RITK Installation Guide for Windows

October 4, 2011

The Range Imaging Toolkit (RITK) is a software framework for the development of range imaging (RI) applications. The toolkit is developed at the Pattern Recognition Lab, University Erlangen-Nuremberg, Germany.

Please send your feedback to this manual to ritk@i5.informatik.uni-erlangen.de

Contents

1 Setting up the Environment

1.1 Applications

Download and **install** the following applications that are required to install the libraries needed for the RITK (32 or 64 bit):

Name	Link	Comment
Visual Studio 2010 (*VS2010*) Express[1]	`http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express`	Download and install Visual 2010 C++ Express
CMake	`http://www.cmake.org`	

1.2 Libraries

Download and **install** the following applications that are required to install the libraries needed for the RITK (32 or 64 bit):

Name	Link	Comment
QT	`http://get.qt.nokia.com/qt/source/qt-everywhere-opensource-src-4.7.4.zip`	
CUDA 4.0	`http://developer.nvidia.com/cuda-toolkit-40`	Download and install only the *Toolkit*
Insight Toolkit (ITK)	`http://itk.org/ITK/resources/software.html`	Source Files Version 3.20 or later
Visualization ToolKit (VTK)	`http://vtk.org/VTK/resources/software.html`	Source Files Version 5.8.0 or later
MS Kinect SDK[2]	`http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/download.aspx`	
OpenNI[2]	`https://github.com/avin2/SensorKinect`	Follow the installation guide here

[1] You will need an emulation program to mount the VS2010 *.iso files. Use e.g. deamon tools

[2] Please note that you have to decide either for installing OpenNI or MS Kinect SDK. In general, both will not work side by side in general.

1.2.1 QT Installation[3]

1. Unpack the archive to a custom destination , let us say <QT_DIR_PATH>
2. Add <QT_DIR_PATH>/bin to the PATH variable
3. Open *VS2010* command prompt go to <QT_DIR_PATH>, type Configure press enter and follow the instructions
4. type nmake (this may take a while)

1.2.2 CUDA Installation

Install the CUDA Toolkit 4.0 with the setup wizard by using its default settings.

1.2.3 ITK Installation

ITK does not provide any setup wizard but has to be compiled. The first step is to unzip the archive into a subfolder in your library folder (e.g. .../libs/InsightToolkit-3.20.0/). In this folder you should now find a file called CMakeLists.txt. This file is now used by CMake to create a *VS2010* Project:

1. Open CMake (cmake-gui)
2. Set the path to the itk root directory (InsightToolkit-3.20.0/)
3. Set the directory where the binaries are build (usually a subfolder of the source folder, as we are using *VS2010* we call this folder VC-10)
4. Click *Configure*. If CMake asks whether to create the specified folder click yes.
5. Now a window will pop up where you have to specify the generator. Select *Visual Studio 2010* and click *Finish*. CMake will now configure the project. This could take a while.
6. After configuring CMake will show you some options (name and value) that depend on the actual project. For compiling ITK make sure that the option

   BUILD_SHARED_LIBS is enabled

   BUILD_TESTING and BUILD_EXAMPLES is disabled (for faster compilation)
7. Again, click *Configure*
8. Click *Generate*. (This will create a *VS2010* Solution .sln file)

Now we are ready to compile ITK.
In the binary folder (e.g. .../libs/InsightToolkit-3.20.0/VC-10) you should find a file ITK.sln, this is the Visual Studio Solution file. A double-click will open the solution. Build the project in both DEBUG and RELEASE configuration. This may take a while.

[3] Taken from http://doc.qt.nokia.com/latest/install-win.html

1.2.4 VTK Installation (V. 5.8.0)

The steps to build VTK are in general the same as for ITK:
Extract files to a subfolder in your library folder, open CMakeLists.txt with CMake (make sure to update the source and binary paths), setting some options, generating a Visual Studio solution file and, finally, building the solution VTK.sln in DEBUG and RELEASE mode.
The following options have to be set:

1. make sure that you enable BUILD_SHARED_LIBS
2. disable BUILD_TESTING, and BUILD_EXAMPLES is disabled and click *Configure* (you may want to enable them but this takes really long to compile)
3. Select *group view*, expand the VTK section
4. enable VTK_USE_PARALLEL and VTK_USE_QT and click *Configure*

CMake will now look for a valid installation of Qt. If a valid installation is found, press one more time Configure and then the Generate. This will create the VTK.sln file in your binary folder (e.g. .../libs/VTK-5.6.1/VC-10).
Building VTK is now the same as for ITK. If no valid installation is found add the the variable QTDIR to your system environment variables (you may want to use an editor like Rapid Environment Editor). The value of this variable has to be the root directory of your Qt installation (e.g. .../libs/<QT_DIR_PATH>). After setting the variable close and re-run CMake.

2 Setting up the RITK

1. Download the sources from http://www5.cs.fau.de/ritk
2. Unzip the files to a custom destination, let us say <RITK_INSTALL_DIR>
3. Open Cmake (cmake-gui) and select

   <RITK_INSTALL_DIR>/SurfaceImaging/RITK/v10/RITK for the source directory

   <RITK_INSTALL_DIR>/SurfaceImaging/RITK/v10/RITK/VC-10 for the build directory

   Click *Configure*

   Set CMAKE_INSTALL_PREFIX to your needs, let us say <RITK_BIN_DIR>.
   (default: C/Program Files/RITK)

   Click *Configure* and then *Generate*
4. Build RITK with *VS2010* (debug and release) in analogy to ITK/VTK
5. Copy all required dll files into the folder <RITK_BIN_DIR>VC-10/bin/debug (release)

   Copy all VTK dll's:
   .../libs/VTK-5.6.1/VC-10/bin/debug (release)

   Copy the ITK dll:
   .../libs/InsightToolkit-3.20.0/VC-10/bin/debug (release)/ITKCommon.dll

Now the RITK is ready to use, the application (debug/release versions) can be found in:

- Debug: <RITK_BIN_DIR>/bin/debug/RITKd.exe
- Release <RITK_BIN_DIR>/bin/release/RITK.exe

If there are any problems with starting RITK, please copy the DLLs for QT, too - in analogy to ITK/VTK.