PCL安装教程请参考网页：https://blog.csdn.net/weixin\_42059276/article/details/106149359

# 1、PCL下载

PCL1.11.0下载地址： https://github.com/PointCloudLibrary/pcl/releases

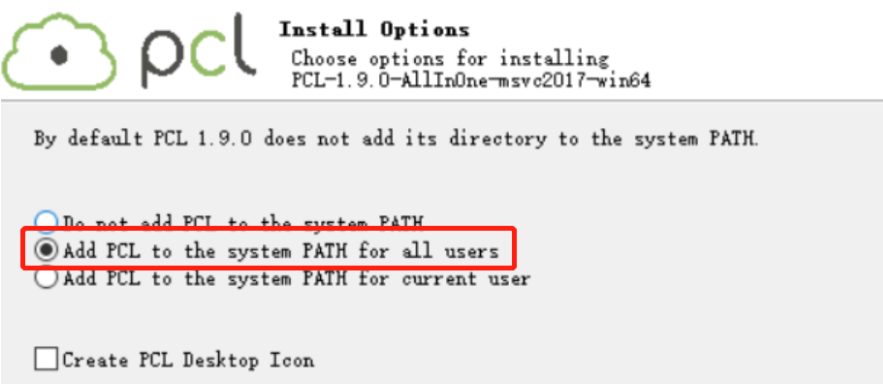
打开上述网页找到与VS2019版本对应的PCL库，包括两个文件：PCL-1.11.0-AllInOne-msvc2019-win64.exe以及pcl-1.11.0-pdb-msvc2019-win64.zip。



# 2、PCL安装

## 1）安装PCL

双击“PCL-1.11.0-AllInOne-msvc2019-win64.exe”，点击下一步，出现下图界面时，选择“Add PCL to the system PATH for all users”，自动把路径添加到系统环境变量中。安装路径选择D盘，其他的默认就行了。





## 2）安装OpenNI

安装完成之后打开文件夹 D:\PCL 1.11.0\3rdParty\OpenNI2。双击OpenNI-Windows-x64-2.2 选择路径（D:\PCL 1.11.0\3rdParty\OpenNI2）安装即可。



## 3）添加PDB文件

解压“pcl-1.11.0-pdb-msvc2019-win64.zip”，将解压得到的.pdb文件拷贝到“D:\PCL 1.11.0\bin”中。

## 4）添加环境变量

设置环境变量：右击计算机—属性—高级系统设置—高级—环境变量—用户变量—Path—编辑!，依次点击“新建”添加下列五个环境变量：

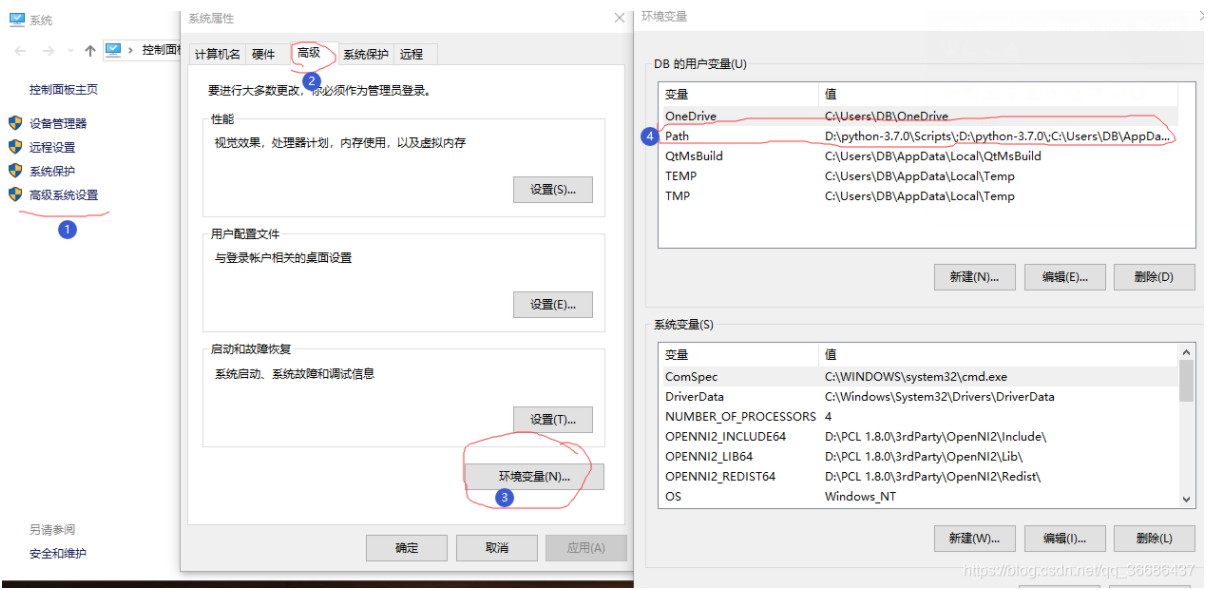
%PCL\_ROOT%\bin

%PCL\_ROOT%\3rdParty\FLANN\bin

%PCL\_ROOT%\3rdParty\VTK\bin

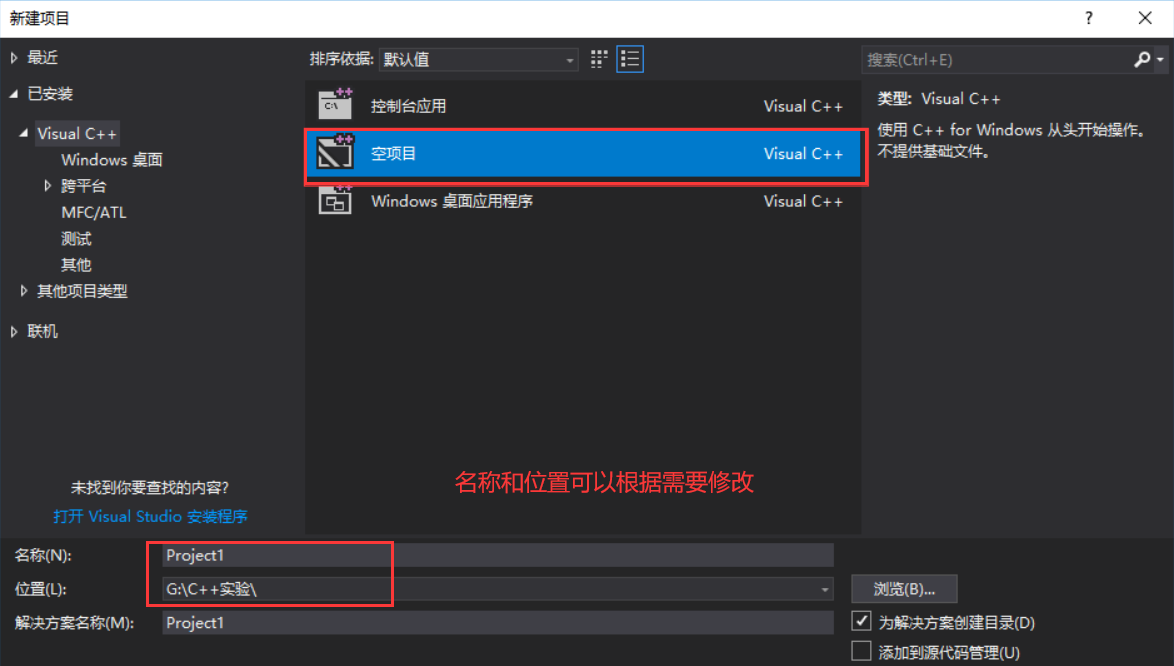
%PCL\_ROOT%\Qhull\bin

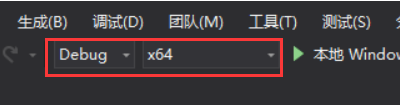
%PCL\_ROOT%\3rdParty\OpenNI2\Tools



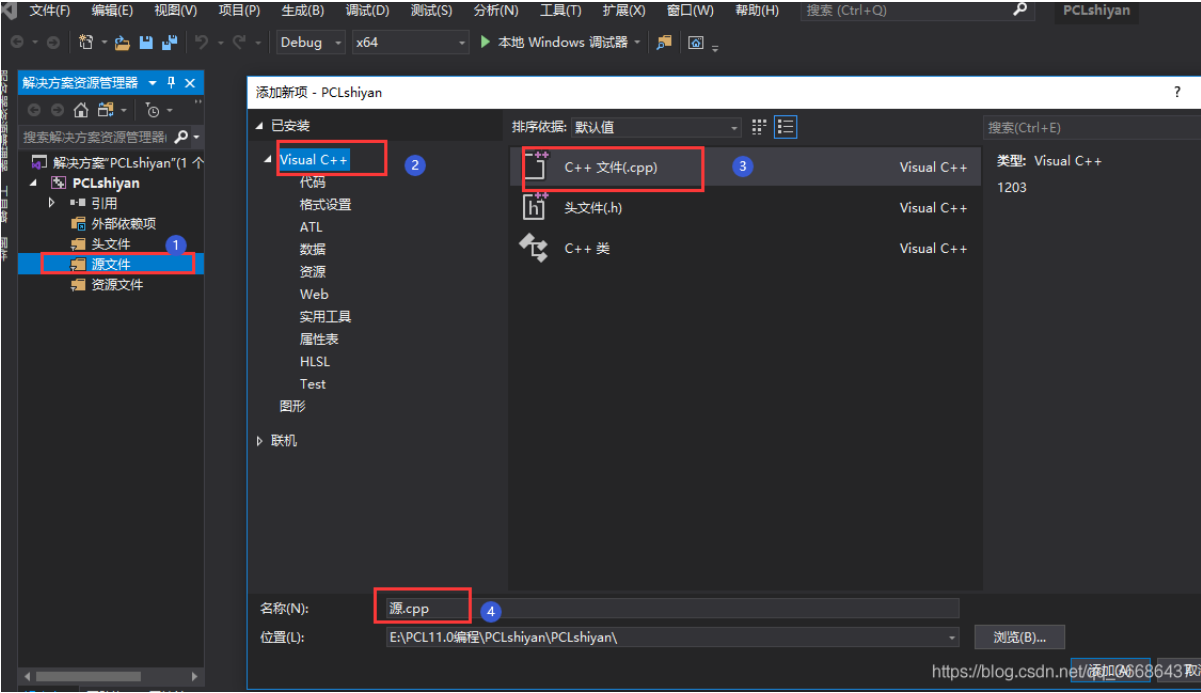
# 3、VS2019中配置PCL库

1）在VS中新建一个空项目，解决方案配准选择Debug，解决方案平台选择x64。

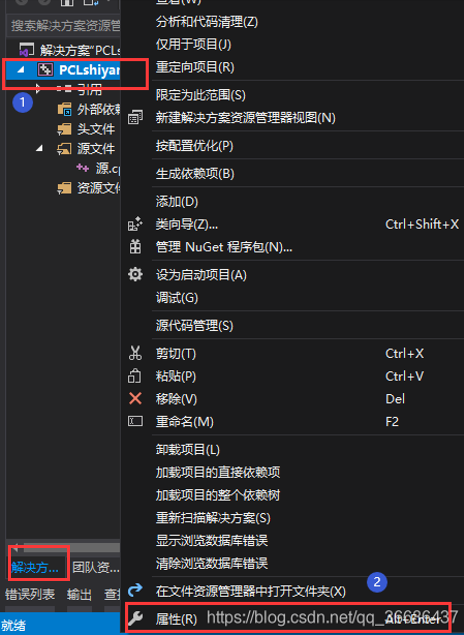




2）新建一个C++源文件



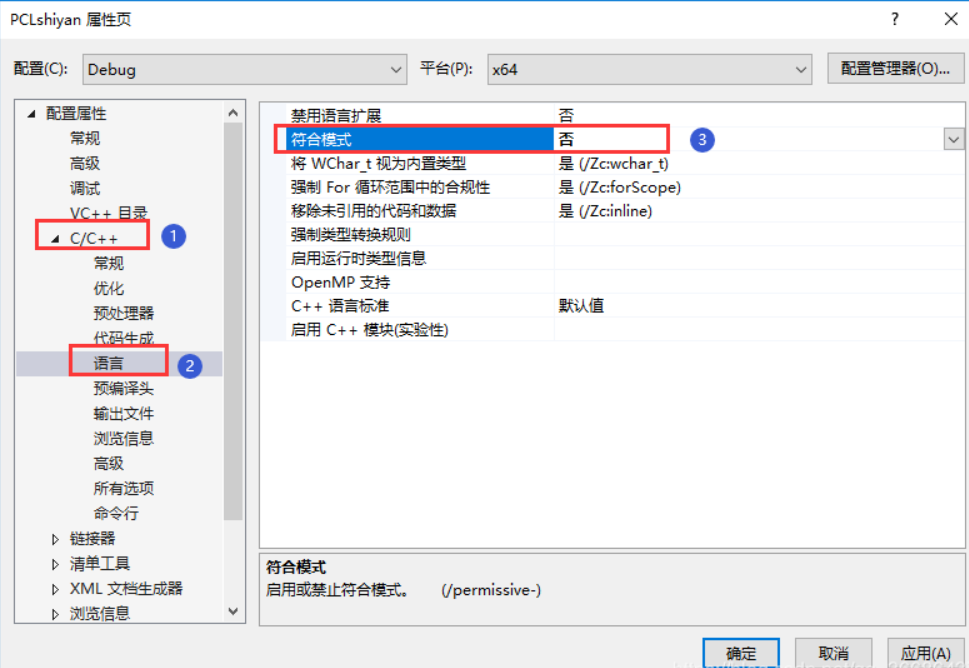
3）其他设置。打开属性表：右击解决方案管理器中新建的项目—>属性。



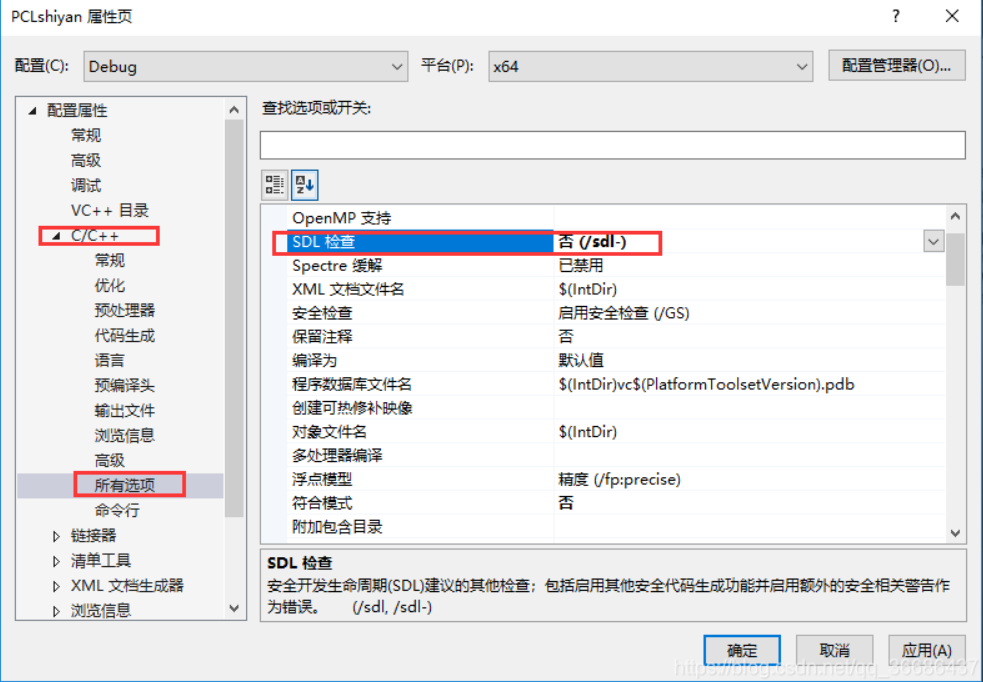
**①**项目属性中找到，配置属性—调试—环境—添加：

PATH=D:\PCL 1.11.0\\bin;D:\PCL 1.11.0\\3rdParty\FLANN\bin;D:\PCL 1.11.0\\3rdParty\VTK\bin;D:\PCL 1.11.0\\3rdParty\OpenNI2\Tools

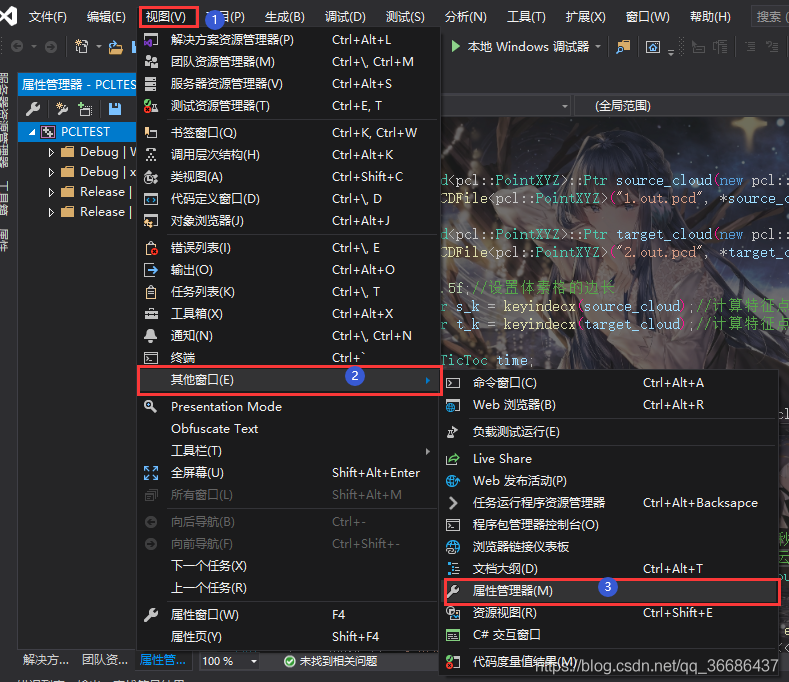
**②**打开C/C++—语言—符合模式：选择否



**③**C/C++ ->所有选项->SDL检查 改为否。



4）视图->其他窗口->**属性管理器**。



5）**添加属性表**。打开属性管理器之后，选择Debug|X64—单击Debug|X64左侧倒三角—右击选择“添加现有属性表”。



# 4、检查PCL是否安装成功

当前项目中添加cpp源文件，复制下列代码测试PCL库是否安装成功。

#include <iostream>

#include <vector>

#include <ctime>

#include <pcl/point\_cloud.h>

#include <pcl/octree/octree.h>

#include <boost/thread/thread.hpp>

#include <pcl/visualization/pcl\_visualizer.h>

using namespace std;

int

main(int argc, char\*\* argv)

{

srand((unsigned int)time(NULL));

pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);

// 创建点云数据

cloud->width = 1000;

cloud->height = 1;

cloud->points.resize(cloud->width \* cloud->height);

for (size\_t i = 0; i < cloud->points.size(); ++i)

{

cloud->points[i].x = 1024.0f \* rand() / (RAND\_MAX + 1.0f);

cloud->points[i].y = 1024.0f \* rand() / (RAND\_MAX + 1.0f);

cloud->points[i].z = 1024.0f \* rand() / (RAND\_MAX + 1.0f);

}

pcl::octree::OctreePointCloudSearch<pcl::PointXYZ> octree(0.1);

octree.setInputCloud(cloud);

octree.addPointsFromInputCloud();

pcl::PointXYZ searchPoint;

searchPoint.x = 1024.0f \* rand() / (RAND\_MAX + 1.0f);

searchPoint.y = 1024.0f \* rand() / (RAND\_MAX + 1.0f);

searchPoint.z = 1024.0f \* rand() / (RAND\_MAX + 1.0f);

//半径内近邻搜索

vector<int>pointIdxRadiusSearch;

vector<float>pointRadiusSquaredDistance;

float radius = 256.0f \* rand() / (RAND\_MAX + 1.0f);

cout << "Neighbors within radius search at (" << searchPoint.x

<< " " << searchPoint.y

<< " " << searchPoint.z

<< ") with radius=" << radius << endl;

if (octree.radiusSearch(searchPoint, radius, pointIdxRadiusSearch, pointRadiusSquaredDistance) > 0)

{

for (size\_t i = 0; i < pointIdxRadiusSearch.size(); ++i)

cout << " " << cloud->points[pointIdxRadiusSearch[i]].x

<< " " << cloud->points[pointIdxRadiusSearch[i]].y

<< " " << cloud->points[pointIdxRadiusSearch[i]].z

<< " (squared distance: " << pointRadiusSquaredDistance[i] << ")" << endl;

}

// 初始化点云可视化对象

boost::shared\_ptr<pcl::visualization::PCLVisualizer>viewer(new pcl::visualization::PCLVisualizer("显示点云"));

viewer->setBackgroundColor(0, 0, 0); //设置背景颜色为黑色

// 对点云着色可视化 (red).

pcl::visualization::PointCloudColorHandlerCustom<pcl::PointXYZ>target\_color(cloud, 255, 0, 0);

viewer->addPointCloud<pcl::PointXYZ>(cloud, target\_color, "target cloud");

viewer->setPointCloudRenderingProperties(pcl::visualization::PCL\_VISUALIZER\_POINT\_SIZE, 1, "target cloud");

// 等待直到可视化窗口关闭

while (!viewer->wasStopped())

{

viewer->spinOnce(100);

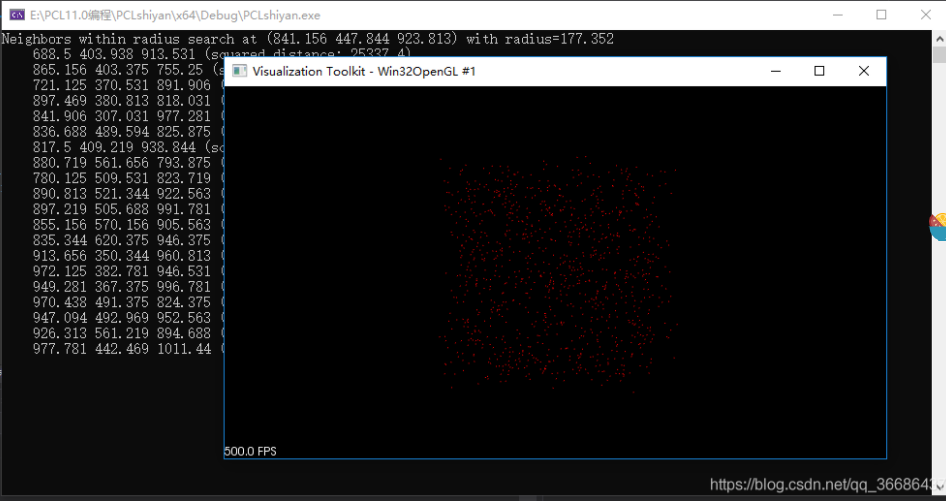
boost::this\_thread::sleep(boost::posix\_time::microseconds(1000));

}

return (0);

}

输出下图（数字可能不同），则表示安装成功！



# 5、附录——属性表设置

打开属性管理器之后，选择Debug|X64—单击Debug|X64左侧倒三角—右击选择“添加现有属性表”。

1. 双击新添加的属性表—VC++目录—包含目录，添加7个include路径

D:\PCL 1.11.0\include\pcl-1.11

D:\PCL 1.11.0\3rdParty\Boost\include\boost-1\_73

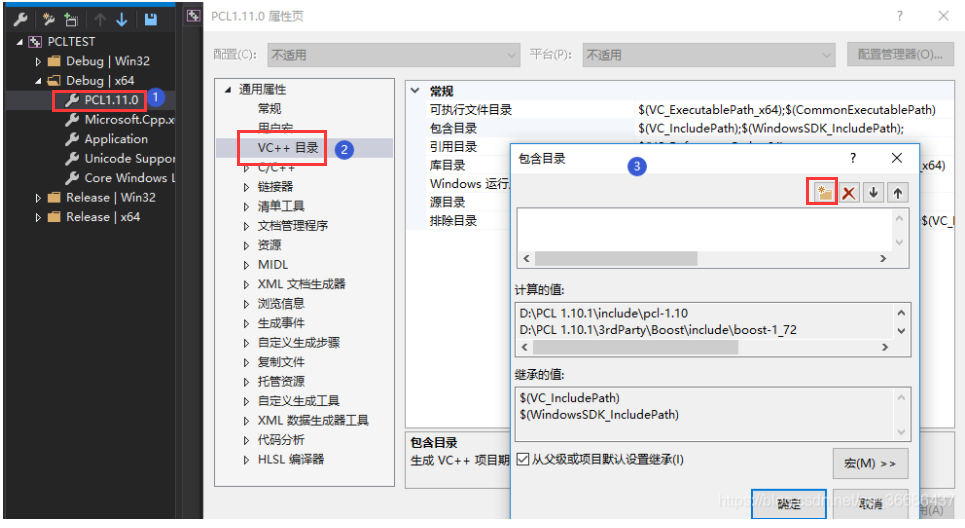
D:\PCL 1.11.0\3rdParty\Eigen\eigen3

D:\PCL 1.11.0\3rdParty\FLANN\include

D:\PCL 1.11.0\3rdParty\Qhull\include

D:\PCL 1.11.0\3rdParty\VTK\include\vtk-8.2

D:\PCL 1.11.0\3rdParty\OpenNI2\Include



**②**VC++目录—库目录，添加6个lib路径

D:\PCL 1.11.0\lib

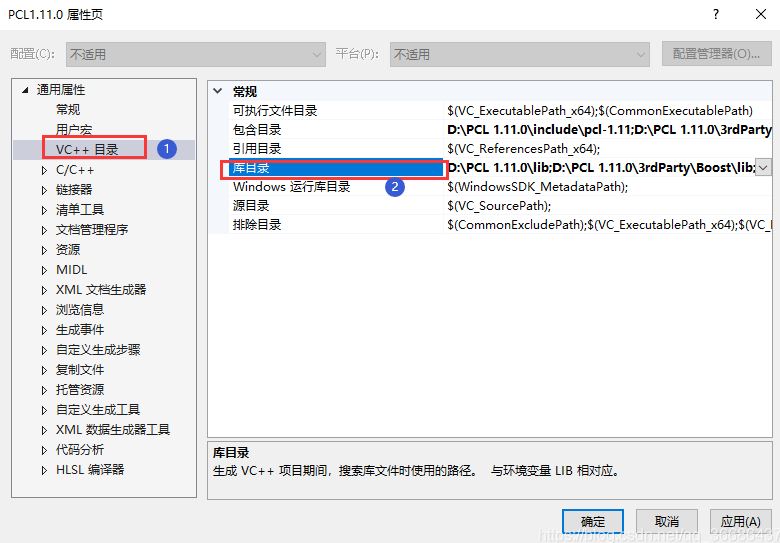
D:\PCL 1.11.0\3rdParty\Boost\lib

D:\PCL 1.11.0\3rdParty\FLANN\lib

D:\PCL 1.11.0\3rdParty\Qhull\lib

D:\PCL 1.11.0\3rdParty\OpenNI2\Lib

D:\PCL 1.11.0\3rdParty\VTK\lib

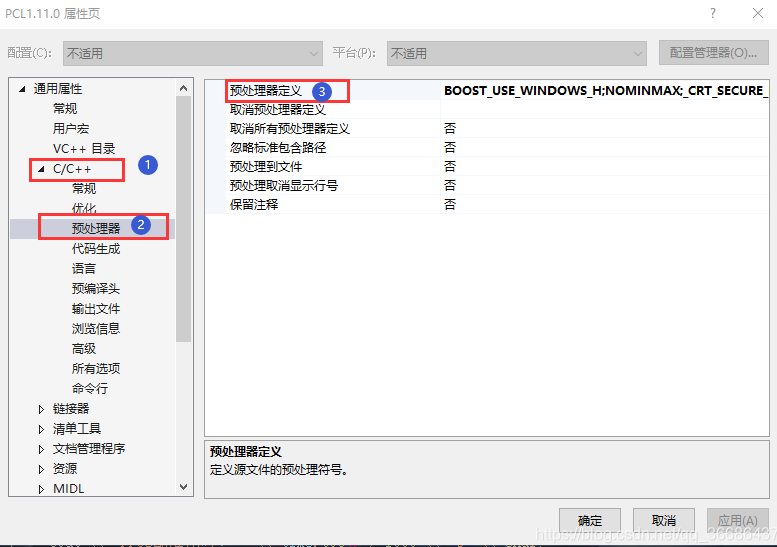


**③**C/C++—预处理器—预处理器定义—添加：

BOOST\_USE\_WINDOWS\_H

NOMINMAX

\_CRT\_SECURE\_NO\_DEPRECATE



**④**链接器—输入—附加依赖项——添加PCL和VTK的相关lib文件。因为解决方案配准为x64的Debug，所以属性表中添加Debug版本的lib文件。

pcl\_common\_debug.lib

pcl\_features\_debug.lib

pcl\_filters\_debug.lib

pcl\_io\_debug.lib

pcl\_io\_ply\_debug.lib

pcl\_kdtree\_debug.lib

pcl\_keypoints\_debug.lib

pcl\_ml\_debug.lib

pcl\_octree\_debug.lib

pcl\_outofcore\_debug.lib

pcl\_people\_debug.lib

pcl\_recognition\_debug.lib

pcl\_registration\_debug.lib

pcl\_sample\_consensus\_debug.lib

pcl\_search\_debug.lib

pcl\_segmentation\_debug.lib

pcl\_stereo\_debug.lib

pcl\_surface\_debug.lib

pcl\_tracking\_debug.lib

pcl\_visualization\_debug.lib

vtkalglib-8.2-gd.lib

vtkChartsCore-8.2-gd.lib

vtkCommonColor-8.2-gd.lib

vtkCommonComputationalGeometry-8.2-gd.lib

vtkCommonCore-8.2-gd.lib

vtkCommonDataModel-8.2-gd.lib

vtkCommonExecutionModel-8.2-gd.lib

vtkCommonMath-8.2-gd.lib

vtkCommonMisc-8.2-gd.lib

vtkCommonSystem-8.2-gd.lib

vtkCommonTransforms-8.2-gd.lib

vtkDICOMParser-8.2-gd.lib

vtkDomainsChemistry-8.2-gd.lib

vtkexoIIc-8.2-gd.lib

vtkexpat-8.2-gd.lib

vtkFiltersAMR-8.2-gd.lib

vtkFiltersCore-8.2-gd.lib

vtkFiltersExtraction-8.2-gd.lib

vtkFiltersFlowPaths-8.2-gd.lib

vtkFiltersGeneral-8.2-gd.lib

vtkFiltersGeneric-8.2-gd.lib

vtkFiltersGeometry-8.2-gd.lib

vtkFiltersHybrid-8.2-gd.lib

vtkFiltersHyperTree-8.2-gd.lib

vtkFiltersImaging-8.2-gd.lib

vtkFiltersModeling-8.2-gd.lib

vtkFiltersParallel-8.2-gd.lib

vtkFiltersParallelImaging-8.2-gd.lib

vtkFiltersPoints-8.2-gd.lib

vtkFiltersProgrammable-8.2-gd.lib

vtkFiltersSelection-8.2-gd.lib

vtkFiltersSMP-8.2-gd.lib

vtkFiltersSources-8.2-gd.lib

vtkFiltersStatistics-8.2-gd.lib

vtkFiltersTexture-8.2-gd.lib

vtkFiltersTopology-8.2-gd.lib

vtkFiltersVerdict-8.2-gd.lib

vtkfreetype-8.2-gd.lib

vtkGeovisCore-8.2-gd.lib

vtkgl2ps-8.2-gd.lib

vtkhdf5-8.2-gd.lib

vtkhdf5\_hl-8.2-gd.lib

vtkImagingColor-8.2-gd.lib

vtkImagingCore-8.2-gd.lib

vtkImagingFourier-8.2-gd.lib

vtkImagingGeneral-8.2-gd.lib

vtkImagingHybrid-8.2-gd.lib

vtkImagingMath-8.2-gd.lib

vtkImagingMorphological-8.2-gd.lib

vtkImagingSources-8.2-gd.lib

vtkImagingStatistics-8.2-gd.lib

vtkImagingStencil-8.2-gd.lib

vtkInfovisCore-8.2-gd.lib

vtkInfovisLayout-8.2-gd.lib

vtkInteractionImage-8.2-gd.lib

vtkInteractionStyle-8.2-gd.lib

vtkInteractionWidgets-8.2-gd.lib

vtkIOAMR-8.2-gd.lib

vtkIOCore-8.2-gd.lib

vtkIOEnSight-8.2-gd.lib

vtkIOExodus-8.2-gd.lib

vtkIOExport-8.2-gd.lib

vtkIOExportOpenGL-8.2-gd.lib

vtkIOGeometry-8.2-gd.lib

vtkIOImage-8.2-gd.lib

vtkIOImport-8.2-gd.lib

vtkIOInfovis-8.2-gd.lib

vtkIOLegacy-8.2-gd.lib

vtkIOLSDyna-8.2-gd.lib

vtkIOMINC-8.2-gd.lib

vtkIOMovie-8.2-gd.lib

vtkIONetCDF-8.2-gd.lib

vtkIOParallel-8.2-gd.lib

vtkIOParallelXML-8.2-gd.lib

vtkIOPLY-8.2-gd.lib

vtkIOSQL-8.2-gd.lib

vtkIOTecplotTable-8.2-gd.lib

vtkIOVideo-8.2-gd.lib

vtkIOXML-8.2-gd.lib

vtkIOXMLParser-8.2-gd.lib

vtkjpeg-8.2-gd.lib

vtkjsoncpp-8.2-gd.lib

vtklibharu-8.2-gd.lib

vtklibxml2-8.2-gd.lib

vtklz4-8.2-gd.lib

vtkmetaio-8.2-gd.lib

vtkNetCDF-8.2-gd.lib

vtknetcdfcpp-8.2-gd.lib

vtkoggtheora-8.2-gd.lib

vtkParallelCore-8.2-gd.lib

vtkpng-8.2-gd.lib

vtkproj4-8.2-gd.lib

vtkRenderingAnnotation-8.2-gd.lib

vtkRenderingContext2D-8.2-gd.lib

vtkRenderingContextOpenGL-8.2-gd.lib

vtkRenderingCore-8.2-gd.lib

vtkRenderingFreeType-8.2-gd.lib

vtkRenderingGL2PS-8.2-gd.lib

vtkRenderingImage-8.2-gd.lib

vtkRenderingLabel-8.2-gd.lib

vtkRenderingLIC-8.2-gd.lib

vtkRenderingLOD-8.2-gd.lib

vtkRenderingOpenGL-8.2-gd.lib

vtkRenderingVolume-8.2-gd.lib

vtkRenderingVolumeOpenGL-8.2-gd.lib

vtksqlite-8.2-gd.lib

vtksys-8.2-gd.lib

vtktiff-8.2-gd.lib

vtkverdict-8.2-gd.lib

vtkViewsContext2D-8.2-gd.lib

vtkViewsCore-8.2-gd.lib

vtkViewsInfovis-8.2-gd.lib

vtkzlib-8.2-gd.lib

如果解决方案配准为x64的Release，所以属性表中添加Release版本的lib文件：

pcl\_common\_release.lib

pcl\_features\_release.lib

pcl\_filters\_release.lib

pcl\_io\_ply\_release.lib

pcl\_io\_release.lib

pcl\_kdtree\_release.lib

pcl\_keypoints\_release.lib

pcl\_ml\_release.lib

pcl\_octree\_release.lib

pcl\_outofcore\_release.lib

pcl\_people\_release.lib

pcl\_recognition\_release.lib

pcl\_registration\_release.lib

pcl\_sample\_consensus\_release.lib

pcl\_search\_release.lib

pcl\_segmentation\_release.lib

pcl\_stereo\_release.lib

pcl\_surface\_release.lib

pcl\_tracking\_release.lib

pcl\_visualization\_release.lib

vtkalglib-8.2.lib

vtkChartsCore-8.2.lib

vtkCommonColor-8.2.lib

vtkCommonComputationalGeometry-8.2.lib

vtkCommonCore-8.2.lib

vtkCommonDataModel-8.2.lib

vtkCommonExecutionModel-8.2.lib

vtkCommonMath-8.2.lib

vtkCommonMisc-8.2.lib

vtkCommonSystem-8.2.lib

vtkCommonTransforms-8.2.lib

vtkDICOMParser-8.2.lib

vtkDomainsChemistry-8.2.lib

vtkexoIIc-8.2.lib

vtkexpat-8.2.lib

vtkFiltersAMR-8.2.lib

vtkFiltersCore-8.2.lib

vtkFiltersExtraction-8.2.lib

vtkFiltersFlowPaths-8.2.lib

vtkFiltersGeneral-8.2.lib

vtkFiltersGeneric-8.2.lib

vtkFiltersGeometry-8.2.lib

vtkFiltersHybrid-8.2.lib

vtkFiltersHyperTree-8.2.lib

vtkFiltersImaging-8.2.lib

vtkFiltersModeling-8.2.lib

vtkFiltersParallel-8.2.lib

vtkFiltersParallelImaging-8.2.lib

vtkFiltersPoints-8.2.lib

vtkFiltersProgrammable-8.2.lib

vtkFiltersSelection-8.2.lib

vtkFiltersSMP-8.2.lib

vtkFiltersSources-8.2.lib

vtkFiltersStatistics-8.2.lib

vtkFiltersTexture-8.2.lib

vtkFiltersTopology-8.2.lib

vtkFiltersVerdict-8.2.lib

vtkfreetype-8.2.lib

vtkGeovisCore-8.2.lib

vtkgl2ps-8.2.lib

vtkhdf5-8.2.lib

vtkhdf5\_hl-8.2.lib

vtkImagingColor-8.2.lib

vtkImagingCore-8.2.lib

vtkImagingFourier-8.2.lib

vtkImagingGeneral-8.2.lib

vtkImagingHybrid-8.2.lib

vtkImagingMath-8.2.lib

vtkImagingMorphological-8.2.lib

vtkImagingSources-8.2.lib

vtkImagingStatistics-8.2.lib

vtkImagingStencil-8.2.lib

vtkInfovisCore-8.2.lib

vtkInfovisLayout-8.2.lib

vtkInteractionImage-8.2.lib

vtkInteractionStyle-8.2.lib

vtkInteractionWidgets-8.2.lib

vtkIOAMR-8.2.lib

vtkIOCore-8.2.lib

vtkIOEnSight-8.2.lib

vtkIOExodus-8.2.lib

vtkIOExport-8.2.lib

vtkIOExportOpenGL-8.2.lib

vtkIOGeometry-8.2.lib

vtkIOImage-8.2.lib

vtkIOImport-8.2.lib

vtkIOInfovis-8.2.lib

vtkIOLegacy-8.2.lib

vtkIOLSDyna-8.2.lib

vtkIOMINC-8.2.lib

vtkIOMovie-8.2.lib

vtkIONetCDF-8.2.lib

vtkIOParallel-8.2.lib

vtkIOParallelXML-8.2.lib

vtkIOPLY-8.2.lib

vtkIOSQL-8.2.lib

vtkIOTecplotTable-8.2.lib

vtkIOVideo-8.2.lib

vtkIOXML-8.2.lib

vtkIOXMLParser-8.2.lib

vtkjpeg-8.2.lib

vtkjsoncpp-8.2.lib

vtklibharu-8.2.lib

vtklibxml2-8.2.lib

vtklz4-8.2.lib

vtkmetaio-8.2.lib

vtkNetCDF-8.2.lib

vtknetcdfcpp-8.2.lib

vtkoggtheora-8.2.lib

vtkParallelCore-8.2.lib

vtkpng-8.2.lib

vtkproj4-8.2.lib

vtkRenderingAnnotation-8.2.lib

vtkRenderingContext2D-8.2.lib

vtkRenderingContextOpenGL-8.2.lib

vtkRenderingCore-8.2.lib

vtkRenderingFreeType-8.2.lib

vtkRenderingGL2PS-8.2.lib

vtkRenderingImage-8.2.lib

vtkRenderingLabel-8.2.lib

vtkRenderingLIC-8.2.lib

vtkRenderingLOD-8.2.lib

vtkRenderingOpenGL-8.2.lib

vtkRenderingVolume-8.2.lib

vtkRenderingVolumeOpenGL-8.2.lib

vtksqlite-8.2.lib

vtksys-8.2.lib

vtktiff-8.2.lib

vtkverdict-8.2.lib

vtkViewsContext2D-8.2.lib

vtkViewsCore-8.2.lib

vtkViewsInfovis-8.2.lib

vtkzlib-8.2.lib