

# Advanced Video Motion Estimation Tutorial

Sample User's Guide

---

*Intel® SDK for OpenCL™ Applications - Samples*

# Contents

Contents .....	2
Legal Information .....	3
About the Advanced Motion Estimation Tutorial.....	4
Introduction .....	4
Motivation .....	4
Accelerator and Motion Estimation Extensions .....	4
Hardware Considerations .....	5
Implementation Details .....	5
Example Results.....	5
Controlling the Sample.....	6
References.....	7

## 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/](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.

Copyright © 2014 Intel Corporation. All rights reserved.

### 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 Advanced Motion Estimation Tutorial

---

The Advanced Video Motion Estimation (VME) tutorial provides step-by-step guidelines on using the Intel's motion estimation extensions for the OpenCL™ standard. The advanced 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, on basic [cl\\_intel\\_motion\\_estimation](#) extension, and on the Intel's "accelerator" extension, which provides an abstraction for the specific hardware-accelerated capabilities.

Notice that to simplify the host logic, the tutorial code example 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. In contrast to basic [cl\\_intel\\_motion\\_estimation](#) extension, advanced extension [cl\\_intel\\_advanced\\_motion\\_estimation](#) provides additional control over motion vectors estimation process. Multiple kernel input arguments give possibility of fine VME algorithm tuning to reach the best results depending on input video stream characteristics. The new extension additionally provides skip-check functionality required by novel video coding algorithms.

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

## Motivation

---

Motion vectors are the 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.

Skip-check functionality helps to determine Macro Blocks (MB) which can be omitted in encoding process because of insignificant change from frame to frame. It results in significant computation and bandwidth reduction.

## Accelerator and Motion Estimation Extensions

---

The [Advanced 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 the [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_advanced_motion_estimation` extensions should be listed in the `CL_DEVICE_EXTENSIONS` string (see [OpenCL 1.2 specification](#) section Table 4.3) for the Intel Processor Graphics device. If not, update the Intel® Graphics Driver via the [Intel® Driver Update Utility](#).

## 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 3<sup>rd</sup> Generation Intel® Core™ processor (and higher) or an Intel® Atom™ processor with the Intel Processor Graphics 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 the Advanced VME at <http://software.intel.com/en-us/articles/intro-to-advanced-motion-estimation-extension-for-opencl>.

## Controlling the Sample

The sample executable is a console application. Use the following sequence of commands to control the sample:

```
MotionEstimationAdvanced --input <string> --output <string> [-width <int>] [-height <int>] --block<string> --penalty<string> --precision <string> --predictors<int> [-h]
```

Notice that the 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 value is 1080.
--block <string>	Set VME macro block type. Can be 4x4, 8x8 and 16x16. Default value is 8x8.
--penalty <string>	Set cost penalty type depending on movement characteristics. Can be none, low, normal and high. Default value is high.
--precision <string>	Set cost precision in pixels. Can be quarter, half, full and double. Default value is quarter.
--predictors <int>	Set number of predictor motion vectors. Valid values are in range [0,8]. Default value is 8.

By default, the 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](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](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-openc>
4. Advanced Motion Estimation Extension for OpenCL™ specification at [https://www.khronos.org/registry/cl/extensions/intel/cl\\_intel\\_advanced\\_motion\\_estimation.txt](https://www.khronos.org/registry/cl/extensions/intel/cl_intel_advanced_motion_estimation.txt)
5. Intro to Advanced Motion Estimation Extension for OpenCL™ at <http://software.intel.com/en-us/articles/intro-to-advanced-motion-estimation-extension-for-openc>