Uber Homework Assignment

Report By:
Arihant Lunawat
Georgia Institute of technology

November 29, 2015

Problem Statement

This world is full of pedestrians trying to get to the other side of the road at a 4-way intersection. The problem is to make their actions look as human as possible. In this world there are 4 corners, crosswalks connecting those corners, and traffic signals for each of the crosswalks. Each actor (pedestrian) has a target corner that it is trying to reach. When it gets there it will receive a new target corner. This world should run forever with the actors constantly trying to get across the road.

My Approach / Flow of Thought

Most motion planning algorithms with a known start and goal locations can be solved using well-known algorithms such as D*, LPA* or other planning algorithms. This would involve building a map of the world (which is fairly easy, because the complete world is visible to the actors), finding the optimal path using prioritized expansion of map nodes and then taking that path. But this problem, specifically, is very deterministic because there are finite number of actions an actor can take. This made me think about finite-state decision making algorithms.

High-Level Design

I have implemented a finite-state controller for the decision making of each actor. Each agent determines if the target corner is across a crosswalk or diagonally across the street.

- If the target corner is across a crosswalk, then the agent simply waits for the signal to turn green and walks across.
- If the target corner is diagonally across the street, the actor walks across the crosswalk which is currently green. After that, the actor waits for the second crosswalk signal to turn green and walks across to the target corner.

Low-Level Behavior of Actor

The actor has the following behavioral characteristics,

- Make decisions based on above mentioned algorithm
- Obey traffic signals
- Avoid collisions with fellow actors (pedestrians)
- Walking Speed

Make decisions

Each actor has its own reference corner and a temporary target corner. The reference corner is the last visited corner and the temporary target corner is the one it is walking towards. If the actor is at its reference corner, it starts moving towards its temporary target corner as soon as the light turns green. Once on the crosswalk, it continues walking towards the temporary target. If the temporary target is the actual target, then it stops and waits for a new assignment. Otherwise, it reassigns itself a new temporary target and starts moving toward it until it has reached the actual target.

The decision to select a temporary target has to be made at the most twice, because, any given combination of start and target corner will be at the most two crosswalks apart.

Obey Traffic Signals

The actor will leave a corner only if the signal state is green on the crosswalk connecting the reference corner and the temporary target corner. If it leaves on a yellow, it has no guarantee that it can make it across the crosswalk in time, so this situation is avoided in the given solution.

Avoid Collisions

The actor tries to avoid collisions with other actors who are within its private space. The actor computes a vector from itself to each of the actors and tries to move in the exact opposite direction of the resultant sum of vectors. This vector is then superimposed to the target velocity vector of the actor (after normalizing both) and the resulting direction is returned as the displacement vector of the actor.

Three parameters play a key role here,

- Size of private space (taken as one unit distance for each actor in this solution)
- Weights to actual velocity and weights to resultant collision avoidance velocity
- Strength of repulsion as a function of distance

The weights determine the contribution of each velocity to the resultant displacement velocity vector of the actor. The strength of repulsion is modeled as a negative exponent, i.e.,

$$repulsion = e^{(param * dist)}$$
$$param \ \epsilon \ (-\inf, 0]$$

Any actor within the private space will be repelled. The strength is uniform if $param = 0$ and it is decaying with distance if $param < 0$, the decay is stronger for more negative value of param.

Walking Speed

Each individual pedestrian is expected to have a unique walking speed. Research has shown that the average walking speed of humans is 1.34 m/s. Each actor has a walking speed of (1.34 ±0.1) m/s in order to capture this uniqueness. At the lower end, we have people with disabilities and at the higher end, we have people walking dogs, for example.

Evaluation/Conclusions

With the high level decision making and low-level behavior of actors as described above, the actors seem to be moving successfully to the target corner.

- Collision avoidance is working, but it is not very clean. The actors sometimes collide when they are inside a corner and stationary. When they are on the crosswalk, they tend to avoid colliding into others very nicely and effectively

- The actors do not break traffic lights and take the shortest path to the target corner. This implicitly results in queuing behavior at the corners
- The variance in walking speed models the uniqueness of the individual actors
- The movement of actors is very smooth and not jittery (it can result from finite-state algorithms)
- The solution is scalable to large number of agents (discussed below)
- The solution is completely distributed. No actor instructs or controls the actions of any other actor. Every actor acts for itself. Distributed action capability makes the solution scalable to a large number of actors

The optimum number of agents for my solution is around 5-15 agents. Due to the small size of the world, fewer agents make the simulation look cleaner.

As a measure of my competence and fitness to the company's needs, I have written the solution to the problem in less than 8 hours (1 working day). Below, I have also discussed the issue of scalability with respect to personal observations and further possible additions to the solution submitted.

Build and Execute Instructions

I have not used any extra libraries. The source code compiles as it is. Below are the steps,

- Change directory to the source code: cd path-to-src/ped-homework
- Run makefile: make
- Run output file: homework-sim

Scalability

In theory, this solution is infinitely scalable, as long as there is enough space in the world and there is enough processing power. I was recently in New York, where I observed a huge number of pedestrians trying to cross the streets. There were fast walking people (residents of the city) and slow walking people (most of them tourists). The pedestrians do cross the street in huge numbers, which results in the need for more sophisticated decisions to be taken. One of them is, the crosswalk might not be able to fit all the pedestrians trying to get onto it, and hence, portion of the crowd has to wait for the next green light. This behavior is implicit in the collision avoidance behavior. Another decision is, to dynamically switch the traffic lights, to match the number of pedestrians on each corner and their target corners with the use of a voting system (as can be seen on our streets in the US with the help of a button on each corner).

Further Additions

- Collision avoidance can be modeled using other functions and more sophisticated decision making algorithms. This has to be the first extension to the submitted solution and the most necessary in terms of feasibility of the solution
- Pedestrians should walk only on one side of the crosswalk so as to avoid collisions with people walking in the opposite direction. This is another immediate extension to the solution submitted
- Pedestrians can be differentiated in more ways than just their speed. Each individual can have a unique size of private space. They can have unique decision making boundaries with respect to the signal states based on their speeds. This is another immediate extension to the solution submitted

- I have not coded decisions for situations where pedestrians are on the crosswalk and the signal goes red. In that case, the pedestrians should increase their speed and get across to the closest corner. This problem could arise when there are a lot of pedestrians on the same crosswalk. When they are few in number, they easily get across, because they have at-least 10 seconds to get across. This is not an immediate extension, but an important consideration for the feasibility of the solution submitted

- A different approach to the problem can be taken. There are at-most three possible ways to go from one corner to another. The three possibilities can be ranked with respect to the current state of the world and the positions of other actors to minimize the time to travel and the distance of travel as an optimal control problem (the distance of travel may not be the most important thing!). The formulation of the problem (cost function) would be such that it implicitly accounts for all the favorable behaviors discussed in this report. I particularly like this method because it is a very natural way of formulating a problem and is also flexible in its solution.