

One-Dimensional Haar Wavelet Transform

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. DwtHaar1D
Runs with default options; x = 262144.
2. DwtHaar1D -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 <code>cpu</code> or <code>gpu</code> .
-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 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	--signalLength	Length of the signal.
-i	--iterations	Number of iterations for kernel execution.

2 Implementation Details

For a basic one-dimensional Haar Wavelet transform, we make the following assumption: we are given a one-dimensional “image” with a resolution of four pixels, having values

$\{9, 7, 3, 5\}$

We can represent this image in the *Haar basis* by computing a wavelet transform. To do this, we first average the pixels pairwise to get the new, lower-resolution image with pixel values

$\{8, 4\}$

Clearly, some information has been lost in this averaging process. To recover the original four pixel values from the two averaged values, we must store some *detail coefficients*, which capture the missing information. In this example, we choose 1 for the first detail coefficient, since the average we computed is 1 less than 9, and 1 more than 7. This single number lets us recover the first two pixels of our original four-pixel image. Similarly, the second detail coefficient is -1, since $4 + (-1) = 3$ and $4 - (-1) = 5$. Thus, we have decomposed the original image into a lower resolution (two-pixel) version and a pair of detail coefficients. Repeating this process recursively on the averages gives the full decomposition.

Resolution	Averages	Detail Coefficients
4	{9, 7, 3, 5}	
2	{8, 4}	{1, -1}
1	{6}	{2}

Finally, we define the *wavelet transform* (also called the *wavelet decomposition*) of the original four-pixel image to be the single coefficient representing the overall average of the original image, followed by the detail coefficients in order of increasing resolution. Thus, for the one-dimensional Haar basis, the wavelet transform of our original four-pixel image is given by:

$\{6, 2, 1, -1\}$

3 Normalized Decomposition Method

See section 2.2 of reference [1] for more details of the normalized decomposition algorithm. The basic steps of this method are:

1. Calculate the levels (resolutions) from the signal length.
2. Calculate the averages and coefficients from the input signal up to nine levels.
3. If levels > 9, calculate the averages and coefficients from the previous averages.
4. Make the decomposed output signal by arranging the overall average of the original signal, followed by the detail coefficients in order of increasing level (resolution).

4 Recommended Input Option Settings

For best performance, enter the following in the command line: `-x 8388608 -i 10 -q -t`

5 References

1. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. **Wavelets for computer graphics: A primer, part 1.** *IEEE Computer Graphics and Applications*, 15(3):76-84, May 1995.
2. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. **Wavelets for computer graphics: A primer, part 2.** *IEEE Computer Graphics and Applications*, 15(4):75-85, July 1995.

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

© 2011 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, Radeon, FireStream, and combinations thereof are trademarks of Advanced Micro Devices, Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos. Other names are for informational purposes only and may be trademarks of their respective owners.