Prelude

A typical dissertation will be structured according to (somewhat) standard sections, described in what follows. However, it is hard and perhaps even counter-productive to generalise: the goal is *not* to be prescriptive, but simply to act as a guideline. In particular, each page count given is important but *not* absolute: their aim is simply to highlight that a clear, concise description is better than a rambling alternative that makes it hard to separate important content and facts from trivia.

You can use this document as a LaTeX-based [1, 2] template for your own dissertation by simply deleting extraneous sections and content; keep in mind that the associated `Makefile` could be of use, in particular because it automatically executes to deal with the associated bibliography.

You can, on the other hand, opt *not* to use this template; this is a perfectly acceptable approach. Note that a standard cover and declaration of authorship may still be produced online via

http://www.cs.bris.ac.uk/Teaching/Resources/cover.html

Dissertation Type: research

DEPARTMENT OF COMPUTER SCIENCE

How effective are Temporal difference learning methods in reducing the number of zero contribution light paths in Path tracing?

Callum Pearce

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Wednesday 10$^{\text{th}}$ April, 2019

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MEng in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Callum Pearce, Wednesday 10th April, 2019

Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

Executive Summary

In the field of Computer Graphics, Path tracing is an algorithm which accurately approximates global illumination in order to produce photo-realistic images. Path tracing has traditionally been known to trade speed for image quality. This is due to the lengthy process of accurately finding each pixels colour, whereby many light rays are fired through each pixel into scene, then directions for each ray are continually sampled until it intersects with a light source. Due to this, a variety of Importance sampling algorithms have been invented to avoid sampling directions which lead to rays contributing no light to the rendered image. The paths formed by sampling rays in these directions are known as zero contribution light paths. By not sampling zero contribution light paths, it is possible to significantly reduce the noise in rendered images using the same number of sampled rays per pixel.

Recently a Temporal Difference learning method was used by Nvidia to achieve impressive results in Importance sampling within a Path tracer. The algorithm essentially learns which directions light is coming from for a given point in the scene. It then uses importance sampling to favour shooting rays those stored directions, reducing the number of zero contribution light paths sampled. Nvidia only explore two classic Temporal Difference learning schemes in doing so. Therefore, there is more potential research to assess the ability of deep reinforcement learning schemes in Temporal Difference learning for reducing the number zero contribution light paths sampled.
My goals are as follows:

1. Reimplement Nvidia's state of the art on-line Temporal Difference learning Path Tracer in order to further investigate its ability to reduce the number of zero contribution light paths, and at what costs the method bears in doing so.

2. Design and implement an on-line Deep Q-Learning variant of the Path tracing algorithm and investigate its ability to reduce the number of zero contribution light paths sampled, and what costs the method bear in doing so.

Both of these goals together contribute to confirming the following hypothesis:

> A Deep Q-learning Path tracer is better able to reduce the number of zero contribution light paths than an Expected SARSA Path tracer with the same memory budget.

Outcomes

- Which is better able to reduce the number of zero contribution light paths expected SARSA or Deep Q-learning

- Can Expected SARSA learning handle multiple lights well in a scene & deep q-learning

Main areas of work

- I have written x lines of code to build a Path tracing engine from scratch which supports a variety of GPU accelerated Path tracing algorithms I have experimented with.

- I have spent x hours researching into the field of efficient light transport simulation for ray-tracing techniques.

- I have spent x hours researching into Reinforcement learning, particularly Temporal Difference learning and Deep Reinforcement learning, neither of which I have been taught before.

- I spent x hours implementing and validating the on-line Expected SARSA Path tracing algorithm proposed by Nvidia, which required me to implement the Irradiance Volume data structure as a prerequisite.
- I have spent x hours designing, implementing and analysing my own on-line Deep Q-learning Path tracing algorithm, along with a neural network architecture designed for the algorithm.

0.0.1 Plan

Breakdown

- **Small Intro**: What is Path tracing (1-2 sentences)? Why is it important (1 sentence)? Seek for real-time ray-tracing (1 sentence). Importance sampling and how temporal difference learning is beginning to be used for importance sampling to avoid sampling rays which do not contribute to the creation of the image(1-2 sentences).
- **Aims & Objectives**: Evaluate the performance of the state of the art temporal difference learning method compared to a simple path tracer. Furthermore, I present a new deep Q-learning on-line path-tracing scheme and evaluate its performance against the state of the art temporal difference learning scheme and the default path tracer.
- **Outcomes**: From my experiments, Temporal difference learning has clear potential to reduce the number of zero contribution light paths sampled by the path tracer, even more so, deep q-learning is shown to outperform temporal difference learning (TODO)
- **Main areas of work**: Built a path tracing engine from scratch using only pixel & basic maths libraries. Researched into efficient light transport simulation. Reimplemented the Irradiance Volume Paper and Nvidia's Learning light the reinforced way paper. Researched into temporal difference learning & deep reinforcement learning. Developed a new deep q-learning path-tracing algorithm.

Preliminary

1. Path tracing is a ray-tracing method for rendering computer generated photo-realistic images by accurately approximating global illumination. Traditionally it has been thought to trade off rendering speeds with image quality.
2. The goal is to design and implement modified path-tracing algorithms which reduces the number of zero-contribution light paths in its estimation of global illumination. This will lead to less noisy image with the same number of samples per pixel. I have integrated different reinforcement learning algorithms into the path-tracing rendering pipeline, which was initially motivated by Nvidia's promising results when integrating Q-learning
3. More specifically, the task of the reinforcement learning AI agent is to learn for any given point in the scene the light power contribution from all incident angles. This is known as the Irradiance Distribution, the term introduced by (cite The Irradiance Volume). It is then possible to importance sample scattering directions from the learned irradiance distribution at a given point in the scene to dramatically reduce the number of rays scattered in directions giving zero-light power contribution, also known as zero-contribution light paths.
4. I have assessed different on-line reinforcement learning techniques for learning the irradiance distribution for any point in a scene by comparing metrics such as average path length, number of ray paths connecting to a light - generalization
5. I have spent x hours researching into reinforcement learning for Dynamic Programming methods, Monte Carlo methods, Temporal Difference methods, Deep reinforcement learning methods including monte-carlo and temporal difference learning approaches
6. I have spent x hours researching into the newly emerging field of learning light transport
7. I have created my own path-tracing graphics engine which supports naive path-tracing, reinforcement learning approach introduced by Ken Dahm, and my newly proposed deep reinforcement learning scheme for learning the irradiance distribution for any point in the scene

Supporting Technologies

1. I used the SDL2 library for displaying and saving rendered images from my Path tracing engine.
2. I used the OpenGL mathematics library to support low level operations in my Path tracing engine. It includes GPU accelerated implementations for all of its functions.
3. I used the CUDA Toolkit 10.1 parallel computing platform for accelerating Path tracing algorithms. This means the CUDA nvcc compiler must be used to compile my Path tracing engine.
4. All experiments were run on my own desktop machine with an Nvidia 1070Ti GPU, Intel i5-8600K CPU and 16GB of RAM.
5. I used the C++ API for the Dynet neural network framework to implement all of my Neural Network code as it is able to be compiled by the CUDA compiler.

Notation and Acronyms

An optional section, of roughly 1 or 2 pages

Any well written document will introduce notation and acronyms before their use, *even if* they are standard in some way: this ensures any reader can understand the resulting self-contained content.

Said introduction can exist within the dissertation itself, wherever that is appropriate. For an acronym, this is typically achieved at the first point of use via "Advanced Encryption Standard (AES)" or similar, noting the capitalisation of relevant letters. However, it can be useful to include an additional, dedicated list at the start of the dissertation; the advantage of doing so is that you cannot mistakenly use an acronym before defining it. A limited example is as follows:

AES	:	Advanced Encryption Standard
DES	:	Data Encryption Standard
	$\vdots$	
$\mathcal{H}(x)$	:	the Hamming weight of x
$\mathbb{F}_q$	:	a finite field with q elements
x_i	:	the i-th bit of some binary sequence x, st. $x_i \in \{0, 1\}$

Acknowledgements

An optional section, of at most 1 page

It is common practice (although totally optional) to acknowledge any third-party advice, contribution or influence you have found useful during your work. Examples include support from friends or family, the input of your Supervisor and/or Advisor, external organisations or persons who have supplied resources of some kind (e.g., funding, advice or time), and so on.

0.0.2 Plan

1. Carl Henrik Ek - Validating my understanding of deep reinforcement learning
2. Neill Campbell - Deep reinforcement learning strategy

Chapter 1

Contextual Background

1.0.1 Plan

Breakdown

- **What is it**: What is path tracing in computer graphics, related to ray-tracing. What is global illumination? Other methods of global illumination. A conceptual description of the path tracing algorithm and where importance sampling comes in and why it reduces image noise. This will definitely require an image of sampling on a diffuse surface and how we can benefit from importance sampling.
- How reinforcement learning links in with on-line learning in a path tracer and how what it learns can be used to reduce the number of zero-contribution light paths. A picture now relating to the one above but there is a blocker in the way of the light and the agent is able to learn that is not a favourable direction to shoot the ray in. Discuss how this work builds on top of Nvidias and on the topic of efficient light path sampling itself.
- Need to have something describing the potential for deep reinforcement learning to be applied compared to standard temporal difference learning.
- **Why is it important**: Where is it used in industry. It's current position and potential future usage in other industries. Why speed is becoming more valuable. While the goal of my thesis is not directly improve the speed of path tracing, it is to reduce the number of samples required in the algorithm to accurately approximate global illumination in hope that these more efficient sampling technique can be refined and optimised in future path tracing engines. Nvidia already showed it is possible to get their method to be only 20% slower than a standard path-tracer yet the superior results in image quality outweigh the time penalty. Reduces path length over time, so there is potential for the method to not only produce better images but be faster in doing so as well.
- **Who and why will they benefit from the project**: Film & Media industry from potentially improved render times. More targeted towards reaching real-time ray-tracing. Opens up more avenues to explore further deep reinforcement learning methods as part of the rendering pipeline e.g. actor-critic algorithm.
- **Challenges involved with the project**: Assessing the costs which the method proposed by Nvidia brings. Researching in how deep reinforcement learning applies to rendering equation. Designing an on-line algorithm for learning and a suitable network architecture. Accelerating the algorithms on a GPU in order to get results in a reasonable amount of time. Accelerating Nvidias algorithm on a GPU to find it's shortcomings.

Preliminary

1. Path-tracing in industry/ray-tracing in general, why is it important and how is the current field moving. Why should we optimise it algorithmically. Why should the reader care about path-tracing? - Usage in films, increasing interest for real-time simulations and gaming industry which is worth lots of money
2. High level overview of path-tracing: specifically must explain why it takes so long and why we care about the number of samples

3. In the path-tracing algorithm, a single pixel's colour is determined by firing multiple rays from the camera, through that pixel into the scene and building a colour value estimate for each one, then averaging their values to get the pixels colour. Each rays colour estimate is computed by estimating a solution to the recursive Rendering Equation (cite). The path-tracing algorithms estimate to this solution involves scattering the ray around the scene until it intersects with a light source. Therefore, if a ray is scattered in a direction with zero-light contribution, but other sampled rays are not, a noisy estimate is achieved for the pixel value unless many rays are sampled to reduce the effect of this noise. Therefore, avoiding scattering rays in directions of zero-light power contribution can reduce the number of samples needed to achieve an accurate estimate of a pixels colour value.

4. Work was primarily motivated by Ken & Dahms paper for modelling the irradiance distribution in order to reduce the number of zero-contribution light transport paths traced. Nvidia are world leaders in GPU manufacturing and drive the computer graphics forward.

5. Literature around efficiently simulating light transport - it's applicability to all modern used off-line rendering techniques

6. Aims & Challenges:

   (a) Implementing a path-tracer for diffuse surfaces from scratch using only maths and pixel libraries as helper functions which can handle imports of a custom scene
   (b) Accelerating path-tracer on Cuda to get results in a reasonable time
   (c) Implementing the irradiance volume data-structure and sampling technique which can adapt to any size scene
   (d) Implementing Ken Dahms proposed path-tracing algorithm with nearest neighbour search of KD-Tree on a GPU efficiently
   (e) Researching reinforcement learning: TD-Learning & deep reinforcement learning - never been taught before, so self taught with resources on-line
   (f) Training a network on pre-computed Q values to check if it is possible for a neural network to learn the irradiance distribution function for a set of points in a scene
   (g) Designing an algorithm to integrate deep reinforcement learning into the rendering pipeline for a path-tracer
   (h) Choosing a set of metrics to evaluate the algorithms performances on
   (i) Accelerating the algorithms via Cuda to run on Nvidia GPU

Chapter 2

Technical Background

2.0.1 Plan

Breakdown

- The rendering equation and what each component is, how this relates to global illumination
- Path tracings use of the rendering equation. How monte carlo comes into play. The iterative version of the path tracing algorithm. Concept of a light path.
- Importance sampling in terms of BRDF and relating this to reducing variance in pixel colours leading to less noise. Give examples of classical importance sampling techniques and their performance. Critic them and clearly present where their shortcomings are and how they are unavoidable.
- Introduce reinforcement learning: Markov Decision Process, Bellman Equation, Temporal Difference Learning and its strong points and weaknesses, how does it differ to traditional monte-carlo (might not be relevant). Proved to converge on the true valuation function for a given state-action pair when run infinitely
- Give Nvidia's derivation of their learning rule. How does the Markov Decision Process relate to a rendered scene, i.e. what is the AI doing for us here. Provide a justification of parameter matching. Essentially cover all reinforcement learning theory of the paper here, with a justification (mathematical) and visual examples of why it works.
- Discretizing the state space is required for Q-learning to be applied, shortcoming is that it may not work very well with infinite state spaces. Introduce the Irradiance volume and how it can be used to rather store actual irradiance values to instead store Q-values. The irradiance distribution for a given point in the scene. Sampling the irradiance volumes around the scene onto geometry.
- Present the full algorithm proposed by Nvidia, displaying irradiance volumes learned Q-values (as an image of hemispheres) throughout the process and stating how these update a cumulative distribution to sample from.
- Introduce concept of Deep Q-learning and how it no longer needs a discretized state space. However it still requires action space to be discretized (unlike an actor-critic setup). What is the role of the network and what other function approximators can be trialled. Explain in quite some detail the DeepMind Atari paper which introduced Deep Reinforcement learning.

Preliminary

1. Define what a ray-tracing rendering algorithm consists of and the difference between global and direct illumination. Acknowledge other ray-tracing algorithm l ike bi-directional path-tracers, Rendermans algorithm, photon mapping.
2. Define terms like BRDF, radiance, irradiance and the rendering equation
3. Explain the details of the path-tracing algorithm in depth. It should be completely clear the relation between path-tracing and the rendering equation. It should be clear where the Monte Carlo approach comes in and why importance sampling within path-tracing can yield less noisy and more accurate results, potentially in the same fixed time-budget

4. Introduce the concept of importance sampling in computing global illumination with some early examples of its success, use in industry and recent papers on efficient light transport simulation. State the reasoning behind why it still continues to accurately simulate global illumination, in other words, why zero-contribution light paths do not contribute to the image.

5. Introduce reinforcement learning: Markov Decision Process, Bellman Equation, Temporal Difference Learning and its strong points and weaknesses, how does it differ to traditional monte-carlo (might not be relevant). Proved to converge on the true valuation function for a given state-action pair when run infinitely

6. State the derived learning rule supplied by Ken Dahm and visualize the matching terms as well as a justification why each parameter matches. What is the value and the incentive, diminishing return for rewards far in the future etc

7. State new on-line algorithm proposed by Ken Dahm and details for discretizing the state and action space into the Irradiance Volume data-structure which was previously introduced

8. Introduce the concept of deep reinforcement learning, describing how DeepMind used the technique for playing Atari games. Given a state give me the state-action values for all actions possible in that state. Then how we can apply this to our scene to model the state space and continuous.

Chapter 3

Deep Q-Learning Path tracer

3.0.1 Plan

Breakdown

1. State learning rule for deep Q-learning and the difference from deep Q-learning to q-learning. Maybe some of the difficulties associated with deep q-learning versus q-learning, and some of the general advantages.
2. Derive the learning rule for deep q-learning network which I used, once again justifying terms throughout the derivation.
3. Explain concept of eta-greedy policy used. Explain exploration vs exploitation (this should be a big section, potentially more later on about this as it heavily dictates how photo-realistic and image looks).
4. Describe how the current method is used for diffuse surfaces. Introduce the pseudo code for the new algorithm. Give a description of each stage and what it does. Relating back to properties such as bias rendering and pointing out assumption made by the path tracer.
5. Present and explain the network architecture. Explain in depth about how the state was modelled as a point relative to all vertices to give the network information about the position of the vertex relative to the rest of the world compared to passing in a single position. Relate this to Atari games, we get an image showing where we are relative to the world rather than just a single position in the world.
6. Present some results side by side against a default path tracer and Nvidia's reinforcement learning approach. Pointing out aspects of the image and reasoning for certain parts.

Chapter 4

Critical Evaluation

A topic-specific chapter, of roughly 15 pages

This chapter is intended to evaluate what you did. The content is highly topic-specific, but for many projects will have flavours of the following:

1. functional testing, including analysis and explanation of failure cases,
2. behavioural testing, often including analysis of any results that draw some form of conclusion wrt. the aims and objectives, and
3. evaluation of options and decisions within the project, and/or a comparison with alternatives.

This chapter often acts to differentiate project quality: even if the work completed is of a high technical quality, critical yet objective evaluation and comparison of the outcomes is crucial. In essence, the reader wants to learn something, so the worst examples amount to simple statements of fact (e.g., "graph X shows the result is Y"); the best examples are analytical and exploratory (e.g., "graph X shows the result is Y, which means Z; this contradicts [1], which may be because I use a different assumption"). As such, both positive *and* negative outcomes are valid *if* presented in a suitable manner.

4.0.1 Plan

1. Show for about 4 different scenes the results for a n different numbers of samples; the images, average path length, number of light paths which actually contribute to the image which are sampled between all techniques. I will have to analyse which reduces the number of zero contribution paths the most, but also still assess if the image is photo-realistic.
2. Also analyse default Q-learnings ability on top of expected SARSA
3. Justify reasoning for choosing to analyse Q-Learning, Expected SARSA and DQN (because they have good results for other cases and TD learning fits the online learning procedure)
4. Assess the number of parameters required, configuration is important for these algorithms, if it is very difficult to get right, then the time spent configuring may not be worth it compared to actually rendering the image. E.g. default path-tracing there are not other parameters apart from the number of samples per pixel, expected SARSA requires the user to specify the memory which is allowed to be used by the program, this requires careful consideration, as well as the threshold the distribution cannot fall below, the deep Q-learning algorithm requires less config but potentially different neural network architectures should be investigated to further reduce the number of zero-contribution light paths.
5. Ease of implementation
6. Parallelisability of each algorithm, path-tracing is far easier to parallelise as it requires minimal memory accesses by the program to infer pixel values, as opposed to expected SARSA which requires many. Deep-q learning has more customizability in terms of parallelizing (needs more research)

7. Memory usage: Path-tracing is minimal, Expected SARSA is unbounded, Deep Q-Learning is bounded by the size of the neural network, but the memory it requires is still significant (needs more research)

8. DQN vs Expected Sarsa: Do not have to wait for an iteration to begin importance sampling on the newly learned Q values for a given point, neural network is continually trained and inferred from. Continuous state space vs discretized required for storage in expected SARSA.

Chapter 5

Conclusion

A compulsory chapter, of roughly 5 pages

The concluding chapter of a dissertation is often underutilised because it is too often left too close to the deadline: it is important to allocation enough attention. Ideally, the chapter will consist of three parts:

1. (Re)summarise the main contributions and achievements, in essence summing up the content.
2. Clearly state the current project status (e.g., “X is working, Y is not”) and evaluate what has been achieved with respect to the initial aims and objectives (e.g., “I completed aim X outlined previously, the evidence for this is within Chapter Y”). There is no problem including aims which were not completed, but it is important to evaluate and/or justify why this is the case.
3. Outline any open problems or future plans. Rather than treat this only as an exercise in what you *could* have done given more time, try to focus on any unexplored options or interesting outcomes (e.g., “my experiment for X gave counter-intuitive results, this could be because Y and would form an interesting area for further study” or “users found feature Z of my software difficult to use, which is obvious in hindsight but not during at design stage; to resolve this, I could clearly apply the technique of Smith [7]”).

5.0.1 Plan

1. Summarise contributions:
   (a) Implementing a path tracer from scratch to analyse in depth the difficulties and issues that come with Ken Dahm’s algorithm. Including memory usage, parallelisation and parameter usage.
   (b) Analysis of different reinforcement learning approaches pitched together clearly on a variety of scenes
   (c) Analysis of neural networks ability to learn the irradiance distribution function
   (d) Online deep-reinforcement learning algorithms effectiveness of learning irradiance distribution function
2. If DQN does not work well provide some further analysis on potential other alternatives which could be used.
3. Future Work: Policy learning to model continuous action & state space
4. DDQN and other deep reinforcement learning strategies

Bibliography

[1] L. Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, 1986.

[2] F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley. *The LaTeX Companion.* Addison-Wesley, 2nd edition, 2004.

Appendix A

An Example Appendix

Content which is not central to, but may enhance the dissertation can be included in one or more appendices; examples include, but are not limited to

- lengthy mathematical proofs, numerical or graphical results which are summarised in the main body,
- sample or example calculations, and
- results of user studies or questionnaires.

Note that in line with most research conferences, the marking panel is not obliged to read such appendices.