**Blue Waters Petascale Semester Curriculum v1.0**

**Unit 7: CUDA**

**Lesson 4: Volume Rendering**

**Exercise Instructions for Students**

*Developed by Michael D. Shah for the Shodor Education Foundation, Inc.*



*Except where otherwise noted, this work by The Shodor Education Foundation, Inc. is licensed under CC BY-SA 4.0. To view a copy of this license, visit*[*https://creativecommons.org/licenses/by-sa/4.0*](https://creativecommons.org/licenses/by-sa/4.0)

*Browse and search the full curriculum at*[*http://shodor.org/petascale/materials/semester-curriculum*](http://shodor.org/petascale/materials/semester-curriculum)

*We welcome your improvements! You can submit your proposed changes to this material and the rest of the curriculum in our GitHub repository at*[*https://github.com/shodor-education/petascale-semester-curriculum*](https://github.com/shodor-education/petascale-semester-curriculum)

*We want to hear from you! Please let us know your experiences using this material by sending email to* [*petascale@shodor.org*](mailto:petascale@shodor.org)

* Download and run the Simple Cubemap Texture example from the NVidia GPU Toolkit.

<http://developer.download.nvidia.com/compute/DevZone/C/html_x64/Volume_Processing.html>

* The recommendation is to get the 'Simple Layered Texture' example.
* Students may need to install some libraries like GLUT

e.g. sudo apt-get install freeglut3-dev

* Students may need to run and install GLEW as well
  + e.g. Visit <http://glew.sourceforge.net/> and run the 'make' command in the root directory.
  + Then copy over any shared libraries needed.
* Compiling can be a little tricky, but here is a simple compile command that should work on linux:

e.g. nvcc -L./../../common/lib/linux/x86\_64/ -I../../common/inc volumeRender\_kernel.cu volumeRender.cpp -o volumeRender -lGL -lGLU -lX11 -lXi -lXmu -lglut -lGLEW

* If there are linker errors for finding any libraries, you may have to try:  
  export LD\_LIBRARY\_PATH=../../common/lib/linux/x86\_64/
* The example can then be run with ./volumeRender