AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Simple Depth Image

1 Overview

1.1 Location

`$<AMDAPPSDKSamplesInstallPath>\samples\opencl\cl\2.0`

1.2 How to Run

See the *Getting Started* guide for how to build samples. You first must compile the sample.

Use the command line to change to the directory where the executable is located. The pre-compiled sample executable is at `$<AMDAPPSDKSamplesInstallPath>\samples\opencl\bin\x86\` for 32-bit builds, and `$<AMDAPPSDKSamplesInstallPath>\samples\opencl\bin\x86_64\` for 64-bit builds. Ensure that the OpenCL 2.0 environment is installed.

Type the following command(s).

1. `SimpleDepthImage`
   This command runs the program with the default options.
2. `SimpleDepthImage -h`
   This command prints the help file.

1.3 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 meanings.
	`--device [cpu|gpu]`	Devices on which the OpenCL kernel 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-related statistics.
`-v`	`--version`	AMD APP SDK version string.
	`--load [filename]`	Load binary SPIR image and execute on device. This option is mandatory for this sample .
	`--flags [filename]`	Specify the filename containing the compiler flags for building the kernel.
`-i`	`--iterations`	Number of iterations.
`-p`	`--platformId`	Select the platformId to be used[0 to N-1 where N is number platform s available].
`-d`	`--deviceId`	Select deviceId to be used[0 to N-1 where N is number devices available].
`-z`	`--threshold`	Threshold for depth image binarization.

2 Introduction

This sample demonstrates the basic usage of depth image APIs.

Depth images are used in image rendering either to change the image perspective or to render images in 3D. OpenCL 2.0 provides APIs that allow OpenCL kernels to read and manipulate these images.

As the purpose of this sample is to demonstrate depth API usage, a depth image is interpreted as a 2D floating point image. An image is read and converted to a floating point image. This image is then passed to an OpenCL kernel as a depth image. The kernel, based on a threshold, does binarization of the depth image. This manipulated image is then read back as a floating point image, and after proper conversion, written back as an output image.

This sample must be run in the OpenCL 2.0 environment.

3 Implementation

The host reads an input image, `SimpleDepthImage_Input.bmp`. This image must be a 24-bit RGB bitmap (`.bmp`) image. The sample also takes in a threshold value (it uses a default threshold value if none is provided). The luma (Y) of the input image is converted to be in the range of [0.0, 1.0], and this image is passed as depth image to the OpenCL kernel. The kernel performs binarization on this depth image.

The output of the binarization by OpenCL kernel is written in the output file, `SimpleDepthImage_Output.bmp`. If verification with host code is performed (by using the `-e` option), a host side verification code writes the output in the `SimpleDepthImage_Ref.bmp` file.

4 References

1. The OpenCL Specification (ver 2.0, rev 22) document.
2. The OpenCL C Programming Language (ver 2.0, rev 22) document.

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