DOI: 10.20396/revpibic.

XXVII Congresso de Iniciação Científica Unicamp
16 a 18 de outubro de 2019 - Campinas | Brasil

Learning Ray Tracing Geometry Intersections

João Vítor Buscatto Silva*, Hélio Pedrini

Abstract
Rendering complex scenes in a realistic and physically correct manner has long been a subject of interest to various areas of research, such as Computer Graphics. In this context, identifying geometry primitives as fast and accurately as possible is of significant importance, as it is responsible for a considerable fraction of the execution time of a ray tracing-based method. This research project reviews existent methods and proposes a new one, based on the results found.

Key words:
Ray Tracing, Machine Learning, Computer Graphics.

Introduction

Knowledge domains, such as Computer Graphics, Animation and Game Development, have long studied how to render realistic and physically correct images. Recent advances both in hardware and software made it possible to obtain plausible images generated by Ray Tracing and Path Tracing techniques in real time, which were deemed impossible before due to their high computational cost.
The objective of this project is to investigate and implement a novel technique to predict geometry primitive intersections in real time through the usage of Deep Learning models. This new approach is expected to be faster than traditional intersection methods.

Results and Discussion

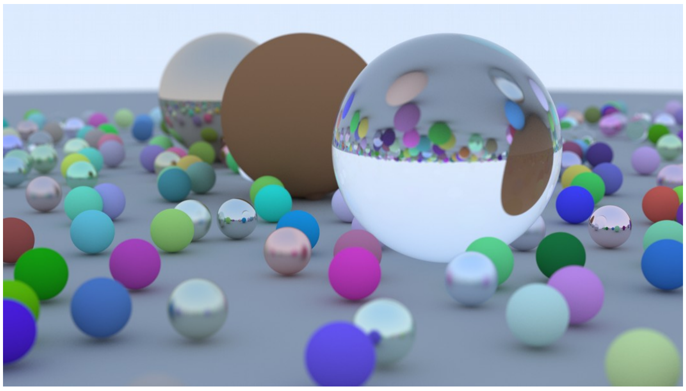

Image 1. 'Spheres' test scene, rendered on a traditional Path Tracer at a resolution of 1920x1080 pixels with 500 samples per pixel.

The initial concept of this research project is based on the work developed by Kraska et al.[1], where the authors suggested that all index-based data structures can be replaced by learned models. They compared different neural network models and analyzed the conditions for a learned model to outperform traditional structures, as well as the challenges these techniques represent. Some of the neural models analyzed by the authors outperformed the B-Trees by up to 70% in speed.
When it comes to Ray Tracing-based algorithms, the most common data structure for storing geometry primitives is a Bounding Volume Hierarchy (BVH) binary tree. Given the outstanding results obtained by Kraska et al., we believe that the techniques described by them can be applied to Ray Tracing to reduce its execution time.
We generated our data sets through a traditional Path Tracing implementation, recording values of ray directions and origins, as well as camera parameters. The target variable is given by indices of the intersected geometry primitives. We separated the data between training and testing at a rate of 3 to 1.
We started by implementing a naive deep learning model, fully connected with four hidden layers of 1024, 512, 256 and 128 neurons. We used 5-fold cross validation and executed the model for 50 epochs, with a learning rate of 0.0001 and regularization factor of 0.1. In Chart 1, we have the R^2, or R Squared, metric values obtained for our two test scenes. This metric indicates how good the predictions of our model are.

Chart 1. R^2 values for training, validation and testing data sets for two different scenes. Higher values imply a more precise model.

Scene	Train	Validation	Test
Spheres	0.642	0.640	0.638
Cornell Box	0.553	0.550	0.564

Conclusions

Based on the results obtained, we conjecture that the model proposed is capable of learning the scene intersections to a certain extent. Although the R^2 values obtained still leave room for improvement, they represent a remarkable precision obtained for a first test, on a naive implementation. Further improvements are expected through the exploration of different and more complex neural network architectures.

Acknowledgements

I would like to thank Prof. Hélio Pedrini, my supervisor, for all the help provided during the preparation of this project. I would also like to thank Peter Shirley and Morgan McGuire for their vital feedback provided on early stages of the project. This work was partially supported by the funding received from PIBIC/CNPq.

[1]Kraska et al. The Case for Learned Index Structures. *arXiv:1712.01208v3 [cs.DB]*, 2018.