

# Introduction

## Environment

The environment of the project is in the path "project/env/"

Windows:

1, install Visual C++ Redistributable 2013 or 2015  
I install both

2, install qt (5.9.1) with **data visualization and MingGW (default is ok, but need confirmation as the figure)**

<https://wiki.qt.io/>

How to setup Qt and openCV on Windows

3, compile and install opencv lib 3.3.0 (install **cmake** first), (try the simple example after installing)

see

<https://wiki.qt.io/>

How to setup Qt and openCV on Windows

<https://www.youtube.com/watch?v=Ny34LpmWijo>

Note:

If you have the following error : ... windres.exe: unknown option -- W ...

try this : in cmake uncheck ENABLE\_PRECOMPILED\_HEADERS

If, in the file opencv/sources/modules/videoio/src/cap\_dshow.cpp, you have the following error : 'sprintf\_instead\_use\_StringCbPrintfA\_or\_StringCchPrintfA' was not declared in this scope ...

try this :put the following line: #define NO\_DSHOW\_STRSAFE, before the line : #include "DShow.h"

4, open project, change the include and lib of project file with your opencv path

My path:

```
LIBS += -L"C:\opencv\release\install\x86\mingw\bin" -lopencv_calib3d330 -lopencv_core330 -lopencv_features2d330 -lopencv_flann330 -lopencv_highgui330 -lopencv_imgcodecs330 -lopencv_imgproc330 -lopencv_ml330 -lopencv_objdetect330 -lopencv_photo330 -lopencv_shape330 -lopencv_stitching330 -lopencv_superres330 -lopencv_video330 -lopencv_videoio330 -lopencv_videostab330
```

```
INCLUDEPATH += C://opencv//release//install//include
```

5, build and run.

Mac:

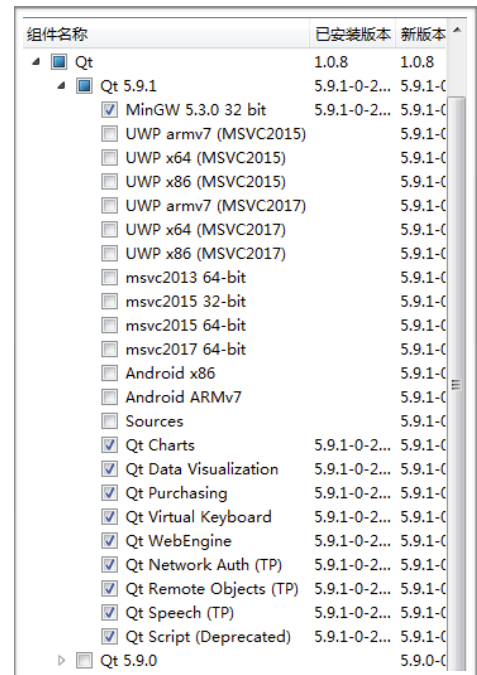
1, install qt (5.9.1), my version is 5.8.0

<https://www.youtube.com/watch?v=X-SJpwvzq0Y>

2, compile and install opencv lib (3.3.0), my version is 3.0.0, (install **cmake** first)

<https://www.youtube.com/watch?v=37RvqZVddAw>

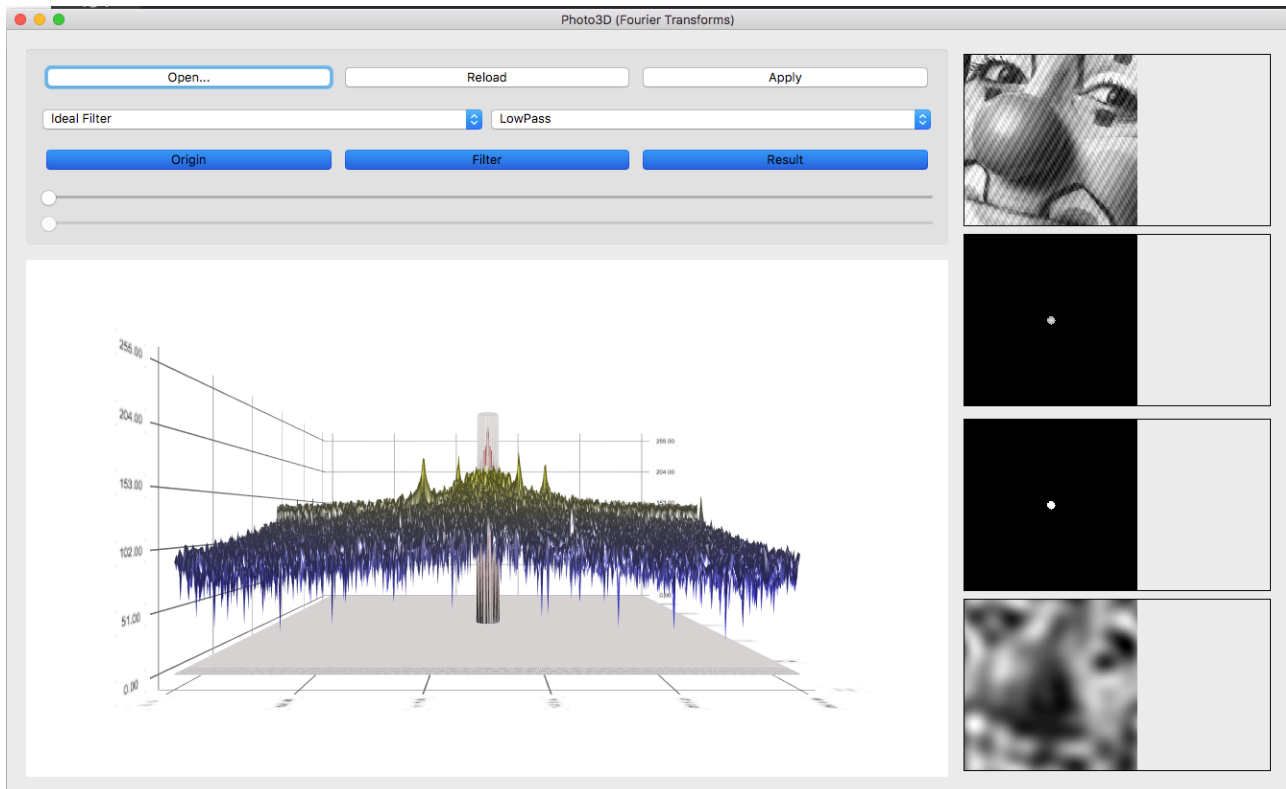
3, open project and build



# Source

The source code of the project is in the path “project/src/”

## Photo3D



Right panel is 2d image includes original image, Fourier transform, filter, filtered image.

Three toggle button “Origin” “Filter”, “Result” , set visible of 3D surface.

Push button :

“Open”: open image file.

“Reload”: currently it used to define custom filter.

“Apply”: filtering for image

Filter type:

Ideal filter: sharp filter

gaussian filter: curve sharp filter

Butterworth filter: similar as gaussian

Threshold filter: make a threshold and keep the low frequency.

Custom filter: make a custom filter, filter mask in the any frequency position. select frequency point on 3D surface. and push “Reload” button to create the mask.

Low pass: keep low frequency.

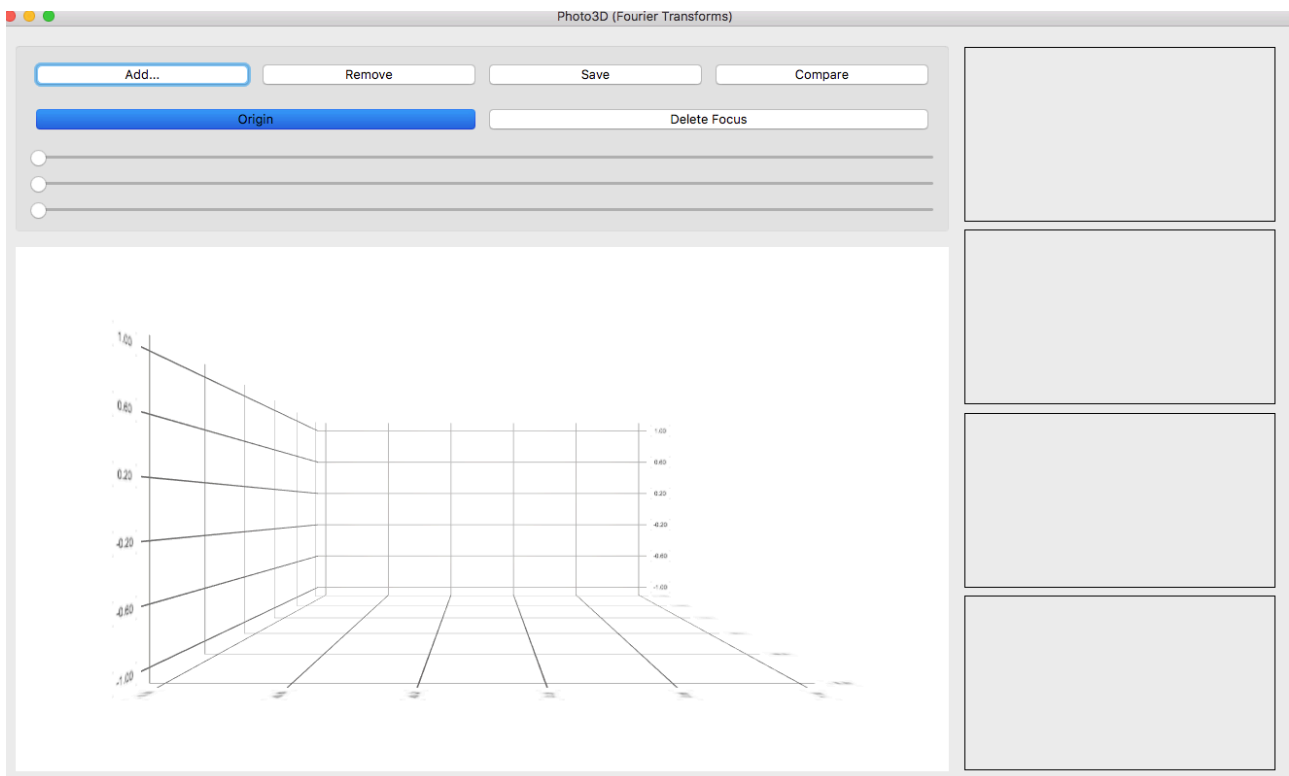
high pass: keep high frequency.

band-pass: make a ring mask.

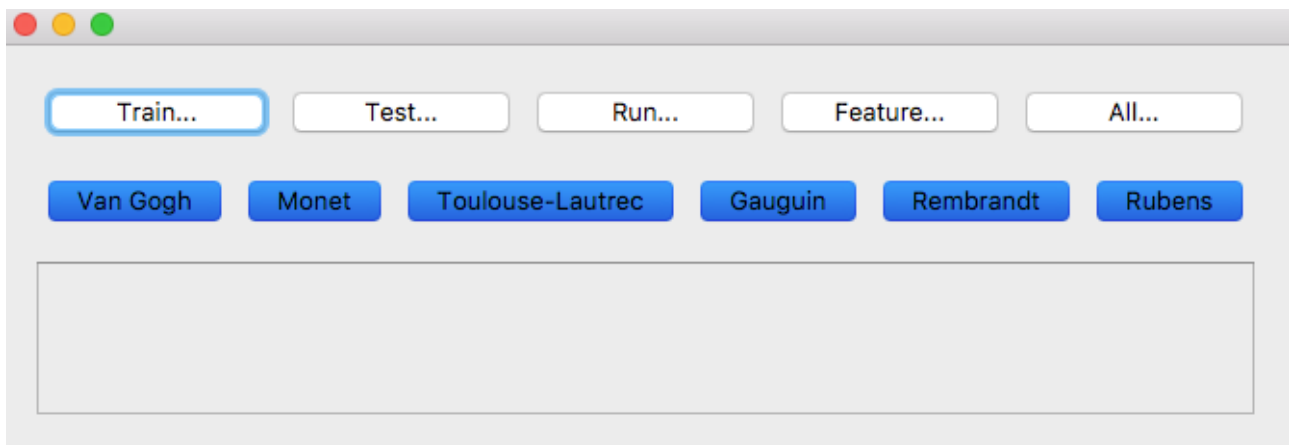
First Slider: control size of filter.

Second Slider: control curve sharp of Butterworth filter.

# Photo3Dmultiple (Painting Classification)



the UI basic UI is almost same as Photo3D.  
Push button “Compare” open UI of Painting classification



The six toggle button of Painter Name: select the Painter for classification.

PushButton

“Train”: select a xml file as a train input. (not related with the toggle button, could configure the experiment for free)

“Test”: select a xml file as a test input. (not related with the toggle button, could configure the experiment for free)

“Run”: run one experiment with the selected painter.

“Feature”: select an image, and return the feature vector.

“All”: run all experiments from 2-class to 6-class.

## Input file format:

```
<?xml version="1.0"?>
<Paintings>
  <Painting>
    <filename>/Users/wejaq/Documents/test/dataset/vangogh/1.jpg</filename>
    <height>34</height>
    <width>55</width>
    <author>0</author>
  </Painting>
  <Painting>
    <filename>/Users/wejaq/Documents/test/dataset/vangogh/2.jpg</filename>
    <height>34</height>
    <width>55</width>
    <author>0</author>
  </Painting>
</Paintings>
```

“filename”: image file path

“height”: image canvas height (current not use)

“width”: image canvas width (current not use)

“author”: author label, different painter need different label.

## Parameter setting:

### imagecropwindow.cpp

```
const QString trainfile = QString("/Users/wejaq/Documents/test/dataset/test/training_set.xml");
const QString testfile = QString("/Users/wejaq/Documents/test/dataset/test/test_set.xml");
const int descriptor_size = 10;
const int fourier_level = 10;
const int crop_size = 300;
```

trainfile and testfile: for the default experiments of the 6 painters. using for “Run” and “All” button with 6 toggle button of painters selection.

descriptor\_size: feature vector size, it current need be equal with fourier\_level.

fourier\_level: the max size level of Fourier transform.

crop\_size: patch size for cropping.

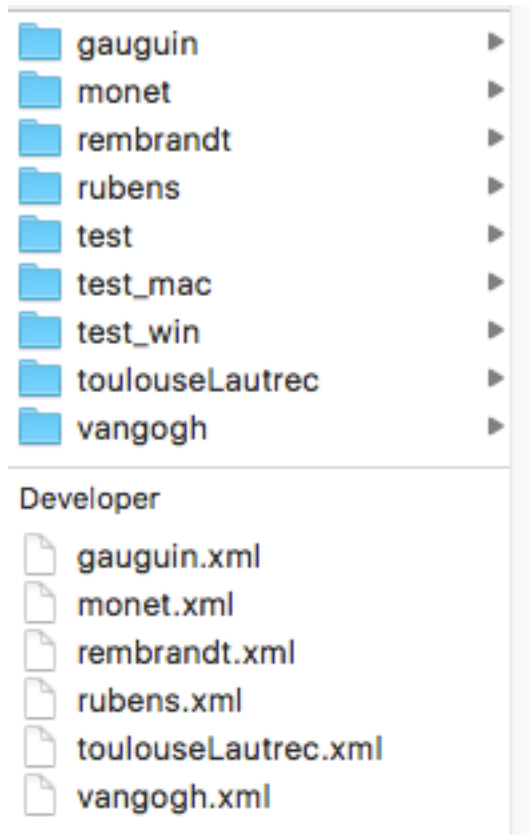
### imageprocess.cpp

```
const double threshold_value = 0.6;
```

threshold\_value: threshold for remove low value of frequency after fourier transform.

# Dataset

The dataset of the project is in the path "project/dataset/"



Painting image is put on the folder of painter's name.

xml file is input file of the project.

test folder includes

"test\_set.xml" and "training\_set.xml" include all the data of the experiments. use as main input of the programing. setting is in file "imagecropwindow.cpp"

"allsets.xml" use for analyze the result. see the method "handleBtnDistance" in file "imagecropwindow.cpp"

"test1.xml" include one entry for checking the code is right or not.