Search Based Software Engineering for Testing Autonomous Cars

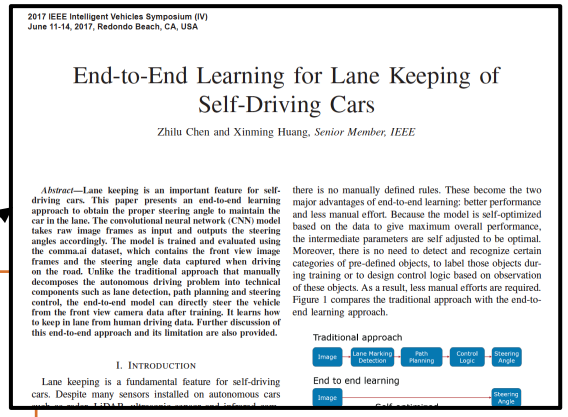
Test generator : Wiggling Road

Yuki Okumura

Motivation

Our AI was designed based on “ lane keeping of self-driving cars”

https://ieeexplore.ieee.org/abstract/document/7995975



2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

End-to-End Learning for Lane Keeping of Self-Driving Cars

Zhilu Chen and Xinming Huang, *Senior Member, IEEE*

Abstract—Lane keeping is an important feature for self-driving cars. This paper presents an end-to-end learning approach to obtain the proper steering angle to maintain the car in the lane. The convolutional neural network (CNN) model takes raw image frames as input and outputs the steering angles accordingly. The model is trained and evaluated using the comma.ai dataset, which contains the front view image frames and the steering angle data captured when driving on the road. Unlike the traditional approach that manually decomposes the autonomous driving problem into technical components such as lane detection, path planning and steering control, the end-to-end model can directly steer the vehicle from the front view camera data after training. It learns how to keep in lane from human driving data. Further discussion of this end-to-end approach and its limitation are also provided.

I. Introduction

Lane keeping is a fundamental feature for self-driving cars. Despite many sensors installed on autonomous cars

there is no manually defined rules. These become the two major advantages of end-to-end learning: better performance and less manual effort. Because the model is self-optimized based on the data to give maximum overall performance, the intermediate parameters are self adjusted to be optimal. Moreover, there is no need to detect and recognize certain categories of pre-defined objects, to label those objects during training or to design control logic based on observation of these objects. As a result, less manual efforts are required. Figure 1 compares the traditional approach with the end-to-end learning approach.

Traditional approach

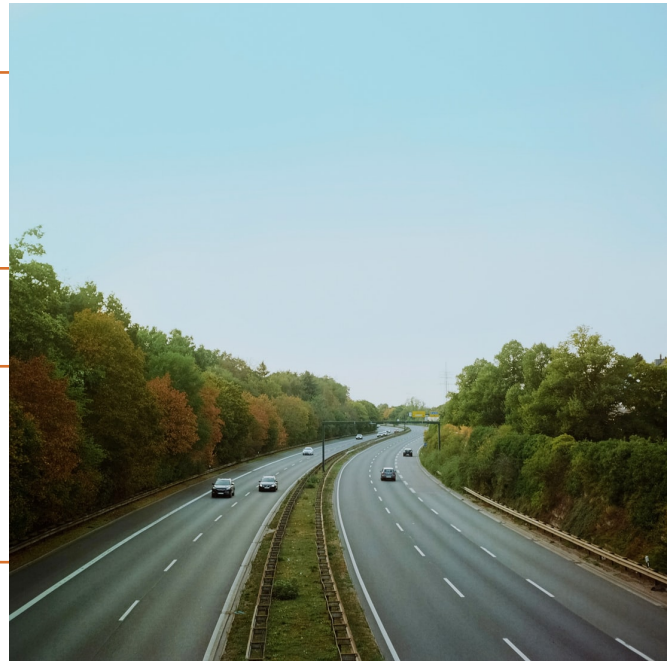
Image → Lane Marking Detection → Path Planning → Control Logic → Steering Angle

End to end learning

Image → Steering Angle

Lane keeping functions play an active role in particularly **highways**

Assume that Test generator creates **highway road** for testing AI’s lane keeping functionality



Test generating Conditions

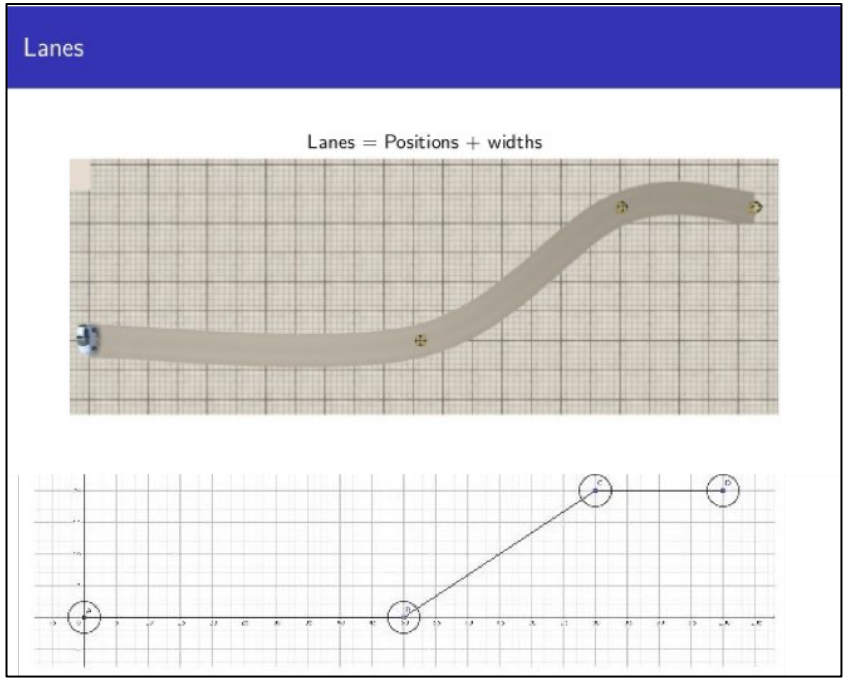
- Creates 16 positions
- Road Widths are 3,5 m which is the same width with real highway road
- Start and Goal are fixed positions
- 2nd position is also a fixed position to generate at least one straight road
- Other positions are generated randomly

  Reason of Random : I set a goal to order courses by course's difficulty
- Single lane road and no static and dynamic objects to avoid too complex situations

  →but a number of lanes should be increased as real highway usually has multiple lanes and static and dynamic objects also should be considered in further step

Drivebuild Lane generation method

Lanes = Positions + Widths



Course Generation Algorithm

Pseudo Code

1. x = 0, y = 0, w = 3.5
2. x += 10, y = 0, w = 3.5

3 ~ 15.

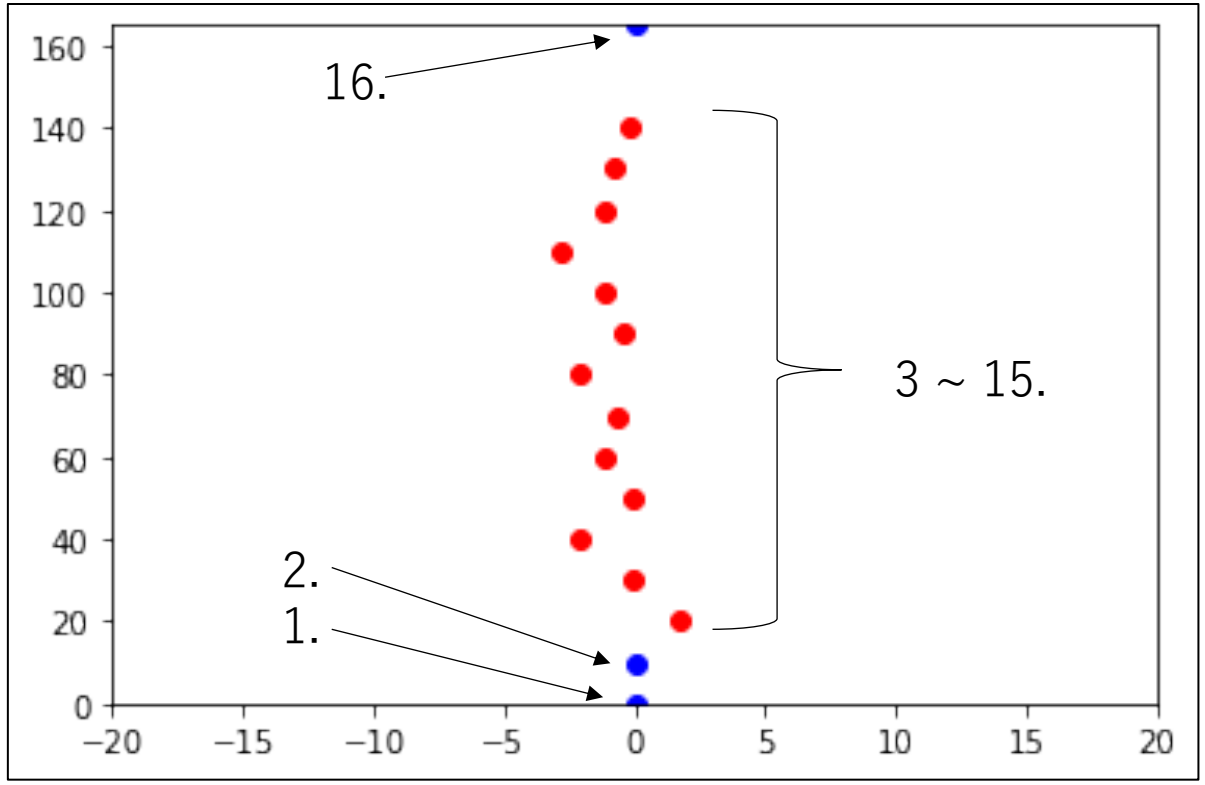
x += 10

y += random.uniform(-2,2)

w = 3.5

16. x = 165, y = 0, w = 3.5

Course Sample

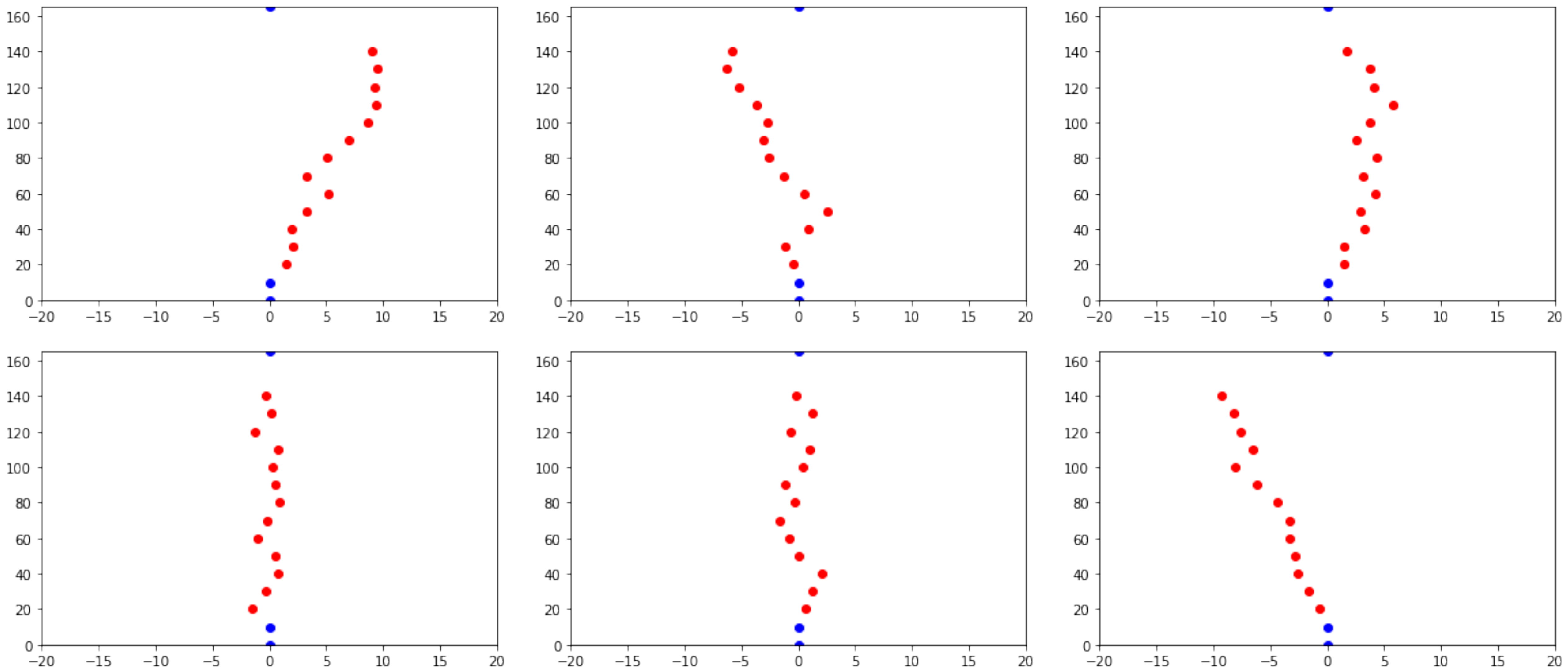


Long : Wide = 4 : 1

● : fixed position

● : randomly generated position

Generated Course Samples



Suggestion 1 for Evaluation
Direct Distance of two positions

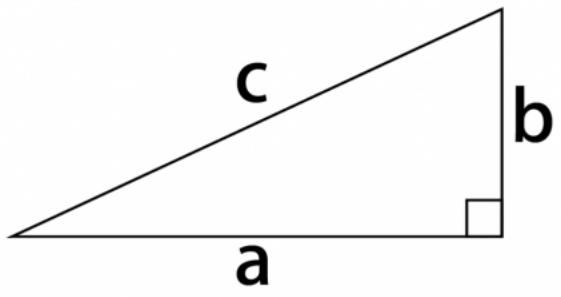
※ I suggest three evaluation methods since there are no clear evaluation methods exist for test course

Conditions

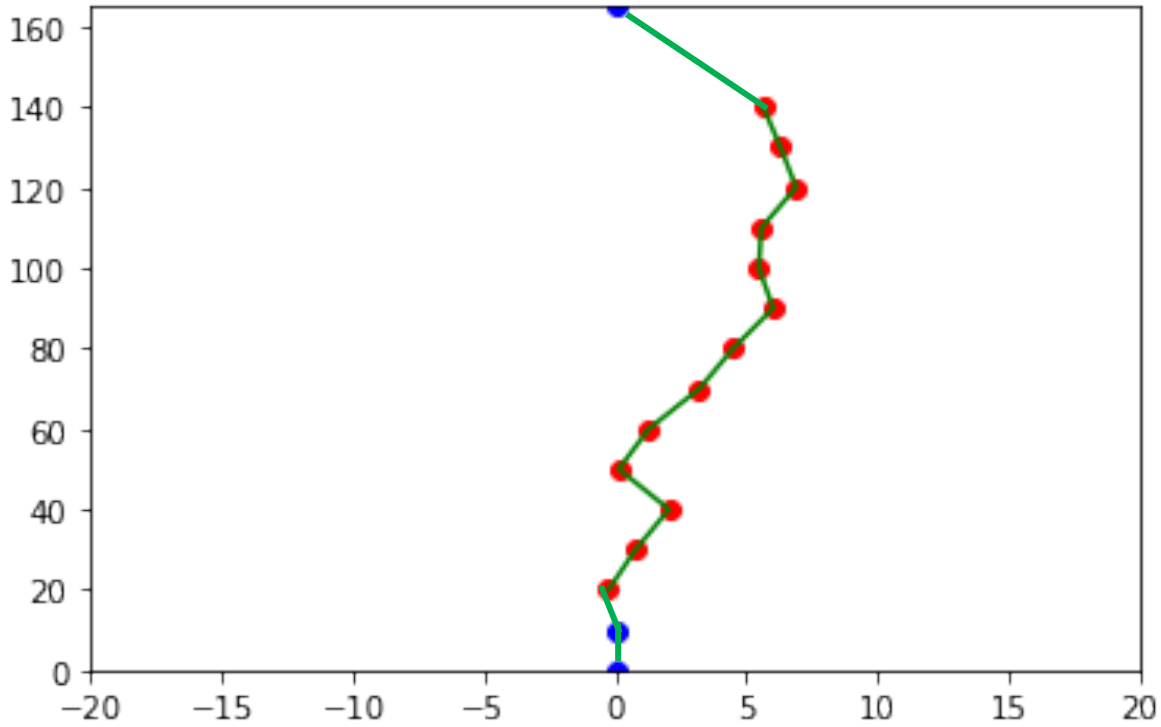
- Total: 15 distances
- Calculate

  Sum of distances, Mean of distances

Calculation method



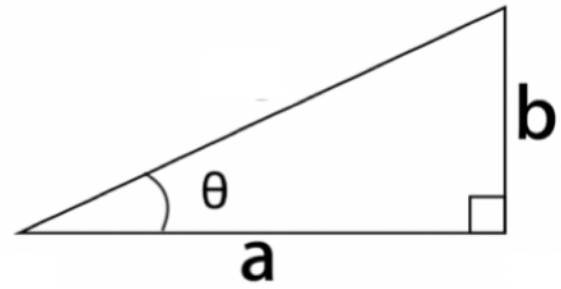
$$c^2 = a^2 + b^2$$



Suggestion 2 for Evaluation
Angle of a position from start point

Calculation method

Use Arc tan θ



$$\tan\theta = \frac{b}{a}$$

Conditions

- Total : 13 angle values

→ except 2nd and goal position

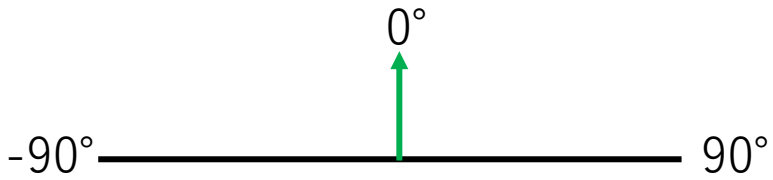
because the angle is always 0°

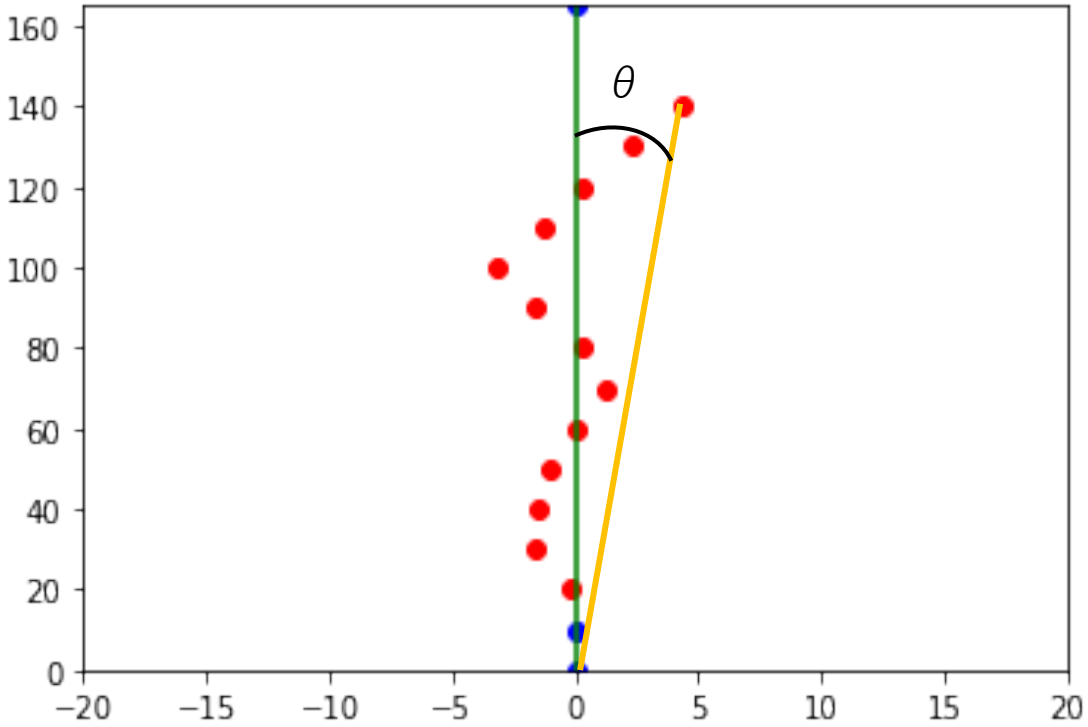
- Calculate

Sum of angles, Mean of angles

Sum of absolute angles, Mean of absolute angles

- Left is negative, Center is 0° , Right is positive





Why absolute value?

Angles change both positive and negative because the center has set as 0°

Suggestion 3 for Evaluation

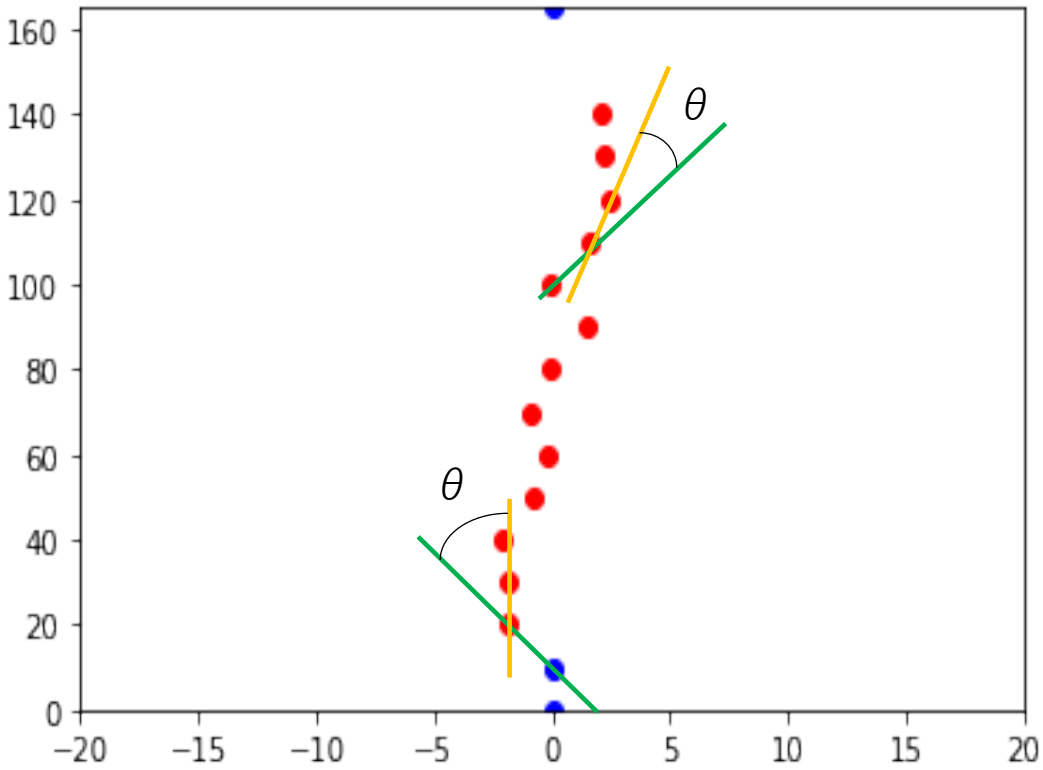
Angle of a position from previous 2 positions

Conditions

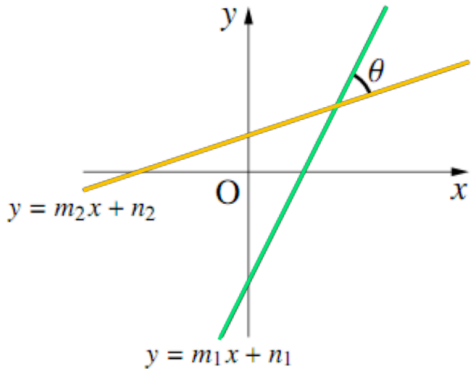
- Total : 14 angles because calculation is available from the 3rd position
- Calculate

  Sum of angles, Mean of angles

  Sum of absolute angles, Mean of absolute angles
- Left is negative, Center is 0°, Right is positive



Calculation method

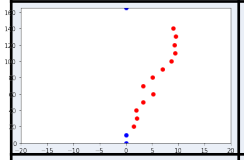
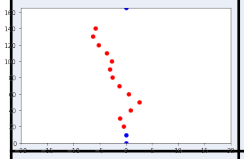
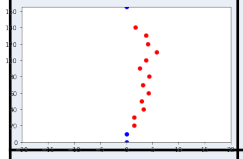
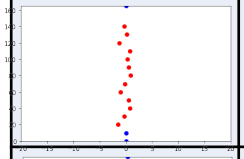
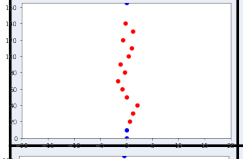
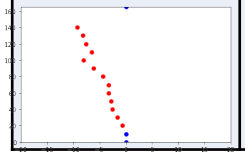


$y = m_1 x + n_1$ $y = m_2 x + n_2$

$$\tan\theta = \frac{m_1 - m_2}{1 + m_1 m_2}$$

Use Arc tan θ

Evaluation Result

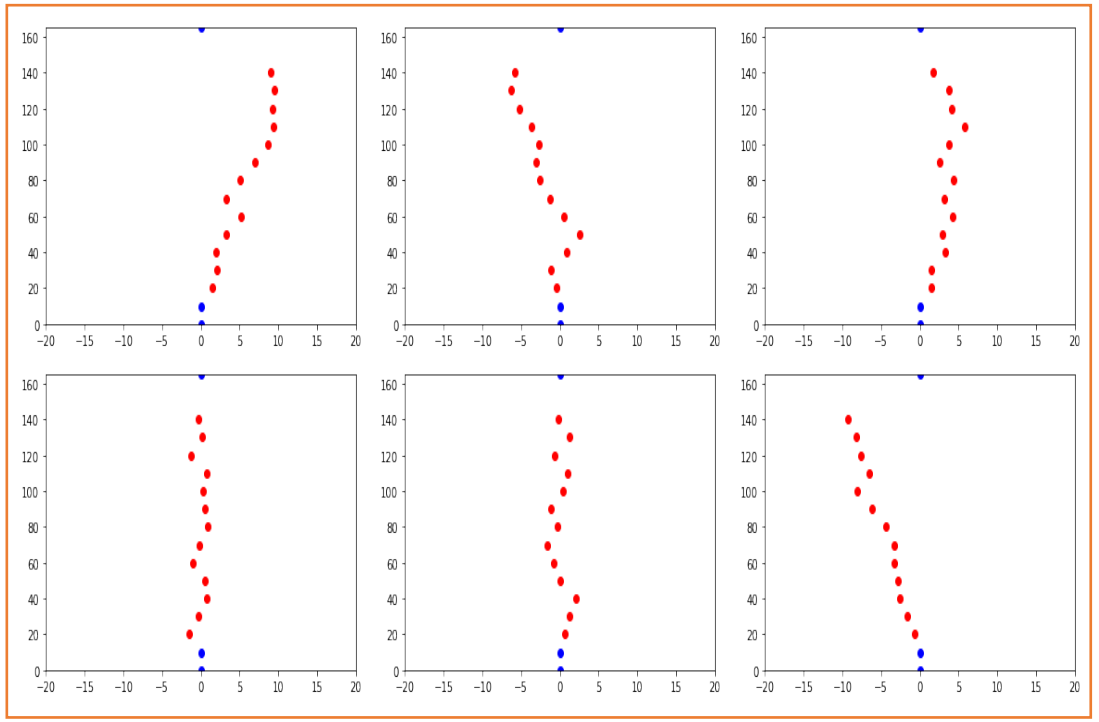
	Direct Distance		Angle of a position from Start position				Angle of a position from previous 2 positions			
	Sum	Mean	Sum	Mean	Absolute Sum	Absolute Mean	Sum	Mean	Absolute Sum	Absolute Mean
	167.666	11.177	52.424	4.032	52.424	4.032	-19.830	-1.416	52.424	3.494
	166.756	11.117	-14.800	-1.138	24.081	1.852	13.197	0.942	24.081	1.605
	166.317	11.087	35.788	2.752	35.788	2.752	-4.045	-0.288	35.788	2.385
	165.785	11.052	-3.655	-0.281	10.148	0.780	0.699	0.049	10.148	0.676
	166.0169	11.067	5.089	0.391	11.982	0.921	0.364	0.026	11.982	0.798
	167.463	11.164	-43.921	-3.378	43.921	3.378	20.353	1.453	43.921	2.928

Problem and further work

Evaluation should consider not only information from positions but also AI's driving data such as vehicle's velocity, steering angle, test succeed or fail etc. because it is hard to determine difficulty level only with information from courses

Need to develop a method which orders generated test courses by difficultly levels that are determined by information from both courses and AI tested result

AI Developer can check the maturity level of AI and AI's weak operations would be exposed



Require more data and require ordering method