

## Beta testing of BROCCOLI with Matlab

1. BROCCOLI currently only supports 64-bit operating systems (so far tested with Windows 7 and Linux (CentOS 6.4)). It is recommended that you have at least 8 GB of CPU memory and 2 GB of GPU memory.
2. BROCCOLI is written in OpenCL (Open Computing Language), OpenCL drivers therefore need to be installed for your hardware. If you have a Nvidia GPU and a Nvidia driver, you probably do not have to install any additional driver, unless you also want to test BROCCOLI with a CPU. Note that you only need a driver and not the SDK.

Intel: <http://software.intel.com/en-us/vcs/source/tools/opencl-sdk>

AMD: <http://developer.amd.com/tools-and-sdks/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/downloads/>

Nvidia: <http://www.nvidia.com/download/index.aspx>

3. Download files from <https://github.com/wanderine/BROCCOLI/>. In Linux you can run “git clone <https://github.com/BROCCOLI.git>” in a folder where you want the files. For Windows, see windows.github.com
4. To be able to load nifti files into Matlab, download the nifti Matlab package from <http://research.baycrest.org/~jimmy/NIfTI/>. Open “test\_T1\_MNI\_registration.m” in BROCCOLI/beta\_testing and change the addpath command in the beginning to the folder where you saved the Matlab nifti files.
5. Start Matlab and go to the folder BROCCOLI/beta\_testing
6. Test if OpenCL works, by running GetOpenCLInfo in Matlab. My output looks like this

```
>> GetOpenCLInfo
Device info
```

```
-----
Platform number: 0
```

```
-----
Platform vendor: Intel(R) Corporation
Platform name: Intel(R) OpenCL
Platform extensions: cl_khr_fp64 cl_khr_icd cl_khr_global_int32_base_atomics
cl_khr_global_int32_extended_atomics cl_khr_local_int32_base_atomics
cl_khr_local_int32_extended_atomics cl_khr_byte_addressable_store cl_intel_printf
cl_ext_device_fission cl_intel_exec_by_local_thread
Platform profile: FULL_PROFILE
-----
```

```
-----
Device number: 0
```

-----  
Device vendor: Intel(R) Corporation  
Device name: Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz  
Hardware version: OpenCL 1.2 (Build 67279)  
Software version: 1.2  
OpenCL C version: OpenCL C 1.2  
Device extensions: cl\_khr\_fp64 cl\_khr\_icd cl\_khr\_global\_int32\_base\_atomics  
cl\_khr\_global\_int32\_extended\_atomics cl\_khr\_local\_int32\_base\_atomics  
cl\_khr\_local\_int32\_extended\_atomics cl\_khr\_byte\_addressable\_store cl\_intel\_printf  
cl\_ext\_device\_fission cl\_intel\_exec\_by\_local\_thread  
Global memory size in MB: 15911  
Global memory cache size in KB: 256  
Local memory size in KB: 32  
Constant memory size in KB: 128  
Parallel compute units: 8  
Clock frequency in MHz: 3500  
Max number of threads per block: 1024  
Max number of threads in each dimension: 1024 1024 1024

-----  
Platform number: 1  
-----

Platform vendor: Advanced Micro Devices, Inc.  
Platform name: AMD Accelerated Parallel Processing  
Platform extensions: cl\_khr\_icd cl\_amd\_event\_callback cl\_amd\_offline\_devices  
Platform profile: FULL\_PROFILE  
-----

-----  
Device number: 0  
-----

Device vendor: GenuineIntel  
Device name: Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz  
Hardware version: OpenCL 1.2 AMD-APP (1214.3)  
Software version: 1214.3 (sse2,avx)  
OpenCL C version: OpenCL C 1.2  
Device extensions: cl\_khr\_fp64 cl\_amd\_fp64 cl\_khr\_global\_int32\_base\_atomics  
cl\_khr\_global\_int32\_extended\_atomics cl\_khr\_local\_int32\_base\_atomics  
cl\_khr\_local\_int32\_extended\_atomics cl\_khr\_int64\_base\_atomics  
cl\_khr\_int64\_extended\_atomics cl\_khr\_3d\_image\_writes cl\_khr\_byte\_addressable\_store  
cl\_khr\_gl\_sharing cl\_ext\_device\_fission cl\_amd\_device\_attribute\_query cl\_amd\_vec3  
cl\_amd\_printf cl\_amd\_media\_ops cl\_amd\_media\_ops2 cl\_amd\_popcnt  
Global memory size in MB: 15911  
Global memory cache size in KB: 32  
Local memory size in KB: 32  
Constant memory size in KB: 64  
Parallel compute units: 8  
Clock frequency in MHz: 1600  
Max number of threads per block: 1024  
Max number of threads in each dimension: 1024 1024 1024

-----  
Platform number: 2  
-----

Platform vendor: NVIDIA Corporation  
Platform name: NVIDIA CUDA  
Platform extensions: cl\_khr\_byte\_addressable\_store cl\_khr\_icd cl\_khr\_gl\_sharing  
cl\_nv\_compiler\_options cl\_nv\_device\_attribute\_query cl\_nv\_pragma\_unroll  
Platform profile: FULL\_PROFILE  
-----

-----  
Device number: 0  
-----

Device vendor: NVIDIA Corporation  
Device name: GeForce GTX 680  
Hardware version: OpenCL 1.1 CUDA  
Software version: 310.44  
OpenCL C version: OpenCL C 1.1  
Device extensions: cl\_khr\_byte\_addressable\_store cl\_khr\_icd cl\_khr\_gl\_sharing  
cl\_nv\_compiler\_options cl\_nv\_device\_attribute\_query  
cl\_nv\_pragma\_unroll cl\_khr\_global\_int32\_base\_atomics  
cl\_khr\_global\_int32\_extended\_atomics cl\_khr\_local\_int32\_base\_atomics  
cl\_khr\_local\_int32\_extended\_atomics cl\_khr\_fp64  
Global memory size in MB: 4095  
Global memory cache size in KB: 128  
Local memory size in KB: 48  
Constant memory size in KB: 64  
Parallel compute units: 8  
Clock frequency in MHz: 1058  
Max number of threads per block: 1024  
Max number of threads in each dimension: 1024 1024 64

-----  
Device number: 1  
-----

Device vendor: NVIDIA Corporation  
Device name: GeForce GTX 680  
Hardware version: OpenCL 1.1 CUDA  
Software version: 310.44  
OpenCL C version: OpenCL C 1.1  
Device extensions: cl\_khr\_byte\_addressable\_store cl\_khr\_icd cl\_khr\_gl\_sharing  
cl\_nv\_compiler\_options cl\_nv\_device\_attribute\_query  
cl\_nv\_pragma\_unroll cl\_khr\_global\_int32\_base\_atomics  
cl\_khr\_global\_int32\_extended\_atomics cl\_khr\_local\_int32\_base\_atomics  
cl\_khr\_local\_int32\_extended\_atomics cl\_khr\_fp64  
Global memory size in MB: 4095  
Global memory cache size in KB: 128  
Local memory size in KB: 48  
Constant memory size in KB: 64  
Parallel compute units: 8  
Clock frequency in MHz: 1058  
Max number of threads per block: 1024  
Max number of threads in each dimension: 1024 1024 64

Three OpenCL platforms are installed for this computer, Intel, AMD and Nvidia. Note that it is possible to run the Intel CPU both on the Intel platform and on the AMD platform. For some reason, the AMD platform seems to work better. The platform and the device to use can easily be selected with the variables “opengl\_platform” and “opengl\_device” in the Matlab scripts.

7. Run the Matlab-script “test\_T1\_MNI\_registration.m” . If you have a small amount of memory, try setting the variable “voxel\_size” to 2.

8. The first time you run the script, the OpenCL driver will (hopefully) compile the OpenCL kernel code in “broccoli\_lib\_kernel.cpp” and save a binary version as a .bin file (e.g. broccoli\_lib\_kernel\_Nvidia\_GeforceGTX680.bin). The processing time will therefore be (much) longer for the first run (it seems to depend on the platform). The processing time should be much shorter the second time, as BROCCOLI then can read the precompiled binary file. If you run BROCCOLI for several platforms or devices, one binary file will be created for each platform and device.
9. Compare the results to the png-files in the beta\_testing folder.
10. Try to run the script for different studies and different subjects. The BROCCOLI github repository contains 3 studies with 3 subjects each, more data can be downloaded from [http://fcon\\_1000.projects.nitrc.org/fcpClassic/FcpTable.html](http://fcon_1000.projects.nitrc.org/fcpClassic/FcpTable.html)