

Back to the Future: Using Past Frames to Optimise Video Games

Honours Proposal

Andrew Allan

1002610

BSc Computer Games Technology (Hons)

# Contents

1. Abstract 2.
2. Introduction 2.
3. Research Question & Objectives 3.
4. Context 3.
5. Methodology 5.
6. Summary 5.
7. References 6.
8. Bibliography 7.

# Abstract

In video games running at real time frame rates, the difference between each frame image is minimal. This occurrence is known as temporal coherence. The aim of this project is to investigate the utilization of temporal coherence to optimise rendering in video games through the use of previously rendered frames.

An application shall be built for the project, which will use previously rendered frames to aid in the rendering of new frames. The application shall be measured against a traditional method, where it is hoped that rendering performance can be improved. It is also anticipated the visual difference between the two methods shall negligible.

# Introduction

The gold standard in which modern AAA video games are held by audiences is 1080p at 60 frame per second. This was first possible from the Xbox 360 and PlayStation 3 generation of consoles. Yet few games have achieved this standard, even with the arrival of newer hardware. The reason for this is the increasingly sophisticated techniques used to create real world fidelity graphics, such as deferred lighting and real-time shadows. These techniques take time to calculate, and by reducing the frame rate or the resolution, the time allowed to carry out these techniques is increased.

To improve performance however, rather than looking for gains in each of these individual techniques, a different approach would be to examine methods of reducing the bandwidth of data calculated each frame. This could provide a more general optimization and help more games meet this gold standard. It could also potentially lower the hardware barrier for some of these techniques allowing them to be used in more games.

It can be observed that in video games running at high frame rates there is a considerable amount temporal coherence between frames, meaning “there is usually very little difference in the camera and lighting parameters, as well as in the set of visible surface points, their properties, and final appearance” (Nehab, D 2007). It can also be noted that current rendering strategies start each frame from scratch, where the exact same calculations could be carried out again (Bowles et al, 2012). By utilising temporal coherence, the pixel data generated in previously rendered frames can be used to assist in the generation of new frames.

This project shall explore exploiting this temporal coherence present in games, with an overall aim to reduce the cost of rendering the pixel shader each frame. It shall draw upon the existing methods from both academia and commercial games to identify and implement a general temporal coherence technique suitable for video games.

# Research Question

“What performance gains could achieved by taking advantage of temporal coherence in video games?”

# Aims & Objectives

Aim**:**

* To implement a rendering pipeline that utilises temporal data
* To assess the performance gains in rendering times by using this pipeline in a real time application
* To analyse the quality of the images produced using this pipeline

Objectives**:**

* To research current techniques and identify the most appropriate for video games
* Implement a temporal based rendering pipeline
* Produce a real time scene containing common features of video games
* Measure the render times each frame in the scene
* Analyse the breakdown of the render times
* Compare the recorded render times against the same scene rendered with a traditional pipeline
* Compare the visual quality of the scene rendered with both pipelines, with particular attention paid to any artefacts or blurring produced

# Context

The term ‘temporal coherence’ has existed since the advent of computer graphics itself. The idea was first discussed by Sutherland et al, (1974) under the term ‘frame to frame coherence’ where it is stated that “in a sequence of movie frames the successive frames are likely to be closely related”. This point is also relevant in video games and is illustrated by consecutive frame screen shots from a video game in figure 1. Nehab et al, (2006) also highlighted the presence of temporal coherence in real time rendering by rendering previous visible surface points in green and new surface points in red each frame, shown in figure 2.



**Figure 1**: Showing the occurrence of temporal coherence in video game (Thief, 2014)



**Figure 2**: A camera moving around a scene containing The Parthenon.(Nehab et al, 2006)

The occurrence of temporal coherence in real time graphics can be viewed as a potential optimisation. Scherzer et al, (2011) states “a higher resolution and frame rate do not necessarily imply a much higher workload, but a larger amount of redundancy and a higher potential for amortizing rendering over several frames”.

Early uses of temporal coherence in computer graphics were used as means to accelerate CPU expense ray tracer algorithms. Havran, (2003) used temporal coherence in a ray casted walkthrough, this to reuses ray object intersections to reduce ray traversals. Walte, (1999) used TC to run a high quality ray based renderer on a conventional CPU without the need for a GPU.

In recent years a number of researchers have taken to researching temporal coherence to improve the efficiency of pixel shaders. Nehab et al, (2006) and Scherzer et al, (2007) both proposed a reverse reprojection caching method. This method stores pixels as they are generated into a frame buffer, as new pixels are generated in successive frames, they are tracked back to this buffer and reused if present. Yang et al, (2011) and Andreev, (2010) offered a system for bi-directional reprojection, which places interpolated frames in between traditionally rendered frames to increase the frame rates. Herzog et al, (2010) and Valient (2014) suggested a temporal up-sampling method, which stores up a history of lower resolution frames and integrates them to create a high quality image each frame.

An ideal temporal coherence method for video games should have minimal lag in displaying new images, (Didyk, 2010), be suitable for fill-bound scenes which are present in all games, (Yang, 2011) and have minimal overhead as resources may be limited (Andreev, 2010). It is also important that the pixel shader of the game consumes more of the render budget then the other shaders in the pipeline to provide the best optimisation, (Sitthi-amorn 2008).

If a video game is to run at 60 frames per second, the screen must be refreshed every 16.6 milliseconds. This however does not mean the whole time is available for rendering. The time is also taken up by other systems vital to the game, this non-exhaustive list includes animation, user input, AI, networking, physics and audio (Gregory, 2009). Therefore any measure on rendering performance should be a measure of time taken and not frames per second.

# Methodology

The initial step of the project shall be to investigate the current methods of utilising temporal coherence in graphics applications. This shall be done in the context of optimisation in video games, with a view to identify the method best suited for game applications. From this investigation the method with the most potential shall be selected to be implemented.

The application shall be built into a DirectX 11 framework where the render pipeline will be altered to utilise temporal data. The application shall also be capable of rendering using convention methods to provide a comparison. A scene shall be constructed within the application to demonstrate a number of key features common in all video games. These features include animation, UI, lighting, a particle system and post-processing.

By building this application it is hoped that we can demonstrate that by utilizing temporal data it is possible to improve the performance of rendering times compared when using conventional methods. The application shall also be used to show that the difference in image quality between the methods is minimal.

The application shall be used to evaluate the effectiveness of using temporal coherence in video games. This effectiveness shall be judged using two different methodologies; a quantitative assessment of the rendering performance and a qualitative evaluation of the quality of image produced.

The quantitative data shall be a collected using Microsoft Visual Studio 2013’s GPU Usage Tool. This tool provides high level performance information and GPU usage for Direct3D applications. The tool shall be utilized to precisely record the render times each frame. The data shall be monitored and recorded when using both the temporal data pipeline and the conventional one. The two sets of data shall be compared to highlight if there is any performance differences between the two pipelines. Each step of the temporal data pipeline will examined, to identify which rendering tasks are consuming the most time. This information shall be utilised to highlight areas for further optimisation

The qualitative information for the evaluation shall be conducted as a visual comparison between the two render pipelines. The comparison shall assess and record any visible difference between images produced by both render pipelines. A further analysis of the images rendered using the temporal data pipeline shall be undertaken, with particular emphasis on the presence of blurring or artefacts.

# Summary

This project shall investigate the performance benefits of utilising temporal coherence to optimise rendering in video games. This will be achieved by building an application that uses temporal data in its renderer to aid the creation of new frames. It is believed that the application will show significant performance benefits over traditional methods in scenes with a certain degree of pixel shader complexity. The application is also expected to show that visually there is no significant difference between either methods of rendering. The goal of the project will be to advise on which temporal coherence method is best suited for optimising video games. The project will also provide recommendation on achieving the best results and advise on areas for future work.

# References

Andreev, D. 2010. *Real-time Frame Rate Up-conversion for Video Games.*[online]. Available from:<http://delivery.acm.org/10.1145/1840000/1837047/a16-andreev.pdf?ip=193.60.168.73&id=1837047&acc=ACTIVE%20SERVICE&key=C2D842D97AC95F7A%2E42BFACAD77C85A68%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=605458846&CFTOKEN=71679790&__acm__=1417788857_46eeff5e9d71d87e8c387d07cee32f9c>.

Bowles, H. et al. 2012. *Iterative Image Warping.*[online]. Available from: <https://graphics.ethz.ch/Downloads/Publications/Papers/2012/Bow12/Bow12.pdf>.

Didyk, P. et al. *Perceptually-motivated Real-time Temporal Upsampling of 3D Content for High-refresh-rate Displays.* <http://resources.mpi-inf.mpg.de/3DTemporalUpsampling/3DTemporalUpsampling.pdf>. [online]. Available from: 2010.

Havran, V., Bittner, J. and Seidel, H. 2003. *Exploiting**Temporal Coherence in Ray Casted Walkthroughs.*[online]. Available from:<http://delivery.acm.org/10.1145/990000/984977/p149-havran.pdf?ip=193.60.168.73&id=984977&acc=ACTIVE%20SERVICE&key=C2D842D97AC95F7A%2E42BFACAD77C85A68%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=605458846&CFTOKEN=71679790&__acm__=1417787963_8ded7475183801eba1a62e8d3c26a241>.

Nehab, D., Sander, P. and Isidoro, J. 2006. *The Real-Time Reprojection Cache.*[online]. Available from: <http://www.csee.umbc.edu/~olano/s2006c03/ch06.pdf>.

Nehab, D. et al. 2007. *Accelerating Real-Time Shading with Reverse Reprojection Caching.*[online]. Available from:<http://gfx.cs.princeton.edu/pubs/Nehab_2007_ARS/NehEtAl07.pdf>.

Scherzer, D. et al. 2011. *A Survey on Temporal Coherence Methods in Real-Time Rendering.*[online]. Available from: [http://www.cse.ust.hk/~psander/docs/tempco.pdf](http://www.cse.ust.hk/~psander/docs/tempco.pdf" \t "_blank).

Scherzer, D., Jeschke, S. and Wimmer, M. 2007. *Pixel-Correct Shadow Maps with Temporal Reprojection and Shadow Test Confidence  .*[online]. Available from: <http://www.cg.tuwien.ac.at/research/publications/2007/Scherzer-2007-PCS/Scherzer-2007-PCS-Preprint.pdf>.

Sitthi-amorn, P. et al. 2008. *An Improved Shading Cache for Modern GPUs.*[online]. Available from:<http://www.cs.virginia.edu/~jdl/papers/reproj/amorn_gh08.pdf>.

Sitthi-amorn, P. et al. 2008. *Automated Reprojection-Based Pixel Shader Optimization.*[online]. Available from:<http://www.cs.virginia.edu/~gfx/pubs/SitthiAmorn_2008_ARP/amorn_siga08.pdf>.

Sutherland, I. et al. 1974. *A Characterization of Ten Hidden-Surface Algorithms.*[online]. Available from:<https://design.osu.edu/carlson/history/PDFs/ten-hidden-surface.pdf>.

Yang, L. et al. 2011. *Image-Based Bidirectional Scene Reprojection.*[online]. Available from: <http://research.microsoft.com/en-us/um/people/hoppe/bireproj.pdf>.

# Bibliography

Drobot, M. Chapter 7: A spatial and temporal coherence framework for real-time graphics. In: E. Lengyel. ed. *Game engine gems 2.*1st ed. A K Peters Ltd. 2011, pp. 97.

Torkan, P. 2014. *Regarding Killzone Shadow Fall and 1080p.*[online]. Available from: <https://www.killzone.com/en_GB/blog/news/2014-03-06_regarding-killzone-shadow-fall-and-1080p.html>.

Valient, M. 2014. *Killzone Shadow Fall: Image Quality into the Next Generation.*[online]. Available from: <http://www.guerrilla-games.com/presentations/GDC2014_Valient_Killzone_Graphics.pdf>.

Yang, L. et al. 2009. *Amortized Supersampling.*[online]. Available from: <http://www.cs.virginia.edu/~gfx/pubs/Yang_2009_AMS/yang2009.pdf>.

Zhu, T. et al. 2005. *A GPU-Accelerated Render Cache.*[online]. Available from:<http://www.cs.virginia.edu/~gfx/pubs/Zhu_2005_GRC/zhu2005.pdf>.