AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Simple Convolution

1 Overview

1.1 Location

`$(AMDAPPSDKSAMPLESROOT)\samples\opencl\cl\app`

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 `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86\` for 32-bit builds, and `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `SimpleConvolution`
   Performs convolution of a 64x64 image with a blur mask of 3x3.
2. `SimpleConvolution -h`
   This 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 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).
`-x`	`--width`	Width of the input matrix.
`-y`	`--height`	Height of the input matrix.
`-m`	`--masksize`	Width of the mask matrix.
`-i`	`--iterations`	Number of iterations for kernel execution.

2 Introduction

Convolution filetering is widely used in image processing applications such as blur, smooth effects, or edge detection. This sample, shows naïve convolution using Opencl.

3 Implementation Details

The overlap between two functions can be quantized using convolution. In image processing, if a small *mask* matrix (say 3x3) can represent an edge, and this is convolved with the image, the resultant image shows all the edges detected.

A convolution filter is just a scalar product of the filer weights with the input pixels within a window surrounding each of the output pixels.

Equation 1 $$(s * k)(i, j) = \sum_m \sum_n s(i-n, j-m)k(n,m)$$

where k is a matrix of size n x m.

A more detailed explanation of convolution can be found at [1]. It is also a heavily data parallel algorithm because the output at a pixel just depends on the input pixels surrounding it.

4 References

1. http://en.wikipedia.org/wiki/Convolution

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