Convert Matlab Codes into Vaa3d Plugin

Pengyu Hong

Zhihao Zheng

Contents

- Compile Matlab Codes into C Library
- Use Matlab Library in C/C++
- Setting Environment Values

Contents

- Compile Matlab Codes into C Library
- Use Matlab Library in C/C++
- Setting Environment Values

Compile Matlab Codes into C Library

- Matlab Toolboxes
  - Matlab Compiler
  - Matlab Compiler SDK
- Compiler Requirements
  - Linux
    - GNU GCC/G++ 4.7.x
  - Windows
    - Microsoft Windows SDK 7.1
    - Microsoft Visual C++ 201X

Compile Matlab Codes into C Library

- Key Steps
  - Integrate your code into one Matlab function
  - Run commands below in Matlab
    - mbuild –setup
    - mcc –B csharedlib:libfoo foo.m
  - Windows
    - libfoo.h libfoo.c libfoo.lib libfoo.dll …
  - Linux
    - libfoo.h libfoo.c libfoo.so …

Contents

- Compile Matlab Codes into C Library
- Use Matlab Library in C/C++
- Setting Environment Values

Use Matlab Library in C/C++

- Linux
  - #include “libfoo.h”
  - libfooInitialize()
  - mlfFoo(1,&mx_result,mx_para1…)
  - libfooTerminate()

Use Matlab Library in C/C++

- Windows
  - #include “libfoo.h”
  - mclInitializeApplication(null,0)
  - libfooInitialize()
  - mlfFoo(1,&mx_result,mx_para1…)
  - libfooTerminate()
  - mclTerminateApplication()

Use Matlab Library in C/C++

- Convert variable types into mxArray
  - dynamic_space = mxCalloc(num, sizeof(TYPE))
  - Copy your data into dynamic_space
  - mx_var = mxCreateUinitNumericArray(
    ndim,dim_vec,
    MATLAB_CLASS,mxREAL)
  - mxSetData(mx_var, dynamic_space)
  - Output = mxGetLogicals(mx_var)
  - mxDestroyArray(mx_var)

Contents

- Compile Matlab code into C Library
- Use Matlab Library in C/C++
- Setting Environment Values

Setting Environment Values

- Windows – Visual Studio
  - qmake –tp vc test.pro
  - Change to “release” mode
  - C/C++ -> General ->Additional Include Directories
    - Add <matlabroot>\extern\include
    - Add <PATH TO THE FOLDER CONTAINS libfoo.h>
  - Linker -> General -> Additional Library Directories
    - Add <matlabroot>\extern\lib\win64\microsoft
    - Add <PATH TO THE FOLDER CONTAINS libfoo.lib>

Setting Environment Values

- Windows – Visual Studio - Cont'
  - Linker -> Input
    - Add mclmcrrt.lib
    - Add libfoo.lib
  - Build Your Solution
  - Copy libfoo.dll into Vaa3d Root Folder
  - Run!

Setting Environment Values

- Linux
  - Edit test.pro
    - INCLUDEPATH += <matlabroot>\extern\include
    - INCLUDEPATH += <PATH TO THE FOLDER CONTAINS libfoo.h>
    - LIBS += -L<matlabroot>/runtime/glnxa64 –lmwmclmcrrt
    - LIBS += -L<matlabroot>/sys/os/glnxa64 –lstdc++
    - LIBS += <PATH TO THE FOLDER CONTAINS libfoo.lib>
  - qmake
  - make

Setting Environment Values

- Linux – Cont'
  - Edit start_vaa3d.sh
    - export LD_LIBRARY_PATH=`pwd`:
      <matlabroot>/runtime/glnxa64:
      <matlabroot>/bin/glnxa64:
      <matlabroot>/sys/os/glnxa64:
      <matlabroot>/sys/opengl/lib/glnxa64:
      <PATH TO THE FOLDER CONTAINS libfoo.h>
  - Run!

Sample Codes

- func.cpp
  - Initialize MCR
  - Create variables for Matlab
  - Set values to those variables
- vaa3d_trace3D.m
  - Sample Matlab Function