Algorithm for Predicting the Trajectory of Road Users to Automate Control of an Autonomous Vehicle

Andrey Azarchenkov[0000-0003-4570-4442], Maxim Lyubimov[0000-0003-0702-3662]

Bryansk State Technical University, Bryansk, Russia
azarchenkovaa@gmail.com, max32@inbox.ru

Abstract. One of the problems faced by developers of artificial intelligence algorithms when creating car control systems is that the actions of other road users are difficult to predict and have a large variability. Even if we assume that all actions comply with traffic rules and participants do not make mistakes, that is, to bring the actual environment closer to the ideal, the task of automating vehicle control still contains many difficulties. This paper describes what difficulties exist in the field of predicting the trajectory of objects, shows concepts that will help in solving this problem, and also describes a particular method of forecasting, which allows you to make a forecast for cars moving along traffic lanes. The main forecasting stages and the results of testing the method collected by using a ready-made data set are given. The results presented in the form of a set of metrics, are compared with another algorithm for predicting trajectories. As a result, the advantages and disadvantages of the created solution were identified.

Keywords: Trajectory Prediction, Autonomous Driving Car, Regression Model, NuScenes Dataset

1 Introduction

Driving a car is a complex process that requires drivers to understand the situation and control many dynamic processes in real time, as well as making decisions to react to these processes in seconds. Unfortunately, we can say that not all people cope with this task, which is confirmed by negative statistics. In 2019, 164,318 accidents were committed on the territory of Russia, in which 16,981 people were killed [1]. Despite all the measures taken to minimize the number of accidents, there is no decreasing trend in these indicators.
In addition, human errors are the cause of up to 94% of accidents [2]. Based on this fact, it can be assumed that eliminating the unreliable human factor could potentially save thousands of lives. Recent advances in artificial intelligence and high-performance computing allow to develop in this direction.

Creating a fully autonomous car requires solving many of the problems faced by a person driving a car. One of these tasks is to assess the trajectory of other road users. If we consider the solution of this problem as a separate subsystem, different types of information can be fed to its input. In the simplest case, this is different information about the physical properties of the object, such as speed, acceleration, orientation, and others [3]. The methods that use this information are the simplest to implementing, but they can't be said to have high accuracy.

The most popular group of methods for predicting the trajectory of movement, in terms of the number of scientific papers, is based on the use of the trajectory of movement from the past. In most cases, this problem is solved using neural networks. This is because they are good at identifying patterns, in this case patterns from the trajectory of movement. In addition, a popular solution for this method is to use recurrent layers [4], especially such recurrent layers as LSTM [5], [6], [7], [8]. There is also a large number of publicly available data sets containing motion paths that can be used for neural network training [9], [10].

The disadvantage of these methods is that they consider the trajectory of a single object, without taking into account others. The solution to this problem is to use the input representation [11] (Fig 1). The principles of generating such an image may differ, but the general points are the presence of layers with objects, as well as the presence of roads in the image [12], [13], [14].

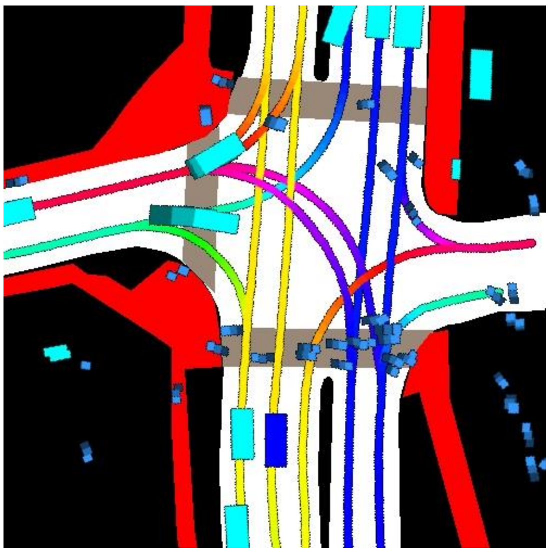

Fig.1. An example of input representation

A number of methods based on using images from the camera should be singled out separately. The idea is that the object is detected in the image, then the neural network predicts its position on the same frame. Most often, recurrent layers are also used for this purpose. [15], [16], [17]. Then, having the calibration parameters of the camera, we can understand the position of the object in space [18].

In addition, it is important to separate the task of trajectory planning for an autonomous driving car and predicting traffic paths for other objects. Although the input and output data for this problem are similar, the methods for solving it are different. The subsystem based on input data returns the expected position of an object or objects. The position can be specified exactly, or it can be a sequence of points or a motion vector. As part of the task, it is planned to determine the sequence of points. The selected time for making the forecast is 6 seconds.

nuScenes set was chosen as the data set for testing the developed method [19]. A special feature of this set is the amount of information marked up. In addition to the classic object detection and segmentation tasks, the data set provides physical characteristics of each object, its exact position in space, and tracking. Special attention is paid to information storage. To work with a data set, a library provided by developers is used that not only provides access to objects, but also allows to filter them by various criteria. The data set is divided into 1000 scenes, each lasting 20 seconds. The vehicle collecting data was equipped with an omnidirectional lidar, cameras, radars, GPS and IMU sensors. In total, the set contains 1,400,000 images, and 390,000 lidar images. It is important to have marked maps for all the sections where data were collected. This allows to define the traffic lane for each object that it moves along.

2 Approach to predict the trajectory

2.1 Description of prediction method

After analyzing existing scientific papers, as well as industrial solutions in the field of trajectory forecasting, several concepts were identified that can improve the results of the work:

- Application of algorithms depending on the class of the object. Since the trajectory is predicted for objects that have different behavior and the class is known in advance, it is possible to predict the trajectory of their movement in different ways.
- Division of algorithms depending on the position of the object. Objects of the same class may have different behaviors depending on their location.
- Use of maps prepared in advance. The main task for which using the module under development is needed is related to predicting the trajectory of the car along the traffic lanes. Information about the position of roads can be considered static, since changes to existing roads are rare. At the same time, while driving on the road, the absolute majority of cars move along the lane. This applies to both straight sections and turns of the road. In this regard, information about the position of lanes can be used to predict the trajectory of traffic in cases where the orientation of the car differs slightly from the direction of the road. To do this, the current position of the car is projected to the middle of the lane, and from this position an angular coordinate is defined, equal to the distance that the car will travel according to the forecast. The resulting road section is shifted to the current position of the car, and an adjustment is made for its orientation.

- Building a regression model for speed prediction. To implement the previous idea, we need to know the position of the road lane in space and the predicted distance of car movement. Since it is planned to use a static time interval when predicting the trajectory, the distance traveled can be determined by knowing the speed and acceleration of the object. However, for a long period of time, the acceleration that we know before making the forecast cannot be considered constant for the entire forecast section. In addition, drivers often adjust the speed of the vehicle using similar models. In this regard, to increase the accuracy, it was decided to use the predicted final speed, which can be obtained using a regression model. To train the model, it was decided to use the initial speed and acceleration of the object, as well as changing the orientation of the object, in comparison with the state from the past.
- Calculating the probabilities of predicted trajectories. Firstly, we can use probabilities to make a decision about using trajectories. Secondly, using a single algorithm, we can get a number of trajectories that also need to be compared. Finally, the obtained probabilities can be used with subsequent modules or the prediction module itself for further calculation.
- Using recurrent layers to predict the trajectory if there is no map. Based on the analysis of existing approaches, we can conclude that recurrent layers are used in most neural networks that perform trajectory prediction.

2.2. Trajectory prediction using maps prepared

Initially, the prediction task was implemented for cars that move along traffic lanes. Within the task, a trajectory forecast is made for 6 seconds, with a frequency of 2 Hertz. Accordingly, the resulting trajectory includes 12 points.

The algorithm consists of the following steps:

- receiving data;
- search for lanes;
- calculation of the final speed;
- calculation of angular coordinates for each suitable lane;
- calculation of trajectory points.

Figure 1 shows examples of the obtained trajectories. Here and further the following rules will be applied when making up drawings: the blue trajectory represents the real trajectory of the object, while the red one is predicted using the algorithm being developed. The green one is predicted using a third-party algorithm where the latter is present. When implementing the algorithm, the following complications were identified:

- Within the selected data set the current lane ends at the moment when there is an alternative path. This raises the problem of choosing a new lane. This problem was solved by constructing a function between the angle of orientation change and the speed. This angle was calculated as the difference between the current orientation and the direction of the alternate lane. This method is effective when the choice is limited to two options, for example, moving straight and turning. However, there is a high probability of error in lanes where multiple turns are possible. In real life, this task is also difficult and is solved by determining the speed of the car by the driver, as well as using turn signals, which is a more effective way. In this regard, in situations where multiple choices are possible, it was decided to consider both trajectories. First, these trajectories have a partial intersection as a part of the current lane, and second, in the case when the turn signal is not visible, the driver in reality also allows both versions of trajectories. However, it should be noted that we are talking about situations when the car is moving along the lane, and is not making a lane change.

- In areas where there is a large number of lanes, the starting lane selection may not be correct, due to the car being too far off-center. It was decided to calculate paths using several lanes within a given radius, with the probability of each of them inversely proportional to the distance from the object to the lane center.

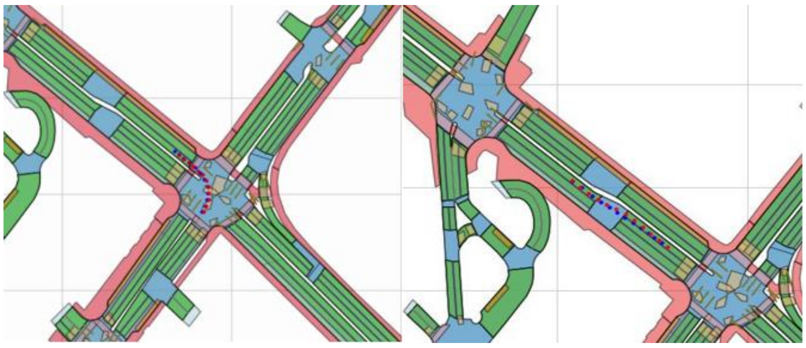

Fig. 2. An example of trajectory prediction

2.3 Prediction of the final speed of objects

The training part of the dataset was used to train the regression model, but since nuScenes test dataset contains many objects, including inanimate objects which location and physical parameters are also different, a number of filters were applied to the entire training sample:

- filtering by class, only transport was used;
- speed filtering, many objects have zero speed;
- filtering by tracking time. For training, the object speed was used after 6 seconds, relative to the prediction;
- filtering by acceleration and direction of the object.

As a result, 65,000 training objects were obtained. 90% percent for training, 10% for accuracy testing. Several regression models with the selected parameters were tested (Table 1). The speed of the models was also taken into account, but all the models under consideration showed acceptable results at the specified parameter values, so the speed comparison was not performed. The standard error between the predicted speed value and the correct one (1) was used to estimate accuracy. This metric allows to evaluate errors with different signs, in addition, its feature is that large errors are taken into account to a greater extent, due to using the square of difference in values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \check{X}_i)^2 \qquad (1)$$

Table 1. Prediction of final speed by various methods

Regression model	Root mean square error
Linear regression	9.246
The method of minimum angles	9.246
The method of orthogonal matching pursuit	9.398
Bayesian linear regression	9.246
Lasso regression method	9.245
RANSAC method	10.404
Theil–Sen estimator	9.899
Huber loss	9.276
Multilayer perceptron	8.2885
k-nearest neighbors algorithm	8.937
Support vector machine	8.368
Elastic map	9.367
Random forests	7.860
Neural network of 10 linear layers and ReLU.	8.26

Having the results obtained, it was decided to use the method of random forests. The resulting model was used to predict the final speed of the object.

2.4 Evaluation of trajectory prediction

As part of the work, the accuracy of the object trajectory prediction algorithm described earlier is estimated. The current position of the object is input into the subsystem, as well as its speed, acceleration, and direction of movement. The subsystem returns a future trajectory consisting of 12 points. The following metrics were used to evaluate the accuracy:

- average displacement error (ADE) is the average distance between two separate points of the trajectory in meters;
- final displacement error (FDE) is the distance between two endpoints in meters;
- minimum displacement error (MDE) is the minimum distance between the endpoint and the path in meters. To calculate this metric, we used 8 trajectory points instead of 12. This was done in order to reduce the impact of the trajectory length on the metric results, in cases where the predicted trajectory is longer.

The first two metrics are classical for the problem under consideration, but they do not have very accurate results, since they strongly depend on the correct definition of the trajectory length.

To compare the results, we used one of the standard algorithms for this problem, which means generation of trajectory points depending on the current speed and orientation

of the object. The algorithm returns an exceptionally straight trajectory. This method was considered as a basic one.
In addition, to demonstrate the advantages of the regression model and show the relationship between metrics and the trajectory length, the results for the algorithm under development were considered, by replacing the regression model with the constant speed, as well as changing the speed based on the initial acceleration. In accordance with the Euler method, integration was performed, that is, half of the object's acceleration was added to the initial speed, since the time between two steps is 0.5 seconds. This operation was carried out for two and for 12 steps. As a result, the following indicators were obtained for 7,600 objects (Table 2).

Table 2. Comparison of metrics for the full test set

	Basic method	Constant speed	2 integration steps	12 integration steps	Regression model
ADE	4.52	4.1449	4.1169	4.089	4.0695
FDE	10.787	9.7575	9.701	9.620	9.5575
MDE	3.613	2.8455	2.841	2.832	2.848

Analyzing the examples, we can conclude that the main reason that negatively affects the accuracy is errors in predicting the length of the trajectory. However, it should be noted that despite its impact on metrics, this problem is less significant in real conditions. In addition, it is important to note the following fact: most of the considered trajectories are straight. Often, the prediction of such trajectories is not problematic and the base method copes with this well. However, the main difficulty is represented by curved paths. Figure 3 shows examples of such paths. Most of the methods of trajectory prediction aim at curved trajectories. At the same time, due to different number of cases of linear motion, the effect of the developed algorithm, in terms of predicting a curved trajectory, is lost in comparison with the basic one. Therefore, it was decided to calculate the metrics considered earlier not on all available objects, but on objects which orientation change is greater than 0.1 radian, that is, the angle of rotation of the object changed by 0.1 radian between the current position and the previous one in the training sample. According to the results of this filtering, 1,772 objects were obtained. Table 3 shows the results. It can be seen that the difference between the two methods has increased significantly.

Table 3. Comparison of metrics for the full test set with curved trajectory

	Basic method	Constant speed	2 integration steps	12 integration steps	Regression model
ADE	5.221	4.2217	4.2038	4.1816	4.1369
FDE	12.3774	9.7505	9.7122	9.6467	9.5215
MDE	4.6327	2.8947	2.9	2.8961	2.8162

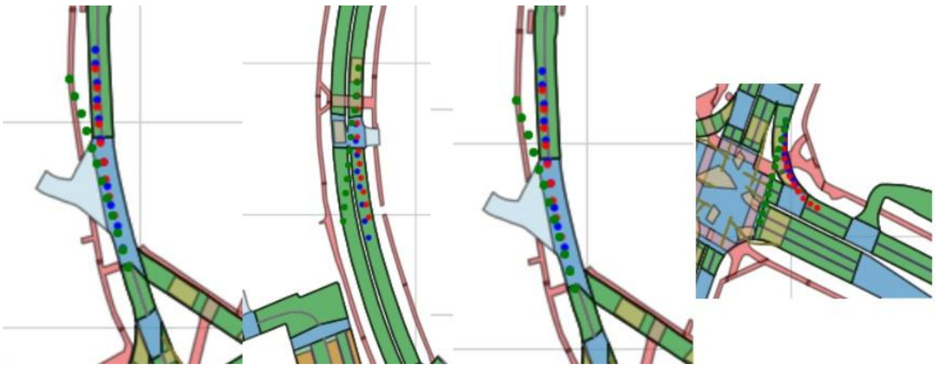
Fig.3. Curved trajectory samples

3 Conclusion

At this stage, the feasibility of the concept was tested by predicting the trajectory of cars based on the use of maps prepared in advance. Analyzing the results of the algorithm on the selected data set, we can conclude that this method allows to predict the direction of the car movement with high accuracy. First of all, the predicted trajectory length has a negative impact on metrics. However, in real-world problems, the accuracy of the trajectory length is less important.

The resulting algorithm can be used to predict car trajectories in unmanned control systems. In such systems, it is necessary to implement this task, since it is not enough for an unmanned vehicle to simply detect an object in order to perform maneuvers. Without information about further movement of the object, it is impossible to move, because the danger of a particular maneuver is not clear.

In the future, it is planned to implement a trajectory prediction method for cars that make changes or move off the road, as well as for other objects, such as pedestrians and two-wheeled transport. The implemented set of algorithms will be tested on nuScenes dataset.

References

1. Road safety indicators, http://stat.gibdd.ru/, last accesses 19/07/2020.
2. Singh, S. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report DOT HS 812 115, National Highway Traffic Safety Administration (2015).
3. Scholler, C., Aravantinos, V., Lay, F., Knoll, A. What the Constant Velocity Model Can Teach Us About Pedestrian Motion Prediction. arXiv:1903.07933v3 (2020).
4. Wenjing, Z., Yuan, L., Tingting, L., Chenyang, Y. Trajectory Prediction with Recurrent Neural Networks for Predictive Resource Allocation. 14th IEEE International Conference on Signal Processing (ICSP) pp. 1-8. (2018).
5. Alahi, A. Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savares, S. Social

LSTM: Human Trajectory Prediction in Crowded Spaces IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2-6. (2016).

6. The Autoware Foundatiion, https://www.autoware.org/, last accessed 19/06/2020.
7. Vemula, A., Muelling, K., Oh, J. Social Attention: Modeling Attention in Human Crowds arXiv:1710.04689v2 (2018).
8. Deo N., Trivedi, M. Convolutional Social Pooling for Vehicle Trajectory Prediction arXiv:1805.06771v1, 2018. – 9 p.
9. US Highway 101 Dataset, https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm, last accesses 19/07/2020.
10. Interstate 80 Freeway Dataset. https://www.fhwa.dot.gov/publications/research/operations/06137/, last accesses 19/07/2020.
11. Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F. C., Lin, T. H., Nitin, S., Schneider, J.: Uncertainty-aware Short-term Motion Prediction of Traffic Actors for Autonomous Driving. CoRR abs/1808.05819, https://arxiv.org/abs/1808.05819 (2020).
12. Cui, H. Radosavljevic, V., Chou, F., Lin, T., Nguyen, T., Huang, T., Schneider J., Djuric, N. Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks 2019 International Conference on Robotics and Automation (ICRA), pp. 1-5 (2019).
13. Grigore, E., Boulton, F., Beijbom, O., Wolff, E. CoverNet: Multimodal Behavior Prediction using Trajectory arXiv:1911.10298v1 (2019).
14. GTC 2020: PredictionNet: Predicting the Future in Multi-Agent Environments for Autonomous Vehicle Applications, https://developer.nvidia.com/gtc/2020/video/s21899-vid, last accessed 06/20.
15. Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., Manocha, D. TrafficPredict: Trajectory Prediction for Heterogeneous Traffic. arXiv:1811.02146v5 (2019).
16. Rasouli, A., Kotseruba, I., Kunic, T., Tsotso, K. PIE: A Large-Scale Dataset and Models for Pedestrian Intention Estimation andTrajectory Prediction. IEEE/CVF International Conference on Computer Vision (ICCV), p. 10 (2019).
17. Yau, Y., Xuo, M., Choi, C., Crandall, D. J., Atkins, E. M., Dariush, B. Egocentric Vision-based Future Vehicle Localization for Intelligent Driving Assistance Systems. arXiv:1809.07408v2 (2019).
18. Siswantoro, J., Prabuwono, A., Abdullah, A. Real World Coordinate from Image Coordinate Using Single Calibrated Camera Based on Analytic Geometry. International Multi-Conference on Artificial Intelligence Technology, pp. 1-11 (2013).
19. Nuscenes dataset, https://www.nuscenes.org, last accesed 19/06/20.