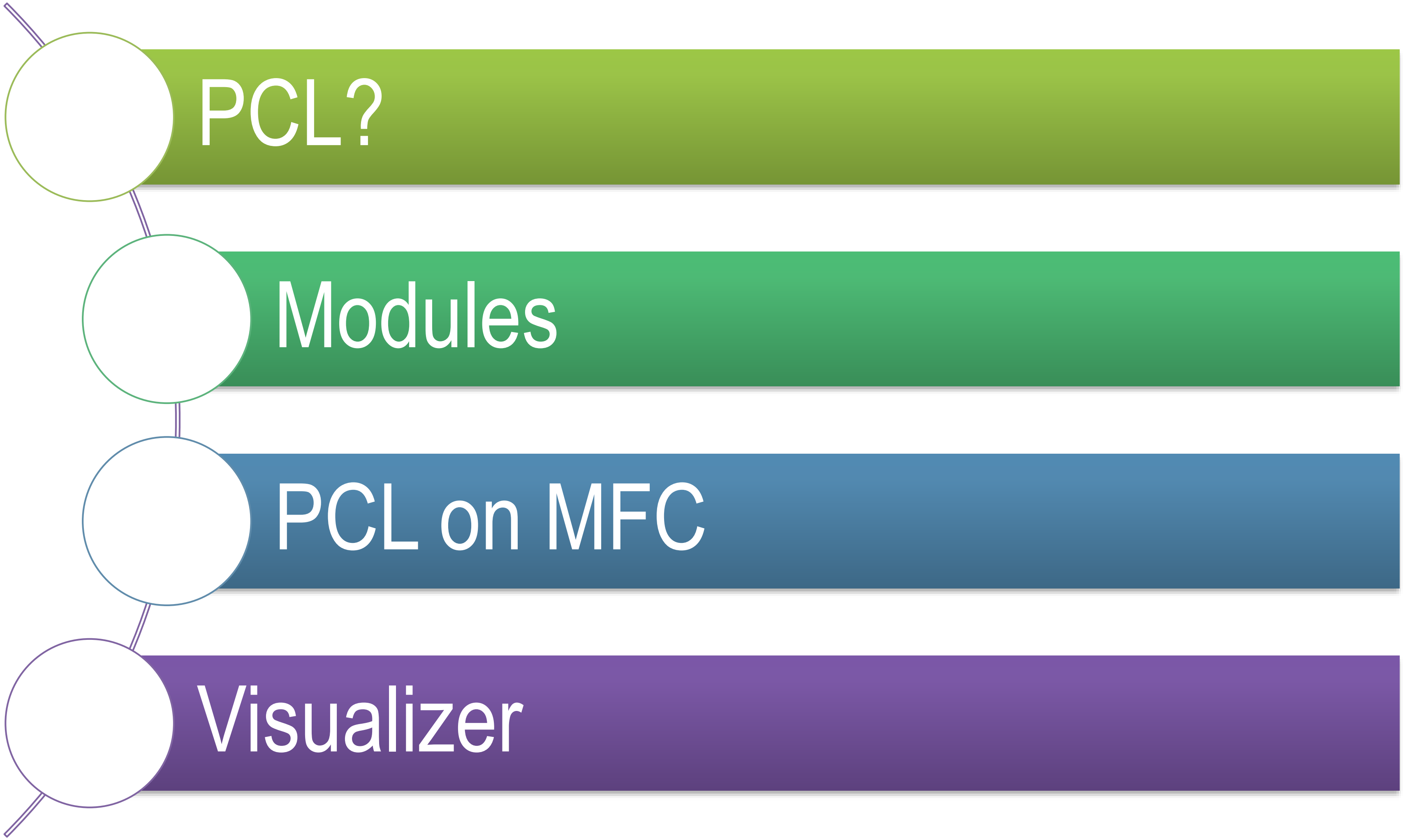
pointcloudlibrary

Introduction

Using on MFC environment

PCLVisualizer for beginners

Contents



- PCL?
- Modules
- PCL on MFC
- Visualizer

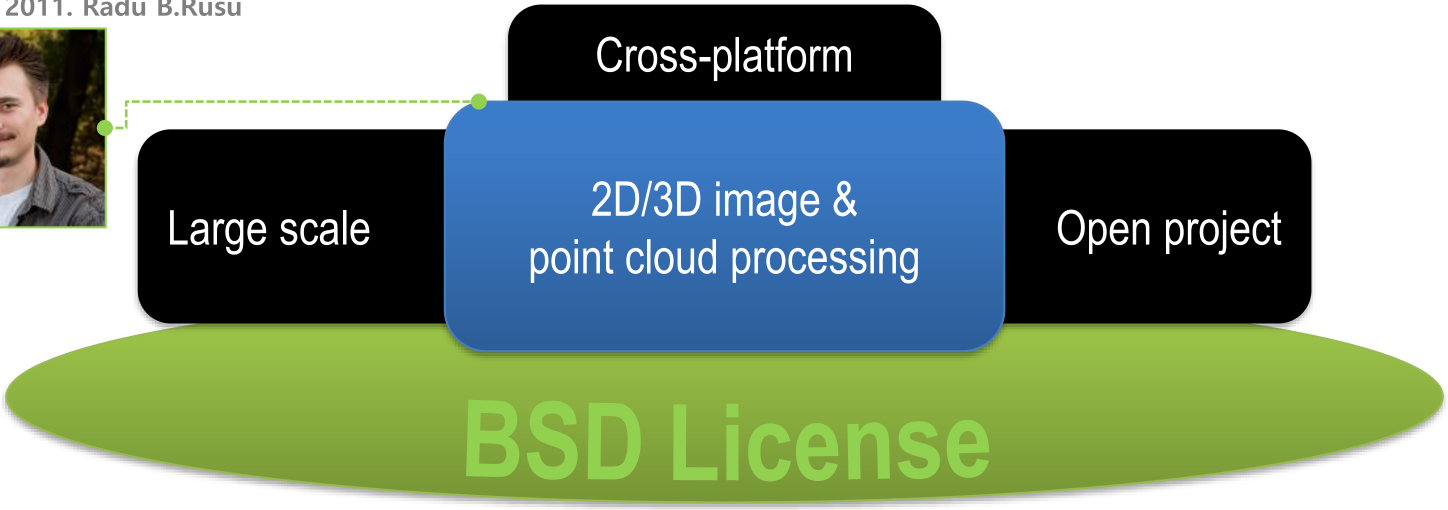
Point Cloud Library

- **Point Cloud**
  - A data structure used to represent a collection of multi-dimensional points
  - Commonly represents 3-dimensional data (X, Y, Z)
  - When color information is present, the point cloud becomes 4D

- **What is PCL?**

Cross-platform
: Linux, Mac OS, Windows, Android/iOS

March, 2011. Radu B.Rusu



Cross-platform

Large scale

2D/3D image &
point cloud processing

Open project

BSD License

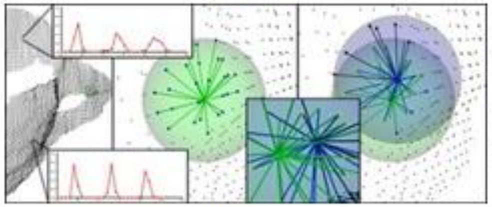
http://www.pointclouds.org

Berkeley **S**oftware **D**istribution License
: 소스코드 공개의 의무가 없으며 상용 소프트웨어에서도 무제한 사용가능

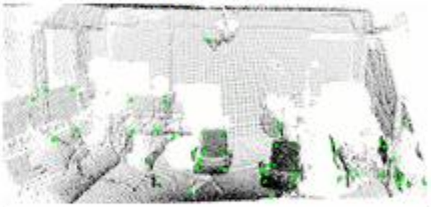
Point Cloud Library : Modules

filters

features



keypoints



registration

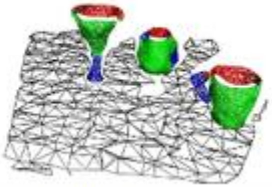
kdtree

octree

segmentation

sample_consensus

surface

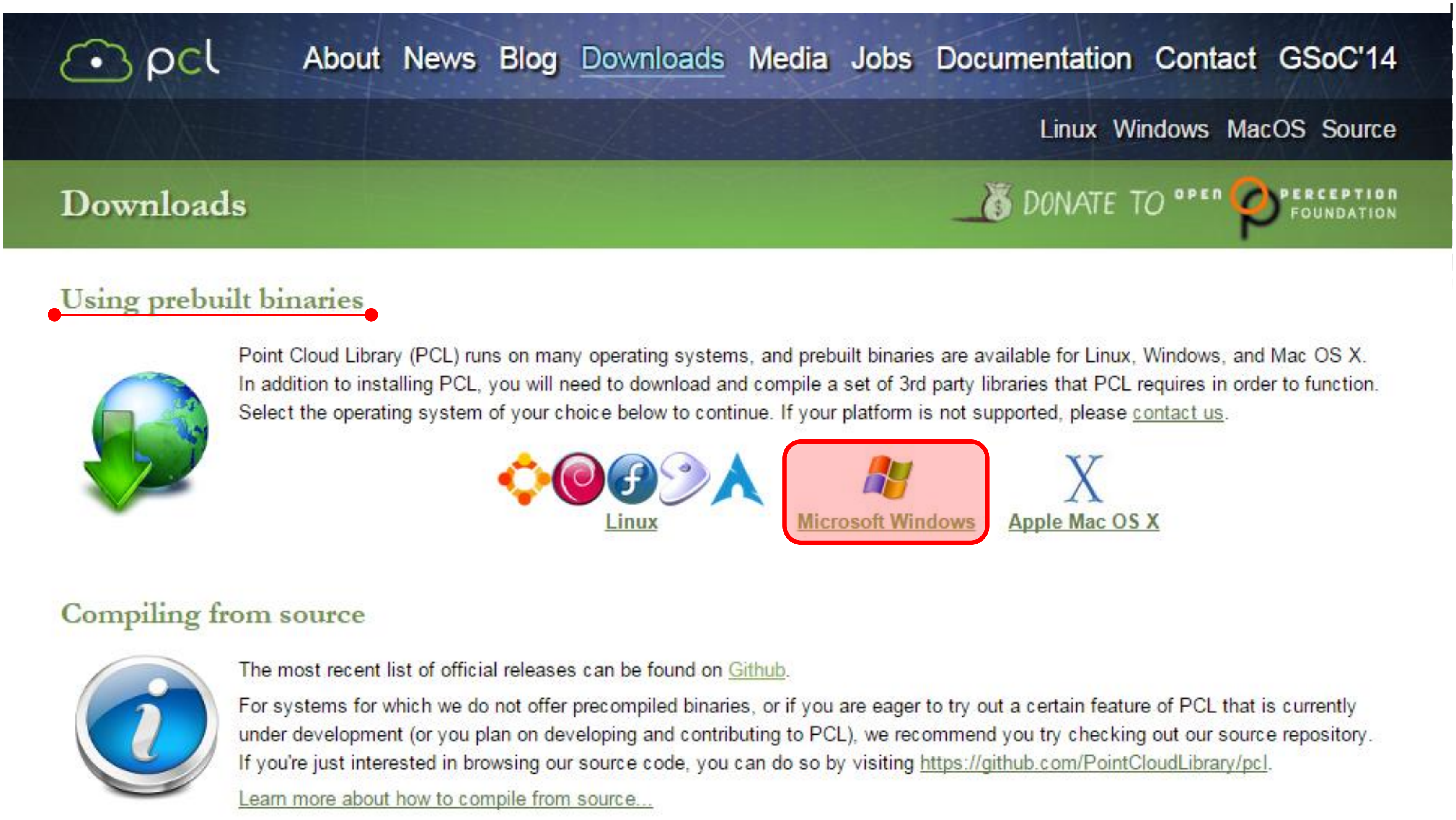


recognition

io

visualization

Installation



Downloads

Using prebuilt binaries

Point Cloud Library (PCL) runs on many operating systems, and prebuilt binaries are available for Linux, Windows, and Mac OS X. In addition to installing PCL, you will need to download and compile a set of 3rd party libraries that PCL requires in order to function. Select the operating system of your choice below to continue. If your platform is not supported, please contact us.

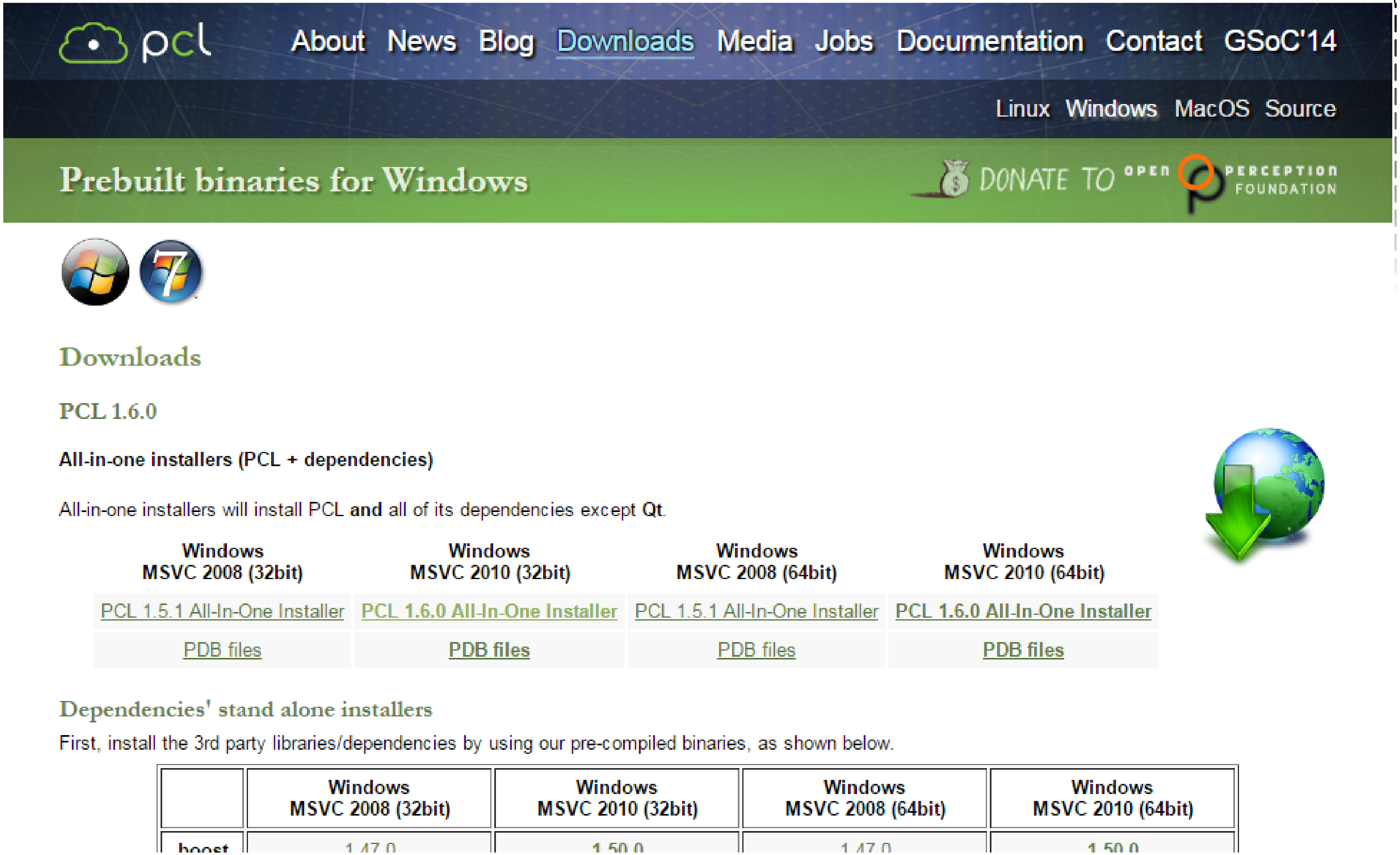
Compiling from source

The most recent list of official releases can be found on Github.

For systems for which we do not offer precompiled binaries, or if you are eager to try out a certain feature of PCL that is currently under development (or you plan on developing and contributing to PCL), we recommend you try checking out our source repository. If you're just interested in browsing our source code, you can do so by visiting https://github.com/PointCloudLibrary/pcl.

Learn more about how to compile from source...

Installation



Prebuilt binaries for Windows

Downloads

PCL 1.6.0

All-in-one installers (PCL + dependencies)

All-in-one installers will install PCL **and** all of its dependencies except **Qt**.

Windows MSVC 2008 (32bit)	Windows MSVC 2010 (32bit)	Windows MSVC 2008 (64bit)	Windows MSVC 2010 (64bit)
PCL 1.5.1 All-In-One Installer	**PCL 1.6.0 All-In-One Installer**	PCL 1.5.1 All-In-One Installer	**PCL 1.6.0 All-In-One Installer**
PDB files	**PDB files**	PDB files	**PDB files**

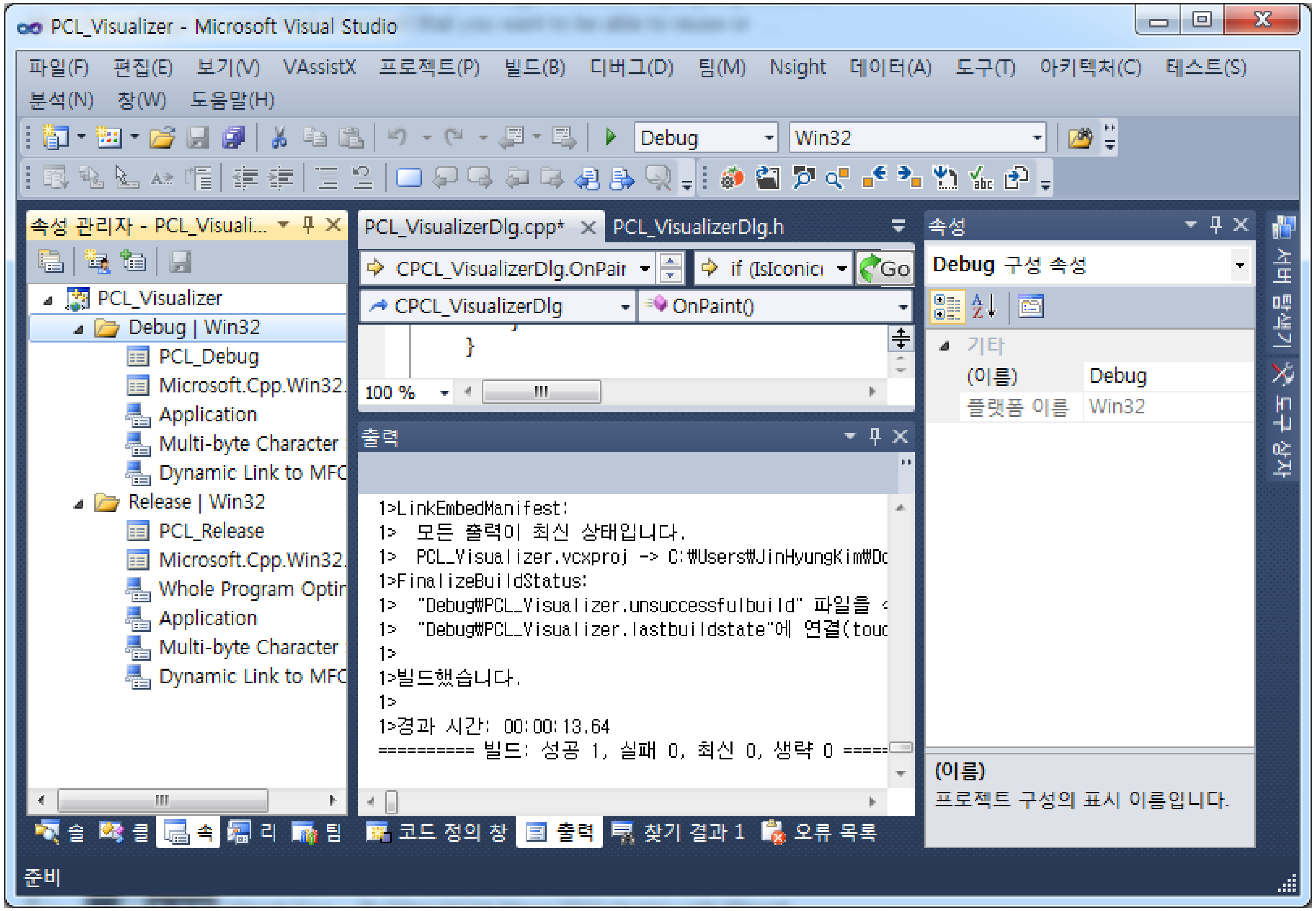
Dependencies' stand alone installers

First, install the 3rd party libraries/dependencies by using our pre-compiled binaries, as shown below.

	Windows MSVC 2008 (32bit)	Windows MSVC 2010 (32bit)	Windows MSVC 2008 (64bit)	Windows MSVC 2010 (64bit)
boost	1.47.0	1.50.0	1.47.0	1.50.0

Setting develop environment

- Include, library, binary directory & additional dependency



```
1>LinkEmbedManifest:
1>  모든 출력이 최신 상태입니다.
1>  PCL_Visualizer.vcxproj -> C:₩Users₩JinHyungKim₩Dc
1>FinalizeBuildStatus:
1>  "Debug₩PCL_Visualizer.unsuccessfulbuild" 파일을 
1>  "Debug₩PCL_Visualizer.lastbuildstate"에 연결(touc
1>
1>빌드했습니다.
1>
1>경과 시간: 00:00:13.64
========== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

(이름) Debug

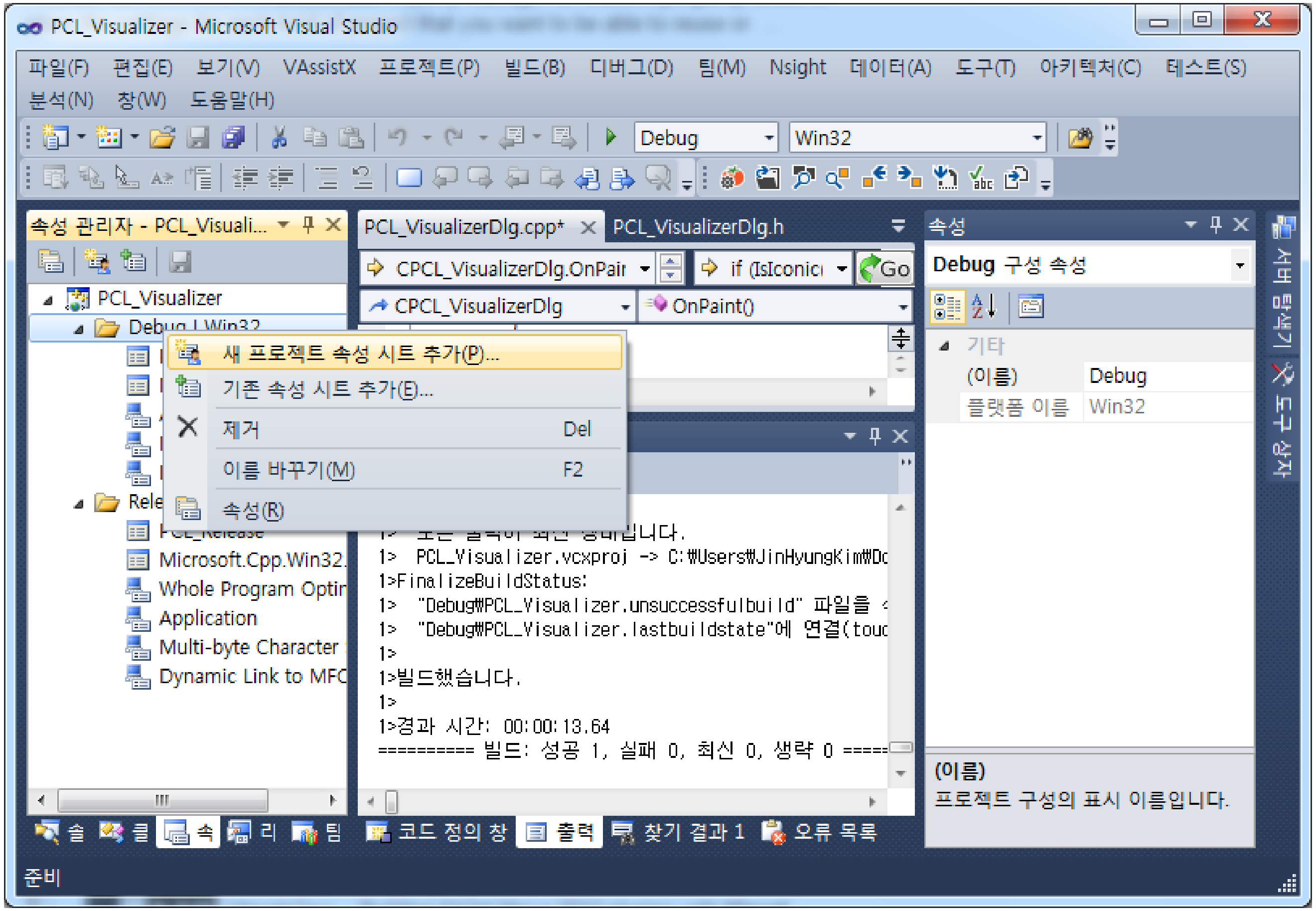
플랫폼 이름 Win32

(이름)

프로젝트 구성의 표시 이름입니다.

Setting develop environment

- Include, library, binary directory & additional dependency



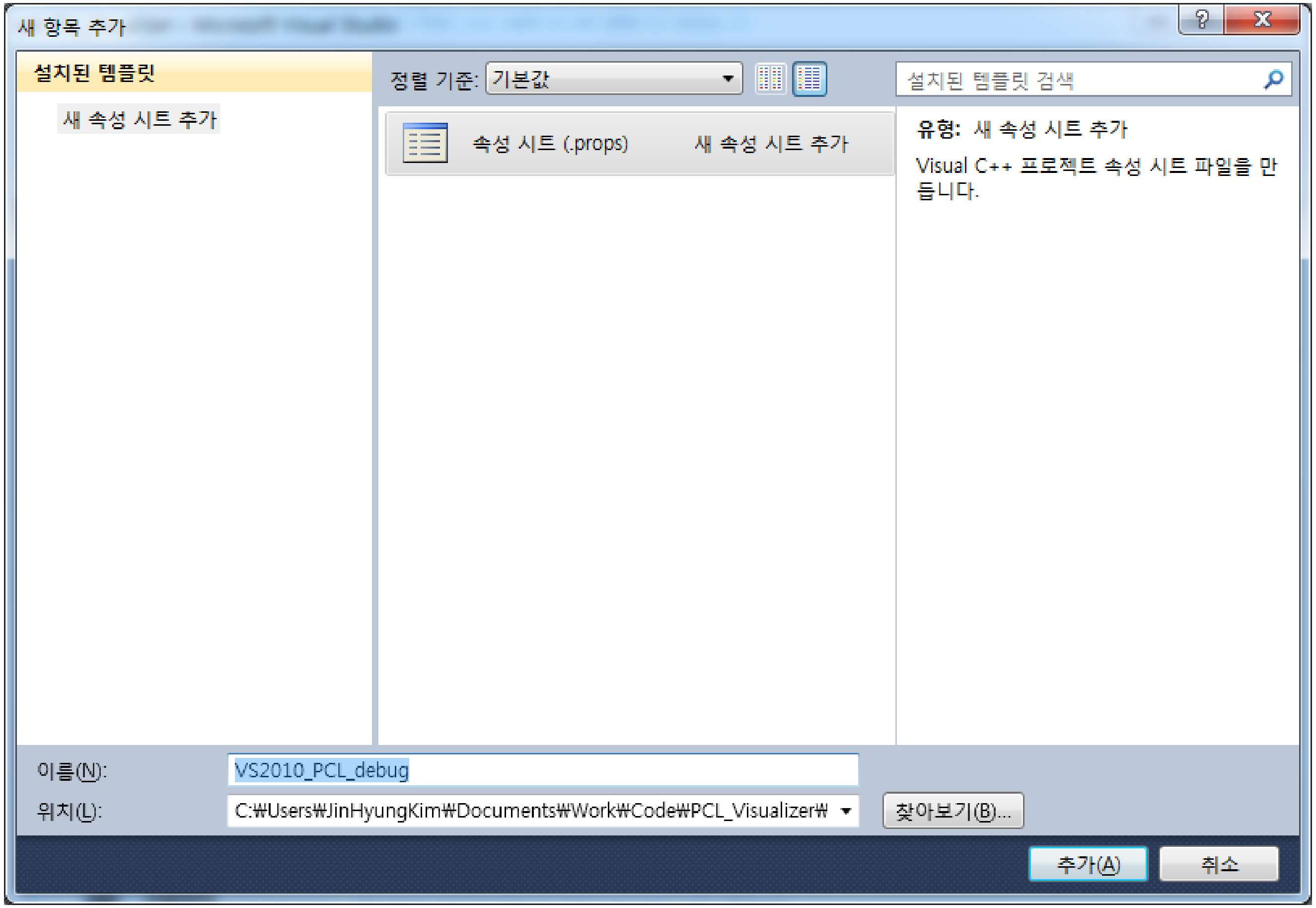
```
1>  모든 출력이 최신 상태입니다.
1>  PCL_Visualizer.vcxproj -> C:₩Users₩JinHyungKim₩Dc
1>FinalizeBuildStatus:
1>  "Debug₩PCL_Visualizer.unsuccessfulbuild" 파일을 
1>  "Debug₩PCL_Visualizer.lastbuildstate"에 연결(touc
1>
1>빌드했습니다.
1>
1>경과 시간: 00:00:13.64
========== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

(이름)	Debug
플랫폼 이름	Win32

(이름)
프로젝트 구성의 표시 이름입니다.

Setting develop environment

- Include, library, binary directory & additional dependency



유형: 새 속성 시트 추가

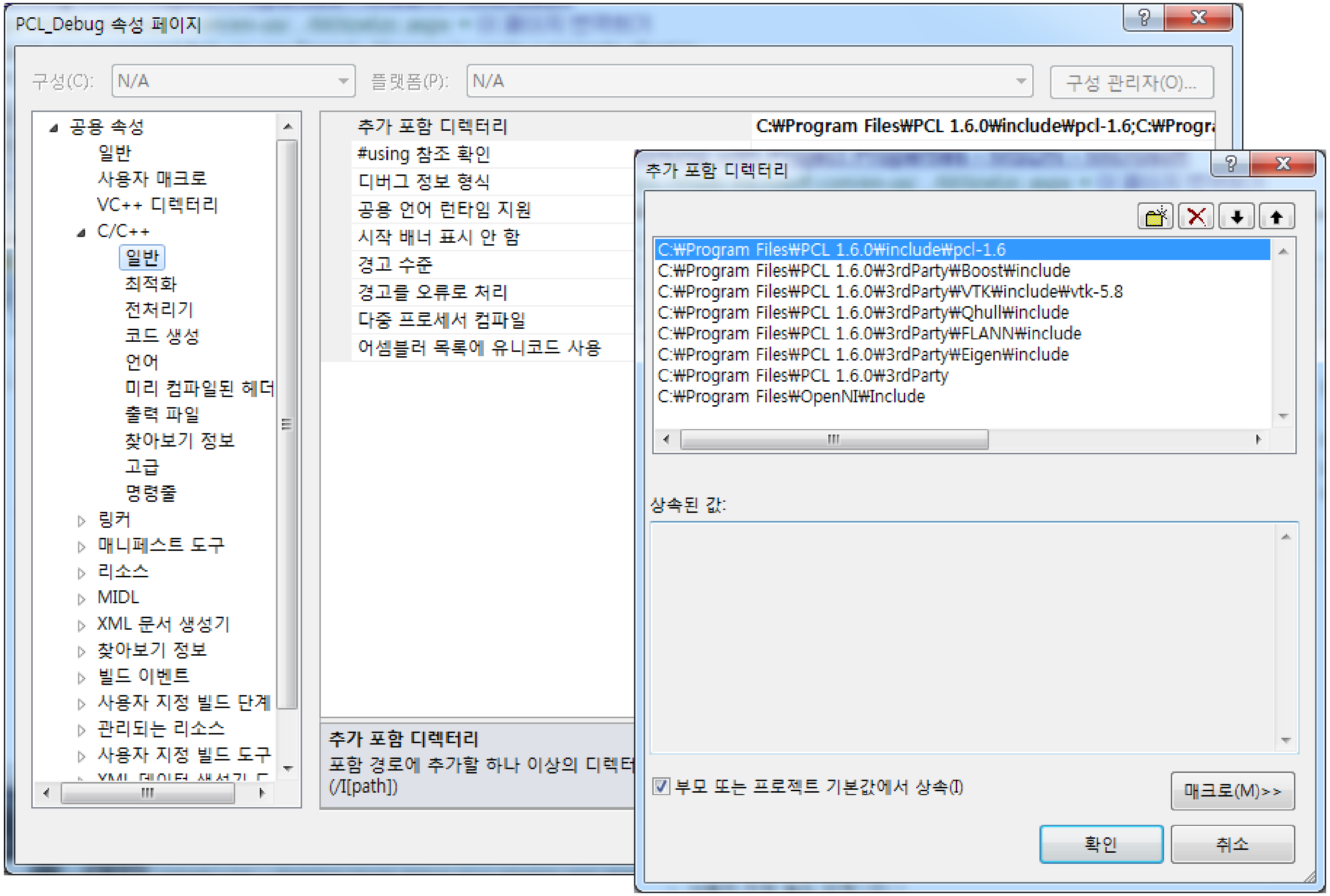
Visual C++ 프로젝트 속성 시트 파일을 만듭니다.

이름(N): VS2010_PCL_debug

위치(L): C:₩Users₩JinHyungKim₩Documents₩Work₩Code₩PCL_Visualizer₩

Setting develop environment

- Include, library, binary directory & additional dependency



PCL_Debug 속성 페이지

구성(C): N/A 플랫폼(P): N/A

추가 포함 디렉터리: C:₩Program Files₩PCL 1.6.0₩include₩pcl-1.6;C:₩Progra

추가 포함 디렉터리
포함 경로에 추가할 하나 이상의 디렉터
(/I[path])

추가 포함 디렉터리

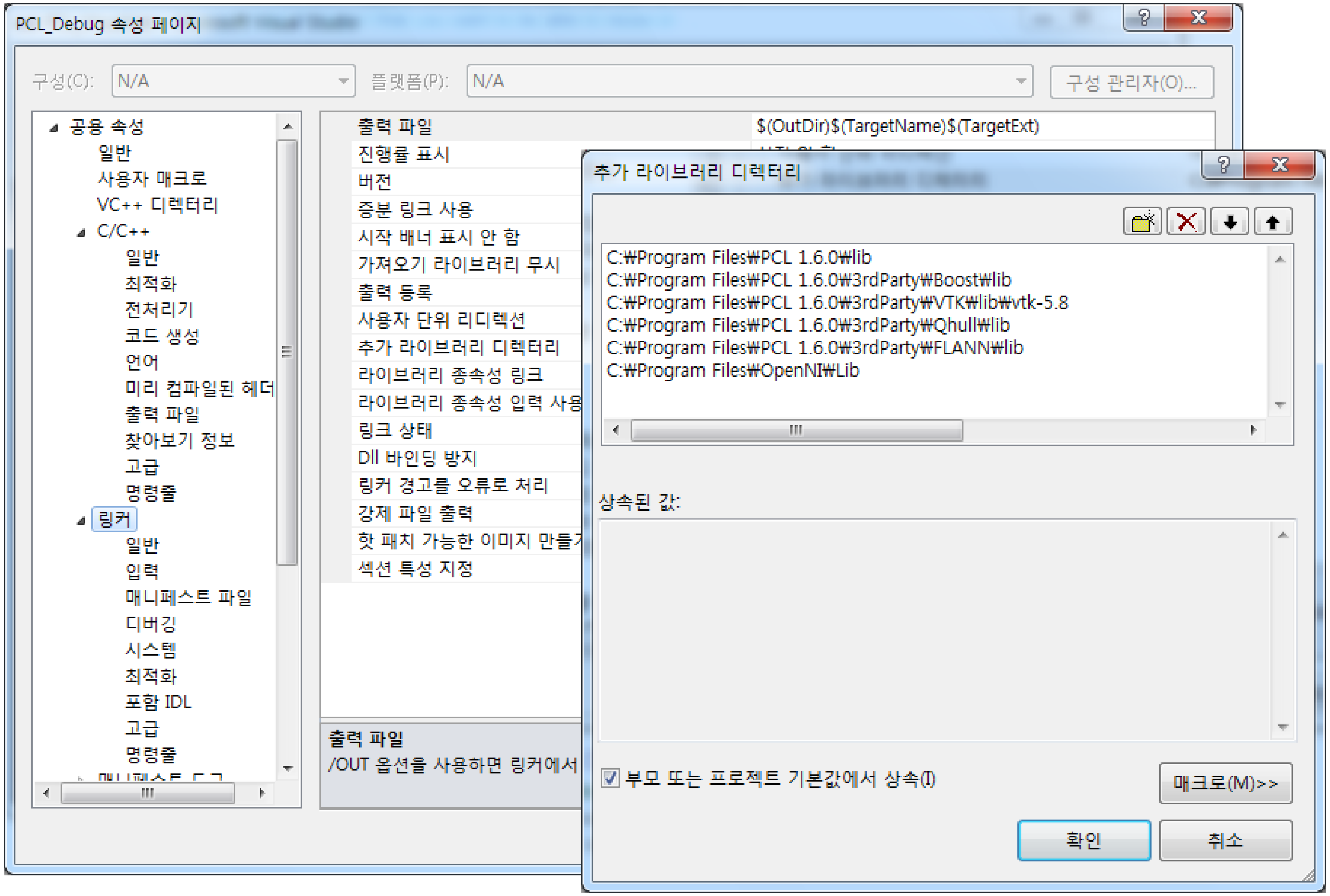
```
C:₩Program Files₩PCL 1.6.0₩include₩pcl-1.6
C:₩Program Files₩PCL 1.6.0₩3rdParty₩Boost₩include
C:₩Program Files₩PCL 1.6.0₩3rdParty₩VTK₩include₩vtk-5.8
C:₩Program Files₩PCL 1.6.0₩3rdParty₩Qhull₩include
C:₩Program Files₩PCL 1.6.0₩3rdParty₩FLANN₩include
C:₩Program Files₩PCL 1.6.0₩3rdParty₩Eigen₩include
C:₩Program Files₩PCL 1.6.0₩3rdParty
C:₩Program Files₩OpenNI₩Include
```

상속된 값:

☑ 부모 또는 프로젝트 기본값에서 상속(I)

Setting develop environment

- Include, library, binary directory & additional dependency



PCL_Debug 속성 페이지

구성(C): N/A 플랫폼(P): N/A

출력 파일: $(OutDir)$(TargetName)$(TargetExt)

추가 라이브러리 디렉터리

C:₩Program Files₩PCL 1.6.0₩lib
C:₩Program Files₩PCL 1.6.0₩3rdParty₩Boost₩lib
C:₩Program Files₩PCL 1.6.0₩3rdParty₩VTK₩lib₩vtk-5.8
C:₩Program Files₩PCL 1.6.0₩3rdParty₩Qhull₩lib
C:₩Program Files₩PCL 1.6.0₩3rdParty₩FLANN₩lib
C:₩Program Files₩OpenNI₩Lib

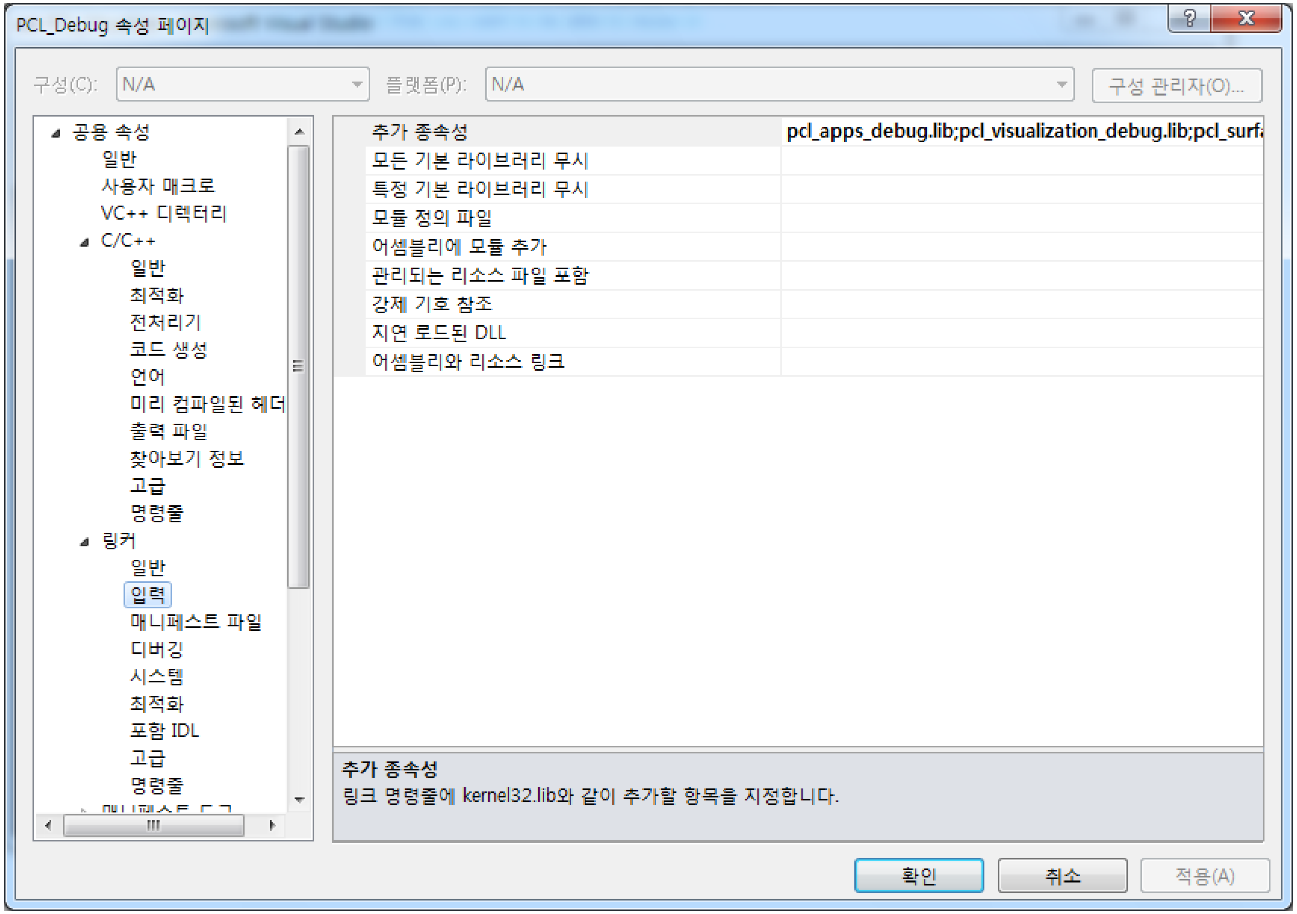
상속된 값:

☑ 부모 또는 프로젝트 기본값에서 상속(I)

출력 파일
/OUT 옵션을 사용하면 링커에서

Setting develop environment

- Include, library, binary directory & additional dependency



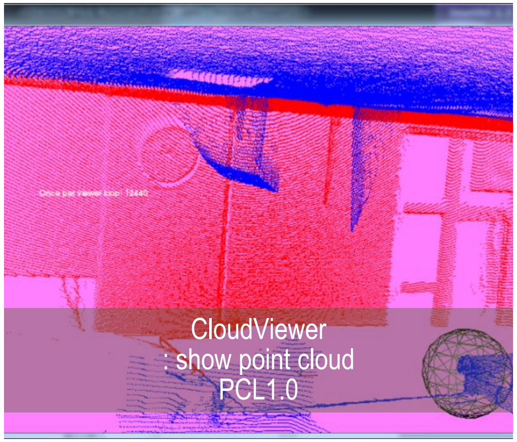
추가 종속성	pcl_apps_debug.lib;pcl_visualization_debug.lib;pcl_surf
모든 기본 라이브러리 무시	
특정 기본 라이브러리 무시	
모듈 정의 파일	
어셈블리에 모듈 추가	
관리되는 리소스 파일 포함	
강제 기호 참조	
지연 로드된 DLL	
어셈블리와 리소스 링크	

추가 종속성
링크 명령줄에 kernel32.lib와 같이 추가할 항목을 지정합니다.

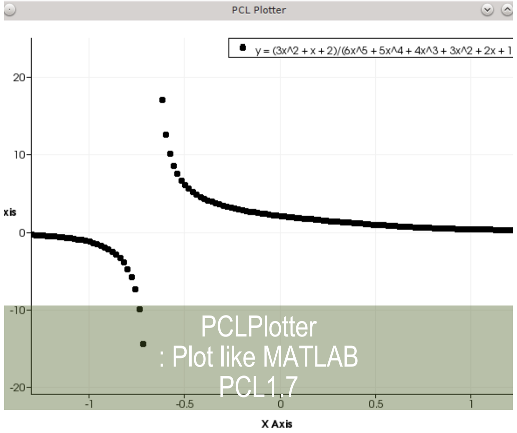
Setting develop environment

Repeat same procedure for Release mode

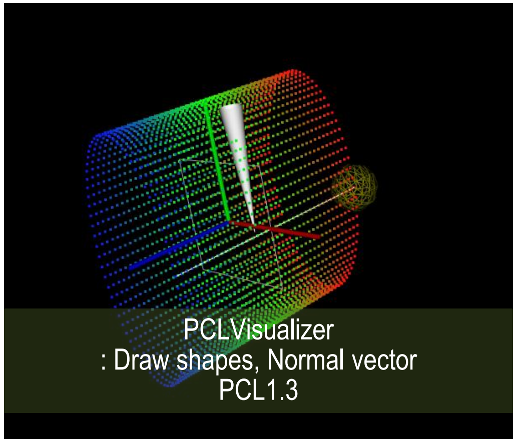
Visualization



CloudViewer
: show point cloud
PCL1.0

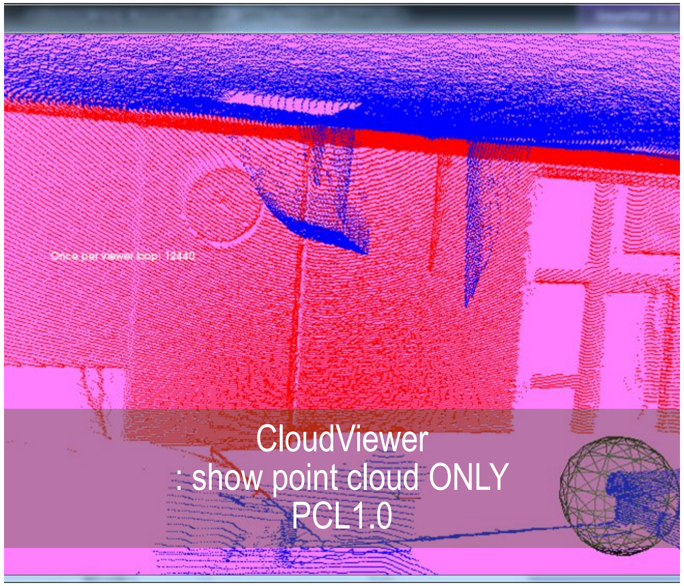


PCLPlotter
: Plot like MATLAB
PCL1.7

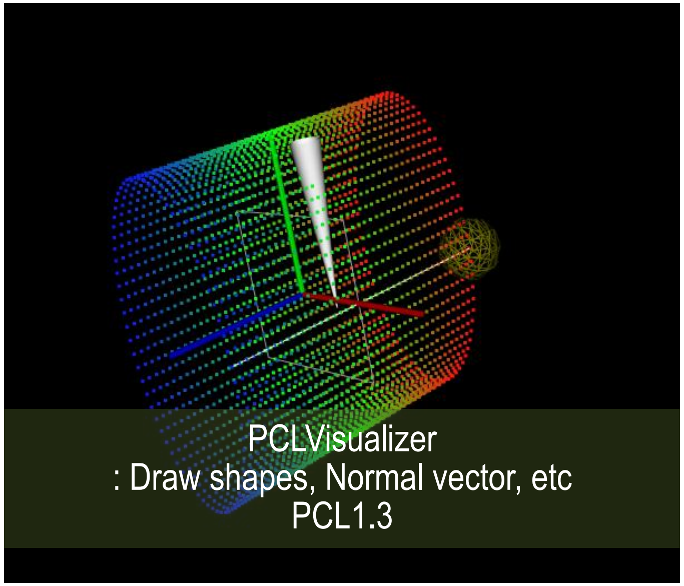


PCLVisualizer
: Draw shapes, Normal vector
PCL1.3

Visualization



CloudViewer
: show point cloud ONLY
PCL1.0

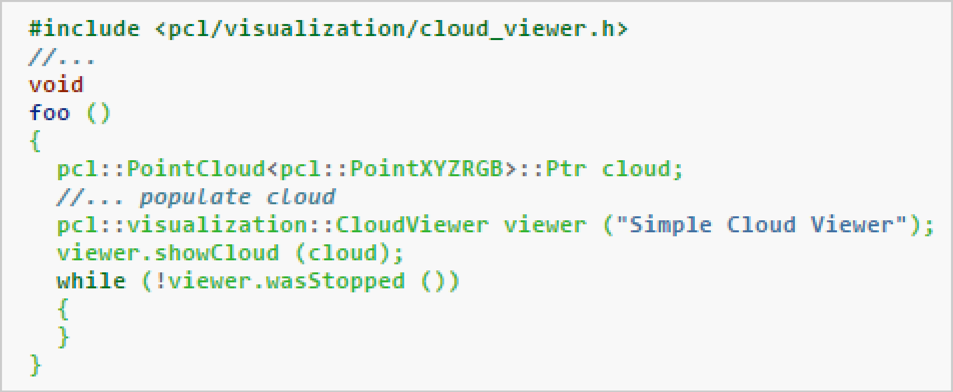


PCLVisualizer
: Draw shapes, Normal vector, etc
PCL1.3

CloudViewer	PCLVisualizer
Simple coding	Complex coding
UI Callback function	UI Callback function
Using thread, PCL Visualizer function can be used	Can Draw shape, Render surface or wire...
	Multiple viewports

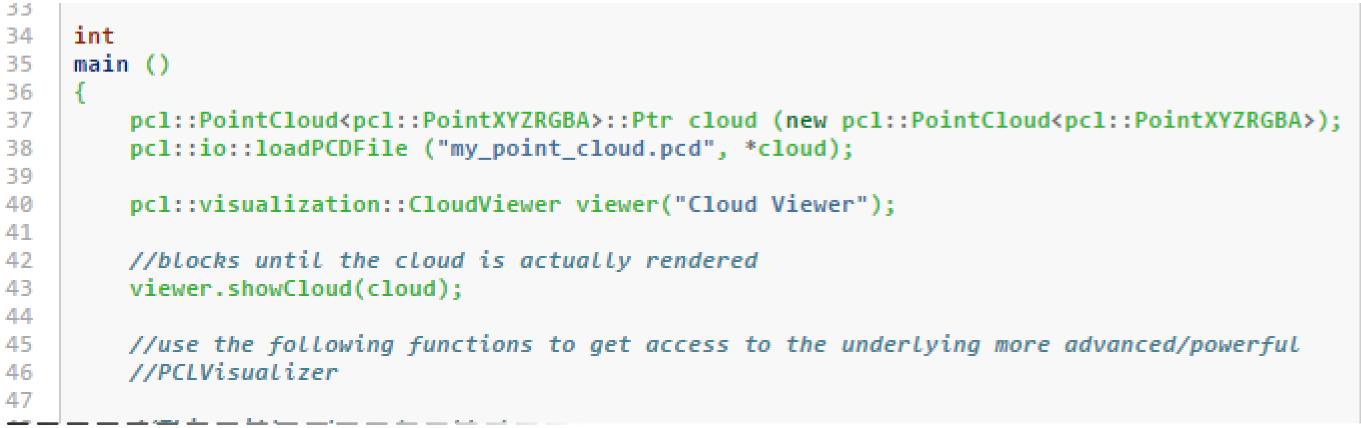
CloudViewer

- Simple cloud visualization



```
#include <pcl/visualization/cloud_viewer.h>
//...
void
foo ()
{
  pcl::PointCloud<pcl::PointXYZRGB>::Ptr cloud;
  //... populate cloud
  pcl::visualization::CloudViewer viewer ("Simple Cloud Viewer");
  viewer.showCloud (cloud);
  while (!viewer.wasStopped ())
  {
  }
}
```

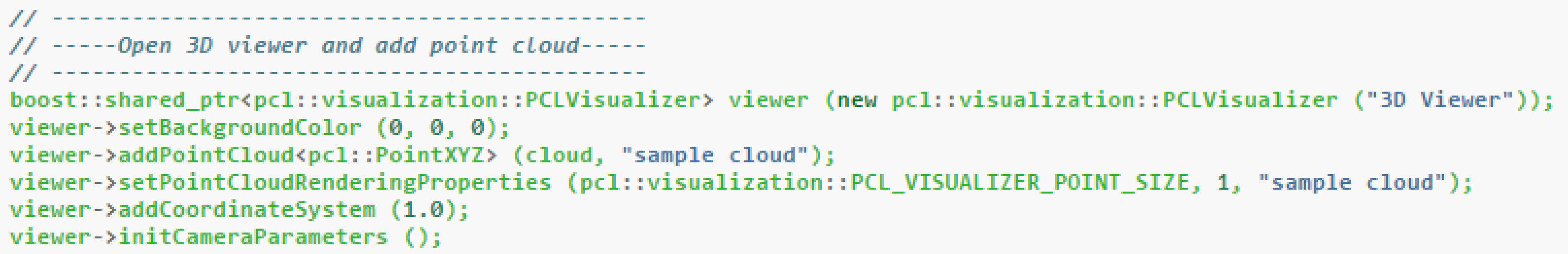
- Using PCLVisualizer function on thread



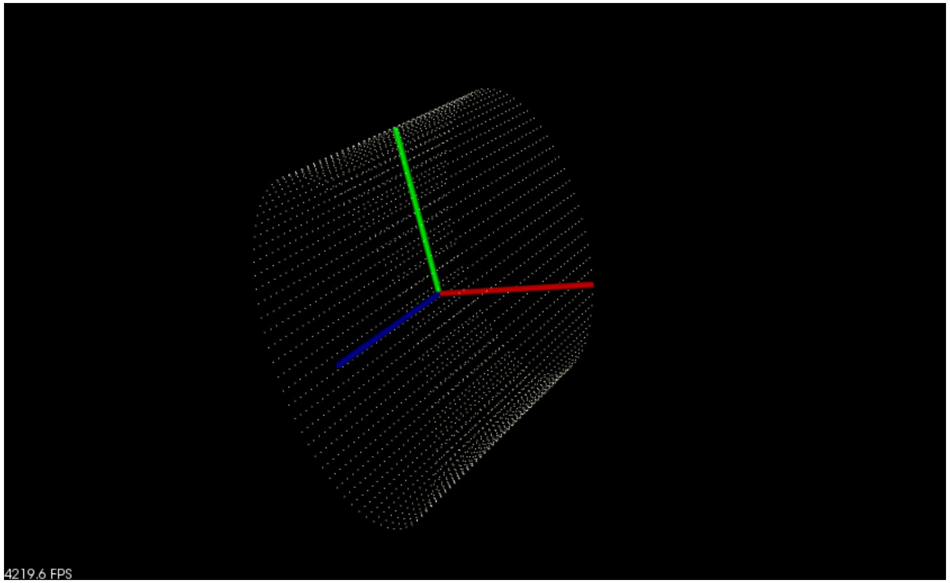
```
int
main ()
{
    pcl::PointCloud<pcl::PointXYZRGBA>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZRGBA>);
    pcl::io::loadPCDFile ("my_point_cloud.pcd", *cloud);

    pcl::visualization::CloudViewer viewer("Cloud Viewer");

    //blocks until the cloud is actually rendered
    viewer.showCloud(cloud);

    //use the following functions to get access to the underlying more advanced/powerful
    //PCLVisualizer

```

PCLVisualizer

- Simple PCLVisualizer code



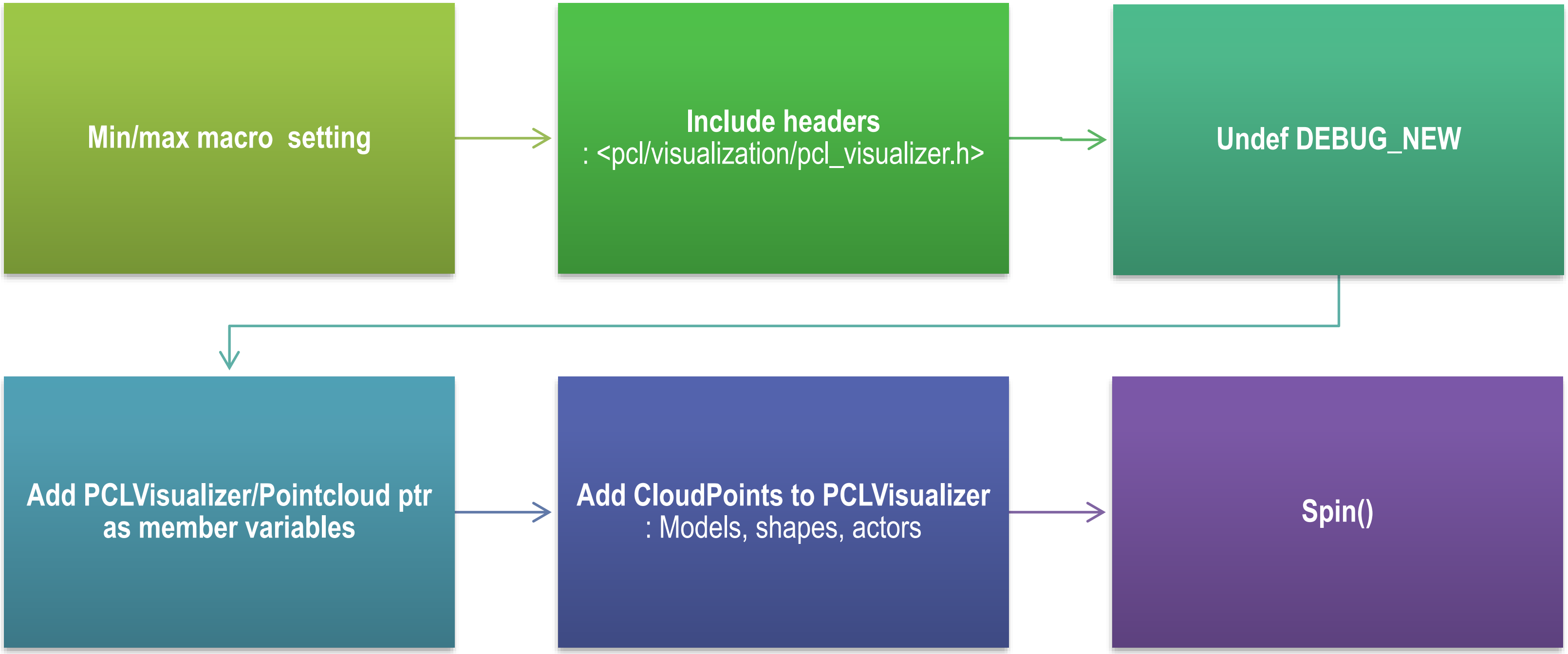
```
// --------------------------------------------
// -----Open 3D viewer and add point cloud-----
// --------------------------------------------
boost::shared_ptr<pcl::visualization::PCLVisualizer> viewer (new pcl::visualization::PCLVisualizer ("3D Viewer"));
viewer->setBackgroundColor (0, 0, 0);
viewer->addPointCloud<pcl::PointXYZ> (cloud, "sample cloud");
viewer->setPointCloudRenderingProperties (pcl::visualization::PCL_VISUALIZER_POINT_SIZE, 1, "sample cloud");
viewer->addCoordinateSystem (1.0);
viewer->initCameraParameters ();
```



4219.6 FPS

PCLVisualizer on MFC Dialog

- General Pipeline of PCLVisualizer on MFC Dialog



Min/max macro setting

Include headers
: <pcl/visualization/pcl_visualizer.h>

Undef DEBUG_NEW

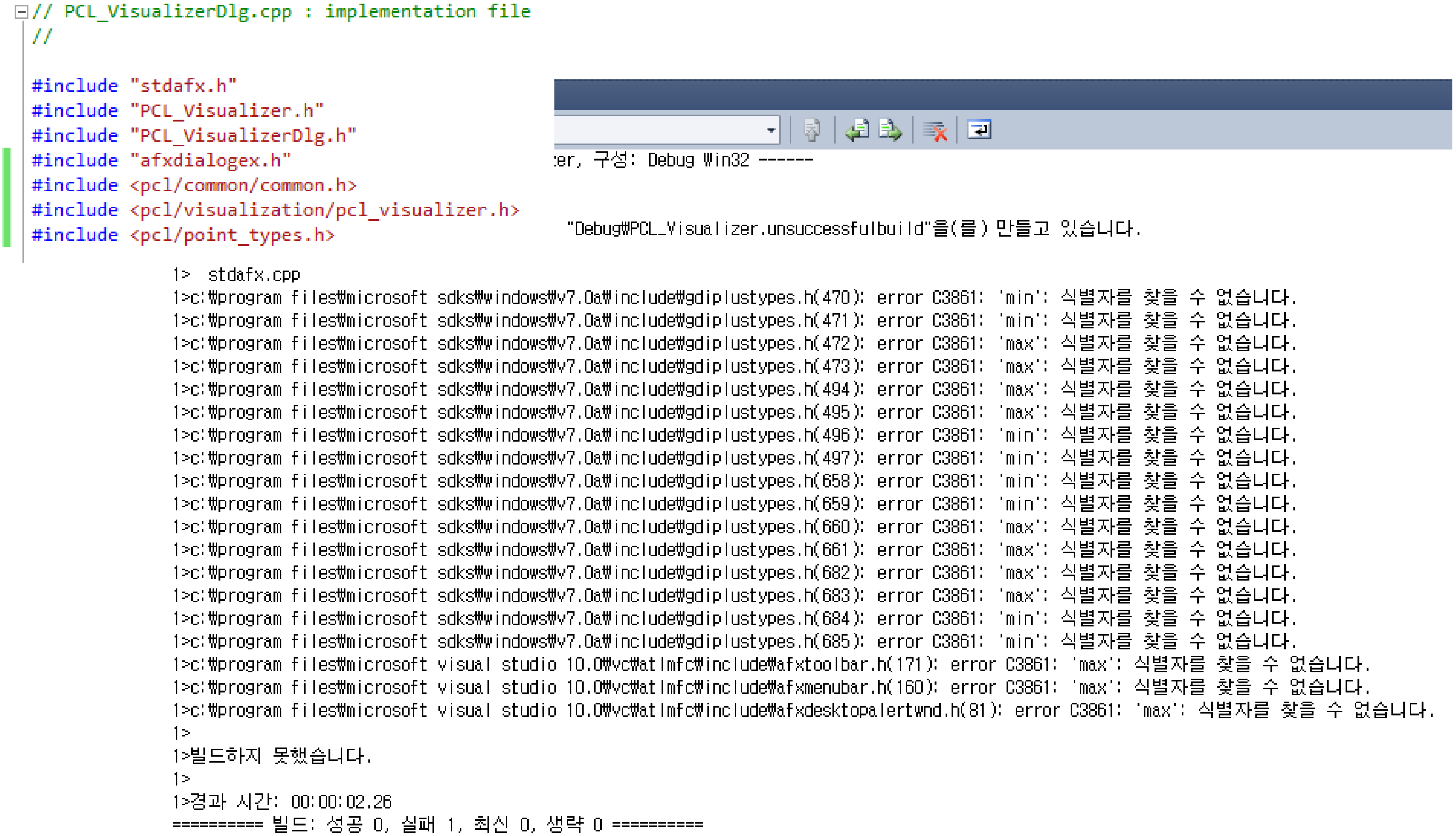
Add PCLVisualizer/Pointcloud ptr as member variables

Add CloudPoints to PCLVisualizer
: Models, shapes, actors

Spin()

PCLVisualizer on MFC Dialog

- Min/max macro prioblem

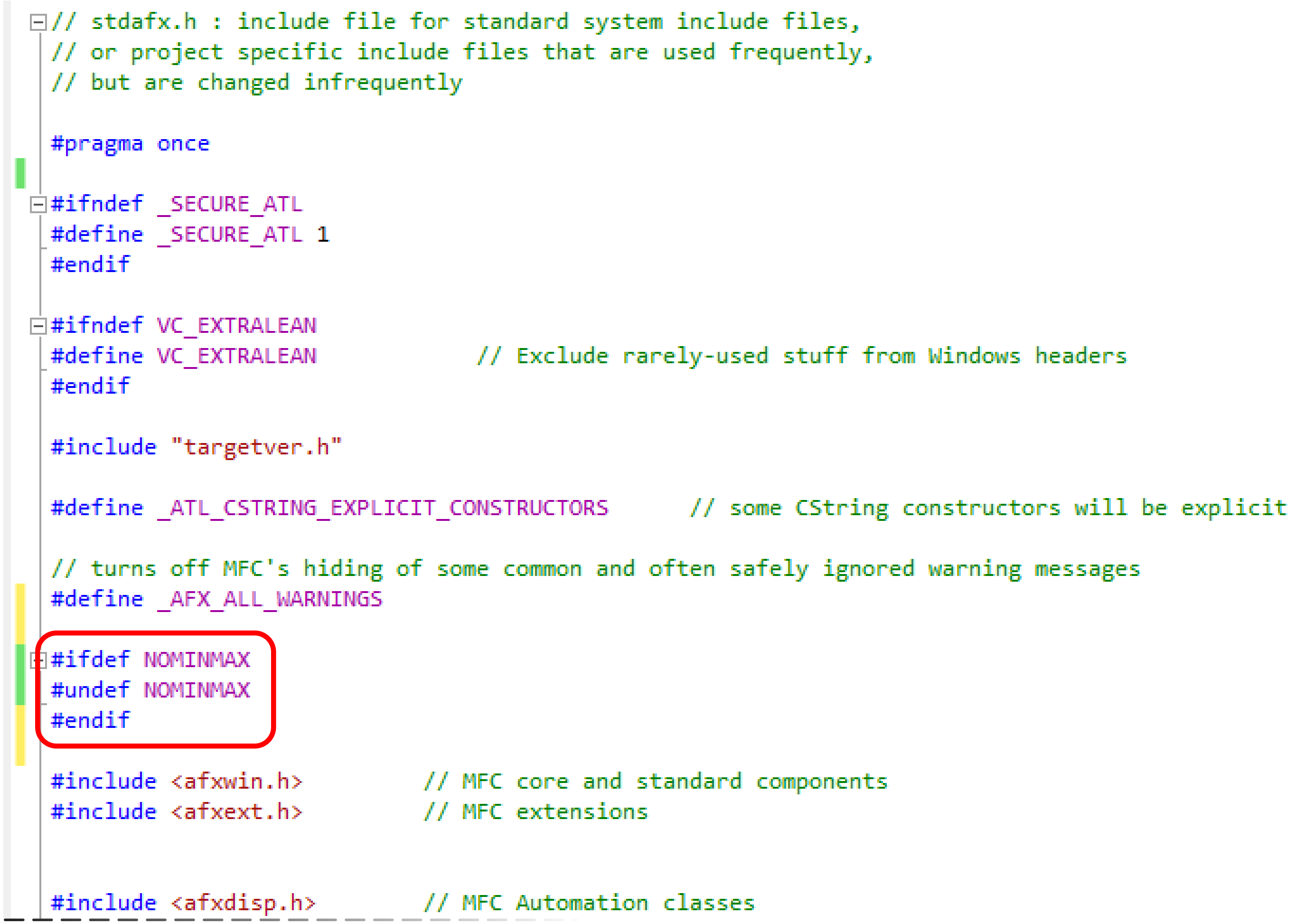


```
// PCL_VisualizerDlg.cpp : implementation file
//

#include "stdafx.h"
#include "PCL_Visualizer.h"
#include "PCL_VisualizerDlg.h"
#include "afxdialogex.h"
#include <pcl/common/common.h>
#include <pcl/visualization/pcl_visualizer.h>
#include <pcl/point_types.h>
```

```
:er, 구성: Debug Win32 ------

 "Debug₩PCL_Visualizer.unsuccessfulbuild"을(를) 만들고 있습니다.
1>  stdafx.cpp
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(470): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(471): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(472): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(473): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(494): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(495): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(496): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(497): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(658): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(659): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(660): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(661): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(682): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(683): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(684): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft sdks₩windows₩v7.0a₩include₩gdiplustypes.h(685): error C3861: 'min': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft visual studio 10.0₩vc₩atlmfc₩include₩afxtoolbar.h(171): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft visual studio 10.0₩vc₩atlmfc₩include₩afxmenubar.h(160): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>c:₩program files₩microsoft visual studio 10.0₩vc₩atlmfc₩include₩afxdesktopalertwnd.h(81): error C3861: 'max': 식별자를 찾을 수 없습니다.
1>
1>빌드하지 못했습니다.
1>
1>경과 시간: 00:00:02.26
========== 빌드: 성공 0, 실패 1, 최신 0, 생략 0 ==========
```

PCLVisualizer on MFC Dialog

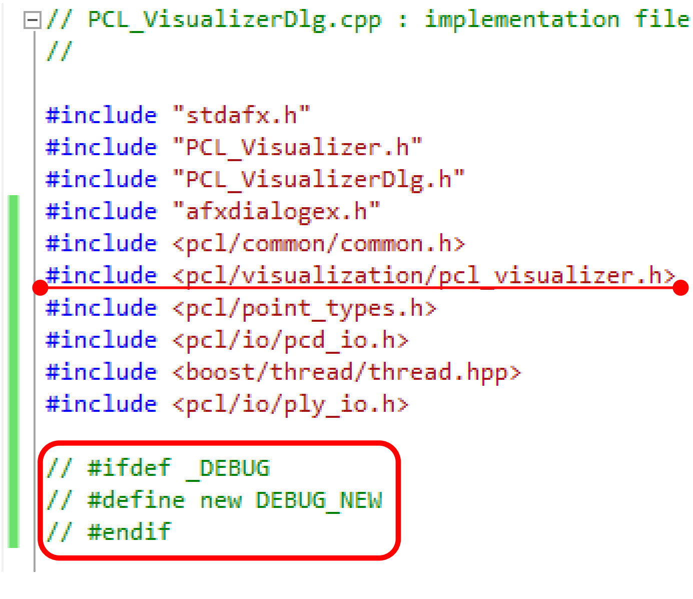
- Do something on Stdafx.h



```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently,
// but are changed infrequently

#pragma once

#ifndef _SECURE_ATL
#define _SECURE_ATL 1
#endif

#ifndef VC_EXTRALEAN
#define VC_EXTRALEAN            // Exclude rarely-used stuff from Windows headers
#endif

#include "targetver.h"

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS      // some CString constructors will be explicit

// turns off MFC's hiding of some common and often safely ignored warning messages
#define _AFX_ALL_WARNINGS

#ifdef NOMINMAX
#undef NOMINMAX
#endif

#include <afxwin.h>         // MFC core and standard components
#include <afxext.h>         // MFC extensions


#include <afxdisp.h>        // MFC Automation classes
```

PCLVisualizer on MFC Dialog

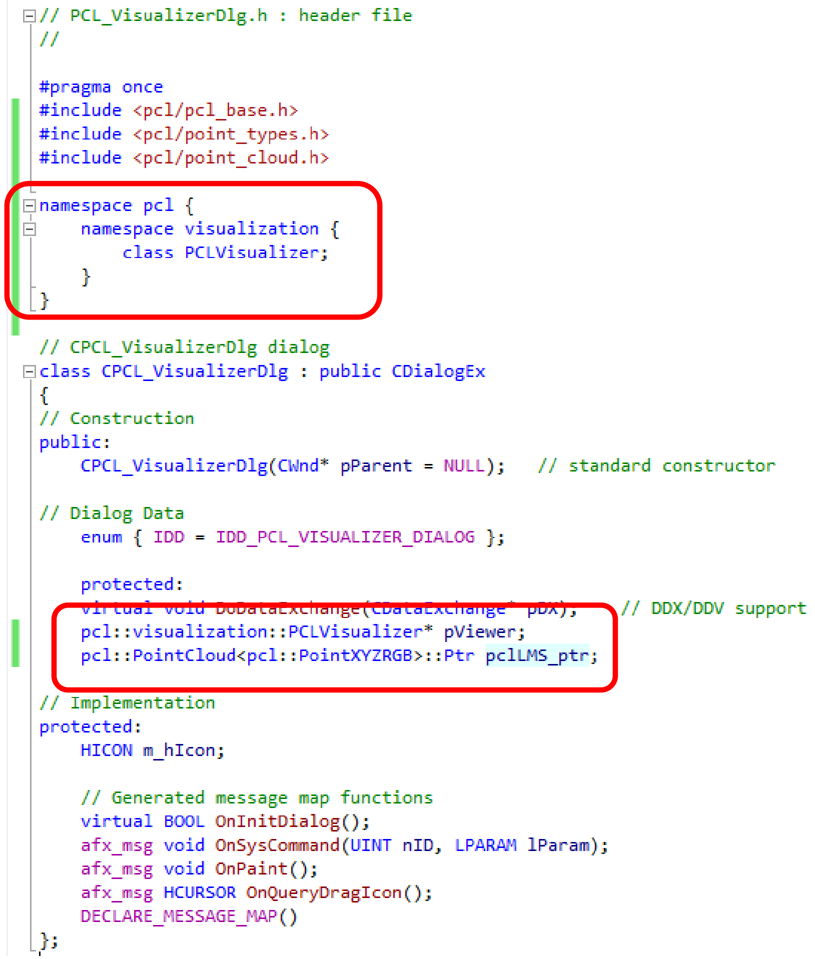
- Add header & undefine DEBUG_NEW



```
// PCL_VisualizerDlg.cpp : implementation file
//

#include "stdafx.h"
#include "PCL_Visualizer.h"
#include "PCL_VisualizerDlg.h"
#include "afxdialogex.h"
#include <pcl/common/common.h>
#include <pcl/visualization/pcl_visualizer.h>
#include <pcl/point_types.h>
#include <pcl/io/pcd_io.h>
#include <boost/thread/thread.hpp>
#include <pcl/io/ply_io.h>

// #ifdef _DEBUG
// #define new DEBUG_NEW
// #endif
```

PCLVisualizer on MFC Dialog

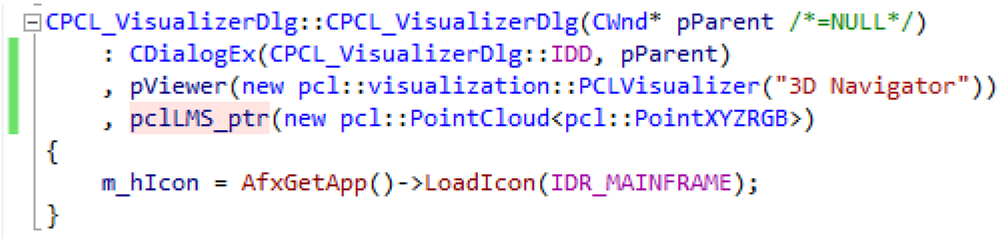
- Add pointers as member variables of dialog class



```
// PCL_VisualizerDlg.h : header file
//

#pragma once
#include <pcl/pcl_base.h>
#include <pcl/point_types.h>
#include <pcl/point_cloud.h>

namespace pcl {
    namespace visualization {
        class PCLVisualizer;
    }
}

// CPCL_VisualizerDlg dialog
class CPCL_VisualizerDlg : public CDialogEx
{
// Construction
public:
    CPCL_VisualizerDlg(CWnd* pParent = NULL);   // standard constructor

// Dialog Data
    enum { IDD = IDD_PCL_VISUALIZER_DIALOG };

    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    pcl::visualization::PCLVisualizer* pViewer;
    pcl::PointCloud<pcl::PointXYZRGB>::Ptr pclLMS_ptr;

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
};
```

PCLVisualizer on MFC Dialog

- Instance creation



```
CPCL_VisualizerDlg::CPCL_VisualizerDlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(CPCL_VisualizerDlg::IDD, pParent)
    , pViewer(new pcl::visualization::PCLVisualizer("3D Navigator"))
    , pclLMS_ptr(new pcl::PointCloud<pcl::PointXYZRGB>)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
```

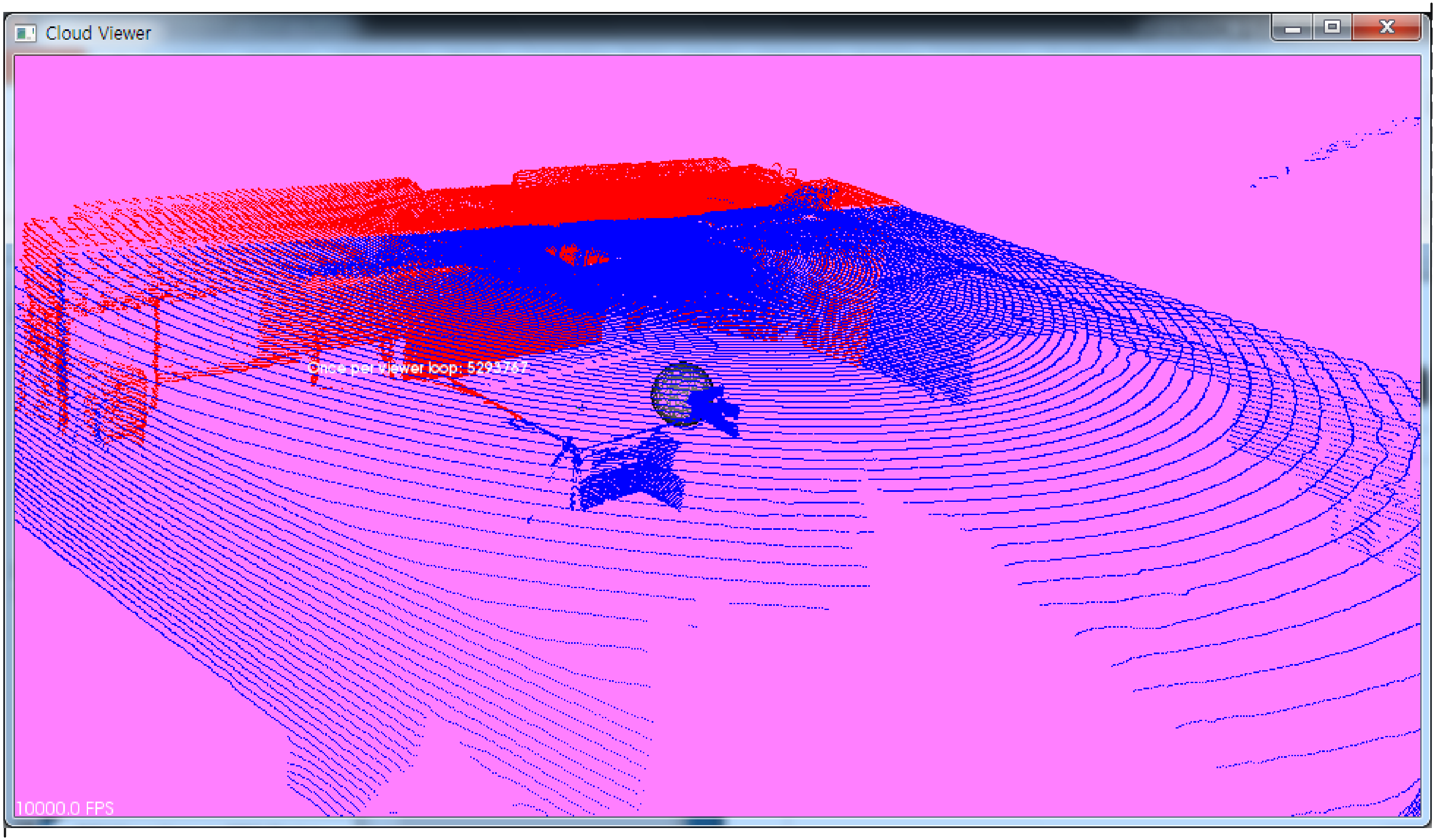
- Useful methods of PCLVisualizer
  - bool addPointCloud (const typename pcl::PointCloud< PointT >::ConstPtr &cloud, const std::string &id="cloud", int viewport=0)
  - bool updatePointCloud (const pcl::PointCloud< pcl::PointXYZ >::ConstPtr &cloud, const std::string &id="cloud")
  - addText (const std::string &text, int xpos, int ypos, int fontsize, double r, double g, double b, const std::string &id="", int viewport=0)
  - bool updatePointCloudPose (const std::string &id, const Eigen::Affine3f &pose)
  - void addCoordinateSystem (double scale, const Eigen::Affine3f &t, const std::string &id="reference", int viewport=0)

Eigen::Affine3f pcl::getTransformation(float x, float y, float z, float roll, float pitch, float yaw)

Eigen::Affine3f ➔ 4x4 affine matrix (4th row ➔ [0 0 0 1])

  - bool updateCoordinateSystemPose (const std::string &id, const Eigen::Affine3f &pose)

Program Demo



Once per viewer loop: 5293767

10000.0 FPS