highway.

Note: Since Python implements "Call by Reference," each car's highway instance always reflects the current highway object, and the highway object's car list reflects the the current car objects.

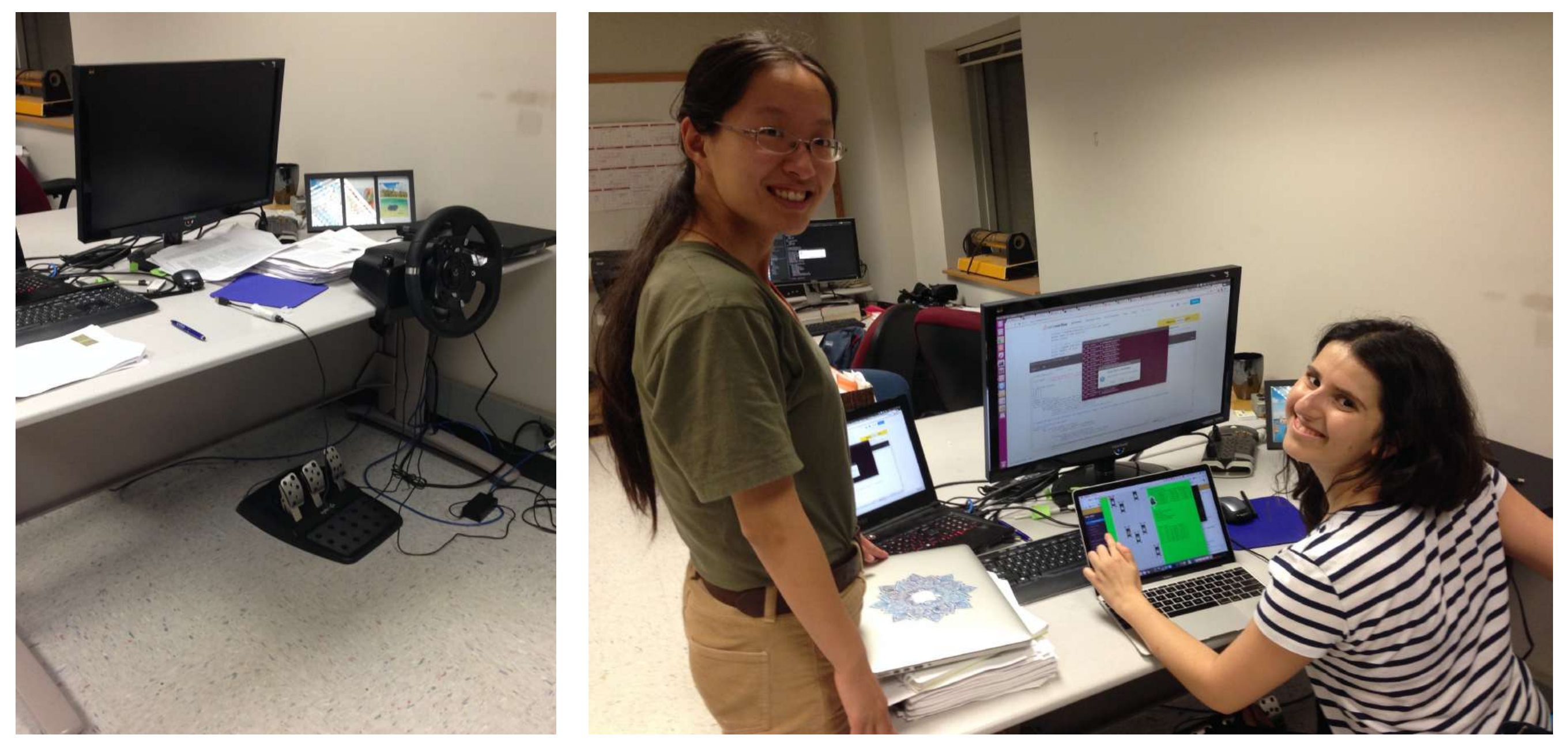
Benefits of Total Information Sharing

- All conceivable state features can be implemented (since all car and highway state information is available).
- Simplifies the implementation of car behaviors that depend on neighboring car state, and simplifies functions such as checking for collisions (simply scan the highway's car object list), projecting the highway state into the future, and playing back prior demonstrations.

Note: Since most scenarios include <20 cars, iterating through the car list is an efficient and simplifying design decision.

Future Directions

- An expanded car behavior library so that researchers could "plug and play" desired car behaviors into their research.
- An Obstacle class to incorporate common car-obstacle interactions and model lane merge and highway exit behavior.
- A Graphical User Interface to ease scenario creation.
- Environment extension to intersections and other situations.



Testing the Driving Simulator

References

[1] Abbeel, Pieter, and Andrew Y. Ng. "Apprenticeship learning via inverse reinforcement learning." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.
[2] Levine, Sergey, Zoran Popovic, and Vladlen Koltun. "Feature construction for inverse reinforcement learning." Advances in Neural Information Processing Systems. 2010.
[3] Information Gathering Actions over Human Internal State Dorsa Sadigh, S. Shankar Sastry, Sanjit A. Seshia, Anca Dragan IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2016

[1]Rachel Gologorsky is a 2017 Robotics Institute Summer Scholar, and is a joint concentrator in Computer Science and Mathematics at Harvard. This work was supported by a National Science Foundation Undergraduate Research Grant.

Acknowledgements

I would like to thank Dr. John Dolan and Dr. Katharina Muelling for their guidance and mentorship this summer (and for obtaining the driving simulator hardware), and Chiyu Dong, Yihuan Zhang, and Shuang Su for their friendship, which made the Learning for Behaviors Lab the setting of many enjoyable memories. Finally, I would like to thank Professor Joseph Blitzstein and Professor Harry Lewis of Harvard University. Without their incredible support, this research experience would not have been possible.