Jacob Kilver
CS7637: KBAI
Assignment 3
19 June 2016

Introduction

The automobile industry is hotly pursuing autonomous vehicle technology. Planning – choosing a sequence of actions to accomplish a goal – is a major part in creating useful autonomous vehicles. This paper explores the details and difficulties of creating a planning algorithm for such vehicles.

Problem Description

The first (and probably simplest) part of creating a planning algorithm is setting a goal. For an autonomous vehicle this will be a set of latitude and longitude coordinates and potentially a directional heading. The second input to the planning algorithm is the initial state. This will be a tuple like the goal state with latitude and longitude coordinates, but the directional heading is required. Since the vehicle does not operate in isolation, the initial state must also contain information about the world such as the obstacles in the vicinity of the vehicle, the speed of these nearby obstacles, and the current road conditions. This presents a difficulty not only because this is a large amount of data, but also because sensors can be unreliable. In lecture it was implicitly assumed that the current state of the environment could be known completely and reliably. In practice this is rarely if ever the case. The planning algorithm must be able to account for data that can be inaccurate or missing at times.

The next input to the planning algorithm is a set of actions that are allowed when transitioning between states. In lecture these were called operators. For the purposes of this discussion the operators for an autonomous vehicle will be restricted to the following: forward, left, and right. Details on these operators can be found in Table 1. These obviously do not cover all the actions that vehicle can take, but it should be adequate for the discussions here. Just as with the uncertainty in sensors, there is uncertainty when taking actions. For instance, the vehicle might predict that it has travelled so far in a certain period of time when in reality it has travelled more or less than that amount. This was also not covered in lecture; when a block was commanded to be moved it was assumed that the block was moved to the specified location. A planning algorithm for an autonomous vehicle cannot make such assumptions and must be able to handle uncertainty in the execution of its operations.

Table 1: Simplified operators for an autonomous vehicle

Operator	Precondition	Postcondition
Forward	• The road must continue in the direction of the current heading	• Robot will be one block further along in the direction it is facing or until the road ends
Left	• Robot must be at an intersection • Road must be present on left	• Robot will be on street 90 degrees CCW • Heading will be 90 degrees CCW
Right	• Robot must be at an intersection • Road must be present on right	• Robot will be on street 90 degrees CW • Heading will be 90 degrees CW

Finally, the vehicle has to have some background knowledge in order to connect the initial state to the goal state using the allowed operators. For a vehicle this would simply be a map. This map adds some additional constraints to the problem like traffic flow and location of roads.

With these pieces in place the vehicle must then decide on the sequence of actions that best achieve its goal of reaching the destination location. Take Figure 1 as an example. The goal is to navigate from the initial state of Home to the goals state of Restaurant. How can this be achieved?

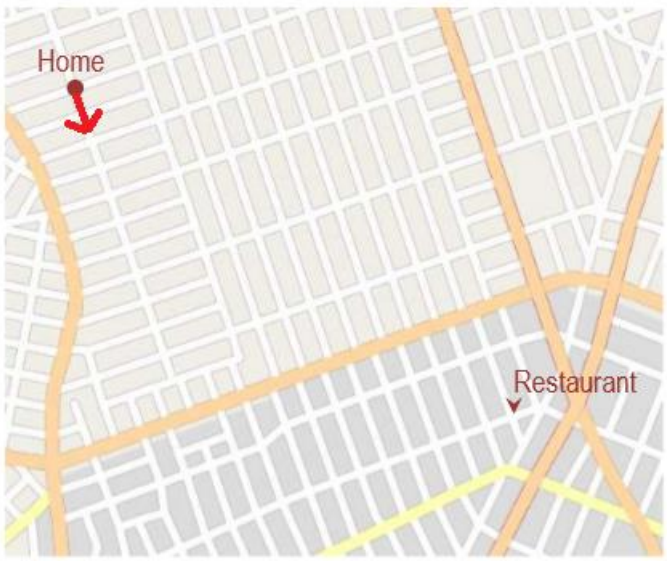

Figure 1: Example of route planning problem

Approach

The goal state will be input by the passenger(s), most likely as a street address. It is assumed that a separate system will be able to translate the street address into the goal coordinates that the vehicle can understand. There are numerous techniques to handle uncertainty in both measurements and movements including Kalman and particle filters. These methods are able to filter out the noise and uncertainty from sensor readings, leaving only the reliable data behind. While these techniques will not be addressed any further here, they do justify the assumption that the vehicle can, with a great deal of certainty, know its location and the location of objects surrounding it.

Now to address planning. Take the example in Figure 1 again and let us add to the goal state that the vehicle should be on the same side of the street as the restaurant. In terms of propositional logic discussed in lecture the goal state becomes

$$LocationX(restaurantX) \wedge LocationY(restaurantY) \wedge (Heading(north) \vee Heading(west)).$$

Meanwhile the initial state is

$$LocationX(homeX) \wedge LocationY(homeY) \wedge Heading(south).$$

From this knowledge we can extract a distance (probably Manhattan in this case rather than Euclidean since this is an urban area). To derive a path the vehicle will use a means-end technique coupled with its knowledge of the goal. For planning it sometimes helps to solve the problem backwards. This technique

is referred to as dynamic programming. Rather than plan from “home” to “restaurant” the robot instead plans from the “restaurant” to “home.” In this case the robot has two equivalent goal states: at the restaurant either heading north or heading west. Meanwhile there are many paths to reach the restaurant from home while not fulfilling the requirement on the heading. It could be the case that means end arrives at a solution where the vehicle is at the proper location but the heading is such that the route must be completely revised, such as if the robot reached the restaurant but was facing south. By starting at the goal and working backwards, dynamic programming ensures that the goal state will be reached without having to do re-planning. This is a way of managing conflict in planning by ordering the operators in such a way to avoid conflict altogether.

For dynamic programming means-end, the algorithm first does a search step: apply each operator (but now in reverse) to the current state (the goal state) and compare the resultant distances from this new location to home. Next the algorithm prunes: it selects the operator that results in the minimum distance. If multiple operators result in the same minimum distance, the planning algorithm can break ties in any number of ways but how it chooses is not important so long as the minimum distance is still achieved.

The method above will result in a stationary plan that the robot can follow successfully *if its knowledge about the world is complete*. In other words, as long as its map is accurate then planning a route is simple. But what if the automobile finds it cannot execute the plan it devised? For instance, what if it finds that a road on its desired route is closed for construction? As anyone who has used a GPS before knows, it is very common for maps to be out of date, and thus the need for a conditional plan arises. A conditional plan is essentially a plan with if statements built into it. These if statements could be computed on the fly, much like many navigation apps do today. The algorithm is constantly thinking “if the pre-condition of my next operator cannot be fulfilled, what operator should be chosen next?” This is essentially a one-step look-ahead. While the search space will be larger than a simple fixed plan, the search space will not be as vast as a complete deep search. Note that this assumes that the background knowledge of the world is still partially correct. If the map is so wrong that it hinders more than it helps, the robot will not be able to do any better than a blind search.

Conclusion

Planning is essential to navigation, but it is not a simple task. The world is a very complex place and understanding it in real-time is difficult especially when incoming data is not always reliable. Techniques like Kalman filtering can help in eliminating noise from sensor inputs. However, these methods are not able to account for knowledge inaccuracies like an outdated map. Fixed planning needs extended with conditional statements to account for these gaps in knowledge by having a pessimistic outlook on the likelihood of the plan to be executed.