Video Motion Estimation Tutorial

Sample User's Guide

OpenCL™™ Code Builder - Samples

Contents

Legal Information

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to:

http://www.intel.com/design/literature.htm.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to:
http://www.intel.com/products/processor_number/.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Intel, Intel logo, Intel Core, VTune, Xeon are trademarks of Intel Corporation in the U.S. and other countries.

* Other names and brands may be claimed as the property of others.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission from Khronos.

Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

About the Motion Estimation Tutorial

The Video Motion Estimation (VME) tutorial provides step-by-step guidelines on the using Intel's motion estimation extension for OpenCL™ standard. The motion estimation extension includes a set of host-callable functions for frame-based Video Motion Estimation.

The motion estimation extension depends on the OpenCL 1.2 notion of the built-in kernels and on the Intel's "accelerator" extension, which provides an abstraction for the specific hardware-accelerated capabilities.

Notice that to simplify host logic, the tutorial accepts raw YUV files in YV12 only. Refer to the http://www.fourcc.org/yuv.php#YV12 for details on the YV12 format.

Introduction

Motion estimation is the process of determining the motion vectors that describe the transformation from one 2D image to another, usually from adjacent frames in a video sequence. The motion estimation functions, considered in this article, accept full-frame single-channel (luma) images as input, perform a motion search operation, and return a motion vector field as output.

The introduced VME functionality exposes part of the hardware acceleration pipeline for video acceleration. Intel's VME extension provides low-level functionality, currently restricted to the single-channel (luma) input images and block matching methods, so motion vectors are computed for rectangular pixel blocks.

For more details on the VME and associated extensions APIs, refer to the technical article on the VME at http://software.intel.com/en-us/articles/intro-to-motion-estimation-extension-for-opencl.

Motivation

Motion vectors are key element in the video compression algorithms. Motion vectors are also useful for several applications. For example, when generating "slow motion effects", motion vectors can provide the basis to generate intermediate frames for frame rate (up)conversion. Another example is increasing the original frame rate of the digitized film (24 fps) to match the TV rate.

Motion vectors are also useful for image stabilization: the motion vectors in the entire frame can be averaged to produce a "global" motion vector, which can serve as an approximation to a real video camera motion.

Accelerator and Motion Estimation Extensions

The motion estimation extension consists of the new OpenCL built-in kernel (see OpenCL 1.2 specification section 5.6.1) which performs motion estimation, as well as motion estimation accelerator object, which represent the state of the underlying acceleration engine. The kernel is queued for execution from the host using the standard ND-range mechanism.

Both `cl_intel_accelerator` and `cl_intel_motion_estimation` extensions should be listed in the `CL_DEVICE_EXTENSIONS` string (see OpenCL 1.2 specification section Table 4.3) for the Intel HD Graphics device. If not, update the Intel® Graphics Driver.

Hardware Considerations

The goal of the sample is to demonstrate hardware-accelerated VME functionality that Intel HD Graphics device offers. Thus, to run the sample, a 3rd Generation Intel® Core™ processor (and higher) or an Intel® Atom™ processor with Intel HD Graphics (formerly codenamed Bay Trail) is required.

Implementation Details

The tutorial utilizes the following APIs:

- `clGetExtensionFunctionAddressForPlatform`
- `clCreateAcceleratorINTEL`
- `clReleaseAcceleratorINTEL`
- `clEnqueueNDRangeKernel`
- `clCreateProgramWithBuiltInKernels`

Example Results

The pictures below show two frames (reference and source) and computed motion vectors overlaid on the second frame. Specifically, the vectors are rendered as the strokes of the appropriate magnitude. So they point to the new (actually best-matched) pixel block positions.

Notice the radial pattern of the motion vectors, which is due to the nature of the transformation between frames (zoom in addition to the camera movement).

Refer to the technical article for some performance considerations related to VME at http://software.intel.com/en-us/articles/intro-to-motion-estimation-extension-for-opencl.

Controlling the Sample

The sample executable is a console application. Notice that sample-specific parameters have default values, so you can run the sample binary without specifying any command-line options.

The following command-line arguments are available:

Option	Description
`-h, --help`	Show this help text and exit.
`--input <string>`	Input video sequence filename - YUV file (YV12 format). Default input file is `1920x1080_5frames.yuv`
`--output <string>`	Output video sequence with overlaid motion vectors filename - YUV file (YV12 format). Default name for the output file is `output.yuv`.
`--nobmp`	Disable output frames to the sequence of BMP files in addition to the YUV file.

	By default the output is on.
`--width <int>`	Set frame width for the input file. Default value is 1920.
`--height <int>`	Set frame height for the input file. Default 1080

By default sample output sequence of bitmap files that duplicates the content of the output YUV stream.

References

1. Motion Estimation Extension for OpenCL™ specification at http://www.khronos.org/registry/cl/extensions/intel/cl_intel_motion_estimation.txt
2. Accelerator Extension for OpenCL™ specification at http://www.khronos.org/registry/cl/extensions/intel/cl_intel_accelerator.txt
3. Intro to Motion Estimation Extension for OpenCL™ at http://software.intel.com/en-us/articles/intro-to-motion-estimation-extension-for-opencl