Code reproduction documentation

Author: Ting Liu

Catalogue

1. Software and library requirements

The following software needs to be installed before running Demo_WSWTNN_PnP:

Matlab 2018a
Matconvnet
Visual Studio 2015
CUDA 9.0
CUDNN v7.6.5
Computer system: Windows 10, Intel Core i7-10870H CPU (2.20 GHz), RTX 2060

(Note: For computers with different performance and different versions of matlab, the CUDA version should be different. You need to select the version of CUDA and CUDNN according to your computer.)

2. Download software

2.1. MATLAB

MATLAB 2018a: https://pan.baidu.com/s/1Rl2pv0cGv27skG_02Dtjgw
[key]: z6so

2.2. Matconvnet

Matconvnet： https://www.vlfeat.org/matconvnet/
Place the matconvnet-1.0-beta25 file under MATLAB/R2018a/toobox in the MATLAB 2018a installation location as shown in Fig. 2.1 below:

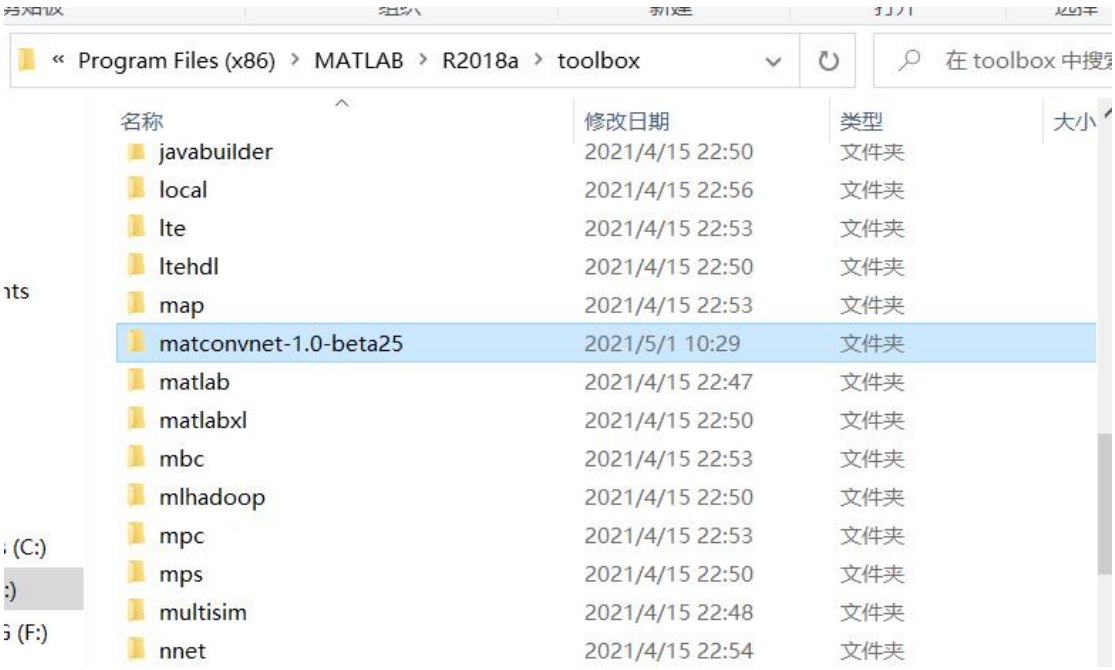


Figure 2.1

2.3. Visual Studio

Visual Studio 2015:
https://pan.baidu.com/s/1bkEexqA8rMiEaTcYkcQdfg
[key]: 5pd5

2.4. CUDA

CUDA 9.0
https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html
The steps for configuring CUDA are as follows:
https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html

2.5. CUDNN

CUDNN v7.6.5: https://developer.nvidia.com/rdp/cudnn-archive
The steps for configuring CUDNN are as follows:
https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html#install-windows

3. Experimental environment construction process

3.1 CPU compilation

(1) Add matconvnet-1.0-beta25 and its sub files to D:\Program Files (x86)\MATLAB\R2018a\toolbox

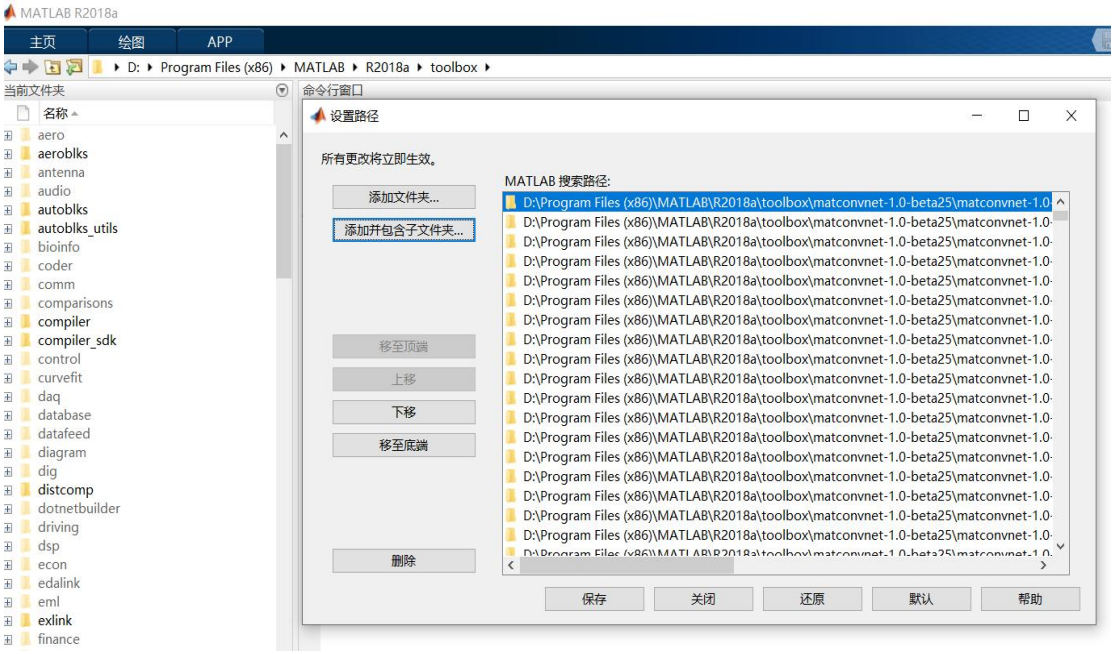


Figure 3.1

(2) Enter mex-setup in the MATLAB command line window to see if mex-setup C++ appears. If no mex-setup C++ appears, Visual Studio 2015 is not successfully installed

```
>> clear all
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2015 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改，现可支持
      包含 2^32-1 个以上元素的 MATLAB 变量。您需要
      更新代码以利用新的 API。
      您可以在以下网址找到更多的相关信息:
      https://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html。

要选择不同的语言，请从以下选项中选择一种命令:
 mex -setup C++
 mex -setup FORTRAN
MEX 配置为使用 'Microsoft Visual C++ 2015' 以进行 C++ 语言编译。
警告: MATLAB C 和 Fortran API 已更改，现可支持
      包含 2^32-1 个以上元素的 MATLAB 变量。您需要
      更新代码以利用新的 API。
      您可以在以下网址找到更多的相关信息:
      https://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html。
MEX 配置为使用 'Microsoft Visual C++ 2015' 以进行 C++ 语言编译。
警告: MATLAB C 和 Fortran API 已更改，现可支持
      包含 2^32-1 个以上元素的 MATLAB 变量。您需要
      更新代码以利用新的 API。
      您可以在以下网址找到更多的相关信息:
      https://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html。
>> vl_compilenn
```

Figure 3.2

My Visual Studio 2015 installation path: C:\Program Files (x86)\VS 2015\professional

（Note: Add the installation directory to the Path of the environment variable → system variable.）

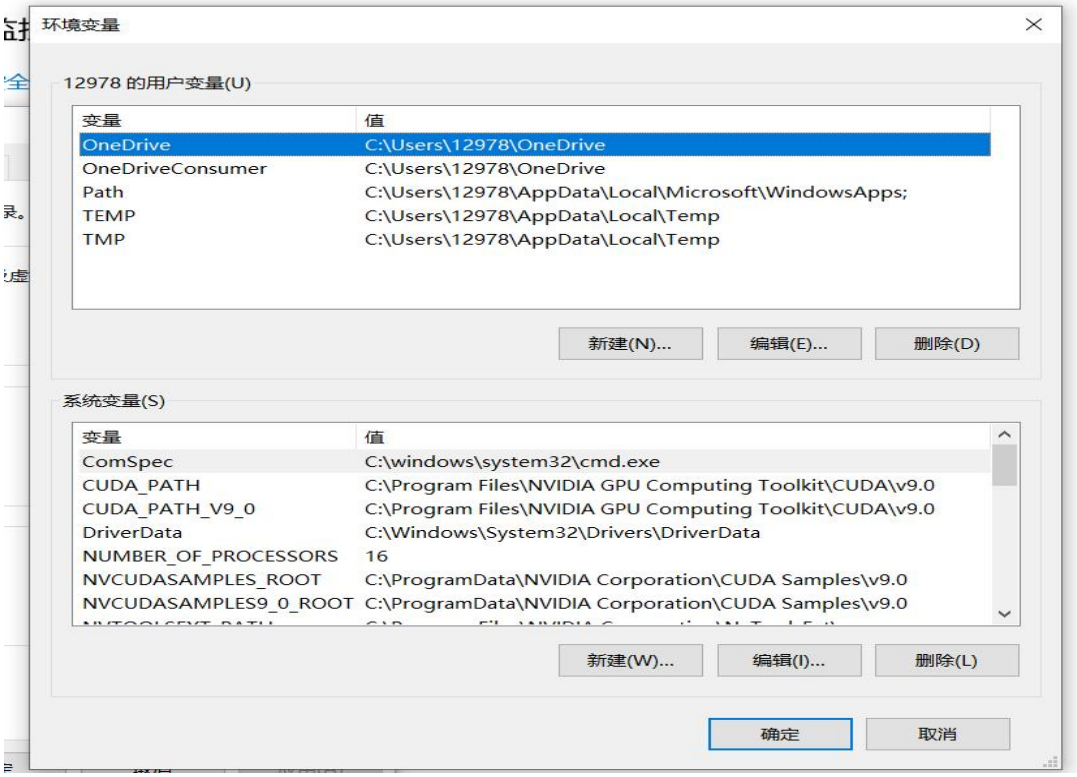


Figure 3.3

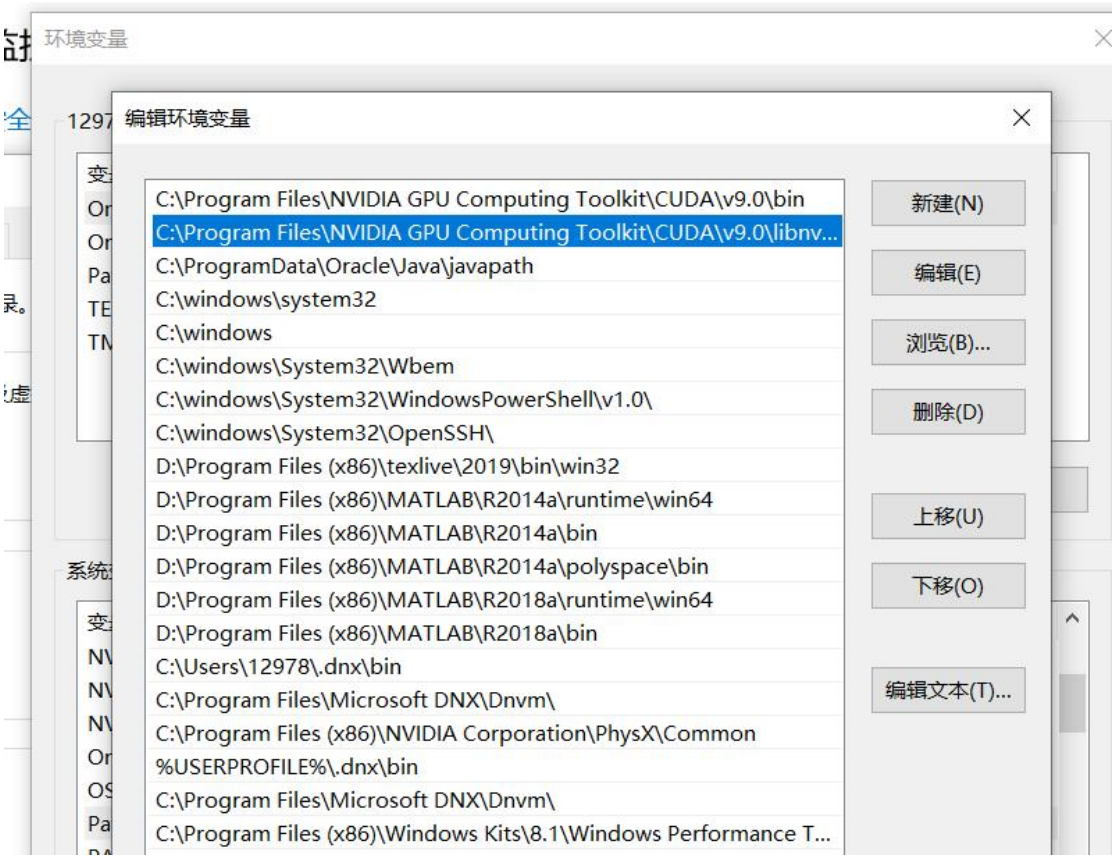


Figure 3.4

(3) Enter vl_compilenn in the command line window of MATLAB， vl_compilenn is successfully compiled.

```
>> vl_compilenn
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015' 编译。
fx MEX 已成功完成。
```

Figure 3.5

(4) Enter test command vl_testnn in the command line window of MATLAB. vl_testnn is successfully compiled.

```
>> vl_testnn
fx
```

```
 正在运行 nnspnorm
  正在设置 nnspnorm[dataType=single, device=cpu]
  设置 nnspnorm[dataType=single, device=cpu] 在 0.0031351 秒内完成
   正在运行 nnspnorm[dataType=single, device=cpu]/basic
   nnspnorm[dataType=single, device=cpu]/basic 在 0.75398 秒内完成
  正在拆解 nnspnorm[dataType=single, device=cpu]
  拆解 nnspnorm[dataType=single, device=cpu] 在 0 秒内完成
 nnspnorm 在 0.75711 秒内完成
----------


result =

  1×1793 TestResult 数组 - 属性:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details

总计:
   1793 Passed, 0 Failed, 0 Incomplete.
   135.0925 秒测试时间。
```

Figure 3.6

Fig 3.6 shows that without using CUDA and CUDNN to drive the GPU, the CPU is used alone to run the program compiled by matconvnet.

3.2 GPU compilation

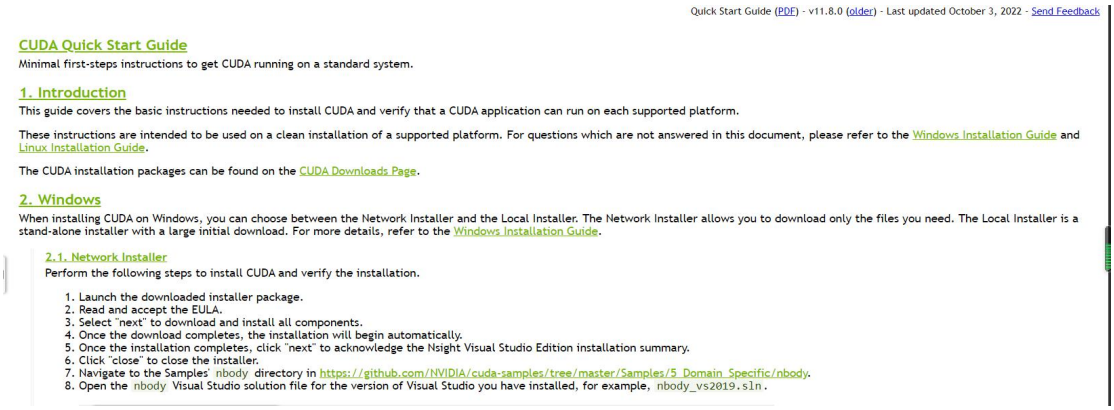
First, look at the computer's CUDA device properties and enter gpuDevice in the MATLAB command line window:

```
>> gpuDevice

ans =

  CUDADevice - 属性:

                      Name: 'NVIDIA GeForce RTX 2060'
                     Index: 1
          ComputeCapability: '7.5'
             SupportsDouble: 1
              DriverVersion: 11.6000
             ToolkitVersion: 9
          MaxThreadsPerBlock: 1024
            MaxShmemPerBlock: 49152
          MaxThreadBlockSize: [1024 1024 64]
                 MaxGridSize: [2.1475e+09 65535 65535]
                   SIMDWidth: 32
                 TotalMemory: 6.4421e+09
             AvailableMemory: 5.2690e+09
       MultiprocessorCount: 30
              ClockRateKHz: 1200000
                 ComputeMode: 'Default'
        GPUOverlapsTransfers: 1
      KernelExecutionTimeout: 1
            CanMapHostMemory: 1
             DeviceSupported: 1
```

Figure 3.7

3.1.1 Install CUDA

Follow the steps on the official website to install CUDA: https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html



Quick Start Guide (PDF) - v11.8.0 (older) - Last updated October 3, 2022 - Send Feedback

CUDA Quick Start Guide

Minimal first-steps instructions to get CUDA running on a standard system.

1. Introduction

This guide covers the basic instructions needed to install CUDA and verify that a CUDA application can run on each supported platform.

These instructions are intended to be used on a clean installation of a supported platform. For questions which are not answered in this document, please refer to the Windows Installation Guide and Linux Installation Guide.

The CUDA installation packages can be found on the CUDA Downloads Page.

2. Windows

When installing CUDA on Windows, you can choose between the Network Installer and the Local Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. For more details, refer to the Windows Installation Guide.

2.1. Network Installer

Perform the following steps to install CUDA and verify the installation.

1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to download and install all components.
4. Once the download completes, the installation will begin automatically.
5. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
6. Click "close" to close the installer.
7. Navigate to the Samples' `nbody` directory in https://github.com/NVIDIA/cuda-samples/tree/master/Samples/5_Domain_Specific/nbody.
8. Open the `nbody` Visual Studio solution file for the version of Visual Studio you have installed, for example, `nbody_vs2019.sln`.

Figure 3.8

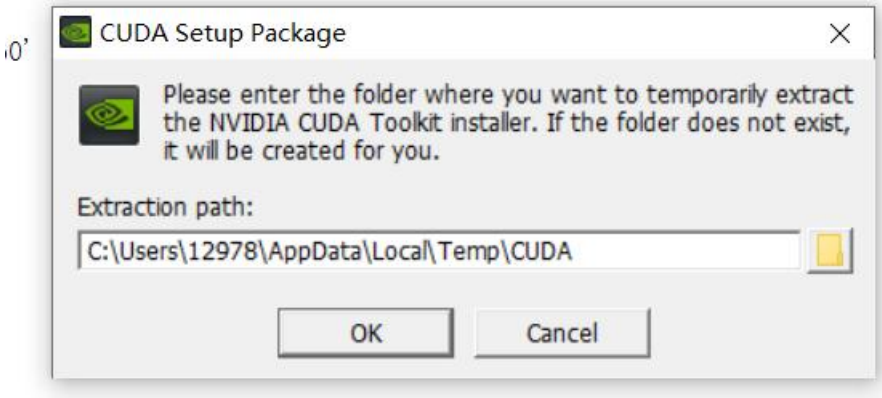


Figure 3.9

Figure 3.10

Figure 3.11

Open the nbody Visual Studio solution file for the version of Visual Studio you have installed. Then, Open the 'Build'menu within Visual Studio and click 'Build Solution'.

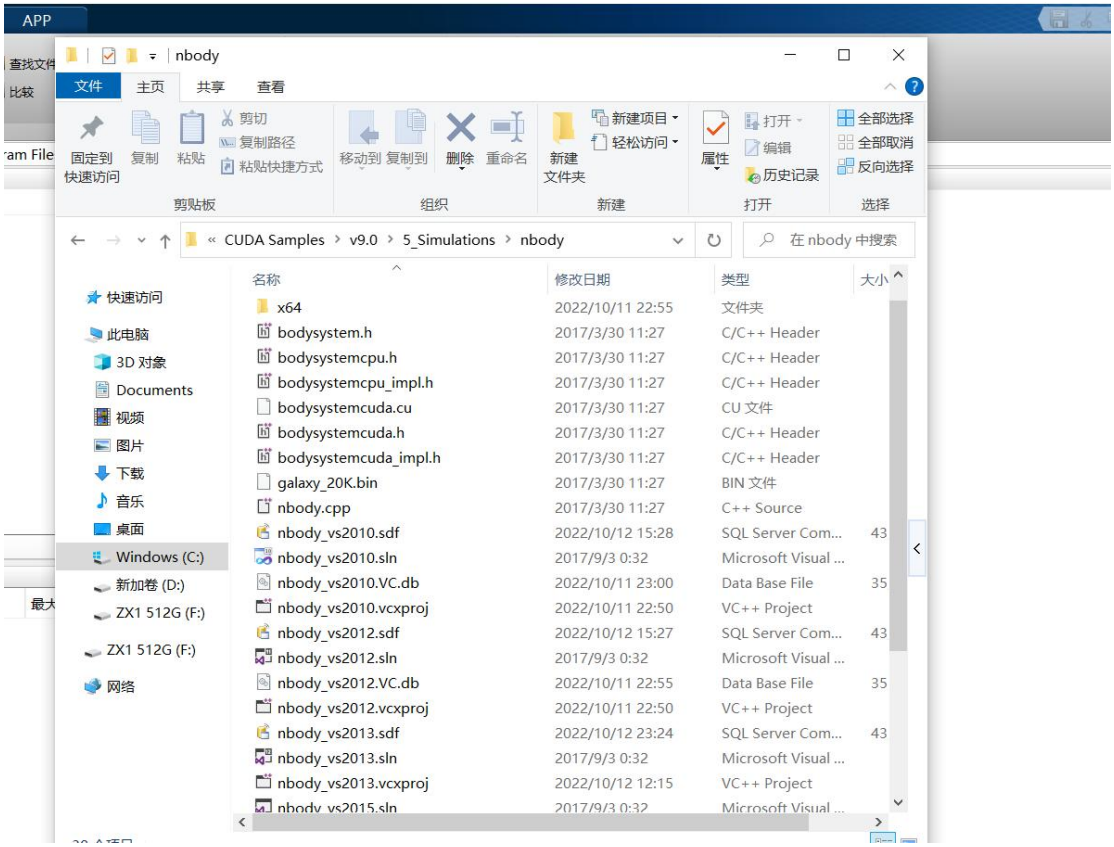


Figure 3.12

Open nbody_ Vs2010. sln as follows:

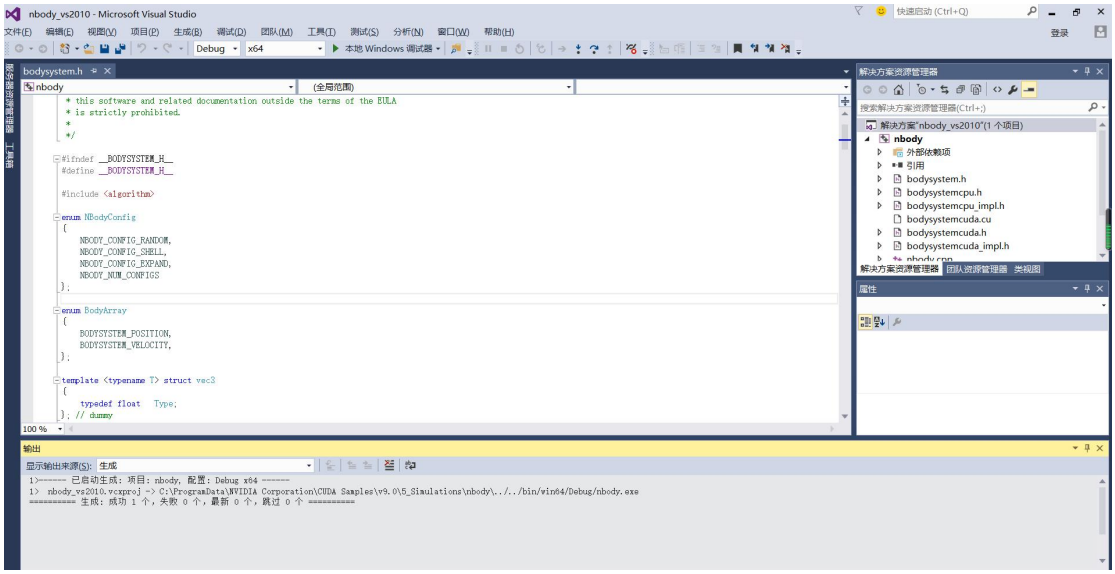


Figure 3.13

Open nbody_ vs2012. sln as follows:

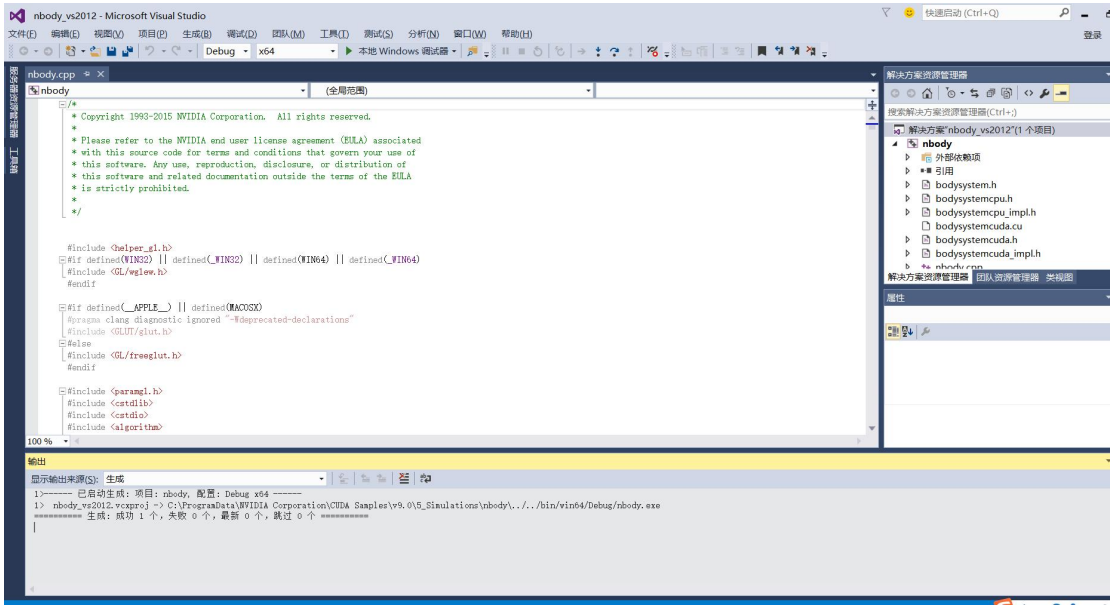


Figure 3.14

Open nbody_ vs2013. sln as follows:

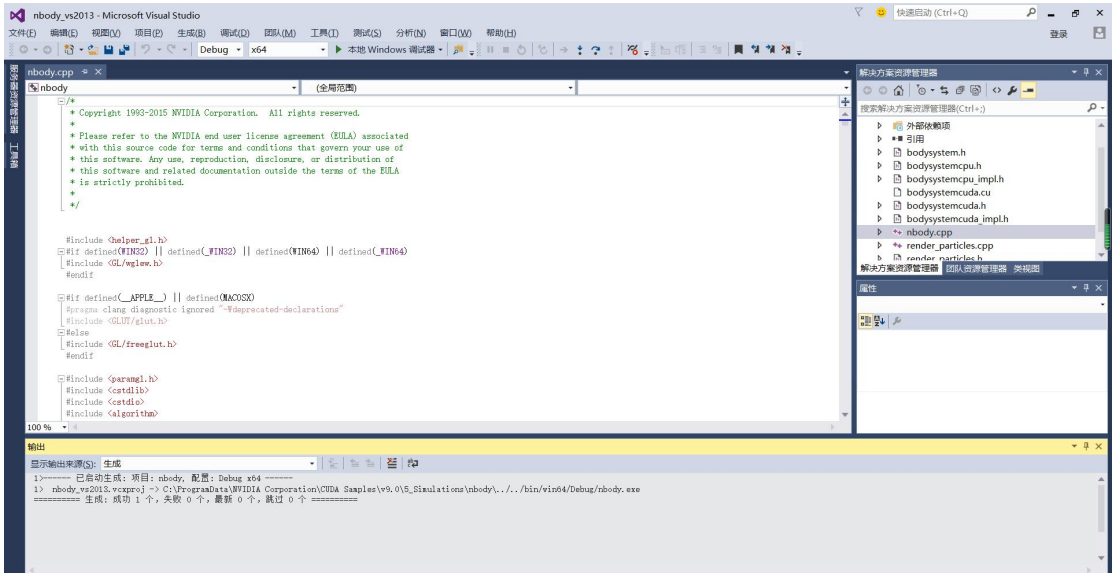


Figure 3.15

Navigate to the CUDA Samples' build directory and run the nbody sample. Note: Run samples by navigating to the executable's location, otherwise it will fail to locate dependent resources.

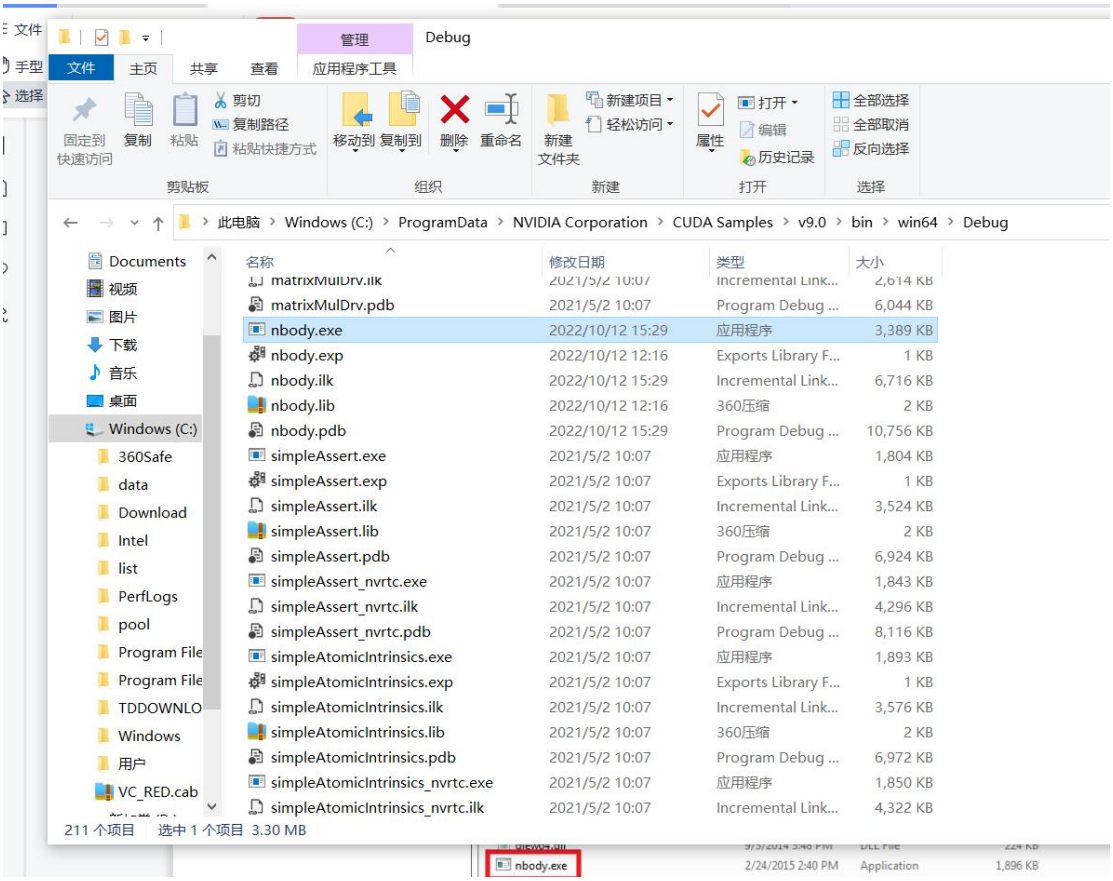


Figure 3.16（图 3.16）

The execution results are as follows:

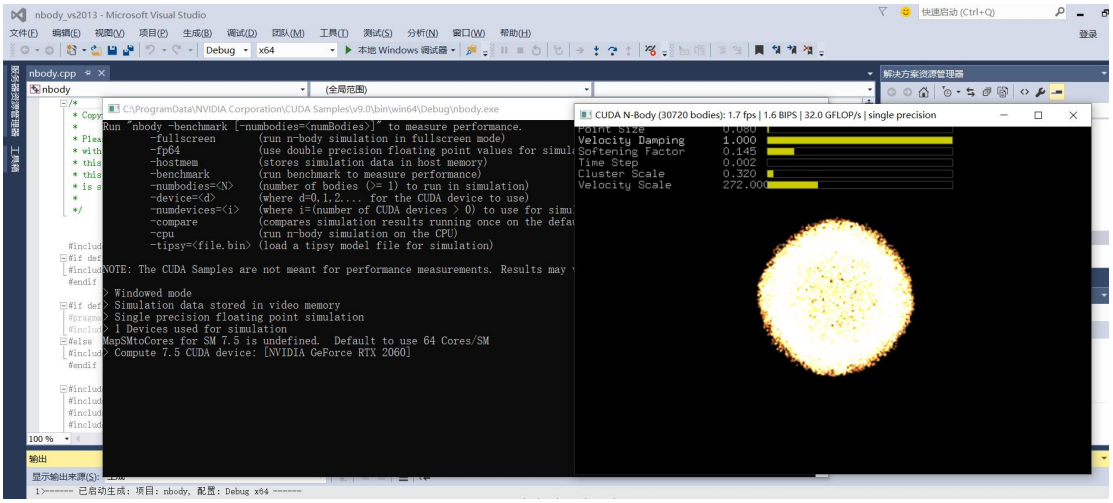


Figure 3.17

The results in Figure 3.17 show that the CUDNN library is installed successfully, and the GPU is used to speed up the operation.

3.1.2 Install CUDNN

Follow the steps on the official website to install CUDNN: https://docs.nvidia.com/deeplearning/cudnn/developer-guide/index.html

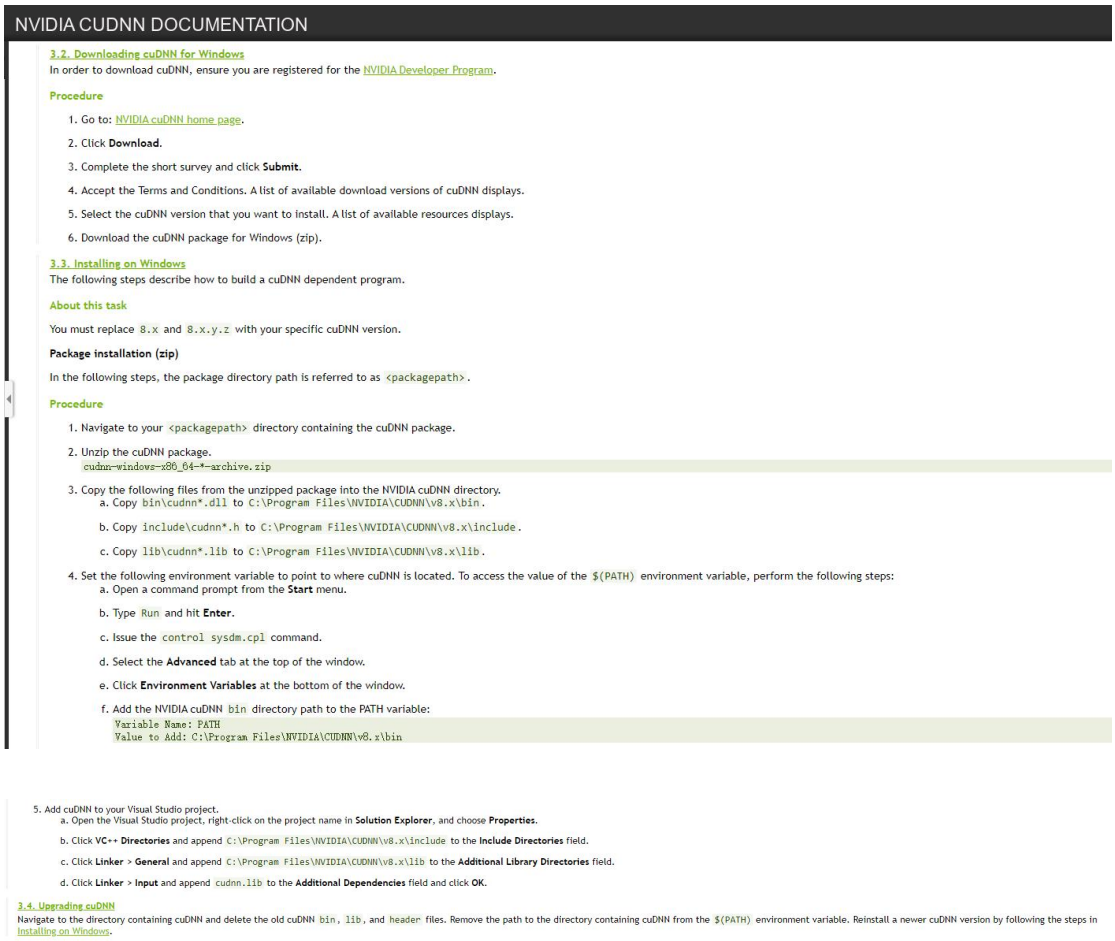


Figure 3.18

Execute the following code in MATLAB:

(1) Enter mex-setup c++ in the MATLAB command line window.

Figure 3.19

(2) Enter vl_compilenn ('enableGpu', true, ...'cudaRoot', 'C:\Program Files\NVIDIA

GPU Computing Toolkit\CUDA\v9.0', ... % Installation path of CUDA 'cudaMethod', 'nvcc', 'enableCudnn', 'true', ... 'cudnnRoot', 'C:\Program Files\NVIDIA GPU Computing Toolkit\CUDNN\cudnn-9.0-windows10-x64-v7\cuda'); % Installation path of CUDNN

```
>> mex -setup c++
MEX 配置为使用 'Microsoft Visual C++ 2015' 以进行 C++ 语言编译。
警告: MATLAB C 和 Fortran API 已更改，现可支持
     包含 2^32-1 个以上元素的 MATLAB 变量。您需要
     更新代码以利用新的 API。
     您可以在以下网址找到更多的相关信息:
     https://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html。
>> vl_compilenn('enableGpu', true, ...
'cudaRoot', 'C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0', ... % CUDA的安装路径
'cudaMethod', 'nvcc', 'enableCudnn', 'true', ...
'cudnnRoot', 'C:\Program Files\NVIDIA GPU Computing Toolkit\CUDNN\cudnn-9.0-windows10-x64-v7\cuda'); % cuDNN的路径
```

Figure 3.20

The compilation results are as follows:

```
变量 代码 SIMULINK 环境 资源
.TLAB ▸ R2018a ▸ bin ▸
命令行窗口
c:\program files\nvidia gpu computing toolkit\cuda\v9.0\include\crt\math_functions.h(5319): warning C4819: 该文件包含不能在当前代码页(936)中表示的字符。请将该文
c:\program files\nvidia gpu computing toolkit\cuda\v9.0\include\crt\math_functions.h(6229): warning C4819: 该文件包含不能在当前代码页(936)中表示的字符。请将该文
c:\program files\nvidia gpu computing toolkit\cuda\v9.0\include\crt\math_functions.h(7104): warning C4819: 该文件包含不能在当前代码页(936)中表示的字符。请将该文
c:\program files\nvidia gpu computing toolkit\cuda\v9.0\include\crt\math_functions.h(7914): warning C4819: 该文件包含不能在当前代码页(936)中表示的字符。请将该文
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
```

Figure 3.21

(3) Enter run vl_setupnn in the command line window of MATLAB（在 MATLAB 的命令行窗口中输入 run vl_setupnn）

```
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
fx >> run vl_setupnn
```

Figure 3.22

(4) Enter vl_testnn('gpu', true) on the command line window of MATLAB

```
使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
>> run vl_setupnn
>> vl_testnn('gpu', true)
```

Figure 3.23

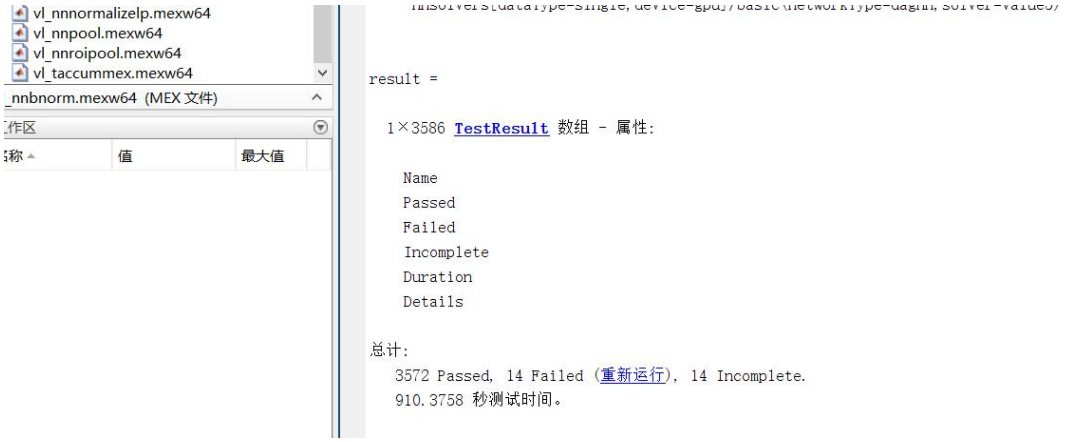


Figure 3.24

Fig 3.24 shows that the matconvnet test code runs successfully in the GPU environment.

After the compilation of CPU and GPU, you can run the code of WSWTNN-PnP.

4. Run the WSWTNN-PnP method code

4.1 Code download：

website：https://github.com/LiuTing20a/WSWTNN-PnP1

4.2 Run Demo_WSWTNN_PnP

The code is running::

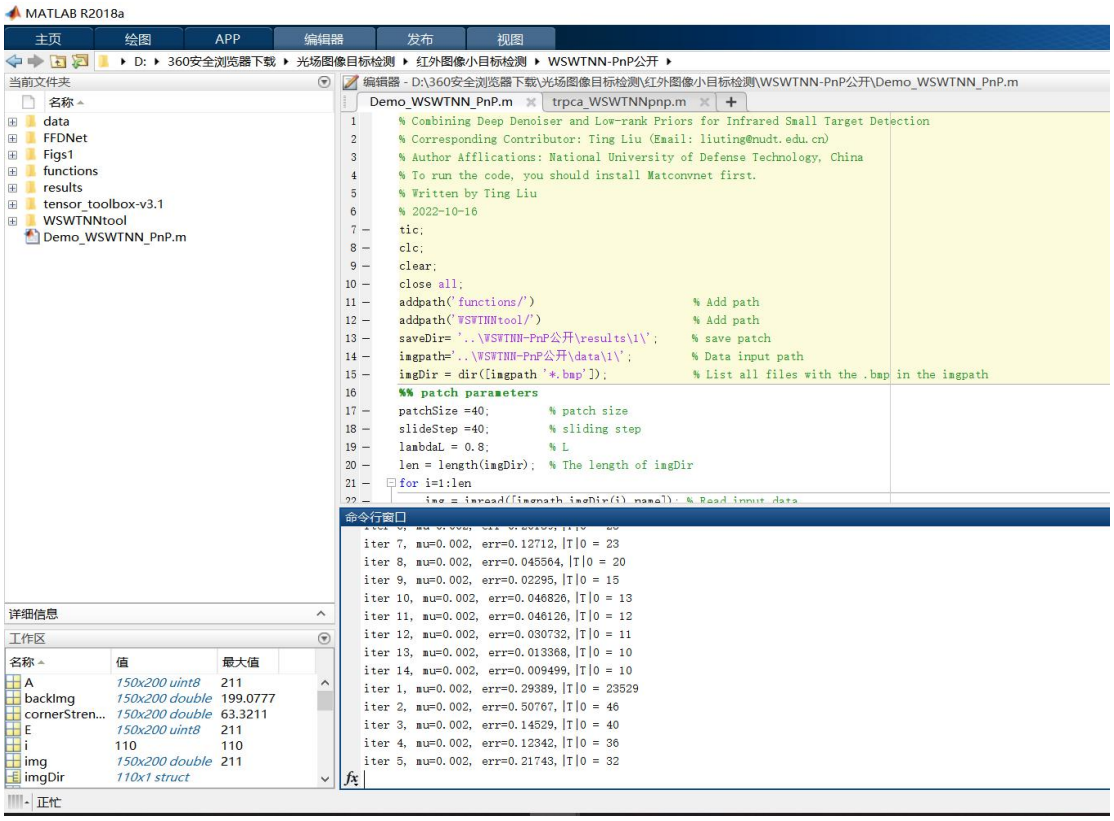


Figure 4.1

Experimental results in the process of code running are shown as follows:

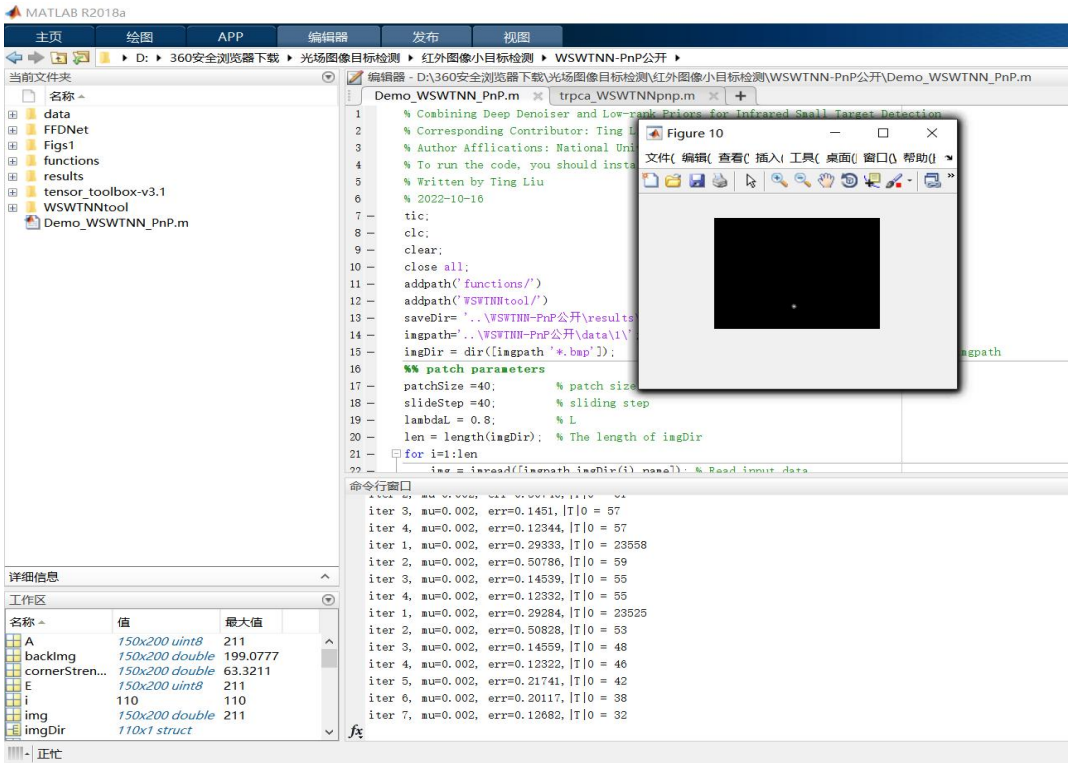


Figure 4.2

Code running completed:

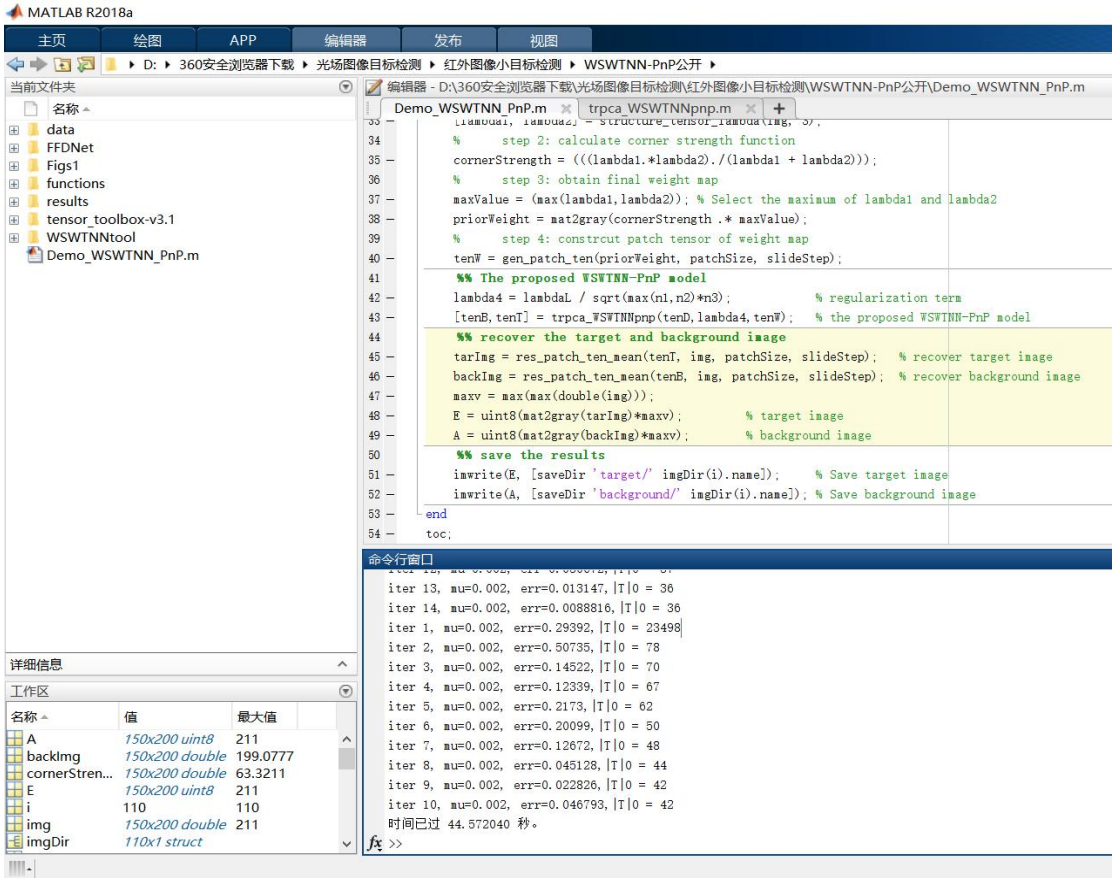


Figure 4.3

The experimental results are stored in the folder

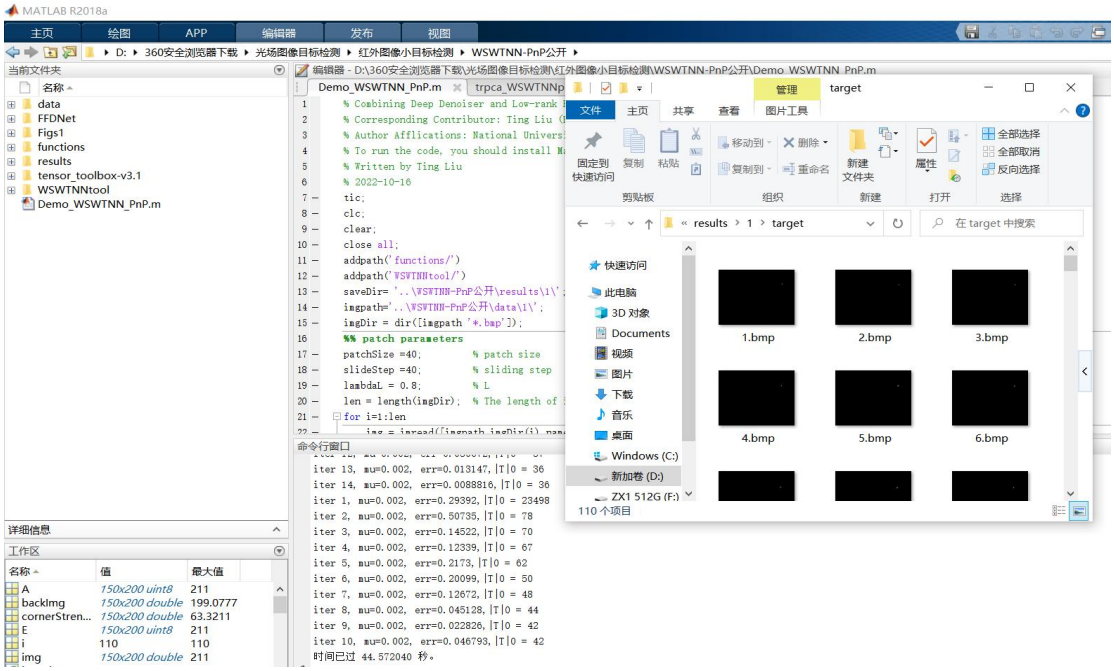


Figure 4.4