

Final Project: Advanced Real-time Rendering

Introduction

In this project, you will get the opportunity to implement one of the advanced algorithms for real-time rendering we talked about in class this semester. This project is more open-ended than the first one – the actual project you decide to undertake is up to you. I just want to see that you are able to understand and implement a sophisticated technique and get it working on the GPU in real-time. This project will be a fair amount of work, so do not leave it until the last minute. I will give you a grade based on the quality of your implementation and your results. The project will be due Wednesday, May 9 at 11:59pm via electronic submission.

Potential Projects

To help make the project more concrete and make it easier to assign you a grade, I now list several projects which I feel would be appropriate for a final project for this class. Choose one of the following projects to implement:

1. Soft shadow algorithms

We spent some time discussing real-time soft-shadow algorithms. Pick either penumbra wedges (Akenine-Moller et al.) or smoothies (Chan et al.) and implement the algorithm. Make sure that it runs at real-time rates and that it gives results similar to those presented in the papers. You will need to test your algorithm on an interesting scene, preferably with animated characters.

2. Pre-computed Radiance Transfer

Illuminating real-time characters from environment maps is an interesting area of research in real-time graphics. If you choose this topic, you will need to implement one of the major algorithms we discussed in class. One possibility is to implement the efficient, spherical-harmonic based irradiance maps of Ramamoorthi et al. Another possibility is to implement the pre-computed radiance transfer as described by Sloan et al. You can use an off-the-shelf global illumination simulator to pre-compute the radiance transfer (ask me if you are looking for one) or you can write your own in OpenGL. Feel free to use the environment maps provided by Paul Debevec.

3. Procedural graphics

We also talked about specifying geometry procedurally for real-time applications. If you choose to do this project, you need to build either an L-system for procedural plants or a terrain modeling system like we talked about in class. The procedural plants should model several different kinds of plants and you should be able to control the amount of detail in each. The final “deliverable” would be an environment you can explore which is filled with these plants. If you choose to do the terrain modeling, you should model things like erosion and shade the terrain appropriately. The deliverable could be a flight simulator that would allow you to fly over the procedural terrain

4. GPU-based Physics

Recently, a lot of folks are porting the physics computations to the GPU to free up the CPU to do other tasks. If you choose to do this project, you will need to write a GPU simulator for a physical process. Some examples include a complex multiple-particle system, cloth simulation, etc. Your physics simulation should run and be rendered on the GPU.

5. GPGPU

The programmability of GPU's has recently made it possible to perform general purpose computing on the GPU (GPGPU). An interesting final project for this class would be to try to solve some general problem on the GPU. The field of the project could be anything, from economics to simulation. However, if you choose to do a GPGPU project please talk to me first to make sure that the project you plan to work on is appropriate.

6. Real-time ray tracing

Ray-tracing has traditionally been an off-line algorithm for generating high-quality, photorealistic images. In recent years, however, people have become interested in implementing real-time ray tracing because it can simulate reflections, refractions, and diffuse illumination better than scanline rendering systems. If you choose this project, you will need to write a ray tracer that either runs on the GPU (like that of Purcell et al.) or runs extremely fast (interactive rates, at least 10fps) on the CPU (or a cluster of CPUs).

7. Choose your own

Finally, I will give you the option to choose your own project. If you choose to do your own project, you will need to talk to me for approval. I need to see what you will be doing to make sure that it is at the appropriate scope and level for the final project in this class.

Getting started

Before you begin implementing anything, I will ask you that you contact me and let me know what project you are thinking of doing. I just want to make sure that we are on the same page. In fact, we should keep in communication through out this project so that I know exactly what you are doing and we can avoid any unpleasant surprises in the end. I will not be in town the week of April 30 and might have limited email access. So try to get as much done on the project before then. Also remember that the week of May 7 is finals week. If you have other final exams, assume that you will get little done this week and so try to get more done before then.

I understand that implementing complex algorithms like the ones described can be quite involved. While you need to write the code for the core ideas of the project, I will let you use existing code for all the "helper" functions that are often needed. For example, if you need to read in a scene for your project you can use any of the existing scene-reader code bases available on the web. If you need environment maps, you can use Paul Debevec's light probes that he has available on his website. However, you will always need to program the main parts of the algorithm yourself. If you have any questions as to

whether or not you can borrow some code, ask me first. Of course, make sure that you cite anything you borrowed in your write-up.

In terms of time, the project is due Wednesday, May 9. I would recommend that you lock down your idea by Wednesday April 18 so that you can start making progress on the project immediately. This will give you three full weeks to work on the project. I would expect this project to take over 100 hours of work, so set aside several hours each day to work on this project. If you don't budget your time wisely, you might not finish!

Final Report

The deliverables of this project will be a zip file with your code and the .exe files of any demos. You should also do a write up, and I will ask you to write it as if you were writing a technical paper. Use the SIGGRAPH template available here:

<http://www.siggraph.org/publications/instructions/>

Tips & Advice

In order to implement the algorithm you choose, you should first read the appropriate papers carefully. Go to the original work, then read some of the white papers and "intro" presentations commonly found at GDC and places like that, then go back and re-read the original. You often don't understand the work until you have read it a few times. If you have any questions about an algorithm, ask me. Although you have over three weeks to work on this project, try to start the implementation by this week if possible. Don't leave things to the last minute or you will not finish the project.

Grading

The following breakdown will be used to grade the project:

Implementation	50
Quality	40
Write up	10
<hr/>	
TOTAL	100

Implementation refers to the functionality of your project. If you show me a working piece of code that implements the proper algorithm, you are sure to get nearly full credit on this. The "quality" score will reflect how good the implementation is. Did you go all out or did you do the bare minimum? This portion of the grade will be strongly influenced by how your peers performed in their own projects. Finally the "write up" would judge the quality of the final report you hand in. If I have trouble running your code, I might ask you to meet with me either on Thursday or Friday (May 10 or 11) to go over your submission. I want to make sure I thoroughly understand what you did!

Warnings

Since this project is so open-ended, this will require close communication between you and I to ensure that you get the grade you expect. If you don't let me know what project you have chosen or what you are doing, you could be unpleasantly surprised with a low score if I felt that the project was not ambitious enough. On the other hand, you can run

into trouble if you chose something too ambitious and cannot finish. I will try to help you find something that is just right.

Due Date

The project will be due Wednesday, May 9 at 11:59pm. No late submissions will be accepted because grades are due shortly afterwards and I don't give incompletes. You must e-mail me the zipped version of your code, executables, resource files (images, environment maps, geometry data, etc), and the report. Like I said earlier, I might ask you to meet me the Thursday or Friday afterwards to go over the project together if I have trouble running anything.