# City Scale Autonomy Learning

Sydney Gibbs, Mitchell A. Thornton , Darrell L. Young
Southern Methodist University, Dallas, Tx;

## ABSTRACT

Reinforcement learning for agent autonomous actions requires many repetitive trials to succeed. The idea of this paper is to distribute the trials across a city-scale geospatial map. This has the advantage of providing rationale for the trial-to-trial variance because each location is slightly different. The technique can simultaneously train the agent and deduce where difficult and potentially dangerous intersections exist in the city. The concept is illustrated using readily available open-source tools.

**Keywords:** Geographic Information Systems, Robotics, Simulation, Situation Awareness, V2X, V2I, I2V, Digital Twin, Collision Avoidance

## 1. INTRODUCTION

Each year, 1.35 million people are killed on roadways around the world[1]. Every day, almost 3,700 people are killed globally in crashes involving cars, buses, motorcycles, bicycles, trucks, or pedestrians. More than half of those killed are pedestrians, motorcyclists, or cyclist [1]. 38,824 people died, and an estimated 2,282,015 people were injured on U.S. roads in 2020[2]. Many of those collisions occurred at intersections due to reduced visibility caused by building blockage. We estimate the visibility blockage using Open Street Maps (OSM) for various cities. We use the blocking parameter in a Python Gym reinforcement learning environment to estimate the braking distance and resultant collision probability and risk.

## 2. METHODS

Figure 1A) shows typical Open Street Map (OSM)[1] street graph network. Figure 1B shows the building footprint polygons available in OSM.

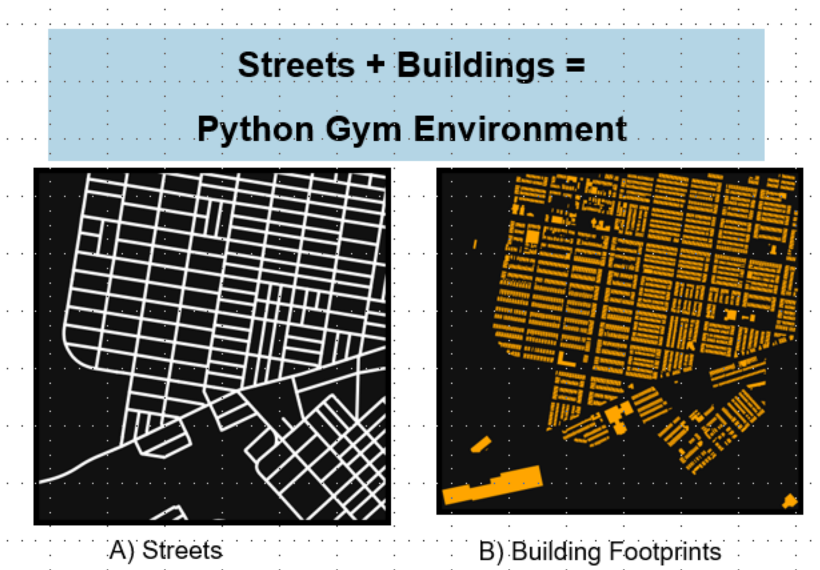


**Figure 1 A) Combining OSM street and footprint data enables calculation of visibility blocking parameter. Street graphs; B) Building footprints polygons. Map data copyrighted OpenStreetMap contributors and available from https://www.openstreetmap.org**

We use the OSM streets and building footprints to provide input conditions for modeling performance of the Python Gym environment for the purpose of collision avoidance using a novel combination of probability of collision and fatality risk.  Figure 2 illustrates how the blocking parameter extracted from OSM is used as a reaction delay in the reinforcement learning simulation.

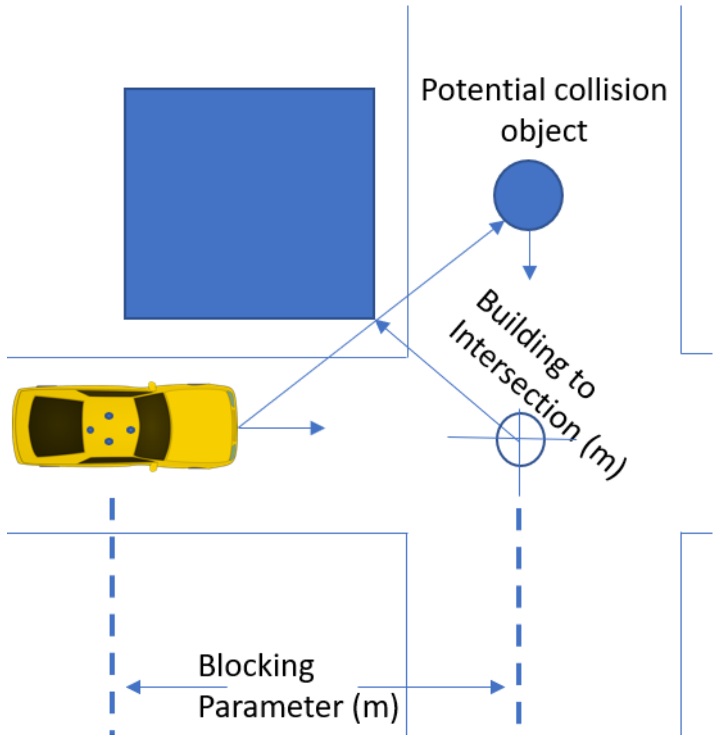


**Figure 2 Illustration of how the distance to building affects onboard sensor visibility**

In the simulations described below, the starting location is -100 meters in the x-direction from the intersection. The vehicle onboard unit (OBU) makes its own observations and estimates the location of the obstacle. After some total reaction delay the vehicle applies the braking action to avoid the collision. A collision is avoided if the vehicle comes to a stop before impact. If it does not come to a stop, the braking action deceleration reduces the impact speed and the resulting risk of a severe accident. In all the simulations, the braking deceleration is set to a constant 0.47g.  Actual braking performance is highly dependent on the vehicle, driver, and environment.

The estimated location error covariance of the vehicle, $\Sigma_v$, is typically much less than the collision object due to the vehicle's onboard GPS and inertial system. We arbitrarily fix the vehicle estimated position standard deviation to be 3 meters in the direction of motion and 1 meter in the cross-axis.  The covariance of the collision object is much larger based on sensing measurements described in [6].

The Python Gym environment consists of four parts: 1) Initialization defines the observation space including vehicle and collision object initial state, the action space consisting of either no change or acceleration change; 2) Reset defines the observation space and states for next trial; 3) Step takes in an action and returns a new set of environment states and observations; 4) Render shows the vehicle and collision object location with estimated error covariance in context of the intersection roadway.

The agent is trained using Stable Baselines 3, Proximal Policy Optimization (PPO) [7] with default parameters of policy=MlpPolicy, seed=0, batch_size=64, ent_coef=0.0, learning_rate=0.0003, n_epochs=10, n_steps=64. Once trained, the agent is evaluated against a rule-based procedure that sets the action to braking whenever the estimated collision risk is above a threshold.

After the vehicle applies the braking action, the vehicle comes to a stop at a braking distance of Dstop= $-\frac{u_v^2}{2a_b}$ . If the collision threat is less than the stopping distance, the impact speed at the center of the intersection is,

$$v_v = \sqrt{u_v^2 - 2a_b x_b} \tag{1}$$

The collision risk is estimated by multiplying the probability of collision based on the size of the objects by the probability of fatality based on speed at impact. We use the algorithm described by Lambert [4] et al. It uses a Monte Carlo (MC) algorithm to estimate the probability of collision that includes the shape of the objects to solve the problem of underestimating the collision probability of point distributions. We multiply the Lambert probability of collision by the Rosen and Sander[5] equation for risk of pedestrian fatality as a function of vehicle impact speed measured in km/h,

$$P_f(v) = \frac{1}{1+e^{(6.9-0.090v)}}. \tag{2}$$

The agent is trained to minimize the product of Lambert's probability of collision based on size of objects, and Rosen's fatality risk based on impact speed.

### 2.1 Results

Figure 3 shows the average distance from the center of the intersection to the nearest building for various cities in the United States. Cities like New York with less distance between buildings and intersection centers cause more visibility blockage.

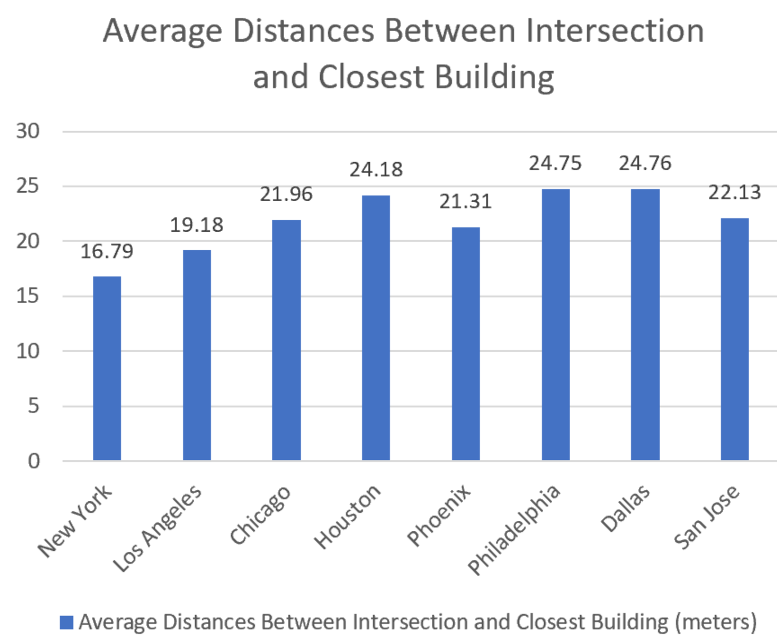


**Figure 3 Average distances from intersection center to nearest building for select US cities**

Table 1 compares the rule-based to machine learning outcomes. The rule-based outcomes fail when there is not enough warning. The machine learning outcomes, in effect, learn to slow down near the intersection, regardless of the presence of a warning. Both approaches reduce the collision risk by applying braking action to reduce impact speed. The initial vehicle speed was 20 m/s for all simulations. The RL agent learns to apply some braking as it approaches the intersection regardless of detecting a collision threat. One interpretation of this result is that autonomous vehicles will learn to slow down near intersections with historical collision threat activity. Additional description of the Python Gym simulation environment is provided in [6].

**Table 1. Comparison of rule-based to RL collision avoidance**

| | Learned Braking Policy | | | Procedural Braking Policy | | | |
|---|---|---|---|---|---|---|---|
| **Reaction Distance (m)** | Probability of Collision | Fatality Risk | Risk | Probability of Collision | Fatality Risk | Risk | Learned Risk Reduction (%) |
| **40** | 0.00125 | 0.001007 | 1.26E-06 | 0.010 | 0.001007 | 1.01E-05 | 800% |
| **50** | 0.0446875 | 0.001007 | 4.50E-05 | 0.355 | 0.001007 | 0.000358 | 795% |
| **60** | 0.053125 | 0.00173 | 0.000354 | 0.415 | 0.006773 | 0.002811 | 795% |
| **70** | 0.0965625 | 0.004681 | 0.001622 | 0.415 | 0.030318 | 0.012582 | 776% |
| **80** | 0.0965625 | 0.008746 | 0.003309 | 0.415 | 0.062747 | 0.02604 | 787% |
| **90** | 0.13944444 | 0.071723 | 0.029488 | 0.415 | 0.234801 | 0.097442 | 330% |
| **100** | 0.13944444 | 0.089713 | 0.036953 | 0.415 | 0.396517 | 0.164554 | 445% |

## 3. FUTURE WORK

Southern Methodist University Darwin Deason Institute for Cybersecurity (SMU-DDI) has recently established the Cybersecurity Autonomy Range (CAR) research facility[8]. Future work intends to address various aspects of autonomy security and information processing.

## 4. SUMMARY

Many useful parameters can be extracted from open-source Geographic Information Systems (GIS) like OSM to enhance the realism of machine learning simulations. We have shown how to extract the building visibility blocking parameter from OSM and include it in a Python Gym reinforcement learning simulation. Initial results indicate an adaptive response can be much better than a fixed rule. The adaptive reinforcement learning response learns to slow down near intersections to decrease impact speed of any potential collisions. This greatly reduces overall risk because impact speed is a non-linear function of fatality risk.

This paper describes preliminary research which should not be used without independent verification. As with all safety-critical applications, extreme care and extensive testing must be done before deployment.

## REFERENCES


[1] Contributors, Open Street Maps, https://planet.osm.org/ (10 May 2023).

[2] Kabil, A., Rabieh, K., Kaleem, F., & Azer, M. A. (2022). Vehicle to Pedestrian Systems: Survey, Challenges and Recent Trends. IEEE Access, 10, 123981-123994.

[3] U.S. Department of Transportation, Bureau of Transportation Statistics, Transportation, Statistics Annual Report 2022 (Washington, DC: 2022). https://doi.org/10.21949/1528354

[4] Lambert, A., Gruyer, D., Saint Pierre, G., & Ndjeng, A. N. (2008, October). Collision probability assessment for speed control. In 2008 11th International IEEE Conference on Intelligent Transportation Systems (pp. 1043-1048).

[5] Rosén, Erik, and Ulrich Sander. "Pedestrian fatality risk as a function of car impact speed." Accident Analysis & Prevention 41, no. 3 (2009): 536-542.

[6] Young, Gibbs, Thornton, Adding RF Situational Awareness to Robotic Simulation Systems, SPIE 2023, Defense and Commercial Sensing Symposium

[7] Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., & Dormann, N. (2019). Stable baselines3

[8] Young, Thornton, et al., (2022, October). SMU-DDI Cyber Autonomy Range. In 2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). IEEE.