Social Navigation of a Mobile Robot

Lahiru Nuwan (lnw8px), Jasmine Sekhon (js3cn)

1 Abstract

The goal of this work is to implement and analyze different methods that enable robots to navigate through crowded environments, such as those that may be encountered in shopping malls, road side walks, or industries. The two key methods considered here for planning a robot's trajectory safe from moving and static obstacles are neural-network based estimation of a human's trajectory and social forces method. We integrate these broad methods with other primitive shortest path-planning algorithms such as A* search to find a collision-free shortest path for the robotic agent through a crowded environment. To analyze the said techniques we use certain existing simulation environments and present a comparison between the approaches.

2 Introduction

Robots are increasingly being used in diverse kinds of workplaces and households to do certain tasks that were initially done by humans. In several other scenarios, robots are often used as cooperative agents with humans, to not substitute a human in a certain task, but instead assist actively. Nonetheless, the increasing acceptance of robotic agents in workplaces and households has led to greater possibility of robots coexisting in environments with human agents. This has made human-robot interaction an active area of research. In crowded workplaces, such as busy offices, airports, industries and even indoor shopping malls, robots may be required to perform tasks like loading or unloading goods, carrying packages, etc. while navigating through moving and static obstacles. Therefore, it is important for the robotic path-planning algorithms to actively consider moving obstacles and re-plan trajectories for collision-avoidance. There exists a large amount of literature that focuses on path-planning for robotic agents in a defined environment and collision avoidance with respect to moving obstacles and static obstacles. Over the last decade, several mobile robots have been deployed successfully in crowded environments such as museums, railway stations and exhibits. These robots are governed by a diverse range of path-planning and collision avoidance algorithms, ranging from simpler A* search [6] to more complex, adaptive approaches such as reinforcement learning and inverse reinforcement learning [5].

In our paper, we touch upon two moderately sophisticated methods for path-planning in environments with static and moving obstacles - neural networks and potential field method. We simulate the models using two simulators as described later. In the long run, we plan to further sophisticate the two approaches presented, and compare them based on the shortest paths computed and the number of collisions (if any).

3 Related Work

There exists significant literature that deals with robotic path-planning in the presence of moving and static obstacles. One of the initial methods proposed for robotic path planning in the presence of static obstacles was A* search. A* search assumes the complete environment to be in the form of a grid and computes shortest path on the basis of the least number of cells that can be taken by an agent to reach the goal. This is computed based on a heuristic function. Classical A* search suffered from several limitations, and hence many variants of A* have been derived [6]. Various other approaches have been used for mobile robot navigation based on the nature of environment and obstacles. One of the first methods for path planning in the presence of static obstacles has been the potential field method. The principle of artificial potential field method is often simple and easy to control, but often local minima can be incurred which become hard to find a way around. Owing to a lot of limitations encountered in classic artificial potential field method, several modifications have been proposed in the approach, which involve combining it with regression-based methods, and other modifications such as modified spline-based methods. Nonetheless, potential field method has active application in navigation of multiple UAVs, path planning in indoor environments and even in medical areas where it is used for robotic needle insertion. Neural networks is a proven method for general path planning of autonomous agents [2]. They are also being used to generate human like maneuvers of the agents [3].

4 Artificial Potential Field Method

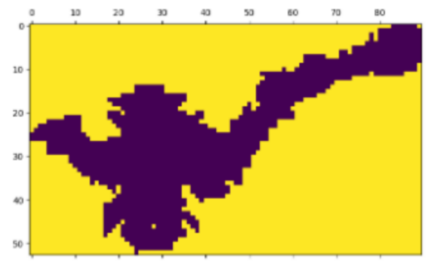
The Artificial Potential Field Method is widely used for robotic path planning owing to its mathematical simplicity and analysis. The basic concept of potential fields in path planning is to fill the robot's workspace with potential fields such that the robot is attracted to it's goal and repulsed by (static and dynamic) obstacles. In our work, we evaluate the artificial potential field method in a dynamic environment where several obstacles are moving, i.e. pedestrians and several other static obstacles exist. The attractive potential is defined in terms of the relative position of the goal of the robot from its own position. The repulsive potential is defined in terms of the relative position and velocity of the robotic agent with respect to the various obstacles. The repulsive potential from the dynamic obstacles is computed by instantiating the dynamic obstacles and considering them as static over an infinitesimally small time period. This approach makes it unnecessary to know the trajectories of other agents over time, and hence is more realistic than approaches that require knowledge about the trajectories [4].

5 Path Planning using Neural Networks

5.1 The dataset

[1] provides a tracking dataset of people in a shopping mall. This dataset provides position (x and y coordinates), velocity, heading angle, face angle at each timestamp and for each person ID. A single-day sample was used for training. First a map of the shopping mall was re-created using the position data of people moving around the place as shown in Figure 1. The dataset was prepared for training as follows. Note this procedure is repeated for all unique people in the dataset. The dataset had 18144 unique person IDs. Since the time constrains we only used 50 people for the training.

Figure 1: Re-created map of the shopping mall



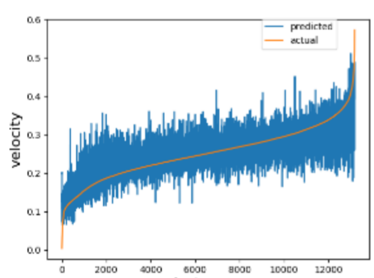
1. Consider a square around a person (this case square size was chosen to be 5000mm)
2. Obtain a feature vector considering obstacles and other people inside this square
3. take the velocity and the heading angle of the person as the quantities to be predicted
4. Train and test data were separated by a ratio of 0.6 : 0.4 and there were 19768 training data points.

5.2 The Neural Network Model

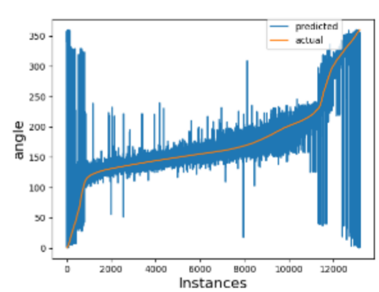
The neural network had two hidden layers with 300 and 100 neurons. For the training a learning rate of 0.001 was chosen. Other parameters were kept as default from python sklearn package. The model was trained and some results on the prediction capabilities were obtained.

Table 1: Prediction Results

prediction type	Average error	Pearson correlation coefficient
velocity	0.02 (normalized velocity)	0.84
angle	0.26 rad	0.83



(a) Velocity prediction



(b) Angle prediction

Figure 2: Prediction Results

6 Evaluation

6.1 Artificial Potential Field Method

The Artificial Potential Field Method is evaluated using the Robotarium Swarm Robotics Testbed MATLAB Simulator. The environment of the testbed was modified to include static obstacles for the purpose of our project (shown in Figure 1), and the inbuilt safety barrier certificates were removed to evaluate the correctness of our algorithm. The robotic agents in the Robotarium follow single-integrator/unicycle dynamics. For the purpose of our project, we used a total of ten agents, nine of which were assumed to be human agents that travelled in a wayward fashion in the environment. One agent was assumed to be a robot with a defined start and goal position. The collision-avoidance technique used for pedestrians was a constraint-optimization problem based algorithm. The robot agent's motion was governed by the artificial potential method, wherein it experienced repulsive potential from all static obstacles and nearby agents within a distance of 0.1 m from itself, and an attractive potential from the Goal. Different values of start and goal position were analyzed to evaluate for collision and local minimum problems. The attractive and repulsive forces and resulting velocity computation is summarized in the equations below: The force on the robot can be calculated as:

$$F(q) = F_{att}(q) + F_{rep}(q) \tag{1}$$

where F_{att} is the attraction force and F_{rep} is the repulsive force and q is the location of the agent
Attraction force can be found by:

$$F_{att}(q) = \zeta(q_{goal} - q) \tag{2}$$

Where $(q_{goal} - q)$ is a vector field directed from q to the goal and ζ is the coefficient of attraction
The repulsive force can be obtained by:

$$f_{rep}(q) = \begin{cases} \eta(\frac{1}{p(q)} - \frac{1}{p_0}) * \frac{1}{p^2(q)} * \frac{q-b}{||q-b||}, & \text{if } p(q) \leq p_0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

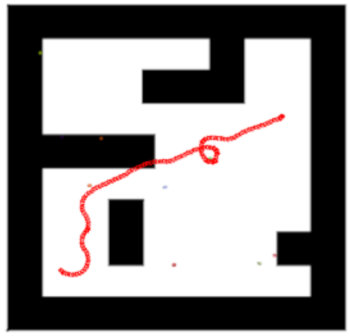
Where p_0 is the cut off distance to the obstacles (repulsive force only exists in the distance closer than p_0). b is the closest point on the boundaries to the agent. $\frac{q-b}{||q-b||}$ is a unit vector directed from b to q.
In general,

$$F(q) = \zeta(q_{goal} - q) + \eta(\frac{1}{p(q)} - \frac{1}{p_0}) * \frac{1}{p^2(q)} * \frac{q-b}{||q-b||} \tag{4}$$

[4]

6.2 The Neural Network Model

The simulation environment was created with python. For the environment, several obstacles were placed and several agents were initialized on the environment. When moving, all the agents made decisions with the same motion prediction model trained in the earlier step. The motivation was the simulated agents could display human-like motion with the learned model. To initialize the environment, several agents were placed in random places. All of them were provided with a goal position which is also random. Then A* is used to obtain a path from the start position to the goal. Then the agent creates a square around it which is the area it observes to make a decision on its motion.



(a) With Artificial Potential Field Method

(b) With Neural Network Model

Figure 3: Motion planing

Afterwards, it estimates the best next motion (velocity and angle) based on the obstacles and other agents inside the square and taking a point in the path found by A* as inputs to the neural network model. when there are multiple agents, all of them do the same.

7 Results

Motion planning using Artificial Potential Field Method worked fairly well, with the robot successfully managing to reach the goal position and evading all other agents successfully. However, the motion planning for the agents posing as humans, which was modeled as a constraint-optimization problem did not fair so well, owing to discrepancies in the computed random way points and maximum linear velocities, but the problem can be removed further on in this work. In certain cases, the robot agent does experience local minimum problem wherein it either incorrectly crashes into an obstacle or gets stuck in a certain position. The local minimum problem can probably be gotten over by tuning the parameters for attractive and repulsion potentials. A sample trajectory obtained by using artificial potentials is shown in Figure 3 (a).

The neural network method works in a limited manner. The agents continued to crash into obstacles and into each other. A path found by A* algorithm before executing the neural network model is shown in Figure 3 (b). This may be attributed to the limitations in the training data. As mentioned earlier, we used a one-day sample from the whole training dataset. Furthermore, we only used 50 out of 18144 unique person IDs available on that day. These data mainly represented people moving only in limited ways. This biased the training dataset making the trained model sub-optimal.

8 Conclusion and Future Work

The artificial potential fields method works well for the environment constructed in the simulator. However, successfully getting over the local minimum problem is still an active area of research in robotic path planning. Also, the kind of defined environment made available to the robot may not always be so in the real world. Current path planning techniques actively look into path-planning in unstructured and undefined environments that pose certain degrees of uncertainty. Future work could probably focus on incorporating uncertainty. Also, the pedestrian model is not so human-like and is more like a robotic agent. Future work could also look into using human walking behavior data to simulate human walking dynamics. Also it is important to look into better ways of modeling human motion. Methods like LSTM and inverse reinforcement learning may be used for this.

References

[1] Drazen Brscic et al. "Person tracking in large public spaces using 3-D range sensors". In: *IEEE Transactions on Human-Machine Systems* 43.6 (2013), pp. 522–534. ISSN: 2168-2291.

[2] Danica Janglová. "Neural networks in mobile robot motion". In: *International Journal of Advanced Robotic Systems* 1.1 (2004), p. 2. ISSN: 1729-8814.

[3] Mingming Li et al. "Role Playing Learning for Socially Concomitant Mobile Robot Navigation". In: *arXiv preprint arXiv:1705.10092* (2017).

[4] Min Gyu Park, Jae Hyun Jeon, and Min Cheol Lee. "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing". In: *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*. Vol. 3. IEEE, 2001, pp. 1530–1535. ISBN: 0780370902.

[5] A Sharma et al. "Model based path planning using Q-Learning". In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. 2017, pp. 837–842. DOI: 10.1109/ICIT.2017.7915468.

[6] Z Yongxiang and Z Lei. "Improvement and application of heuristic search in multi-robot path planning". In: *2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS)*. 2017, pp. 1–4. DOI: 10.1109/EIIS.2017.8298561.