

Installing CUDA Volume Rendering into 3D Slicer

Benjamin Grauer

June 2008

Chapter 1

Slicer3 Using CUDA and Ultrasound Installation Manual

This chapter talks about the installation of 3D Slicer, how to embed the CUDA support and finally enabling the ultrasound scanner and minibirds.

There is a precompiled version of Slicer that has full support for CUDA and Cuda Volume Rendering and features a data dimulator for the 3D ultrasound scanner from the **resources** section on the CUDA Webpage:

http://www.slicer.org/slicerWiki/index.php/Slicer3:Volume_Rendering_With_Cuda

1.1 3D Slicer

At present the CUDA volume renderer is located in a special branche for general CUDA development. This branche is located at the following link:

http://www.na-mic.org/svn/Slicer3/branches/igt_cuda_merge

Note that the following section of the guide up to section 1.2 can also be followed at the next URL (Just check out the `igt_cuda_merge` branche):

http://www.slicer.org/slicerWiki/index.php/Slicer3:Build_Instructions

1.1.1 Pre-Install Procedures

First you will need the following tools:

- **Subversion:** <http://svn.tigris.org>
- **CVS:** <http://www.nongnu.org/cvs>
- **tk/tcl:** <http://www.tcl.tk>
- **curl** <http://curl.haxx.se/>

- **A compiler** either GCC or visual studio

1.1.2 Compiling Slicer3

Check out 3D Slicer from the repository with the following command:

```
svn co http://www.na-mic.org/svn/Slicer3/branches/igt_cuda_merge Slicer3
```

Start the default build instructions by executing:

```
./Slicer3/Scripts/getbuildtest.tcl
```

Test 3D Slicer by changing into the build directory and starting Slicer3.

```
cd Slicer3-build./Slicer3
```

1.2 Enabling Volume Rendering with CUDA

This section of the guide explains how to get CUDA support and Volume Rendering in an already compiled version of 3D Slicer.

1.2.1 Downloading and Installing CUDA

Download and install the CUDA toolkit and SDK from nvidia's CUDA web page at the following URL:

```
http://www.nvidia.com/object/cuda_get.html
```

Please follow the instructions on how to install the software packages on your specific operating system.

You might also consider installing the supplied drivers for CUDA graphics cards that are also located on the above mentioned webpage.

1.2.2 Enabling CUDA

Enabling general CUDA support for 3D Slicer has to be done manually. The easiest way is using the gui by starting the *ccmake* on UNIX or the Windows graphical interface of CMake. Both of them are located in *Slicer3 – libs/CMake – build/bin*.

Now enable the flag for CUDA called **CUDA_SUPPORT_ENABLED** and press *configure* or 'c'.

Now an error message saying '*Specify CUDA_INSTALL_PREFIX*' will be displayed stating that the CUDA libraries are missing. Press 'ok'.

To fix this error the CUDA installation prefix for the toolkit and the SDK have to be set correctly. Point each of the values to the corresponding destinations as follows:

- Set **CUDA_BUILD_TYPE** to **Device** (Emulation will run CUDA in device emulation mode, and is intended for heavy debugging only).
- Set the **CUDA_INSTALL_PREFIX** to the top directory of the CUDA toolkit installation. (In Windows this is usually *C:/Cuda*, in UNIX */usr/local/cuda*)

[width=0.40]../Data/UltrasoundGUI

Figure 1.1: Ultrasound GUI Window

- Set the **FOUND_CUT** to the library of the SDK called *common/lib/libcutil.a* in UNIX or *common/lib/libcutil.dll* in Windows.
- Set the **FOUND_CUT_INCLUDE** to the directory in *common/inc* of the SDK.

Hit *configure* again, and the error message should be gone.

1.2.3 Enabling Volume Rendering with CUDA

Enabling volume rendering with CUDA works in the same way as enabling CUDA itself. Just enter the *ccmake environment* and set the **VOLUMERENDERINGCUDAMODULE** to **ON** and hit *configure* and *generate*. Rebuild the entire solution and volume rendering will be supported¹.

1.2.4 Testing

To test if the CUDA volume rendering installation was successful run *slicer* and test it by loading any dicom dataset or by using the tutorial dataset of Slicer3 located at the following URL:

<http://wiki.na-mic.org/Wiki/images/f/f8/Tutorial-with-dicom.zip>

1.3 Ultrasound Scanner and Ascension Minibird Support

This section shows how to enable the Philips Ultrasound Scanner support module and the Ascension Minibird trackers. To enable the ultrasound module you have to enter the *ccmake environment* again. This time you must enable the **ULTRASOUNDMODULE** by setting it to **ON**. Again hit 'c' or *configure*. Now a simulation environment is setup to see if the module is inserted into Slicer after the next solution rebuild. Figure 1.1 shows how the selected ultrasound GUI will look like after this step. There are two sections, the first one is to control the ultrasound data, and the speed of aquisition of data. The second section is to quickly configure and adjust the tracking information.

Since the libraries for the scanner and the minibird are both proprietary we will not go into the detail on how to setup a running environment. It is said however, that if you have the library for both the Ascension Minibird and the Philips Sonos ultrasound scanner reader, the procedure is the same as with the one mentioned to set the CUDA support.

¹Only the Slicer3-real and the VolumeRendering Module have to be rebuilt for this step