Stanford University

High-Speed Autonomous Driving through Unknown Map

Xiaobai Ma (maxiaobai@stanford.edu), **Zhenkai Wang** (zackwang@stanford.edu), **Siyan Guo** (guosiyan@stanford.edu)
Advised by Marco Pavone, Lucas Janson

Overview

Based on the pioneering study done by the MIT Robust Robotics Group, our team reproduced and improved the algorithm to navigate a vehicle through unknown maps with high speed while avoiding collision. The algorithm extracts features from vehicle configuration, observed map, and potential actions to predict future collision probability and select the optimal path to goal.

Model

Training Procedure

- An arbitrary set of n-cell occupancy maps, $\boldsymbol{M}$, with 10cm resolution is used as background.
- Generate training data: (1) **observed features**, $\boldsymbol{\phi}$, (2) **indicators of collision**, $\boldsymbol{y}$.
  - i. A feasible configuration, $\boldsymbol{q}$, is randomly generated;
  - ii. Given $\boldsymbol{q}$ and partially observed map, $\boldsymbol{m}$, randomly choose an action, $\boldsymbol{a_t}$, and extract feature $\boldsymbol{\phi_t}$;
  - iii. If there exist three consecutive actions $(\boldsymbol{a_{t+1}}, \boldsymbol{a_{t+2}}, \boldsymbol{a_{t+3}})$ that does not lead into a collision, then $\boldsymbol{y} = 0$; otherwise, $\boldsymbol{y} = 1$.

Testing Procedure

The learned information is used to navigate the vehicle through unknown environments by selecting the actions that minimizes a specifically designed cost function:

$$\boldsymbol{a_t^*(s_t) = argmin_{a_t}\{J_a(a_t) + h(s_t, a_t) + J_c \cdot f_c(\phi(s_t, a_t))\}}$$

, where $\boldsymbol{J_a}$ = time duration of $\boldsymbol{a_t}$;
$\boldsymbol{h}$ = heuristic function, time-to-goal;
$\boldsymbol{J_c}$ = collision penalty factor, 1.0;
$\boldsymbol{f_c}$ = probability of collision.

Algorithm repeats until an inevitable collision state (ICS) or the goal is reached.

Features

Four features were selected for the non-parametric Bayesian inference model, as suggested by the MIT study, with modifications:

a) **Minimum distance to the nearest known obstacle along the action;**
b) **Mean range to obstacle or frontier in 60° cone ahead of the vehicle at the end of the action;**
c) **Straight free path length ahead of the vehicle at the end of the action;**
d) **Speed at the end of the action.**

Results

Training Procedure

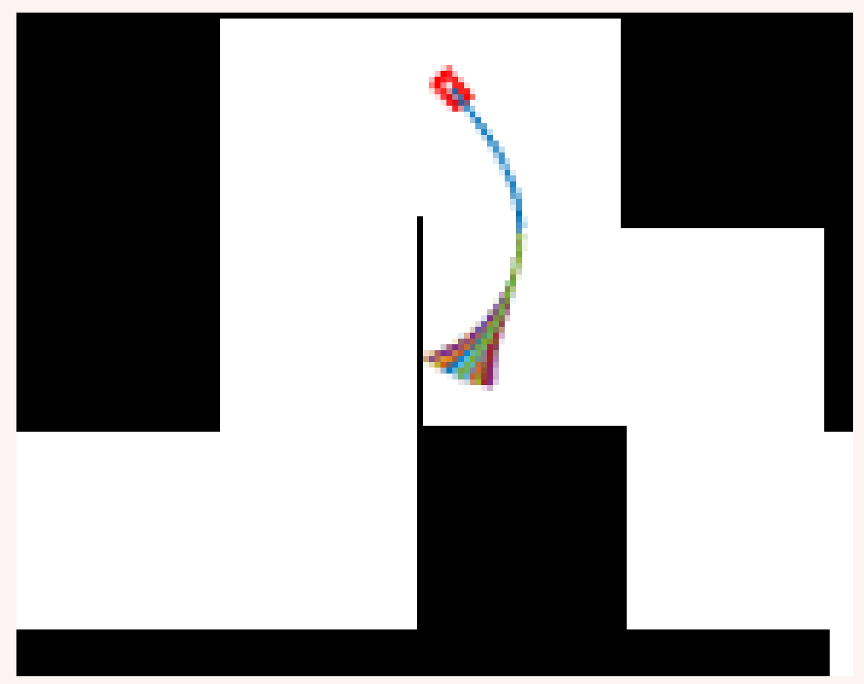

Figure 1. Training examples with $y = 1$

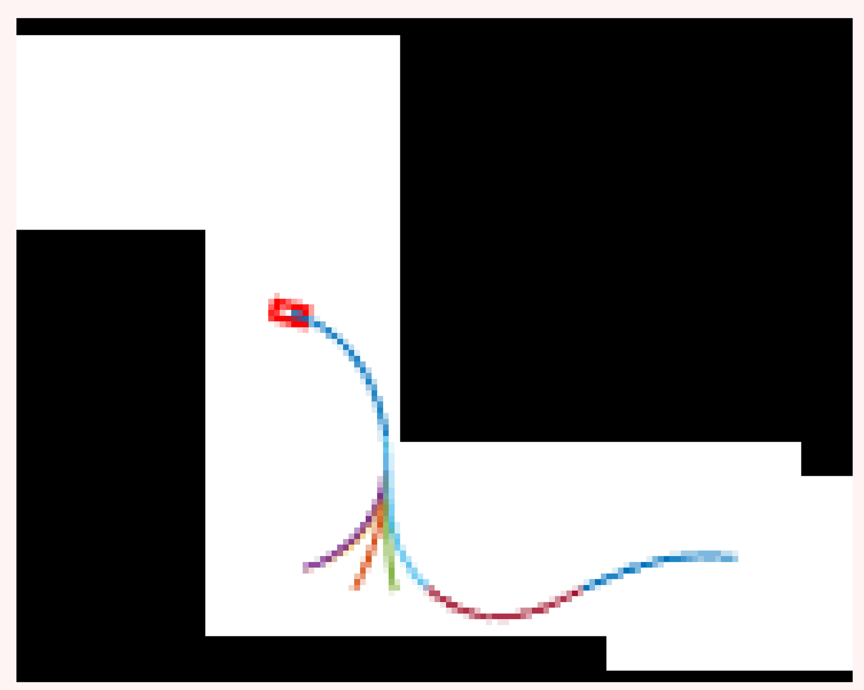

Figure 2. Training examples with y =0

Testing Procedure

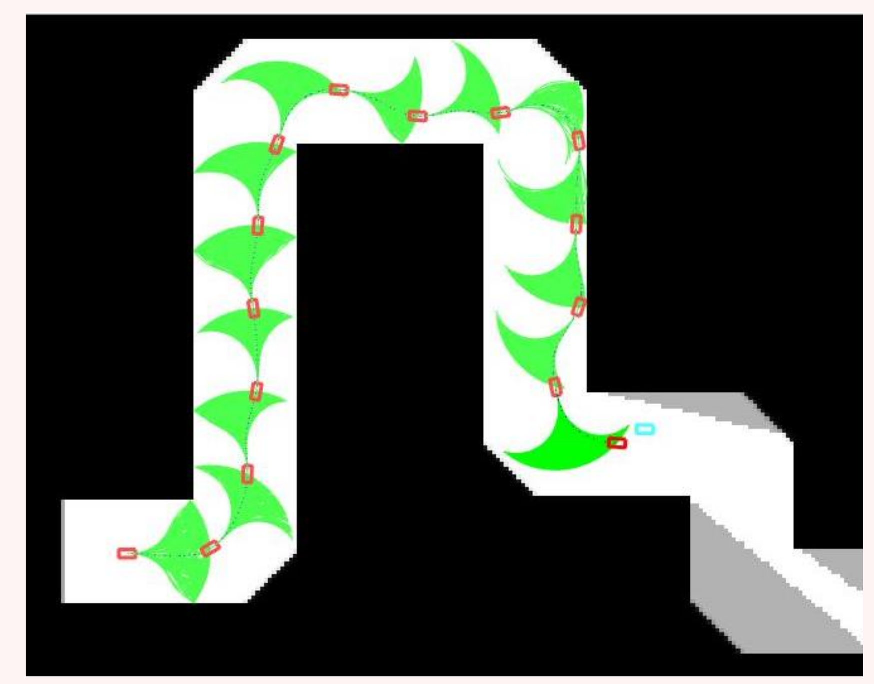

Figure 3. Baseline planner performance

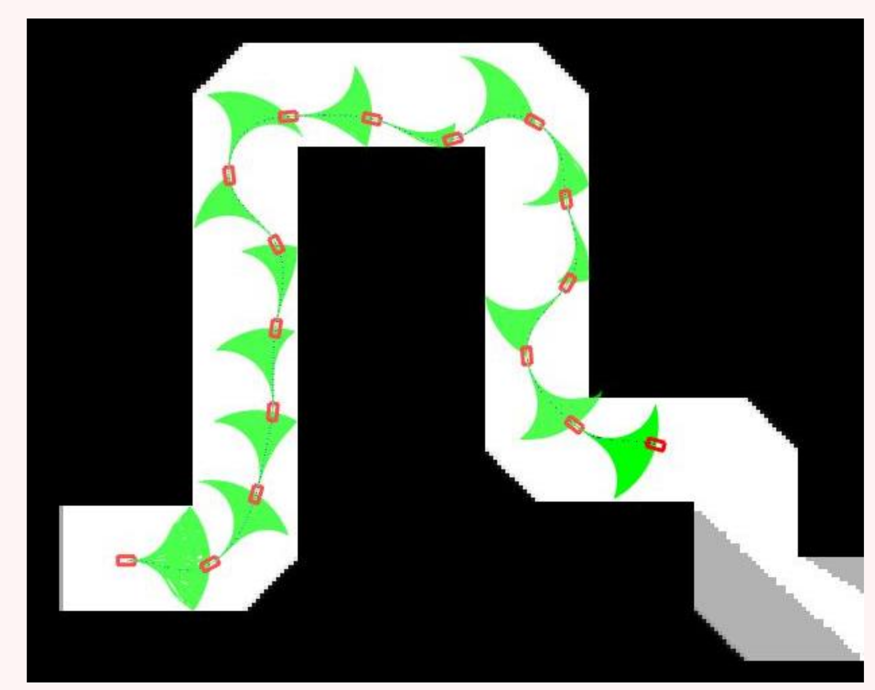

Figure 4. Learned planner performance

Learned planner takes much shorter time than the baseline planner to reach the goal location while maintaining collision free.

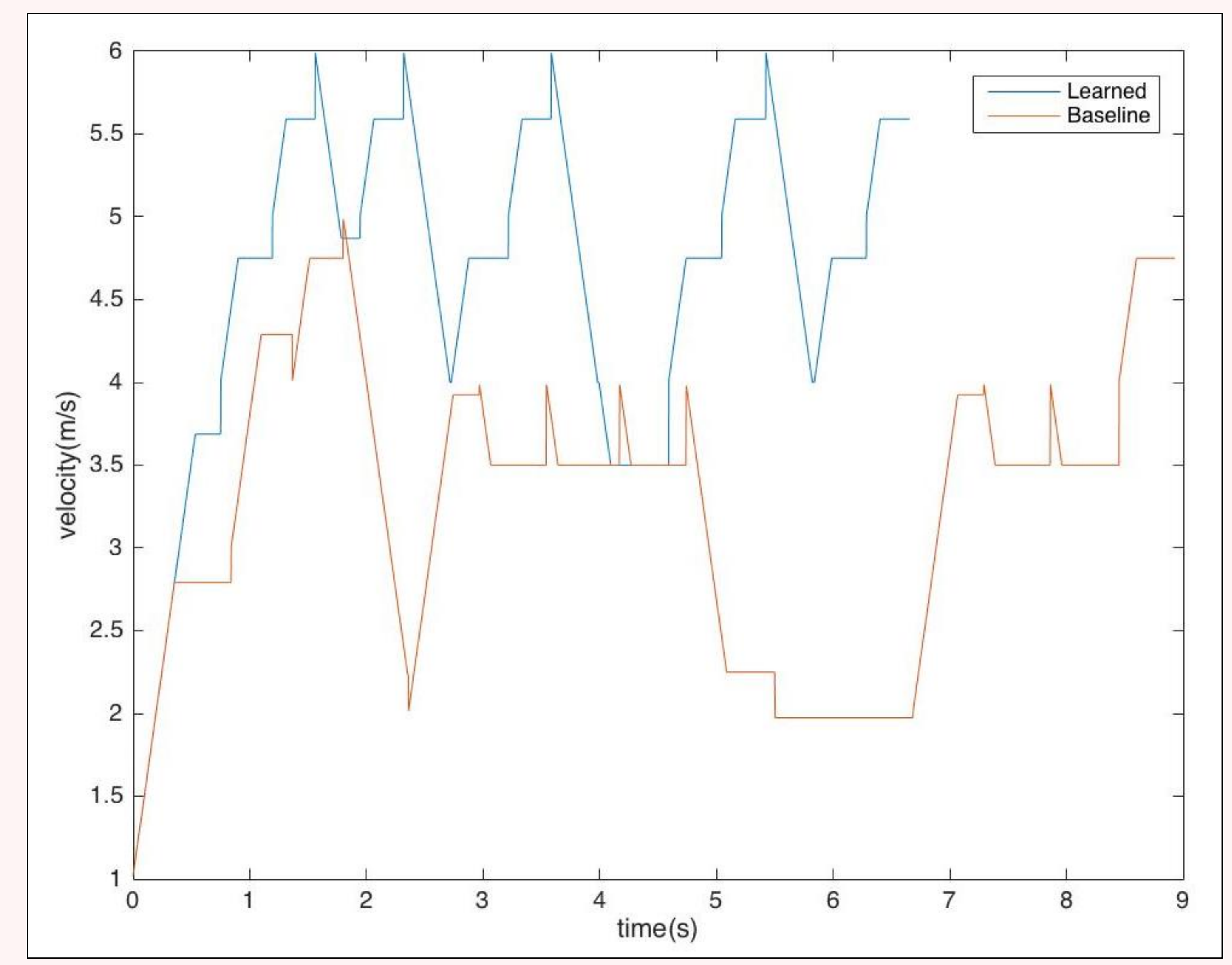

Figure 5. Velocity of learned planner compared to baseline planner

Future Work

- Parameter optimization: J_c, action sample resolution, etc.
- Feature selection techniques application.
- Training on different type of maps: maze, forest, etc.
- Reinforcement learning application.

References

Charles Richter, William Vega-Brown, Nicholas Roy, "Bayesian Learning for Safe High-Speed Navigation in Unknown Environments," in International Symposium on Robotics Research, 2015.

Thomas M. Howard, Colin J. Green, Alonzo Kelly, "State Space Sampling of Feasible Motions for High-Performance Mobile Robot Navigation in Complex Environments," Journal of Field Robotics, vol. 25, no. 6-7, pp. 325-345, 2008.