AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Volume Rendering

1 Overview

1.1 Sample

This sample can be used as a complete stand-alone sample. It can be placed anywhere in the file system. To incorporate this sample into the SDK samples, copy the contents of the `\\VolumeRendering\samples\opencl\cl\app\VolumeRendering` folder into the `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\cl\app\VolumeRendering` folder.

To add this sample in `OpenCLSamples.sln`:

1. Open `OpenCLSamples.sln` in Microsoft Visual Studio 2008 or `OpenCLSamplesVS10.sln` in Visual Studio 2010.
2. Right-click on the `app` folder in the Solution Explorer, and select *add->existing project*.
3. Browse to `VolumeRendering.vcproj` (in Visual Studio 2008) or `VolumeRendering.vcxproj` (in Visual Studio 2010), and add it.
4. Right-click the Solution `OpenCLSamples` in the Solution Explorer, and select Properties.
5. Select the Project Dependencies tab, and add `SDKUtil` as a dependency for the VolumeRendering project.

To add this sample to the main `make` file, so it compiles with other SDK samples:

1. Go to the `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\cl\app` folder.
2. Add VolumeRendering to the `SUB_DIRS` list.

1.2 How to Build

Compile the sample.

To do this under Windows

1. With Visual Studio: Open `VolumeRendering.sln` with Visual Studio 2008 Professional Edition, or open `VolumeRenderingVS10.sln` with Visual Studio 2010 Professional Edition; then, select `build` by right-clicking on the solution name.
2. With Make files: Type `make` to build the sample from the Visual Studio command prompt. If no options are specified, `make` builds for the native platform in debug configuration. To select the release configuration, add the option `release = 1`. To force a 32-bit build on a 64-bit platform, add the option `bitness=32`.

To do thus under Linux

Type `make` to build the sample. If no options are specified, make builds for the native platform in debug configuration. To select release configuration, add the option `release = 1`. To force a 32-bit build on a 64-bit platform, add the option `bitness=32`.

Note : Ensure that AMDAPPSDKROOT and AMDAPPSDKSAMPLESROOT are set to the appropriate location.

1.3 How to Run

Typing VolumeRendering renders 3D volume data that is provided in an input RAW file.

1.4 Command Line Options

Table 1 lists, and briefly describes, the command line options.

Table 1 Command Line Options

Short Form	Long Form	Description
-h	--help	Shows all command options and their respective meaning.
	--device	Devices on which the program is to be run. Acceptable values are cpu or gpu.
-q	--quiet	Quiet mode. Suppresses all text output.
-e	--verify	Verify results against reference implementation.
-t	--timing	Print timing.
	--dump	Dump binary image for all devices.
	--load	Load binary image and execute on device.
	--flags	Specify compiler flags to build the kernel.
-p	--platformId	Select platformId to be used (0 to N-1, where N is the number of available platforms).
-d	--deviceId	Select deviceId to be used (0 to N-1, where N is the number of available devices).
-i	--iterations	Number of iterations for kernel execution.

2 Implementation Details

The sample renders a 3D volume box from a data file using ray casting. The data is loaded into 3D textures that are provided in OpenCL; these originally consist of a number of 2D slices representing a 3D volume. The normalized 3D texture data lies in a box between end points (0, 0, 0) and (1, 1, 1).

A camera and a near location is placed in –Z direction. For each pixel on the near plane, a ray is fired from the camera through the near plane. If the ray does not intersect the volume box, the kernel returns; otherwise, data is sampled using a fixed step size and pixel color. This data is accumulated using a linear blending equation.

Note - The sample works with other volume data available at volvis.org. The volume dimensions may have to be changed in the VolumeRendering.hpp as these are hardcoded to 256x256x256, according to the provided volume data aneurism.

3 References

The OpenCL code is based on a pixel shader code available at:
http://graphicsrunner.blogspot.com/2009/01/volume-rendering-101.html

The volume data aneurism.raw was acquired from www.volvis.org.

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For AMD Accelerated Parallel Processing:

URL:	**developer.amd.com/appsdk**
Developing:	**developer.amd.com/**
Support:	**developer.amd.com/appsdksupport**
Forum:	**developer.amd.com/openclforum**

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Copyright and Trademarks