Lab 5

In this lab, you’ll be measuring the effect (good and bad) of using constant memory.

Constant memory

We begin by looking at the speedup from using constant memory for the book’s ray tracing example. Begin by using ssh with the -Y flag to reach huygens; you need -Y because the program will have graphical output and huygens because that is the machine on which I installed the graphics library (OpenGL). Copy the lab5 directory from the course directory:

```
   cp -R /home/courses/cs180j/lab5 .
```

(The -R makes it a recursive copy so that this command will copy the directory and its contents.) The files ray.cu and ray_noconst.cu in this directory give the code for the last two programs in the chapter, the complete timed ray tracer with and without constant memory, respectively. The other files are headers included with the book. (I did modify how the headers are included so that the compiler looks for them in the same directory rather than where the book’s source code puts them.) To compile these files you’ll need to add the flags -lpthread -lGL -lglut to the compilation line. The last two are for the graphics; the first one is for the pthread library (POSIX-standard threads), which I don’t believe we use, but is apparently used in the common included code. Run these two programs and compare their running time to compute the speedup achieved for this problem by using constant memory. (On huygens, I see a more modest improvement than the 50% suggested in the book, though I do get this much speedup on my laptop.)

After that, see if you can create a program which is slowed when constant memory is used. According to the book, memory performance will suffer when different threads in a half-warp simultaneously read from constant memory at different addresses. Modify the program so that different threads request the spheres in different orders. Alternately, create multiple copies of each sphere and have different threads use different copies. (You won’t have 1 order or 1 copy per thread; just ensure that threads sharing a halfwarp are requesting different spheres.) Think about how to do this and talk to me if you have any questions. You shouldn’t need to make many modifications to the program to see significant slowdowns from this effect.