Pedestrian Motion Classification for Autonomous Vehicles (6.867 Final Project)

Michael Everett & Björn Lütjens

***Abstract*— TODO**

I. INTRODUCTION

Autonomous vehicles use onboard sensors to perceive their surroundings, and use the data to make decisions about where it is safe to drive. Today's research vehicles typically rely on Lidar and cameras as the main sensors, as these provide information about where/what obstacles are, and can help the vehicle maintain safe position in its lane and with respect to obstacles. In addition to static obstacles, vehicles interact with pedestrians in a variety of environments: people in crosswalks, jaywalkers, kids running across neighborhood streets. A campus shuttle could even drive on sidewalks among pedestrians, so the vehicle would be interacting with pedestrians almost all the time. These regular interactions mean the vehicle must be able to predict pedestrians' next moves in order to maintain safety.

Pedestrian motion prediction is difficult because people often behave very unexpectedly. Humans use many strategies to predict pedestrian intent while driving, such as body language, eye contact, and other non-verbal cues between human driver and human pedestrian, but autonomous vehicles can't easily communicate or read these signals. The information from sensors is typically fused: Lidar is used to measure the pedestrian position, and the camera is used to label the particular obstacle as a pedestrian. Therefore the only measurements collected are position over time (from which velocity can also be computed).

This project uses a dataset collected on MIT's campus over several months by three golf cart shuttles providing Mobility on Demand service to students, while simultaneously collecting pedestrian trajectory data to optimize vehicle routing strategy. The dataset contains about 65,000 pedestrian trajectories as well as the vehicles' trajectories, all in a global frame across the MIT campus.

M. E. **related works**

The objective of this project is to develop a classifier that can determine whether a person will step in front of a vehicle, based on a few seconds of their trajectory. We use a portion of our data set to train different classifiers, optimize hyperparameters with a separate portion, and evaluate performance with yet another portion. This classifier could be a useful component of an autonomous vehicle, or part of an active safety feature on a human-driven vehicle that could take over in case the driver does not see a pedestrian in time.

The main contributions of this work are (i) a pedestrian trajectory dataset with 65,000 trajectories over three months, (ii) an SVM classifier for predicting when pedestrians will step in front of a vehicle, and (iii) M. E. **Which?** a classifier using Deep RNNs / a pedestrian motion prediction algorithm using Deep RNNs.

II. DATASET

The raw dataset's fields are shown in Table I, where (easting,northing) are the latitude/longitude global coordinates, and (x,y) are the coordinates in our global campus map. Veh id indicates which of the three vehicles corresponds to that data point, or in the pedestrian case, which vehicle sensed that pedestrian, and ped id is a unique id given to each pedestrian seen.

Type	Fields						
Vechicle	time	easting	northing	x	y	veh id	
Pedestrian	time	easting	northing	x	y	veh id	ped id

TABLE I: Raw data fields

There are some noise-related issues with the raw data, as it was collected on a research vehicle under development. One issue is that the vehicle's (x,y) position sometimes jumps, because the vehicle's localization system does not use GPS and is imperfect.

In addition to addressing noise, the data also needs to be pre-processed to be useful for our classifier. Specifically, the pedestrian trajectories must be converted into the vehicle's local frame in order to determine if they cross in front of the vehicle.

The global-to-local transformation relies on knowledge of vehicle orientation (heading angle) and smooth vehicle trajectories, neither of which we have by default.

Algorithm 1: Algorithm for extracting local trajectories

Input: V_g, P_g: global vehicle, pedestrain trajectories (Table I)

Output: P_l: pedestrian trajectories in local vehicle frame

for each vehicle **do**

$I_{posjump} = \{i \in [1, len(V_g) - 1] \mid euclid\_dist(V_g(i) - V_g(i+1)) > 1.0\}$

$I_{timejump} = \{i \in [1, len(V_g) - 1] \mid time(V_g(i) - V_g(i+1)) > 0.5\}$

J = $I_{posjump} \cup I_{timejump}$

$T_{valid} = \{[J(i), J(i+1)] \mid time(J(i+1) - J(i)) > 5.0\}$

return pedestrian trajectories