Visualization Tool Kit

VTK & OpenCV

안재원

목차

- OpenCV 3.0
  - Installation

- VTK
  - Installation

- Visualization
  - *.ply file
  - Widget
  - Make a widget
  - Smart Pointer

01 OpenCV

- OpenCV?

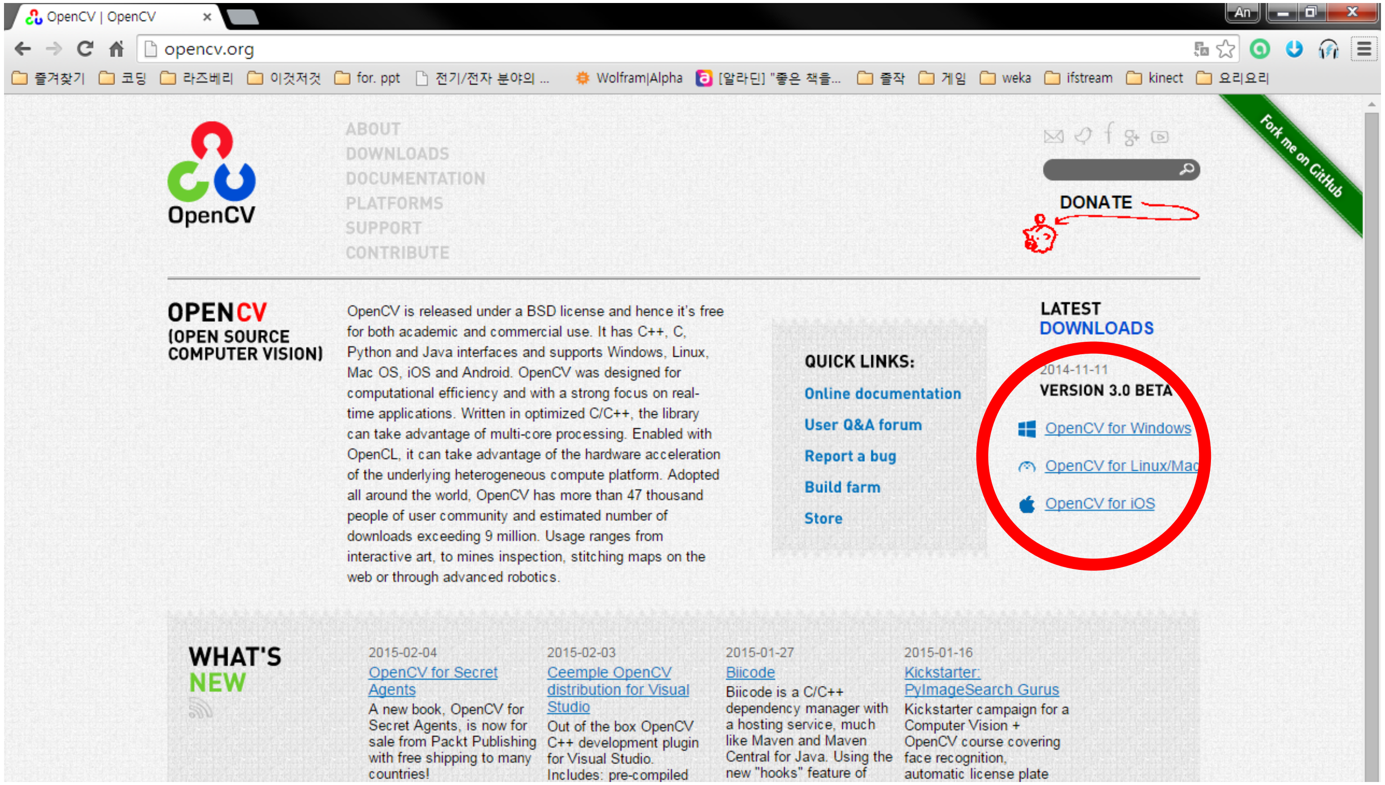
Open-source Computer Vision library

- 2,500+ Argorithms & Functions
- Real-time performance
- C, C++, Python, Java
- Windows, Linux, Mac OS, iOS, Android
- BSD License

Berkeley Software Distribution License
: 소스코드 공개의 의무가 없으며 상용 소프트웨어에서도 무제한 사용가능

01 Installation

- opencv.org



OPENCV (OPEN SOURCE COMPUTER VISION)

OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 9 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

QUICK LINKS:

- Online documentation
- User Q&A forum
- Report a bug
- Build farm
- Store

LATEST DOWNLOADS

2014-11-11
VERSION 3.0 BETA

- OpenCV for Windows
- OpenCV for Linux/Mac
- OpenCV for iOS

WHAT'S NEW

2015-02-04
OpenCV for Secret Agents
A new book, OpenCV for Secret Agents, is now for sale from Packt Publishing with free shipping to many countries!

2015-02-03
Ceemple OpenCV distribution for Visual Studio
Out of the box OpenCV C++ development plugin for Visual Studio. Includes: pre-compiled

2015-01-27
Biicode
Biicode is a C/C++ dependency manager with a hosting service, much like Maven and Maven Central for Java. Using the new "hooks" feature of

2015-01-16
Kickstarter: PyImageSearch Gurus
Kickstarter campaign for a Computer Vision + OpenCV course covering face recognition, automatic license plate

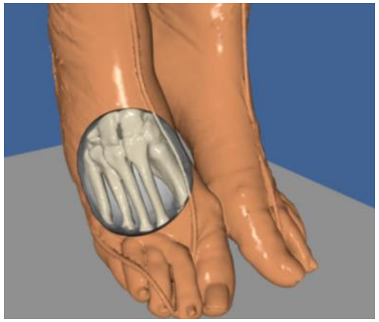
02 VTK

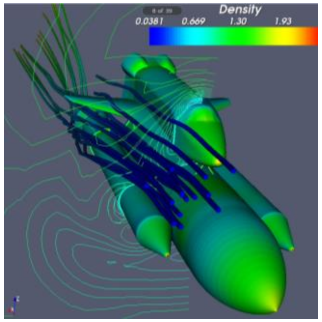
- VTK?

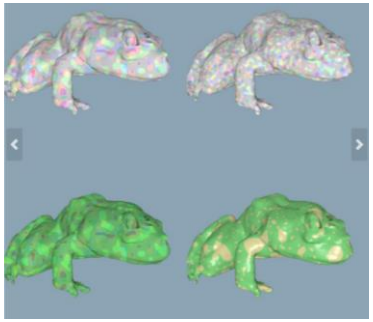
Visualization Tool Kit

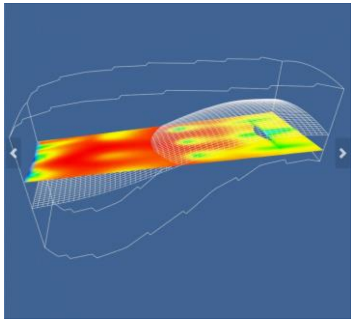
- For 3D computer graphics, modeling etc…
- C/C++, Python, Java
- BSD License

Berkeley Software Distribution License
: 소스코드 공개의 의무가 없으며 상용 소프트웨어에서도 무제한 사용가능



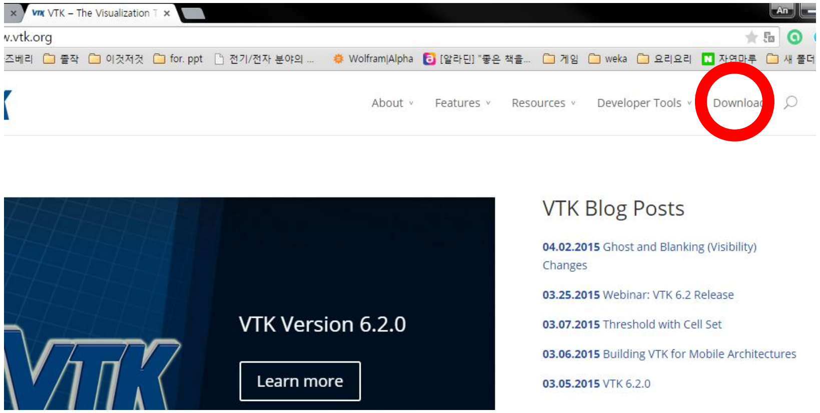






02 Installation

- vtk.org



VTK Blog Posts

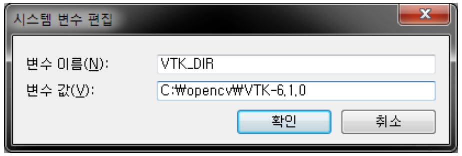
04.02.2015 Ghost and Blanking (Visibility) Changes

03.25.2015 Webinar: VTK 6.2 Release

03.07.2015 Threshold with Cell Set

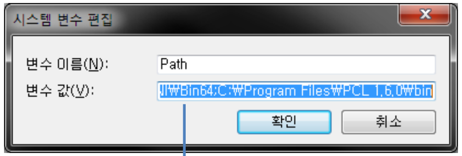
03.06.2015 Building VTK for Mobile Architectures

03.05.2015 VTK 6.2.0



변수 이름(N): VTK_DIR

변수 값(V): C:₩opencv₩VTK-6.1.0



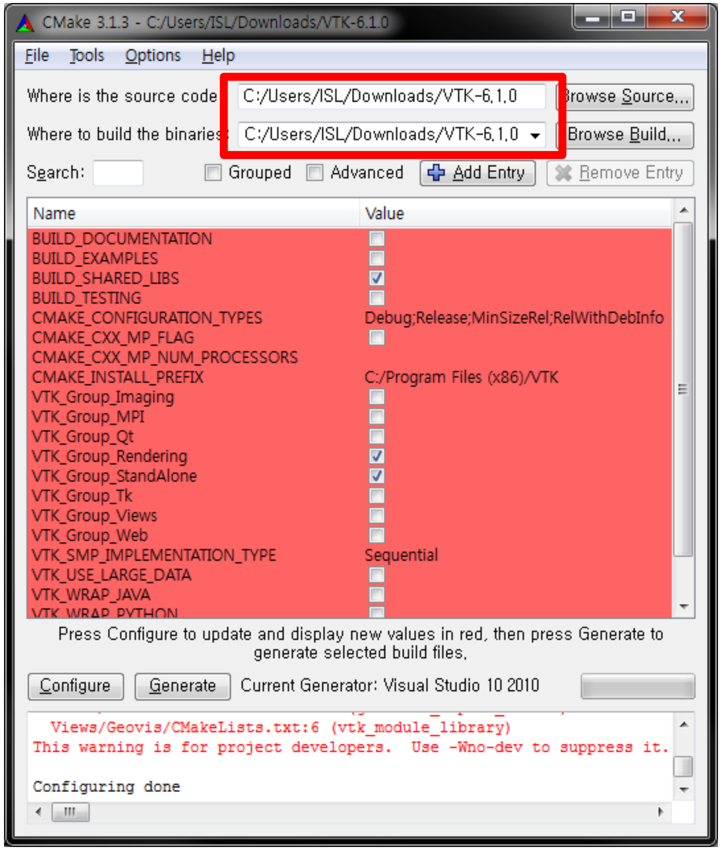
변수 이름(N): Path

변수 값(V): ..l₩Bin64;C:₩Program Files₩PCL 1.6.0₩bin

;%VTK_DIR%\bin\Debug;

;%VTK_DIR%\bin\Release;

- CMake



Where is the source code: C:/Users/ISL/Downloads/VTK-6.1.0

Where to build the binaries: C:/Users/ISL/Downloads/VTK-6.1.0

Search:

□ Grouped □ Advanced

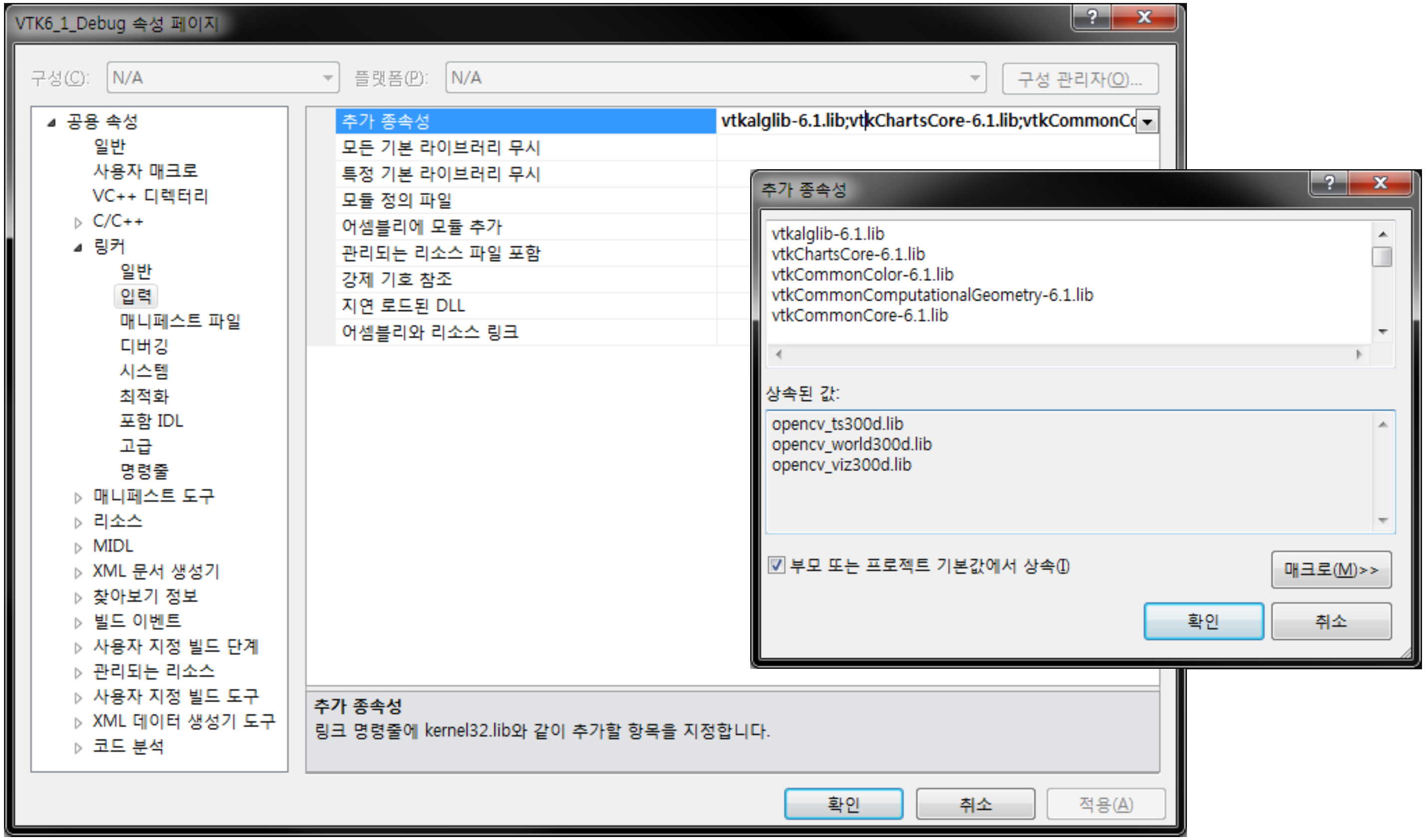
Name	Value
BUILD_DOCUMENTATION	□
BUILD_EXAMPLES	□
BUILD_SHARED_LIBS	☑
BUILD_TESTING	□
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_CXX_MP_FLAG	□
CMAKE_CXX_MP_NUM_PROCESSORS	
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/VTK
VTK_Group_Imaging	□
VTK_Group_MPI	□
VTK_Group_Qt	□
VTK_Group_Rendering	☑
VTK_Group_StandAlone	☑
VTK_Group_Tk	□
VTK_Group_Views	□
VTK_Group_Web	□
VTK_SMP_IMPLEMENTATION_TYPE	Sequential
VTK_USE_LARGE_DATA	□
VTK_WRAP_JAVA	□
VTK_WRAP_PYTHON	□

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 10 2010

```
  Views/Geovis/CMakeLists.txt:6 (vtk_module_library)
This warning is for project developers.  Use -Wno-dev to suppress it.

Configuring done
```

02 Installation



추가 종속성	vtkalglib-6.1.lib;vtkChartsCore-6.1.lib;vtkCommonCo
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	

vtkalglib-6.1.lib
vtkChartsCore-6.1.lib
vtkCommonColor-6.1.lib
vtkCommonComputationalGeometry-6.1.lib
vtkCommonCore-6.1.lib

상속된 값:

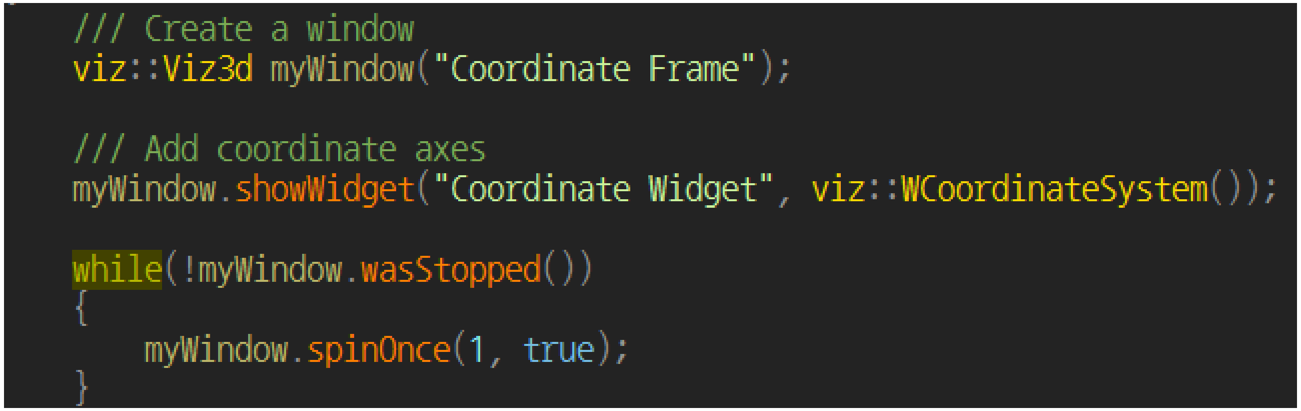
opencv_ts300d.lib
opencv_world300d.lib
opencv_viz300d.lib

☑ 부모 또는 프로젝트 기본값에서 상속(I)

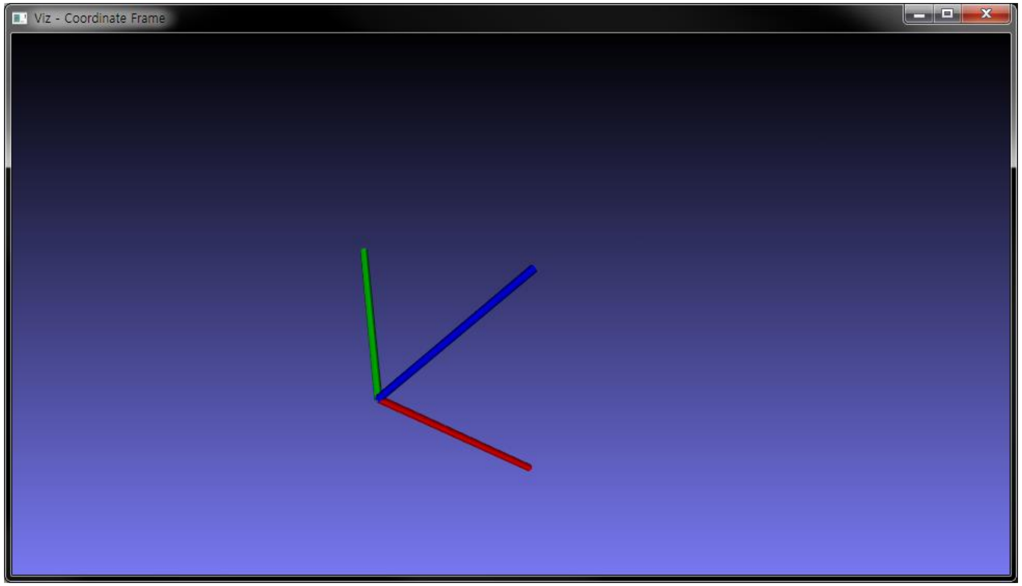
추가 종속성
링크 명령줄에 kernel32.lib와 같이 추가할 항목을 지정합니다.

02 Installation

※OpenCV 3.0.0-dev documentation page 참고(http://docs.opencv.org/trunk/index.html)



```
/// Create a window
viz::Viz3d myWindow("Coordinate Frame");

/// Add coordinate axes
myWindow.showWidget("Coordinate Widget", viz::WCoordinateSystem());

while(!myWindow.wasStopped())
{
    myWindow.spinOnce(1, true);
}
```



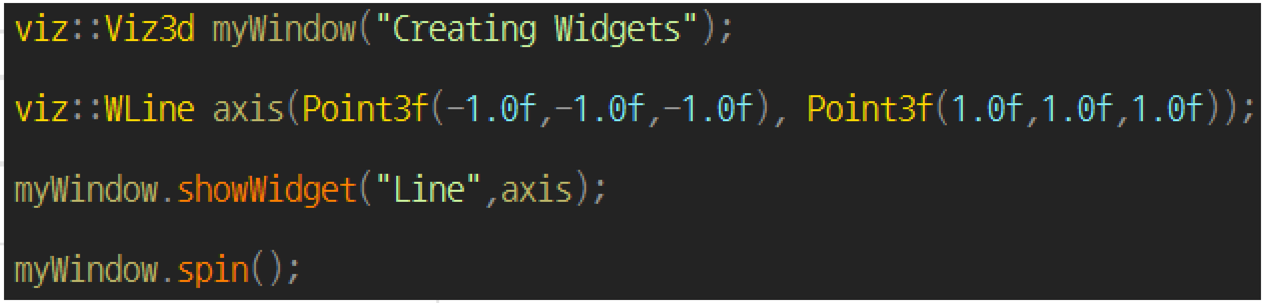
03 Visualization

```
ply
format [형식] – ASCII or Binary
comment [파일에 관한 설명]
element vertex [N]
property [자료형] x
property [자료형] y
property [자료형] z
element face [N]
property list uint8 int32 vertex_indices
end_header
0 0 0
0 1 0
……
4 0 1 2 3
4 0 4 5 1
……
```

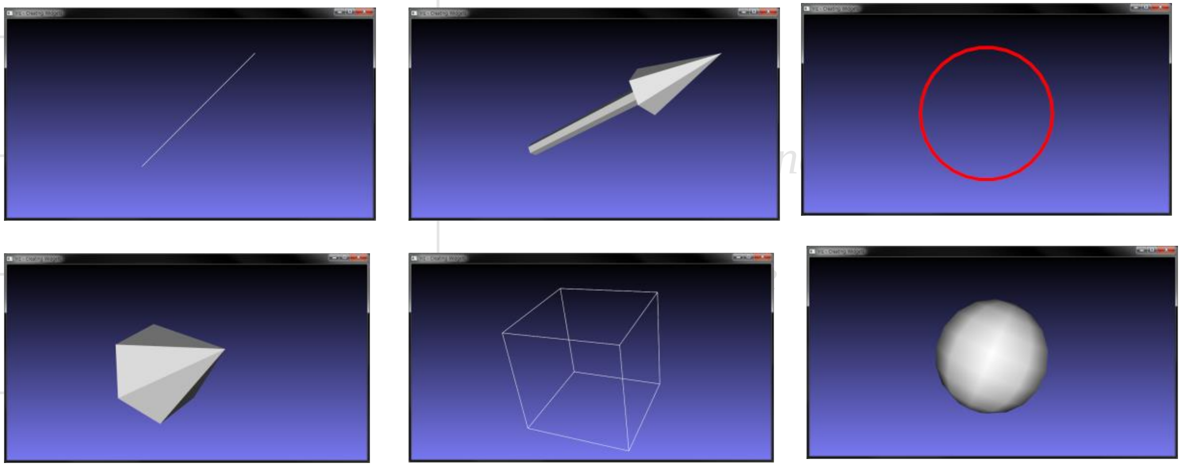
```
ply
format ascii 1.0
element vertex 8
property float32 x
property float32 y
property float32 z
element face 6
property list uint8 int32 vertex_index
end_header
0 0 0
0 0 1
0 1 1
0 1 0
1 0 0
1 0 1
1 1 1
1 1 0
4 0 1 2 3
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
```

03

Visualization

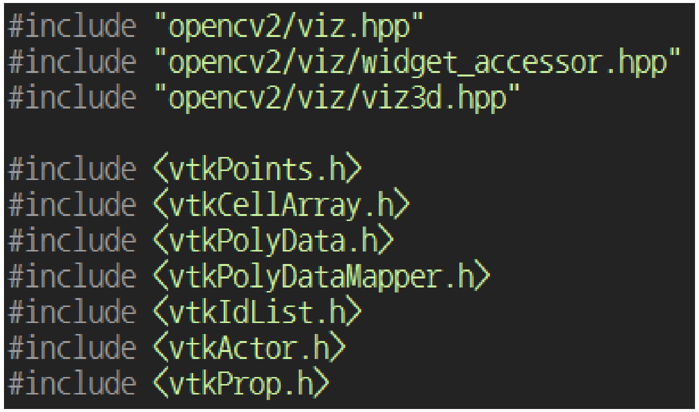


```
viz::Viz3d myWindow("Creating Widgets");

viz::WLine axis(Point3f(-1.0f,-1.0f,-1.0f), Point3f(1.0f,1.0f,1.0f));

myWindow.showWidget("Line",axis);

myWindow.spin();
```

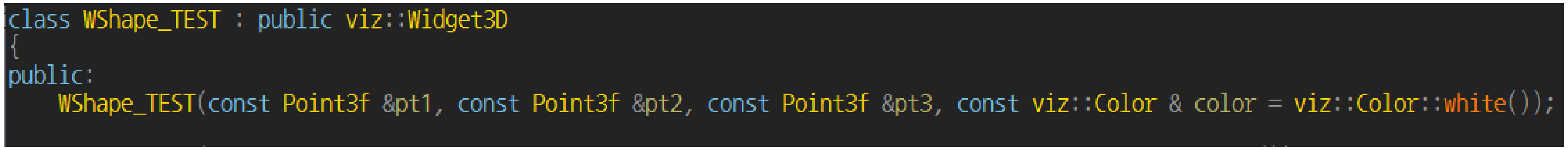


03 Visualization

- Make a widget



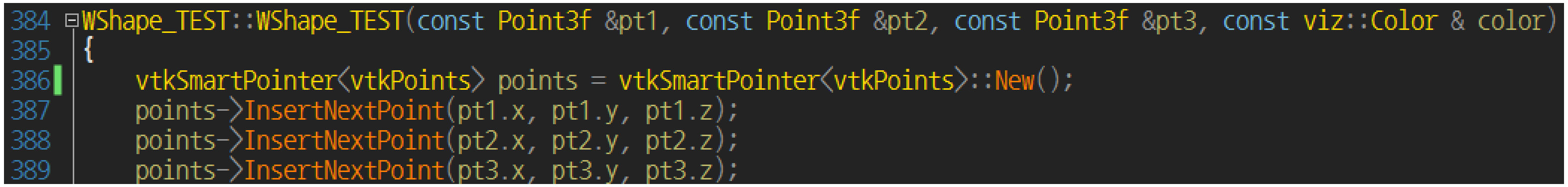
```
#include "opencv2/viz.hpp"
#include "opencv2/viz/widget_accessor.hpp"
#include "opencv2/viz/viz3d.hpp"

#include <vtkPoints.h>
#include <vtkCellArray.h>
#include <vtkPolyData.h>
#include <vtkPolyDataMapper.h>
#include <vtkIdList.h>
#include <vtkActor.h>
#include <vtkProp.h>
```



```
class WShape_TEST : public viz::Widget3D
{
public:
    WShape_TEST(const Point3f &pt1, const Point3f &pt2, const Point3f &pt3, const viz::Color & color = viz::Color::white());
```

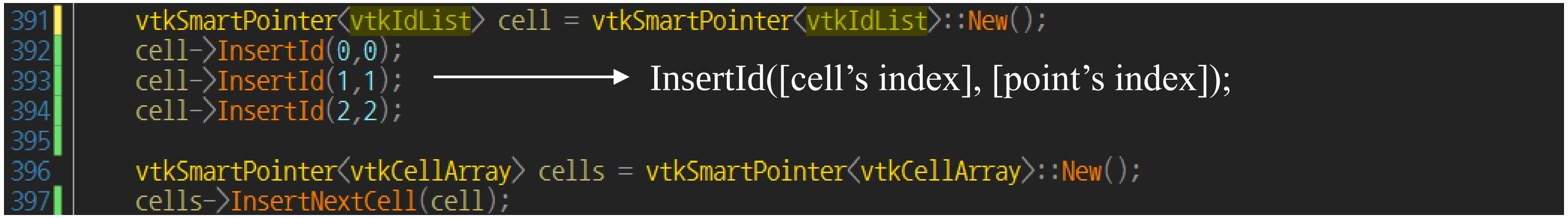
03 Visualization

- Make a widget



```
WShape_TEST::WShape_TEST(const Point3f &pt1, const Point3f &pt2, const Point3f &pt3, const viz::Color & color)
{
    vtkSmartPointer<vtkPoints> points = vtkSmartPointer<vtkPoints>::New();
    points->InsertNextPoint(pt1.x, pt1.y, pt1.z);
    points->InsertNextPoint(pt2.x, pt2.y, pt2.z);
    points->InsertNextPoint(pt3.x, pt3.y, pt3.z);
```

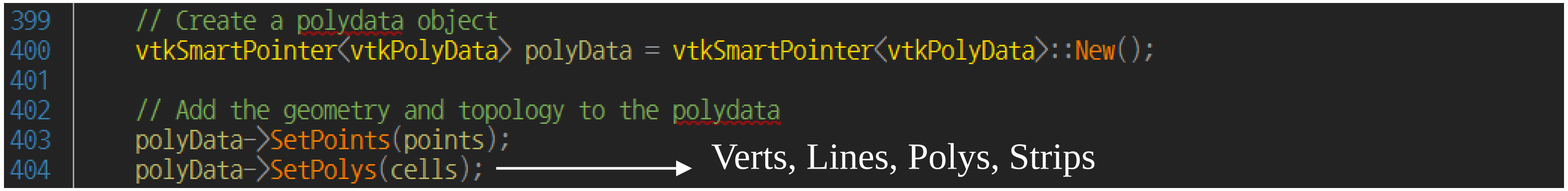
-> Insert point information.



```
    vtkSmartPointer<vtkIdList> cell = vtkSmartPointer<vtkIdList>::New();
    cell->InsertId(0,0);
    cell->InsertId(1,1);
    cell->InsertId(2,2);

    vtkSmartPointer<vtkCellArray> cells = vtkSmartPointer<vtkCellArray>::New();
    cells->InsertNextCell(cell);
```

InsertId([cell’s index], [point’s index]);

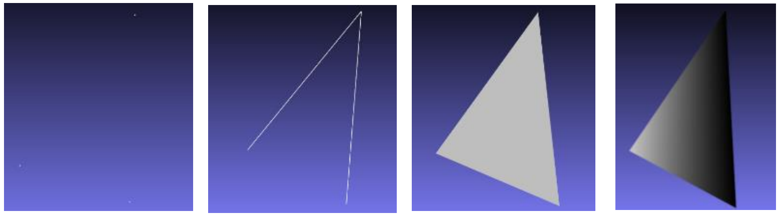
-> Set cell information.



```
    // Create a polydata object
    vtkSmartPointer<vtkPolyData> polyData = vtkSmartPointer<vtkPolyData>::New();

    // Add the geometry and topology to the polydata
    polyData->SetPoints(points);
    polyData->SetPolys(cells);
```

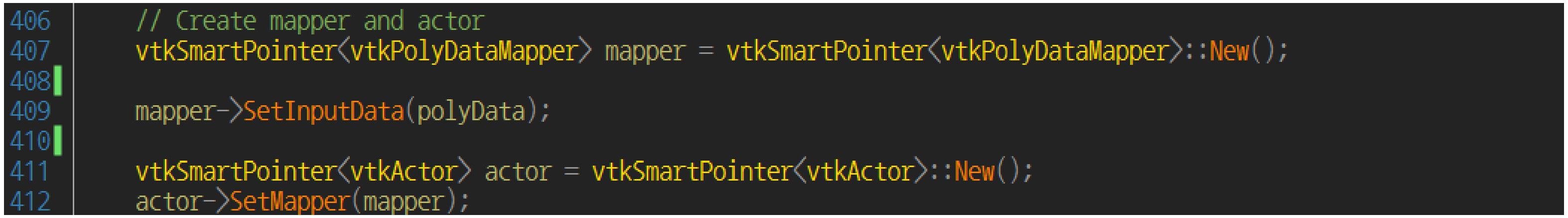
Verts, Lines, Polys, Strips

-> Set Points and Shape.



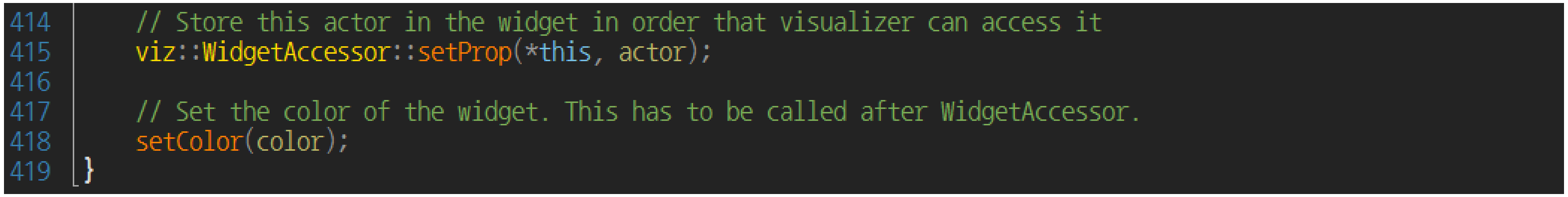
03 Visualization

- Make a widget



```
    // Create mapper and actor
    vtkSmartPointer<vtkPolyDataMapper> mapper = vtkSmartPointer<vtkPolyDataMapper>::New();

    mapper->SetInputData(polyData);

    vtkSmartPointer<vtkActor> actor = vtkSmartPointer<vtkActor>::New();
    actor->SetMapper(mapper);
```

-> Create mapper & actor.

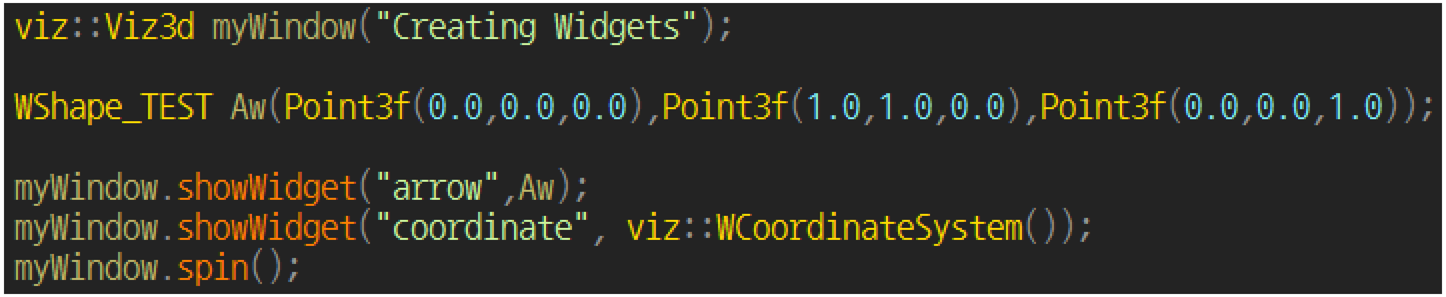


```
    // Store this actor in the widget in order that visualizer can access it
    viz::WidgetAccessor::setProp(*this, actor);

    // Set the color of the widget. This has to be called after WidgetAccessor.
    setColor(color);
}
```

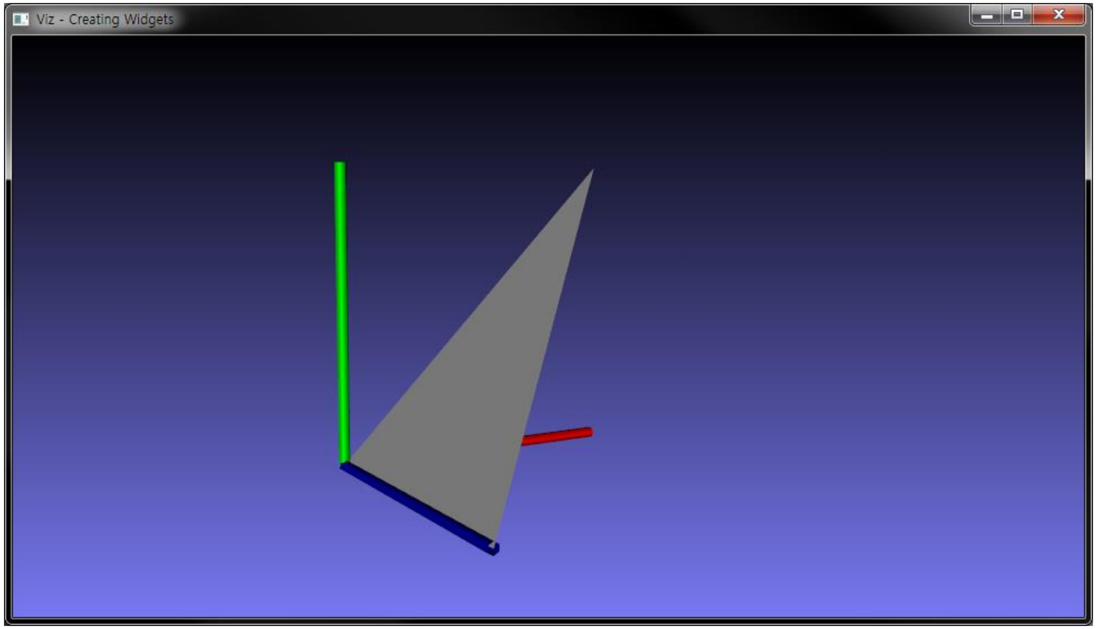
-> Store this actor in the widget

03 Visualization

- Make a widget



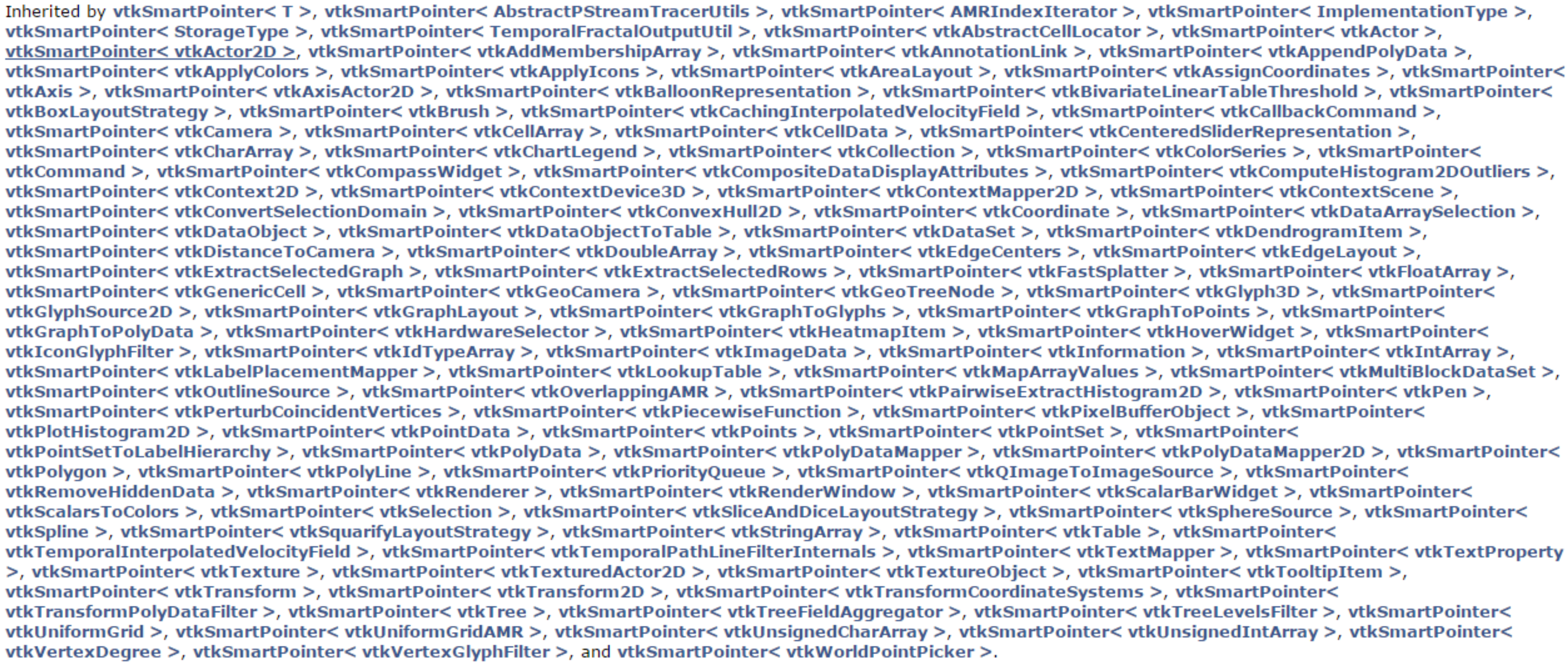
```
viz::Viz3d myWindow("Creating Widgets");

WShape_TEST Aw(Point3f(0.0,0.0,0.0),Point3f(1.0,1.0,0.0),Point3f(0.0,0.0,1.0));

myWindow.showWidget("arrow",Aw);
myWindow.showWidget("coordinate", viz::WCoordinateSystem());
myWindow.spin();
```



03 Visualization

- Smart Pointer

-> 해제 할 필요가 없는 포인터.

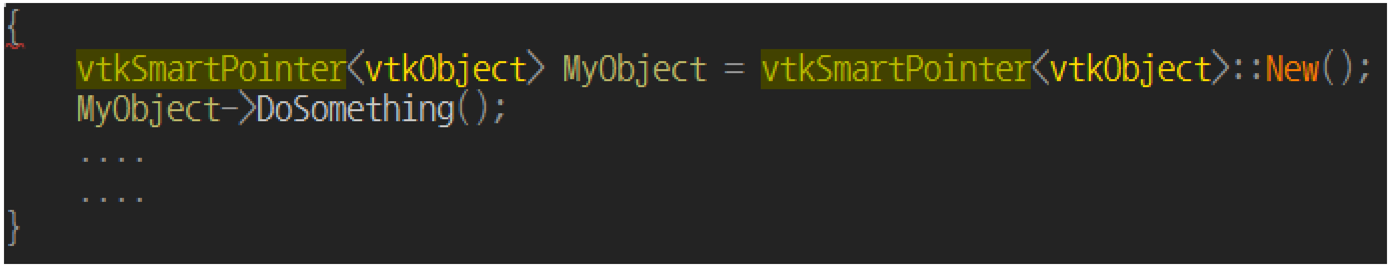


Inherited by vtkSmartPointer< T >, vtkSmartPointer< AbstractPStreamTracerUtils >, vtkSmartPointer< AMRIndexIterator >, vtkSmartPointer< ImplementationType >, vtkSmartPointer< StorageType >, vtkSmartPointer< TemporalFractalOutputUtil >, vtkSmartPointer< vtkAbstractCellLocator >, vtkSmartPointer< vtkActor >, vtkSmartPointer< vtkActor2D >, vtkSmartPointer< vtkAddMembershipArray >, vtkSmartPointer< vtkAnnotationLink >, vtkSmartPointer< vtkAppendPolyData >, vtkSmartPointer< vtkApplyColors >, vtkSmartPointer< vtkApplyIcons >, vtkSmartPointer< vtkAreaLayout >, vtkSmartPointer< vtkAssignCoordinates >, vtkSmartPointer< vtkAxis >, vtkSmartPointer< vtkAxisActor2D >, vtkSmartPointer< vtkBalloonRepresentation >, vtkSmartPointer< vtkBivariateLinearTableThreshold >, vtkSmartPointer< vtkBoxLayoutStrategy >, vtkSmartPointer< vtkBrush >, vtkSmartPointer< vtkCachingInterpolatedVelocityField >, vtkSmartPointer< vtkCallbackCommand >, vtkSmartPointer< vtkCamera >, vtkSmartPointer< vtkCellArray >, vtkSmartPointer< vtkCellData >, vtkSmartPointer< vtkCenteredSliderRepresentation >, vtkSmartPointer< vtkCharArray >, vtkSmartPointer< vtkChartLegend >, vtkSmartPointer< vtkCollection >, vtkSmartPointer< vtkColorSeries >, vtkSmartPointer< vtkCommand >, vtkSmartPointer< vtkCompassWidget >, vtkSmartPointer< vtkCompositeDataDisplayAttributes >, vtkSmartPointer< vtkComputeHistogram2DOutliers >, vtkSmartPointer< vtkContext2D >, vtkSmartPointer< vtkContextDevice3D >, vtkSmartPointer< vtkContextMapper2D >, vtkSmartPointer< vtkContextScene >, vtkSmartPointer< vtkConvertSelectionDomain >, vtkSmartPointer< vtkConvexHull2D >, vtkSmartPointer< vtkCoordinate >, vtkSmartPointer< vtkDataArraySelection >, vtkSmartPointer< vtkDataObject >, vtkSmartPointer< vtkDataObjectToTable >, vtkSmartPointer< vtkDataSet >, vtkSmartPointer< vtkDendrogramItem >, vtkSmartPointer< vtkDistanceToCamera >, vtkSmartPointer< vtkDoubleArray >, vtkSmartPointer< vtkEdgeCenters >, vtkSmartPointer< vtkEdgeLayout >, vtkSmartPointer< vtkExtractSelectedGraph >, vtkSmartPointer< vtkExtractSelectedRows >, vtkSmartPointer< vtkFastSplatter >, vtkSmartPointer< vtkFloatArray >, vtkSmartPointer< vtkGenericCell >, vtkSmartPointer< vtkGeoCamera >, vtkSmartPointer< vtkGeoTreeNode >, vtkSmartPointer< vtkGlyph3D >, vtkSmartPointer< vtkGlyphSource2D >, vtkSmartPointer< vtkGraphLayout >, vtkSmartPointer< vtkGraphToGlyphs >, vtkSmartPointer< vtkGraphToPoints >, vtkSmartPointer< vtkGraphToPolyData >, vtkSmartPointer< vtkHardwareSelector >, vtkSmartPointer< vtkHeatmapItem >, vtkSmartPointer< vtkHoverWidget >, vtkSmartPointer< vtkIconGlyphFilter >, vtkSmartPointer< vtkIdTypeArray >, vtkSmartPointer< vtkImageData >, vtkSmartPointer< vtkInformation >, vtkSmartPointer< vtkIntArray >, vtkSmartPointer< vtkLabelPlacementMapper >, vtkSmartPointer< vtkLookupTable >, vtkSmartPointer< vtkMapArrayValues >, vtkSmartPointer< vtkMultiBlockDataSet >, vtkSmartPointer< vtkOutlineSource >, vtkSmartPointer< vtkOverlappingAMR >, vtkSmartPointer< vtkPairwiseExtractHistogram2D >, vtkSmartPointer< vtkPen >, vtkSmartPointer< vtkPerturbCoincidentVertices >, vtkSmartPointer< vtkPiecewiseFunction >, vtkSmartPointer< vtkPixelBufferObject >, vtkSmartPointer< vtkPlotHistogram2D >, vtkSmartPointer< vtkPointData >, vtkSmartPointer< vtkPoints >, vtkSmartPointer< vtkPointSet >, vtkSmartPointer< vtkPointSetToLabelHierarchy >, vtkSmartPointer< vtkPolyData >, vtkSmartPointer< vtkPolyDataMapper >, vtkSmartPointer< vtkPolyDataMapper2D >, vtkSmartPointer< vtkPolygon >, vtkSmartPointer< vtkPolyLine >, vtkSmartPointer< vtkPriorityQueue >, vtkSmartPointer< vtkQImageToImageSource >, vtkSmartPointer< vtkRemoveHiddenData >, vtkSmartPointer< vtkRenderer >, vtkSmartPointer< vtkRenderWindow >, vtkSmartPointer< vtkScalarBarWidget >, vtkSmartPointer< vtkScalarsToColors >, vtkSmartPointer< vtkSelection >, vtkSmartPointer< vtkSliceAndDiceLayoutStrategy >, vtkSmartPointer< vtkSphereSource >, vtkSmartPointer< vtkSpline >, vtkSmartPointer< vtkSquarifyLayoutStrategy >, vtkSmartPointer< vtkStringArray >, vtkSmartPointer< vtkTable >, vtkSmartPointer< vtkTemporalInterpolatedVelocityField >, vtkSmartPointer< vtkTemporalPathLineFilterInternals >, vtkSmartPointer< vtkTextMapper >, vtkSmartPointer< vtkTextProperty >, vtkSmartPointer< vtkTexture >, vtkSmartPointer< vtkTexturedActor2D >, vtkSmartPointer< vtkTextureObject >, vtkSmartPointer< vtkTooltipItem >, vtkSmartPointer< vtkTransform >, vtkSmartPointer< vtkTransform2D >, vtkSmartPointer< vtkTransformCoordinateSystems >, vtkSmartPointer< vtkTransformPolyDataFilter >, vtkSmartPointer< vtkTree >, vtkSmartPointer< vtkTreeFieldAggregator >, vtkSmartPointer< vtkTreeLevelsFilter >, vtkSmartPointer< vtkUniformGrid >, vtkSmartPointer< vtkUniformGridAMR >, vtkSmartPointer< vtkUnsignedCharArray >, vtkSmartPointer< vtkUnsignedIntArray >, vtkSmartPointer< vtkVertexDegree >, vtkSmartPointer< vtkVertexGlyphFilter >, and vtkSmartPointer< vtkWorldPointPicker >.

03 Visualization

- Smart Pointer

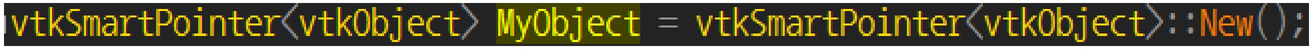
1) 사라지는 타이밍 1



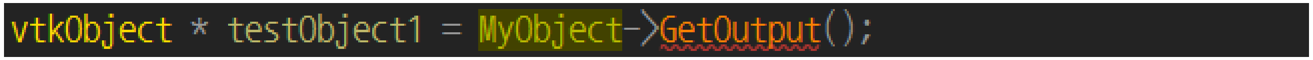
```
{
    vtkSmartPointer<vtkObject> MyObject = vtkSmartPointer<vtkObject>::New();
    MyObject->DoSomething();
    ....
    ....
}
```

-> scope를 나가면 MyObject는 삭제 된다.

2) 사라지는 타이밍 2

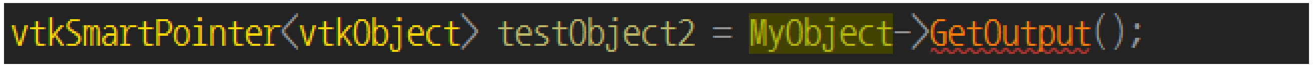


```
vtkSmartPointer<vtkObject> MyObject = vtkSmartPointer<vtkObject>::New();
```



```
vtkObject * testObject1 = MyObject->GetOutput();
```

-> MyObject가 scope를 나가면 삭제.



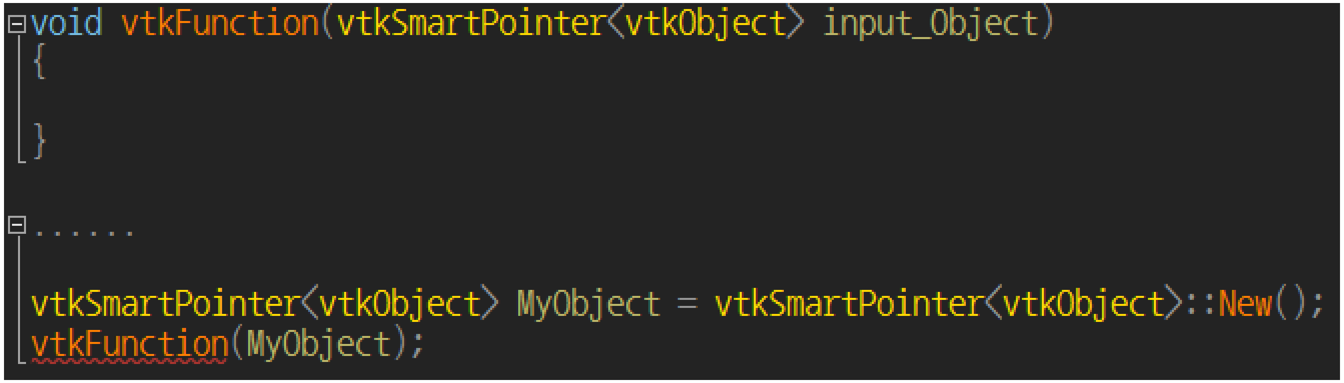
```
vtkSmartPointer<vtkObject> testObject2 = MyObject->GetOutput();
```

-> MyObject와 testObject2가 모두 scope를 나가면 삭제.

03 Visualization

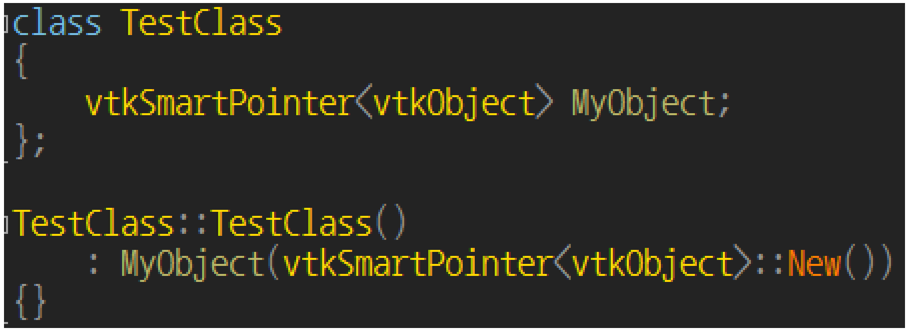
- Smart Pointer

3) 함수 매개변수



```
void vtkFunction(vtkSmartPointer<vtkObject> input_Object)
{

}


......

vtkSmartPointer<vtkObject> MyObject = vtkSmartPointer<vtkObject>::New();
vtkFunction(MyObject);
```

4) 클래스 멤버 변수.



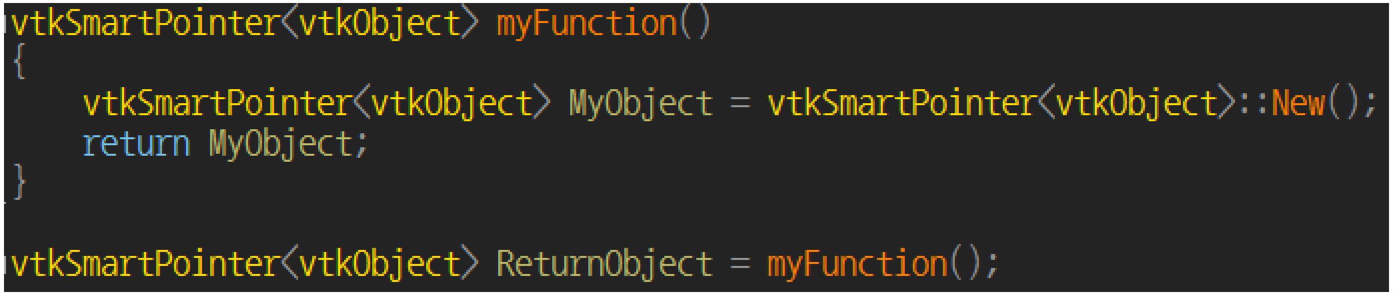
```
class TestClass
{
    vtkSmartPointer<vtkObject> MyObject;
};

TestClass::TestClass()
    : MyObject(vtkSmartPointer<vtkObject>::New())
{}
```

03 Visualization

- Smart Pointer

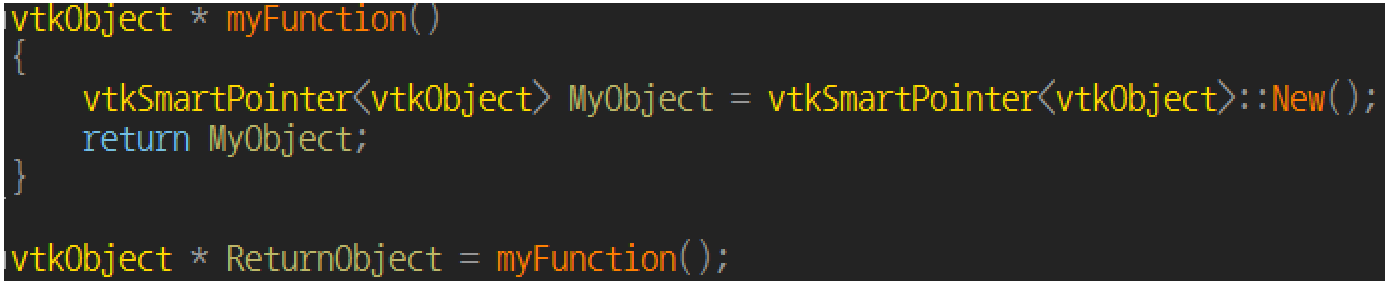
5) Smart Pointer 반환.

○



```
vtkSmartPointer<vtkObject> myFunction()
{
    vtkSmartPointer<vtkObject> MyObject = vtkSmartPointer<vtkObject>::New();
    return MyObject;
}

vtkSmartPointer<vtkObject> ReturnObject = myFunction();
```

✕



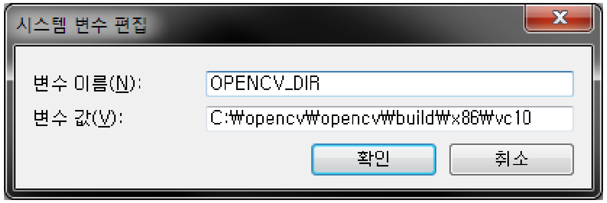
```
vtkObject * myFunction()
{
    vtkSmartPointer<vtkObject> MyObject = vtkSmartPointer<vtkObject>::New();
    return MyObject;
}

vtkObject * ReturnObject = myFunction();
```

감사합니다.

01

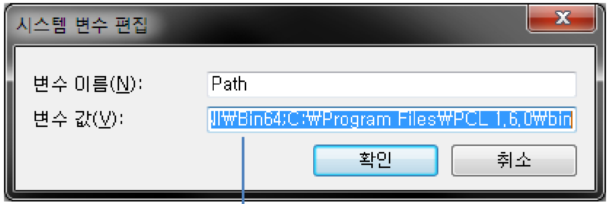
OpenCV 3.0 Installation

- 환경변수 설정


시스템 변수 편집

변수 이름(N): OPENCV_DIR

변수 값(V): C:₩opencv₩opencv₩build₩x86₩vc10

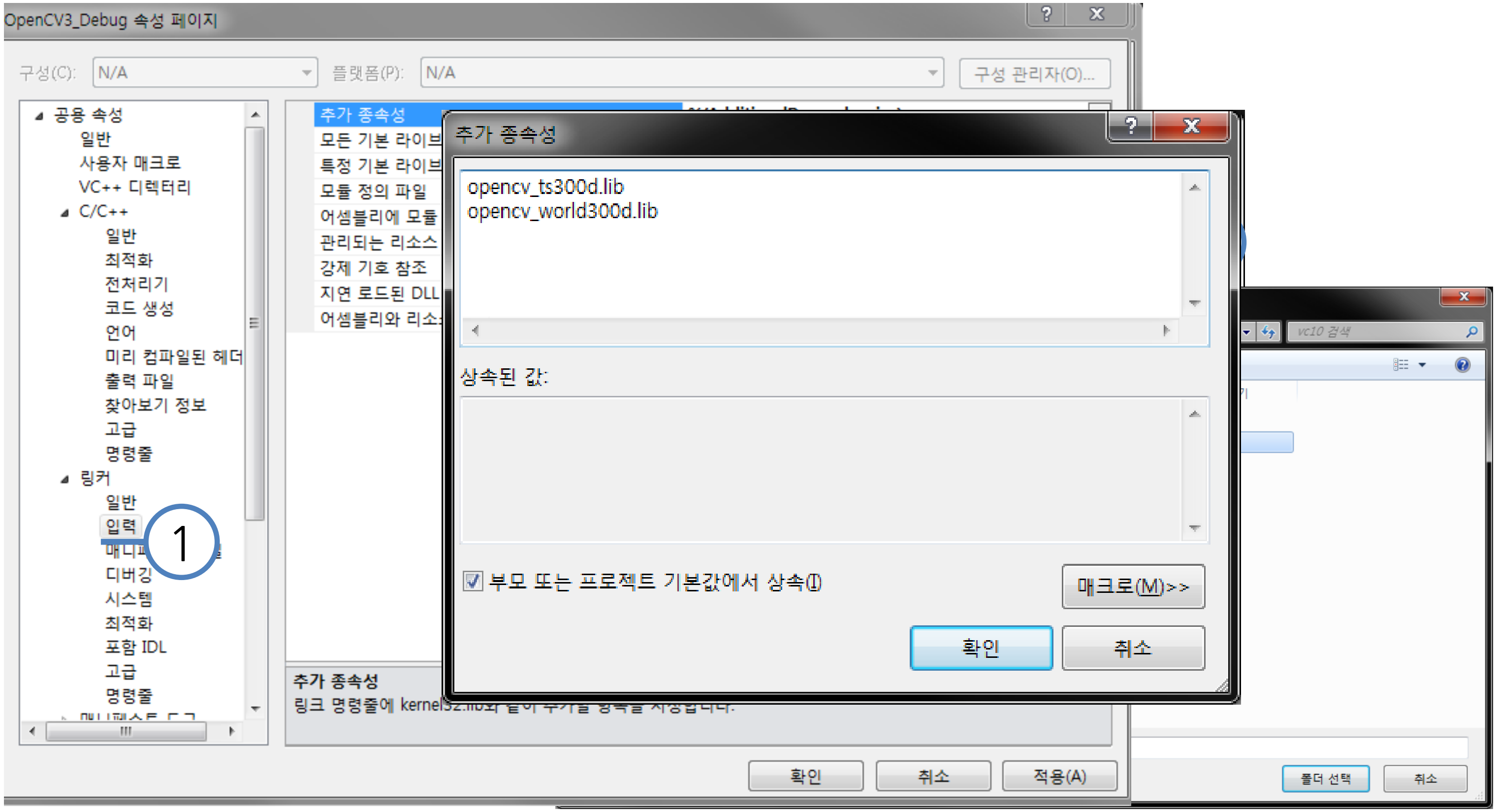

시스템 변수 편집

변수 이름(N): Path

변수 값(V): ;₩Bin64;C:₩Program Files₩PCL 1.6.0₩bin

;%OPENCV_DIR%\bin;

00 OpenCV 3.0 Installation



OpenCV3_Debug 속성 페이지

구성(C): N/A
플랫폼(P): N/A

추가 종속성

opencv_ts300d.lib
opencv_world300d.lib

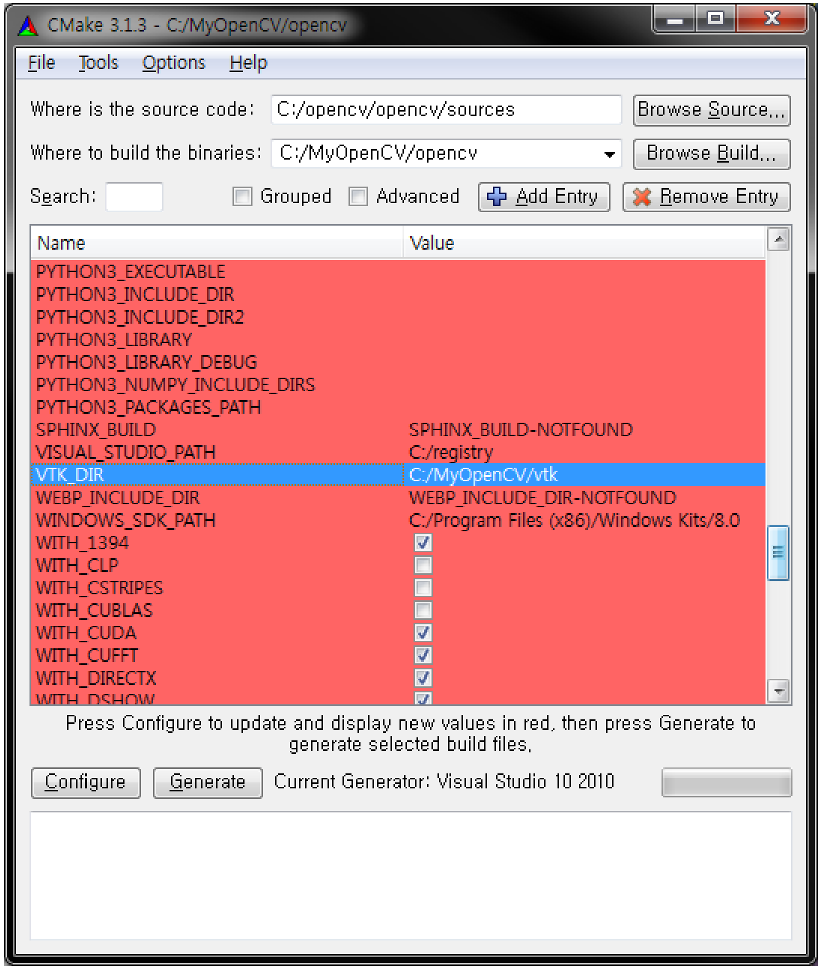
상속된 값:

☑ 부모 또는 프로젝트 기본값에서 상속(I)

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

00

OpenCV 3.0 Installation 2



Where is the source code: C:/opencv/opencv/sources

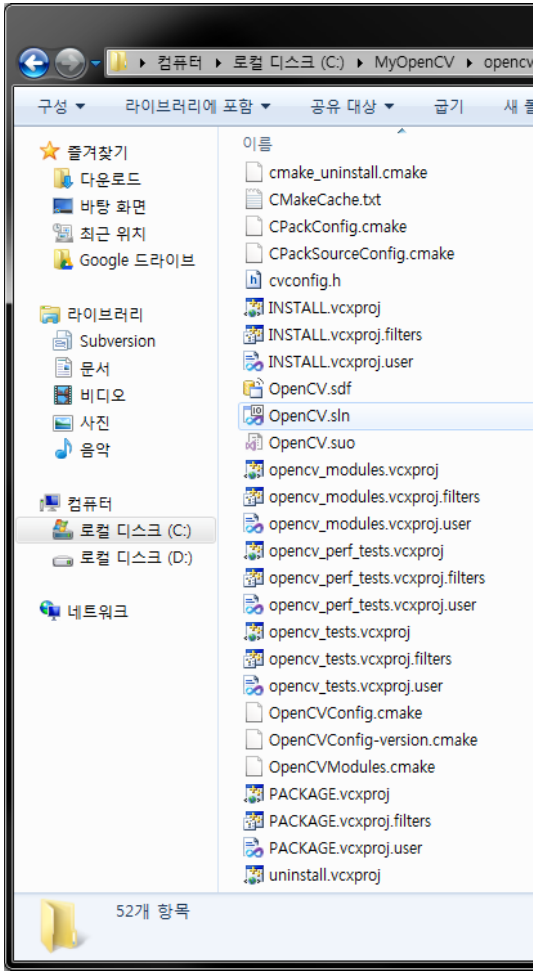
Where to build the binaries: C:/MyOpenCV/opencv

Search: □ Grouped □ Advanced

Name	Value
PYTHON3_EXECUTABLE	
PYTHON3_INCLUDE_DIR	
PYTHON3_INCLUDE_DIR2	
PYTHON3_LIBRARY	
PYTHON3_LIBRARY_DEBUG	
PYTHON3_NUMPY_INCLUDE_DIRS	
PYTHON3_PACKAGES_PATH	
SPHINX_BUILD	SPHINX_BUILD-NOTFOUND
VISUAL_STUDIO_PATH	C:/registry
VTK_DIR	C:/MyOpenCV/vtk
WEBP_INCLUDE_DIR	WEBP_INCLUDE_DIR-NOTFOUND
WINDOWS_SDK_PATH	C:/Program Files (x86)/Windows Kits/8.0
WITH_1394	☑
WITH_CLP	□
WITH_CSTRIPES	□
WITH_CUBLAS	□
WITH_CUDA	☑
WITH_CUFFT	☑
WITH_DIRECTX	☑
WITH_DSHOW	☑

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 10 2010

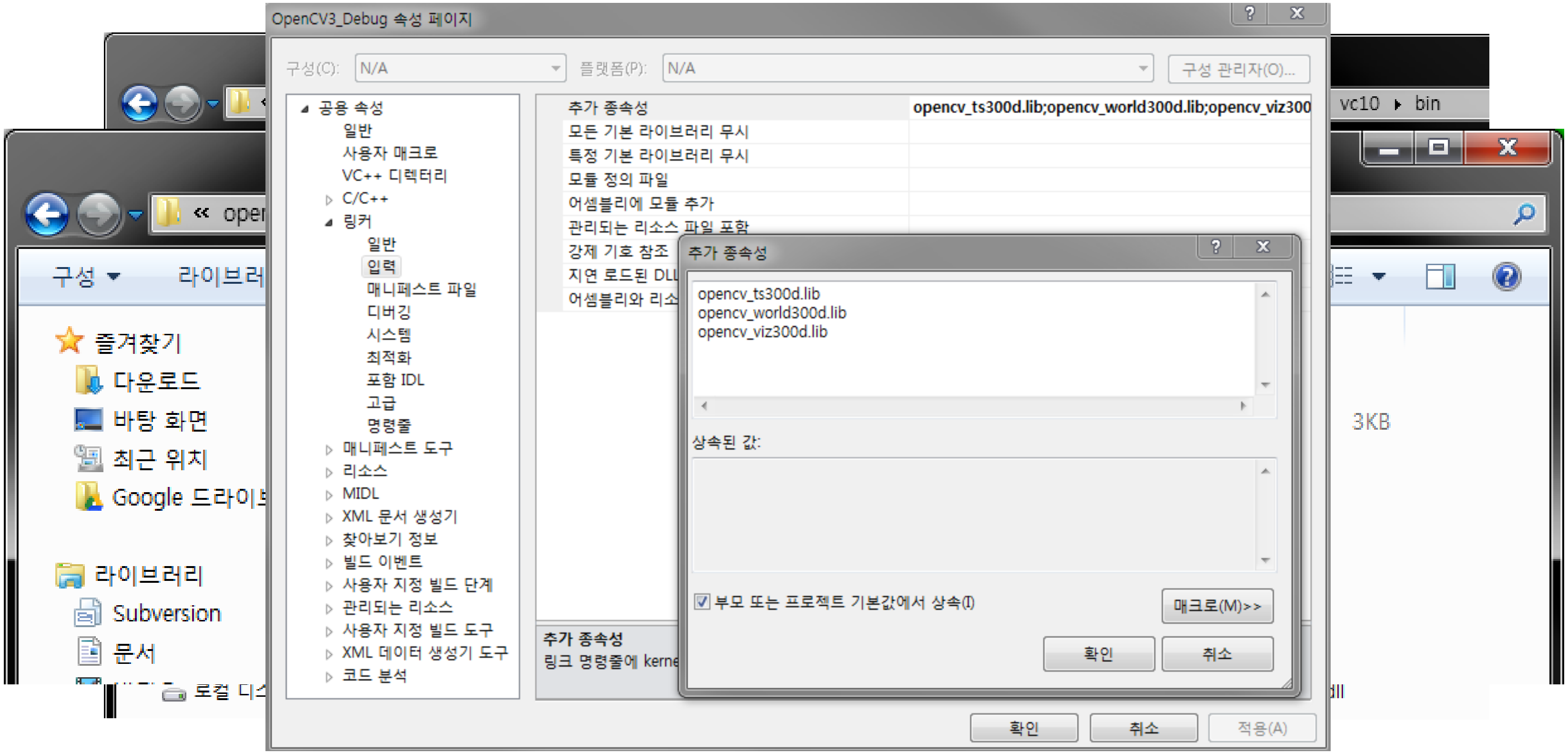


이름

- cmake_uninstall.cmake
- CMakeCache.txt
- CPackConfig.cmake
- CPackSourceConfig.cmake
- cvconfig.h
- INSTALL.vcxproj
- INSTALL.vcxproj.filters
- INSTALL.vcxproj.user
- OpenCV.sdf
- OpenCV.sln
- OpenCV.suo
- opencv_modules.vcxproj
- opencv_modules.vcxproj.filters
- opencv_modules.vcxproj.user
- opencv_perf_tests.vcxproj
- opencv_perf_tests.vcxproj.filters
- opencv_perf_tests.vcxproj.user
- opencv_tests.vcxproj
- opencv_tests.vcxproj.filters
- opencv_tests.vcxproj.user
- OpenCVConfig.cmake
- OpenCVConfig-version.cmake
- OpenCVModules.cmake
- PACKAGE.vcxproj
- PACKAGE.vcxproj.filters
- PACKAGE.vcxproj.user
- uninstall.vcxproj

52개 항목

00 OpenCV 3.0 Installation 2



OpenCV3_Debug 속성 페이지

구성(C): N/A 플랫폼(P): N/A

추가 종속성: opencv_ts300d.lib;opencv_world300d.lib;opencv_viz300

추가 종속성

opencv_ts300d.lib
opencv_world300d.lib
opencv_viz300d.lib

상속된 값:

☑ 부모 또는 프로젝트 기본값에서 상속(I)

추가 종속성
링크 명령줄에 kerne