11/06/2013

ITK v4 GPU Acceleration: OpenCL

Michael Brehler

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

OpenCL: Basics

*"**Open Computing Language** (**OpenCL**) is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, DSPs and other processors."*

OpenCL application:

- runs on a host (CPU)
- submits commands from the host to execute computations on the processing elements within a device (GPU or CPU)
- task and data parallelism are coordinated via command queues
- Memory hierarchy defined regardless the actual memory architecture for the device

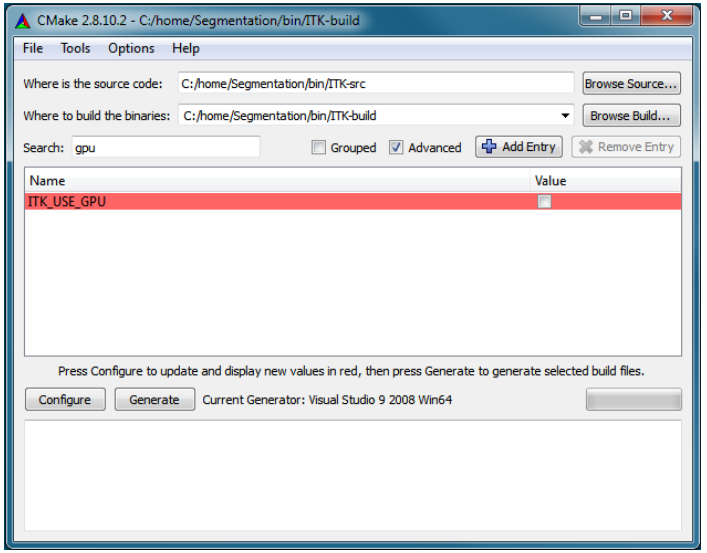
OpenCL is an open standard maintained by the non-profit technology consortium Khronos Group.

OpenCL: ITK v4 / Installation

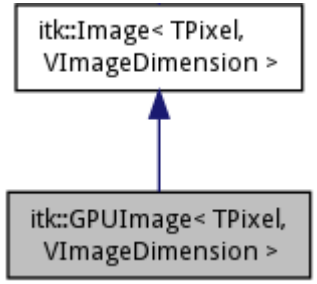
"Although many ITK image filters can benefit from the GPU, there has been no GPU support in ITK as of today."

- ITK support since v4.1 (not working in v4.3.2)
- new data structure, framework and basic operations

- **Installation:**
  - Compatibility list on http://www.khronos.org
  - OpenCL / drivers:
    - NVIDIA CUDA Toolkit
    - AMD APP SKD
  - ITK Cmake flag: "ITK_USE_GPU"



OpenCL: ITK v4



manages GPU and CPU data transparently

GPU memory manager (serves as a base class for GPU data container)

manages GPU contexts and command queues

manages GPU programs and kernels, and execute kernels

OpenCL with ITK: Example 1/3

```
typedef itk::GPUImage<float, 2> GPUImage1f;

//
// Create GPU images as normal itk image
//
GPUImage1f::Pointer srcA, srcB, dest;
srcA = GPUImage1f::New();
srcB = GPUImage1f::New();
dest = GPUImage1f::New();

//
// Initialize GPU images as you normally do for regular itk images
//
srcA->FillBuffer(1.0f);
srcB->FillBuffer(2.0f);
dest->FillBuffer(3.0f);
```

OpenCL with ITK: Example 2/3

```
//
// Create GPU program object
//
GPUKernelManager::Pointer kernelManager = GPUKernelManager::New();
```

```
//
// Load OpenCL source code and compile
//
kernelManager->LoadProgramFromFile("ImageOps.cl");
```

```
//
// Create kernel
//
int kernel_add = kernelManager->CreateKernel("ImageAdd");
```

OpenCL with ITK: Example 3/3

```
//
// Set parameters
//
unsigned int nElem = 65536;
kernelManager->SetKernelArgWithImage(kernel_add, 0, srcA->GetGPUDataManager());
kernelManager->SetKernelArgWithImage(kernel_add, 1, srcB->GetGPUDataManager());
kernelManager->SetKernelArgWithImage(kernel_add, 2, dest->GetGPUDataManager());
kernelManager->SetKernelArg(kernel_add, 3, sizeof(unsigned int), &nElem);
```

```
//
// Launch Kernel
//
kernelManager->LaunchKernel2D(kernel_add, 16, 16, 16, 16);
```

ITK v4.x: GPU filter

- **ITK v4.x:**
  - GPUMeanImageFilter
  - GPUBinaryThresholdImageFilter
  - GPUGradientAnisotropicDiffusionImageFilter
  - GPUDemonsRegistrationFilter

- **ITK v4.5:**
  - **Interpolators:** Nearest Neighbor, Linear, BSpline
  - **Transforms:** Identity, Translation, Euler 2D/3D, Similarity 2D/3D, Affine, Bspline, Composite
  - www.itk.org/Wiki/ITK_Release/GPU_Acceleration/Wish_List

- Get further information on **OpenCL** and **OpenCL with MITK**:
  - OpenCL-Book
  - Eric
  - Jan

- Get further information on **OpenCL with ITK v4.x**:
  - from me
  - and (very rudimentary) here: http://www.itk.org/Wiki/ITK/Release_4/GPU_Acceleration