Optimisation of parallel KD-trees using heuristics for neural networks simulation
Project Specification - Degree Project in Computer Science, DD150X

Daniel Benedí
Supervisor: Alexander Kozlov

February 2022

Introduction

In the last decade neurology has been benefited of simulating the nerve cell behaviour or neural networks. This made use of large-scale, high-precision experimental methods of data acquisition which allowed to develop realistic tools to understand the brain functions and diseases [3]. Due to the amount of data collected, this simulations have to be handled with supercomputers to integrate different levels of simulation. Even though they are using high-performance computers, it is not possible to fully simulate a human brain [1], and because of that it is used only sub-regions of the hole structure. This study will try to optimise the performance obtained by this simulations focusing on how to improve the acquisition of sub-regions using some techniques from computer graphics.

Problem Statement

In computer graphics, there are several approaches for constructing kd-trees for ray-tracing. The reasons they are used are also applied for this problem so they require a spatial localisation around a big cloud of points. Some of this are parallel KD-trees that are optimised with some heuristic to get a better performance [2]. Using this, we pose the question: "Which parameterised heuristics can work better with parallel kd-trees for neurological simulation?".

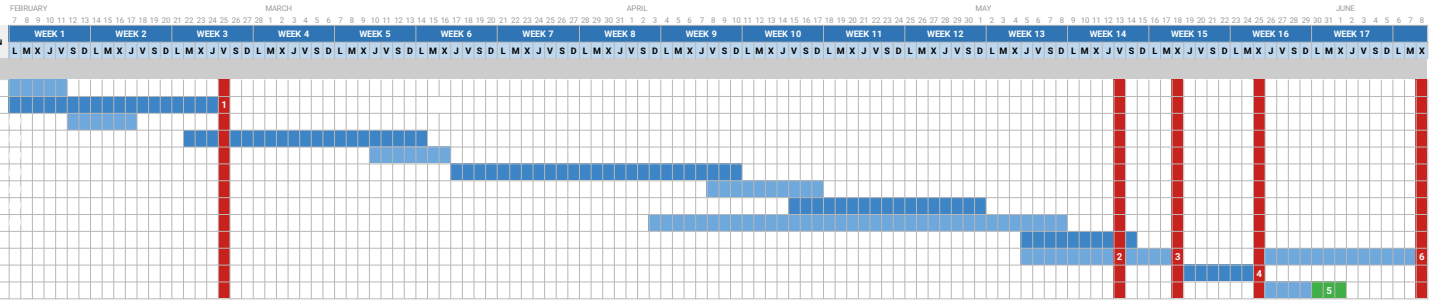
Approach

To check how effective are this heuristics, we are going to have several approaches to the problem. First of all, it will be implemented with the traditional algorith [5], massively parallel KD-tree construction [4] and some other approaches of constructing this massively parallel KD-trees aided with some heuristics such as Surface Area Heuristic [2] or Curve Complexity Heuristic [6], this heuristics may need some little changes to be more related to the actual problem.

The evaluation of this methods will be based on a dataset that will be based on some existing real cases [8] and some little cases that will be created by hand using the python library treem [7]. From this dataset we will analyze different metrics such as execution time and cache misses

TIME PLANNING

AUTHOR	Daniel Benedí García	COURSE	DA150X
SUPERVISOR	Alexander Kozlov	DATE	25/02/22



TASK	BEGINNING	END	DURATION
Choose research topic	07/02/22	11/02/22	4
Write Project Plan	07/02/22	25/02/22	18
Study previous research	12/02/22	17/02/22	5
Resources Retrieval	22/02/22	14/03/22	22
Environment Setup	10/03/22	16/03/22	6
Implement basic models and test	17/03/22	10/04/22	23
Define properly heuristics to use	08/04/22	17/04/22	9
Extend basic models with heuristics	15/04/22	01/05/22	16
Experiment and acquire metrics	03/04/22	08/05/22	35
Analysis	05/05/22	14/05/22	9
Report	05/05/22	08/06/22	33
Peer review	19/05/22	25/05/22	6
Presentation	25/05/22	01/06/22	6

NOTES

1. Hand-in of project specification
2. Submission of Public Science Abstract
3. Hand-in of students' reports for peer review
4. Hand-in of students' peer reviews
5. Student conference
6. Hand-in of students' final reports

References

[1] S. Furber, S. Temple, and A. Brown, "High-performance computing for systems of spiking neurons," vol. 2, Jan. 2006.

[2] B. Choi, R. Komuravelli, V. Lu, *et al.*, "Parallel sah k-d tree construction," in *Proceedings of the Conference on High Performance Graphics*, ser. HPG '10, Saarbrucken, Germany: Eurographics Association, 2010, pp. 77–86.

[3] H. Markram, K. Meier, T. Lippert, *et al.*, "Introducing the human brain project," *Procedia Computer Science*, vol. 7, pp. 39–42, 2011, Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2011.12.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050911006806.

[4] L. Hu, S. Nooshabadi, and M. Ahmadi, "Massively parallel kd-tree construction and nearest neighbor search algorithms," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 2752–2755. DOI: 10.1109/ISCAS.2015.7169256.

[5] M. Adamsson and A. Vorkapic, "A comparison study of kd-tree, vp-tree and octree for storing neuronal morphology data with respect to performance," Ph.D. dissertation, 2016. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-187026.

[6] Y. Lu, L. Cheng, T. Isenberg, *et al.*, "Curve complexity heuristic kd-trees for neighborhood-based exploration of 3d curves," *Computer Graphics Forum*, vol. 40, no. 2, pp. 461–474, 2021. DOI: https://doi.org/10.1111/cgf.142647. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.142647. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.142647.

[7] A. Kozlov. "Treem - neuron morphology processing tool." (), [Online]. Available: https://pypi.org/project/treem/ (visited on 02/25/2023).

[8] G. M. University. "Neuromorpho.org - a centrally curated inventory of digitally reconstructed neurons and glia." (2006-2022), [Online]. Available: http://neuromorpho.org/ (visited on 02/25/2022).