SAMPLE

OpenVideo Decode Render

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\OVDecodeRender` folder into the `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\cl\app\OVDecodeRender` 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 tab, and select *add->existing project*.
3. Browse to the existing project (`OVDecodeRender.vcproj` in Visual Studio 2008, or `OVDecodeRender.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 OVDecodeRender 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 OVDecodeRender to the `SUB_DIRS` list.

1.2 How to Build

Compile the sample. To do this in Windows:

1. With Visual Studio: Open `OVDecodeRender.sln` with Visual Studio 2008 Professional Edition, or open `OVDecodeRenderVS10.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`.

   Ensure `AMDAPPSDKROOT` and `AMDAPPSDKSAMPLESROOT` are set to appropriate location.

1.3 How to Run

Unzip `OVDecodeDataH264.rar` and `OVDecodeDataMPEG.rar` before running this sample.

1. `OVDecodeRender –h`
   Shows all command options and their respective meaning.

2. OVDecodeRender
   Runs on the GPU with `--codec h264`.

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).
	`--codec`	Select the codec type to be decoded: `h264` (H.264 format) or `mpeg` (MPEG2VLD format). Default is `h264`.

2 Implementation Details

The sample uses the AMD Open Video API to decode an input video stream, then run a post-processing kernel which is displayed through a GL-renderer using GL-interop extension.

The Open Video API uses UVD hardware on AMD GPUs for decoding an input stream. The decoded data is written to a specified OpenCL buffer. The horizontal-pitch is removed using an OpenCL kernel; also, the format conversion from YUV (NV12 format) to RGB is done on a device using an OpenCL kernel.

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.

The post-processing kernel is a recursive Gaussian blur which writes the output to an image (2D texture) and which is interop'ed with GL to display the result through a GL renderer.

Note:

1. By default the post-processing is off. Press the space bar to toggle between normal video stream and blurred output.
2. Only devices with UVD3 capability are able to decode MPEG-2 AND H.264. UVD2-capable devices are only able to decode H.264. To check the UVD capability of your device, go to http://en.wikipedia.org/wiki/Unified_Video_Decoder.