AUTHOR, OLDDEGREE

TITLE

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Telematik

submitted to

Graz University of Technology

Supervisor

Dr. Some Body

Institute for Softwaretechnology
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Some One

Graz, November 2013

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Abstract

This is a placeholder for the abstract. It summarizes the whole thesis to give a very short overview. Usually, this the abstract is written when the whole thesis text is finished.

Contents

List of Figures

1 Introduction

This Thesis will use a Genetic Algorithm in order to generate Driving Scenarios for testing ADAS/AD Functionality in vehicles. While generating these scenarios is the objective, the main task of the thesis will evolve around the implementation of the Genetic Algorithm as well as the Optimization of its Hyperparemter.

1.1 Genetic Algorithm

Genetic Algorithms are a popular search algorithm that utilizes the principle of Darwin. They have been used successfully in various areas. Some of their strengths are .... However we will also look at shortcomings, which mainly evolve around performance.

Define a vocabulary

1.2 Traffic Manager

The Genetic Algorithm will control the simulation of a custom developed Traffic Manager. This Traffic Manager was developed closely to fit the needs of the Genetic Algorithm. It, however is not part of this Thesis and will thus will only get a brief introduction. In general, it will simulate traffic starting from a predefined scenarios where the positions and types of Vehicles and Pedestrians are given (i.e. actors). It also allows for an Interface for aplying actions on all actors in the simulation, which will be discussed in section 3.1.

A simulation consits of multpile NPCs and exactly one EGO vehicle. While the NPCs are only controlled by the Traffic Manager (and dadurch also by its action interface), the ego vehicle can be either partly or even completly controlled by an ADAS/AD Function. This function can then be tested inside the simulation on errors.

might not fit into this section:

The task of the Genetic Algorithm is to search for sequences of actions that will result in the most interesting Scenarios according to its cost function.

1.2.1 OpenDrive Standard

The OpenDrive Standart is used to define the roads and traffic signs. It is a widely accepted standard and is, for example, also used by Carla.

1.2.2 Graphics

During the simulation, usually no graphics engine is used in order to save performance. In order to visualize the results, two options can be chosen. The more lightweigth Esmini, as well as Carla, which is using Unreal Engine to render realistic graphics.

1.2.3 used PC

The PC used for the simulation as the following specs:

1.3 Shortcomings

This Master Thesis started with the developement of the Traffic Manger and thus progress was closely linked. Without a working simulations, no genetic alogirthmis could be tested. Due to time and performance constraints, it is not possible to test a full driving stack like autoware, as well as other professional ADAS/AD functions. In this Thesis, internal functions like Time-To-Collision and Emergency Braking will be optimized. The learned information on e.g. optimal hyperparameter settings can then be applied in further steps to test these functions.

Performance is also a problem and will lead to many shortcuts that need to be taken. There is a hughe number of possible compations of hyperparamter, so only a handful can be tested. In further chapers, these shortcuts will be explained and their relevancy will be dicussed.

2 Tools

2.1 Genetic Algorithm

While a short intrododuction was already given, the idea of a Genetic Algorithm will now be explained in greater detail. We will have a look at its History and then discussing the most important parameters.

Usage of GA

2.1.1 History

The GA was invented by....

2.1.2 Different Hyperparameter

Hyperparamter have a huge influence on the performance of a Genetic Algorihm. They have an impact on the "convergin" ... It has been shown, that there is no universal hyperparamter set and that it needs to be optimized on a per "problem" basis.

Num of generations

The Number of Generation defines the duration of a GA. As long as the algorihtm has not converged, ...? For my testing, using a generation size of 40 was almost always sufficient, and will thus mostly be used.

Pop Size

Pop size will set the number of Individuals of a GA per Generation. The higher the pop size, the bigger the less change of premature converging. It will however also lead to a longer convergin time.

Selection

Selection defines how which individuals are allowed to mate and move into the next generation.

pros and cons of roulette vs Tournament

tournament was chosen to be used for this works because of this paper (and also because of pros and cons list)

Other ideas are evolve around having a flexible selection system debending on fitness

cite paper

Crossover

Crossover is the mating process.

Discuss all used crossover methods

Mutation

Mutation is responsible for introducing new information into the gene pool.

Discuss individual mutation

Discuss all used mutation methods

Other

More to come....

2.2 Behavior Tree

A behavior tree is a decision tree.

insert a good introduction to BT

2.2.1 Usage for GA

Due to the fact, , that there is no full stack available for the EGO vehicle, a solution had to be found. In order to have the Genetic Algorithm controll only NPCs and not the EGO vehicle itselve, a behaviour tree is used. The behaviour tree is used to controll the EGO vehicle over the action interface provided by the Traffic Manager. This is the same as the Genetic Algorithm is doing.

insert ref to discussion

The behaviour tree will define which direction the EGO should take at junctions and it will realistically dodge obstacles intoduced by the Genetic Algorithm. The main goal of the BT is to make the EGO vehicle behave in a realistic way.

In a further chapter it will be dicussed if a GA with controll of the EGO (i.e. no BT will be used) lead to better cost.

3 Implementation

3.1 Action Interface

The Traffic Manager has already been discussed in some detail, however it can be thought of as a blackbox, which returned simulation is used by the cost function of the Genetic Algorithm. To interface with this blackbox, actions have to be used. An action will request a certain behaviour from an actor. If no action is set, the actor will behave in a normal manner inside the simulation. An action can be set to at any timestep (for this thesis, the simulation is running with 100 Hz) for any actor. Pedestrians and vehicles however have different actions.

The following list are now all actions provided by the traffic manager that were available for the genetic algorithm at the time of this master thesis.

- JunctionSelection
  - Parameters: Vehicle ID: int, Junction_selection_angle: float
  - Angle is set in radiant. Default value is 0. Vehicles will chose which direction to take at a junction based on this angle.
- LaneChange
  - Parameters: Vehicle ID: int, ...
  - Initiates a LaneChange based on its given parameters.
- AbortLaneChange
  - Parameters: Vehicle ID: int, ...
  - If a LaneChange is currently happening, it will get aborted.
- ModifyTargetVelocity
  - Parameters: Vehicle ID: int, ...

  - Modifies the interal Target Velocity of the Traffic Manager by a percentage. If it is for example 0, the vehicle will stop.
- TurnHeading
  - Parameters: Pedestrian ID: int, ...
  - The pedestrian will turn 180 degrees and walk in the oposite direction
- CrossRoad
  - Parameters: Pedestrian ID: int, ...
  - The pedestrian will cross the road immediately.
- CrossAtCrosswalk
  - Parameters: Pedestrian ID: int, ...
  - The pedestrian will cross the road at the next crosswalk.

All these actions are accessed by the Genetic Algorithm to maximize a given Cost Function.

3.2 Map and Starting Scenario

The map is Town10 from Carla. It was chosen, because 1. its roads are self contained, 2. its not too big, yet still complex and 3. its supported by Carla and thus visualization looks better.

The Starting Scenario defines the number and type of all actors as well as their position. It needs to be created manually. Changing the scenario will have a great impact on the Genetic Algorithms performance. For time and complexity reasons, it was thus decided to first stick with one scenario and do all hyperparameter testing there. And finally test the performance for a handfull different scenarios.

3.3 Genetic Algorithm

explain why pygad was NOT chosen

cite

For implementing the Genetic Algorithm, DEAP was chosen. It is a popular tool for accademia .

cite 3 examples

3.3.1 Encoding

When implementing a Genetic Algorithm, it is necessary to implement a Encoding that fits to the problem.

cite what makes an encoding good: eg. simplicity,...

Gene

Genes are the building blocks of a GA.

Chromosome

Each Individual has 1 chomosome which consits of a list of genes. Starting out, 2 different encodings came to mind, in both cases, the genes position in the chromosome defined the time an action is set.

generate images

Encoding 1 has the idea that each gene stands for 1 time step. Because multiple actors exist in the simulation, a gene thus needs to be a list of actions. This list always has the length of the number of all actors. This means that crossover can only move all actions of a timestep at once, modifing between actions of the same timestep can only be done using mutation.

Encoding 2 has not only the time step encoded in the position, also the actor ID is encoded. This makes a chromosome now much longer than in the previous encoding, whith the eqatuion beeing: number of timesteps * number of actors. Now crossover has more possibilities.

In the chapter 4 these two chromosome types will be compared.

3.3.2 Rules

Often, actions are not possible if specific requirements are not met. The obvious example is that it is not possible to perform the action AbortLaneChange if there is no current LaneChange happening. LaneChange during a LaneChange is not possible as well. Also Pedestrians can not CrossRoad shortly in a Row. The hypothesis is that implementing Rules that dont allow for these behaviours will reduce the searchspace and will thus make GA converge quicker.

be carefull, lanechange after lanechange or crossroad after crossroad might be possible if prev did not happen. Good to explain

3.3.3 Cost Function

Cost function is a bit difficult, as we are only using interal values. No ADAS/AD system is tested and we thus have to work with what we got. Currently 3 different cost functions are tested

3.4 Behavior Tree

The Behavior Tree will controll the ego vehicle

4 Evaluation

in this chapter, we evaluate and compare various different settings In this chapter, we will incrementally move to an optimized Genetic Algorithm

4.1 Hyperparameter tuning

4.1.1 Comparing Pop Size

The mentioned performance limitations forced a lower population size for most simulations than optimal.

4.1.2 Different Cost Function

this will be more tricky to explain as everything is just interal functionality

4.1.3 Crossover vs MutProp

What is the best configuration?

4.2 Behaviour Tree enabled

Will the scenarios be more interesting if the ga is not allowed to have controll over the Ego Vehicle?

4.3 Rules enabled

5 Conclusion

5.1 Test

Altough lots of shortcommings Results look very vielversprechend. This thesis hopes to have emphasised that this approach has lots of advanatages

Appendix

Bibliography

Andrews, Keith (Dec. 2011). *Writing a Thesis: Guidelines for Writing a Master's Thesis in Computer Science*. URL: http://ftp.iicm.edu/pub/keith/thesis/.

Bringhurst, Robert (1993). *The Elements of Typographic Style*. first edition. Hartley and Marks Publishers.

Chemnitz, TU (2004). *German-English Dictionary*. URL: http://dict.tu-chemnitz.de/.

dictionary.com (2004). *dictionary.com*. URL: http://dictionary.com/.

Dupré, Lyn (1998). *Bugs in Writing: A Guide to Debugging Your Prose*. Second. Addison-Wesley. ISBN: 020137921X.

Leo (2004). *Leo English-German Dictionary*. URL: http://dict.leo.org/.

McCaskill, Mary K. (Aug. 1998). *Grammar, Punctuation, and Capitalization: A Handbook for Technical Writers and Editors*. NASA Langley Research Center SP-7084. URL: http://stipo.larc.nasa.gov/sp7084/.

Phillips, Estelle M. and Derek S. Pugh (2005). *How to Get a PhD*. Fourth. Open University Press. ISBN: 0335216846.

Roget (1995). *Roget's II: The New Thesaurus*. URL: http://www.bartleby.com/62/.

Roget (2004). *Roget's Interactive Thesaurus*. URL: http://www.thesaurus.com/.

Strunk Jr, William (1918). *The Elements of Style*. URL: http://www.bartleby.com/141/.

Strunk Jr, William and Elwyn Brooks White (1999). *The Elements of Style*. Fourth. Longman. ISBN: 020530902X.

Voit, Karl (July 2020). *tagstore — Project home page*. URL: https://Karl-Voit.at/tagstore (visited on 12/10/2011).

Zobel, Justin (2004). *Writing for Computer Science*. Second. Springer. ISBN: 1852338024.