EE392A

Intelligent ground vehicle motion planning using demand-based mapping

Under Graduate Project

Shubh Gupta (14670)
Supervisors:
Prof. Mangal Kothari, AE
Prof. S. R. Sahoo, EE

Abstract

The project aims at developing a pipeline for combining motion planning and ground-plane semantic mapping, with both assisting each other. Map generation queries are performed on image patches using online learning classifiers to accommodate for concept drift. The implementation is done in ROS frame-work, with code parts in both C++ and python.

Introduction

- **Motion Planning:** Produce a continuous path that connects a start configuration and a goal configuration while avoiding passing over obstacles.

- **Semantic mapping:** Generation of an abstraction of space for perception of environment by robot.

- **Combined planning and mapping** for faster path generation and robust semantic mapping.

Motivation

- Sequential approach to semantic mapping and motion planning is **slow**

src: The Open Motion Planning Library

Motivation

- **Data** for label generation via patch classification is easier to obtain than full image classification, and is more **flexible** than classical approaches

Motivation

- **Changing environment state** in semantic mapping requires methods capable of adapting.

src: Shutterstock

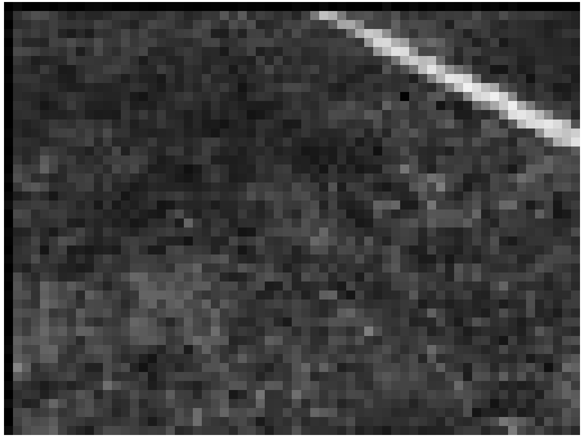
Sensor Modelling

- Image is acquired using a camera mounted over a ground robot

- Each image contains patches belonging to multiple ground grid-cells

- Probabilistic modelling of measurement for each grid-cell using classifier confidence

Sensor Modelling



Patch-wise confidence-measurements over image

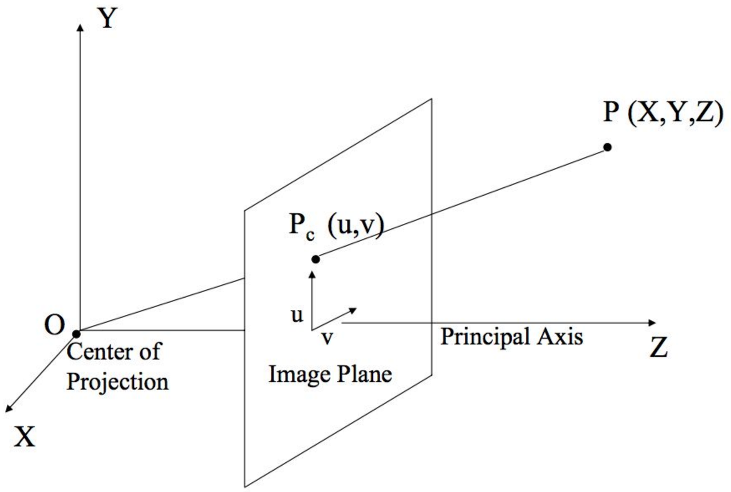
Sensor Modelling

$$b_k^{m^i} = \frac{p(m^i|z_k^i,t_k)p(z_k^i|t_k)p(m^i|z_{0:k-1}^i,t_{0:k})}{p(m^i)p(z_k^i|z_{0:k-1}^i,t_{0:k})}$$

where, $b_k^{m^i}$ denotes belief value of grid-cell m^i after k frames, z_k^i denotes measurement on ith grid-cell from kth frame and t_k denotes camera position during kth frame.

Ground to image projection

- Mapping using Camera projection model



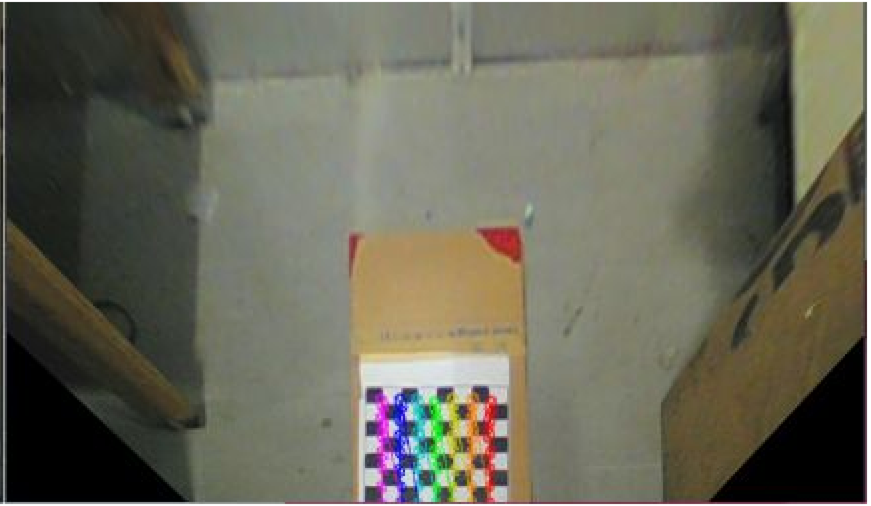
src: https://prateekvjoshi.com/2014/05/31/understanding-camera-calibration/

Ground to image projection

$$P_h = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_x f & 0 & m_x c_x \\ 0 & m_y f & m_y c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where, P_h is a vector of homogenised pixel coordinates, f is camera focal length,
$({}_x, m_y))$ are scale factors, (c_x, c_y) is image origin offset,
r_{ij}, t_i are extrinsic rotation and translation parameters and
X, Y, Z are real world coordinates

Camera calibration

Methodology

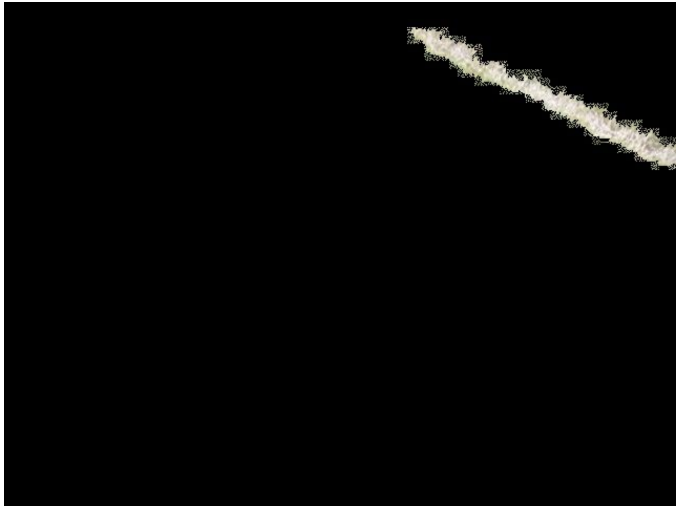
- **Classifier initialization:** Train a classifier with initial estimates of traversable and non-traversable paths.
- **Motion planning:** Generate queries of specific grid-cells for map updation and cost determination to plan a path.
- **Mapping:** Update the map voxel beliefs based on classifier response.
- **Classifier update:** Update the classifier to incorporate for concept drifts

Classifier initialization: Clustering

- Mean Shift Clustering with gaussian kernel to reduce noise and ease lane-grass separation.

Classifier initialization: Data generation

- Threshold based on max value to get mask
- Morphological operations of closing and erosion to extract lane
- Patch generation using sliding window



Motion Planning

- Goal configuration chosen as a circle of radius of 3 grid-cells

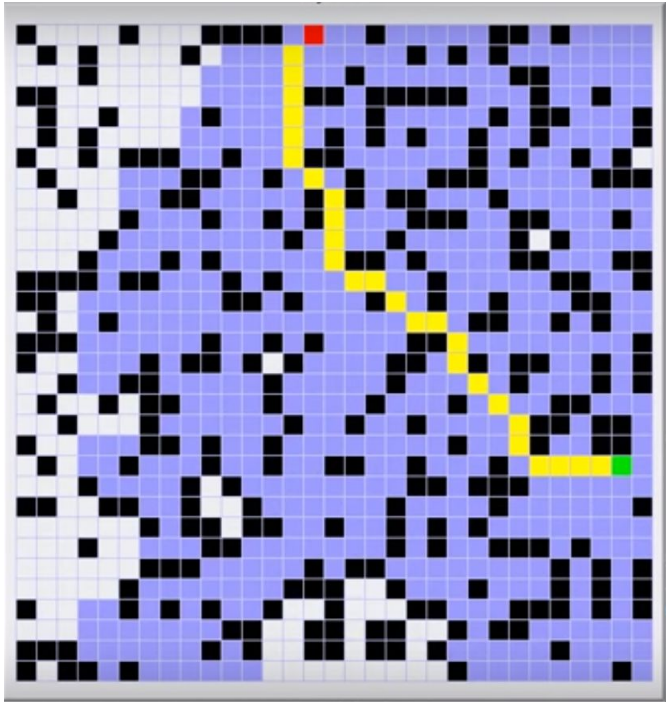
- Two algorithms - **Dijkstra's algorithm** and **A* algorithm**

- Planning algorithm generates a **query** for a grid-cell

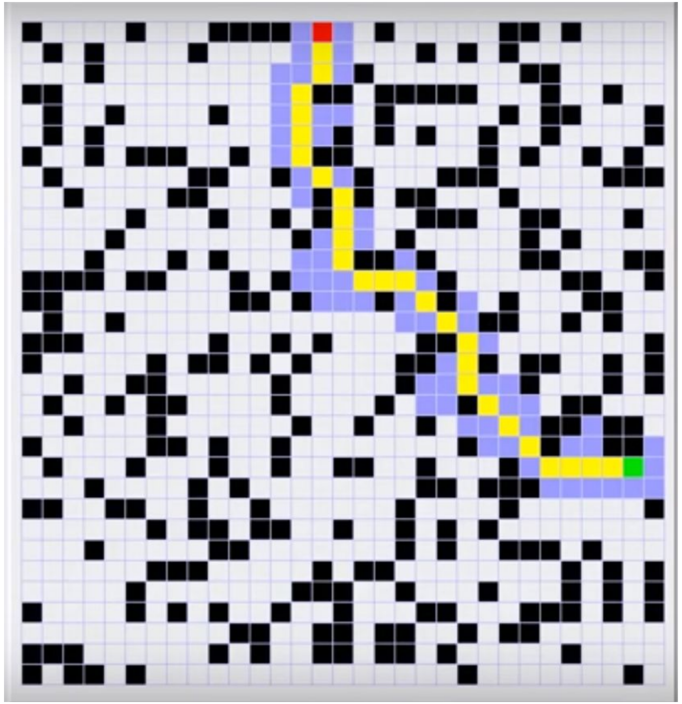
Motion Planning

Dijkstra's Algorithm	**A* Algorithm**
Favours exploration	Favours path finding
Slower computation	Faster computation
Minimizes cost	Minimizes cost and heuristic
Dense map	Sparse map

Motion Planning



Dijkstra’s Algorithm



A* Algorithm

src: https://github.com/kevinwang1975/PathFinder

Mapping

- Motion planner query for an unclassified grid-cell triggers classification for it and its neighbours

- Classification confidence used as measurement belief

$$b_k^{m^i} = \eta p(m^i | z_k^i, t_k) p(m^i | z_{0:k-1}^i, t_{0:k})$$

Classification and Update

- **Classifier:** Confidence-weighted linear classifier

- Patch gets the label of clear majority neighbours

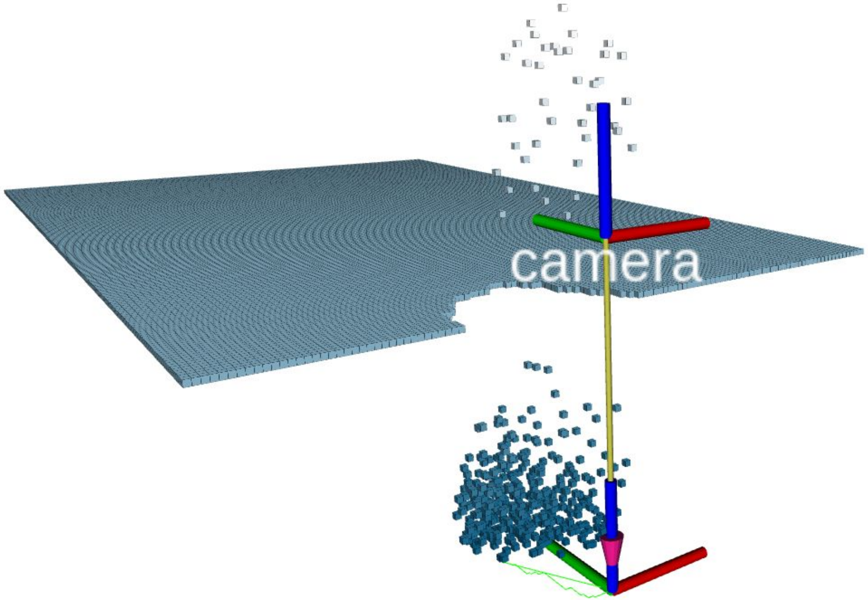
- Classifier update equations:

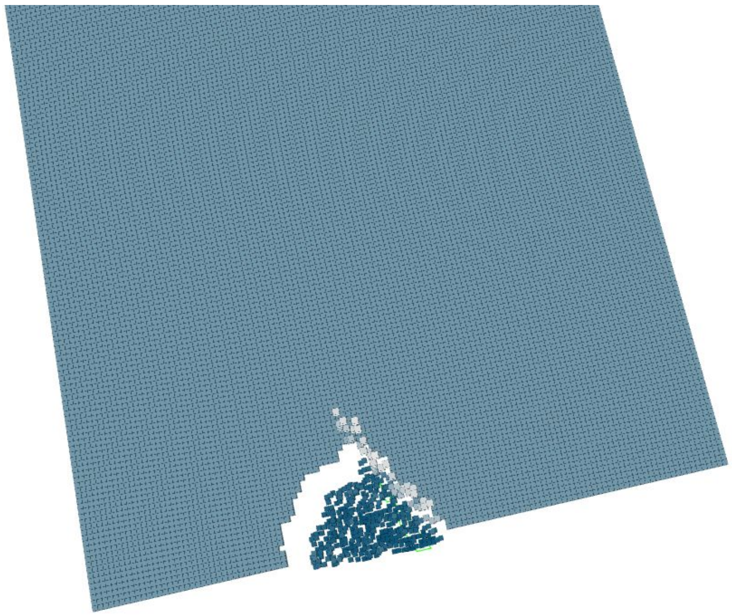
$$\mu_{i+1} = \mu_i + \alpha_i y_i \Sigma_i x_i$$
$$\Sigma_{i+1}^{-1} = \Sigma_i^{-1} + 2\alpha_i \phi diag(x_i)$$

Results

Input image

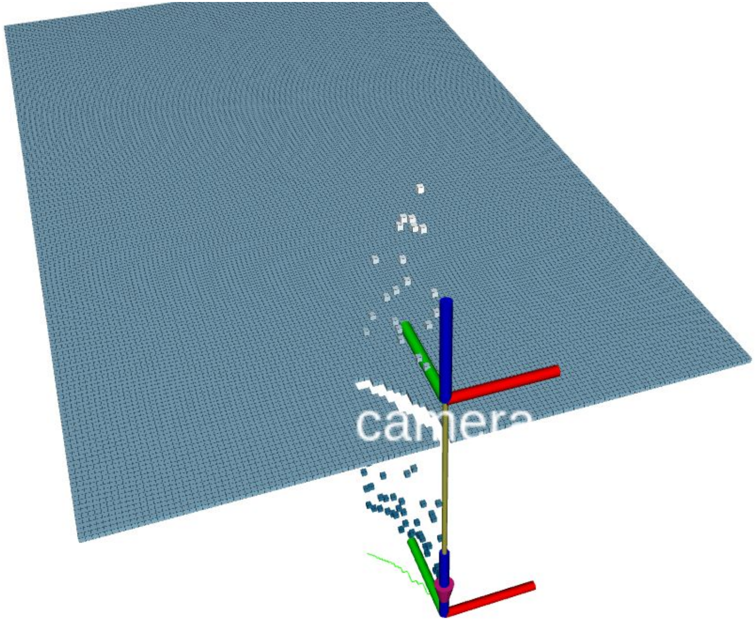
Results

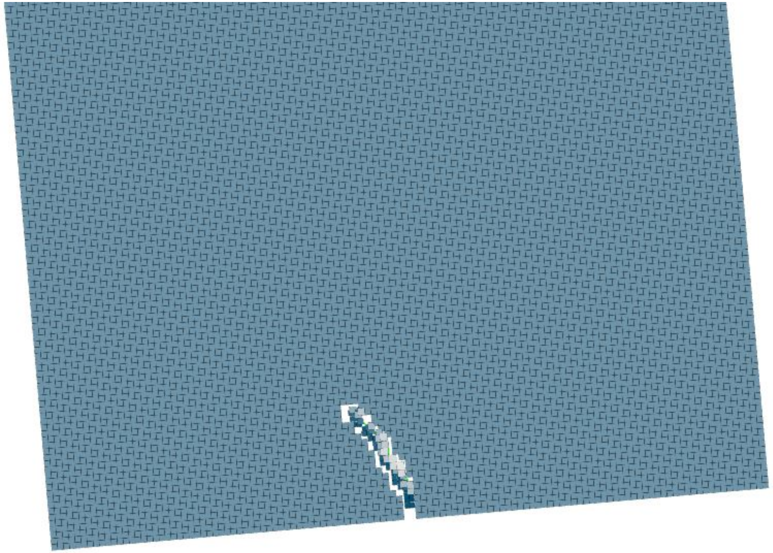




Dijkstra’s Algorithm

Results





A* Algorithm

Results

Algorithm	Time (seconds)
Online classifier (Full Image)	4.7
Offline classifier (Full Image)	21.2
Dijkstra's Algorithm	**3.4**
A* Algorithm	**0.6**

Algorithm Performance

Conclusion and Future work

- Incorporating for uncertainty in position of camera
- Handle non-ground plane objects
- Incorporate motion planning algorithms like RRT

References

- Ali-akbar Agha-mohammadi. *SMAP: simultaneous mapping and planning on occupancy grids*.CoRR, abs/1608.04712, 2016.
- Prof. Gaurav Pandey. *Lecture slides.EE698G*, Probabilistic Mobile Robotics,2016
- Yizong Cheng. *Mean shift, mode seeking, and clustering*.IEEE Transactionson Pattern Analysis and Machine Intelligence, 17(8):790–799, Aug 1995
- Mark Dredze, Koby Crammer, and Fernando Pereira. *Confidence-weighted lin-ear classification*. In Proceedings of the 25th International Conference on Ma-chine Learning, ICML ’08, pages 264–271, New York, NY, USA, 2008. ACM.

Thank You