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 necessarily the SDK.

   Intel: http://software.intel.com/en-us/vcsource/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/wanderine/BROCCOLI.git" in a folder where you want the files. For Windows, see windows.github.com

4. Download the FSL software to obtain the MNI templates (MNI152…), copy them to the folder brain_templates in BROCCOLI.

   http://fsl.fmrib.ox.ac.uk/fsldownloads/

5. 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.

6. Start Matlab and go to the folder BROCCOLI/beta_testing

7. Test if OpenCL works, by running GetOpenCLInfo in Matlab. My output looks like this

>> GetOpenCLInfo

Device info

Platform number: 0

Platform vendor: NVIDIA Corporation
Platform name: NVIDIA CUDA

Platform extentions: cl_khr_byte_addressable_store cl_khr_icd cl_khr_gl_sharing cl_nv_d3d9_sharing cl_nv_d3d10_sharing cl_khr_d3d10_sharing cl_nv_d3d11_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 690
Hardware version: OpenCL 1.1 CUDA
Software version: 320.49
OpenCL C version: OpenCL C 1.1

Device extensions: cl_khr_byte_addressable_store cl_khr_icd cl_khr_gl_sharing cl_nv_d3d9_sharing cl_nv_d3d10_sharing cl_khr_d3d10_sharing cl_nv_d3d11_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: 2048
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: 1019
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 690
Hardware version: OpenCL 1.1 CUDA
Software version: 320.49
OpenCL C version: OpenCL C 1.1

Device extensions: cl_khr_byte_addressable_store cl_khr_icd cl_khr_gl_sharing cl_nv_d3d9_sharing cl_nv_d3d10_sharing cl_khr_d3d10_sharing cl_nv_d3d11_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: 2048
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: 1019
Max number of threads per block: 1024
Max number of threads in each dimension: 1024 1024 64

Platform number: 1

Platform vendor: Intel(R) Corporation
Platform name: Intel(R) OpenCL

Platform extentions: 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 cl_khr_gl_sharing cl_intel_dx9_media_sharing cl_khr_dx9_media_sharing cl_khr_d3d11_sharing

Platform profile: FULL_PROFILE

Device number: 0

Device vendor: Intel(R) Corporation
Device name: Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
Hardware version: OpenCL 1.2 (Build 63463)
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 cl_khr_gl_sharing cl_intel_dx9_media_sharing cl_khr_dx9_media_sharing cl_khr_d3d11_sharing

Global memory size in MB: 16360
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: 3400
Max number of threads per block: 1024
Max number of threads in each dimension: 1024 1024 1024

Platform number: 2

Platform vendor: Advanced Micro Devices, Inc.

Platform name: AMD Accelerated Parallel Processing

Platform extentions: cl_khr_icd cl_amd_event_callback cl_amd_offline_devices cl_khr_d3d10_sharing cl_khr_d3d11_sharing

Platform profile: FULL_PROFILE

Device number: 0

Device vendor: Advanced Micro Devices, Inc.

Device name: Tahiti

Hardware version: OpenCL 1.2 AMD-APP (1124.2)

Software version: 1124.2 (VM)

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_atomic_counters_32 cl_amd_device_attribute_query cl_amd_vec3 cl_amd_printf cl_amd_media_ops cl_amd_media_ops2 cl_amd_popcnt cl_khr_d3d10_sharing cl_amd_c1x_atomics

Global memory size in MB: 2048

Global memory cache size in KB: 16

Local memory size in KB: 32

Constant memory size in KB: 64

Parallel compute units: 32

Clock frequency in MHz: 1000

Max number of threads per block: 256

Max number of threads in each dimension: 256 256 256

Device number: 1

Device vendor: GenuineIntel
Device name: Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
Hardware version: OpenCL 1.2 AMD-APP (1124.2)
Software version: 1124.2 (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 cl_khr_d3d10_sharing

Global memory size in MB: 16360
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: 3411
Max number of threads per block: 1024
Max number of threads in each dimension: 1024 1024 1024

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 “opencl_platform” and “opencl_device” in the Matlab scripts.

8. 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.
9. 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_GeforceGTX690.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.

10. Compare the results to the png-files in the beta_testing folder.
11. 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