AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Advanced Convolution

1 Overview

1.1 Location

`$<AMDAPPSDKSamplesInstallPath>\samples\opencl\cl\1.x`

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.

Type the following command(s).

1. `AdvancedConvolution`
   Performs convolution of a 64x64 image with a blur mask of 3x3.
2. `AdvancedConvolution -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).
`-v`	`--version`	AMD APP SDK version string.
`-l`	`--useLDSPass1`	Use LDS for Pass1 for Separable Filter.
`-f`	`--filterType`	Filter Type - 0: Sobel Filter, 1: Box Filter 2: Gaussian Filter.

Short Form	Long Form	Description
-m	--Filter Size	Dimension of Convolution Filter (3 or 5).
-i	--iterations	Number of iterations for kernel execution.

2 Introduction

This sample demonstrates the optimal implementation of separable and non-separable filters. The sample implements 3X3 and 5X5 Sobel, Box, and Gaussian Filters.

A 2D filter is separable if and only if, viewed as a matrix, it is of rank 1. And hence a 2D separable can be written as product of two simple filters. Typically a 2-dimensional convolution operation is separated into 2 1-dimensional filters.

Computational Advantage of Separable Convolution:

Filtering an MXN image with a PXQ filter kernel requires roughly MNPQ multiplies and adds. If the kernel is separable, it can be filtered in two steps. The first step requires about MNP multiples and adds. The second step requires about MNQ multiplies and adds, for a total of MN(P+Q).

The computational advantage of separable convolution versus non-separable convolution is therefore:

```
PQ / (P+Q)
```

So, the larger the dimensions of filter, the faster separable convolution will be.

3 Implementation Details

The sample includes two kernels:

- **Non-separable Filter [one-pass with and without LDS]:** Non-separable filter is a one-pass convolution operation. This kernel implements 2D convolution with and without using LDS. When using LDS, it first copies the input data from global memory to local memory and then performs the convolution operation on local memory.
- **Separable Filter [using LDS]:** The one-pass implementation of separable filter uses one of the advanced features of GPU (and OpenCL): Local Memory, as it is defined by OpenCL specification, is a block of memory available for sharing all work-items belonging to the same OpenCL work-group. This local memory is used to store the result of first pass, which is later used as an input for the second pass.
  The Separable filter kernel implements both the passes of 2D separable convolution in a single kernel using LDS for intermediate copies between the passes. Here again, there are two approaches:
  - **Approach 1:** During Pass 1, kernel reads the input pixels from global memory and performs row-wise filtering. Output of pass 1 is stored in a local memory.. During pass 2, it performs column-wise filtering on output of pass 1 (stored in local memory) and the final output is stored to global memory.
  - **Approach 2:** This approach is similar to Approach 1, with the only difference being that, before pass 1 starts, the kernel copies the input pixels from global to local memory.

4 References

1. http://en.wikipedia.org/wiki/Convolution
2. http://en.wikipedia.org/wiki/Separable_filter
3. http://developer.amd.com/resources/documentation-articles/articles-whitepapers/tiled-convolution-fast-image-filtering/

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**