

Concurrency and Parallelism 2018-19

(6) Ray Tracer

João Lourenço

October 3, 2018

Abstract

In this class you will improve your knowledge on Cilk+, on C++, on parallel patterns, on development of parallel programs, and on the understanding of existing third-party programs.

1 Introduction

Ray Tracing is a image rendering technique for generating an image by tracing the path of light as pixels in an image plane and simulating the effects of its encounters with virtual objects. The technique is capable of producing a very high degree of visual realism, usually higher than that of typical scanline rendering methods, but at a greater computational cost. This makes ray tracing best suited for applications where the image can be rendered slowly ahead of time, such as in still images and film and television visual effects, and more poorly suited for real-time applications like video games where speed is critical. Ray tracing is capable of simulating a wide variety of optical effects, such as reflection and refraction, scattering, and dispersion phenomena (such as chromatic aberration)¹.

Please use Google to learn more about Ray Tracing.

¹([https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)))

2 Lab Work

In this lab work you are given a working sequential version of a Ray Tracer written using the C++ programming language. You are asked to study the given code and create an optimized (parallel) version of the code using Cilk+.

2.1 Given Version

You may find at

`https://bitbucket.org/cp201819/raytracer.git`

a running sequential version of the Ray Tracer.

Clone the given version in your Linux device (own laptop, lab workstation, or as a last resort, in the “node9” server used in the last lab class) with

`git clone https://bitbucket.org/cp201819/raytracer.git`

and compile it using the command `make`. Make is a command that builds a project given in a project specification file named `Makefile`. Have a look at this text file and learn a bit about `make` and `Makefile`. You’ll need it again in the short future.

Try running the sequential version of the ray tracer and see the results.

2.2 Work plan

The given version of the Ray Tracer is sequential.

Your job is to make an optimized (parallel) version of this program using Cilk+!

Please follow these steps:

1. Clone the given project.
2. Compile and experiment with the given version. Study the source code and understand how it works.
3. Optimize the given code.
4. Experiment with different numbers of processors (by setting the environment variable `CILK_NWORKERS`).

Please remember to use GIT appropriately. Learn to work with branches and make a branch when you start a new phase from the list above, and merge your branch to your main version when the development is finished and appropriately tested/validated.

Acknowledgments

The text from the Introduction is an adaptation from the text in [https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)).