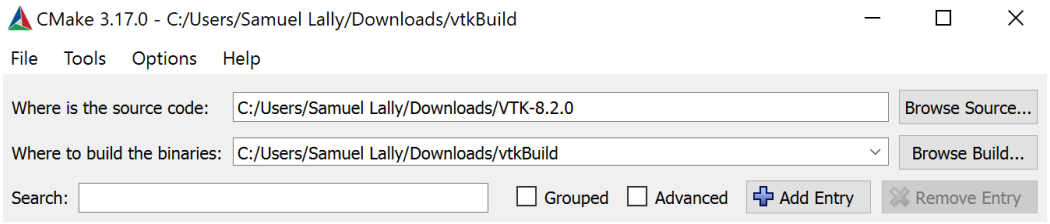
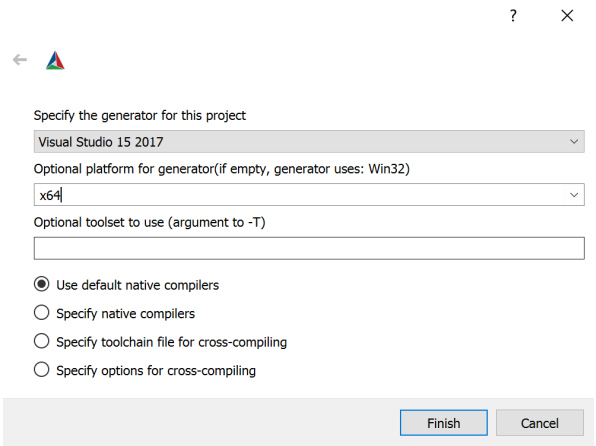
Azure Kinect Kinfu Setup – Sam Lally

Installation with Visual Studios 2017

1. Download and extract kinfu project from https://github.com/microsoft/Azure-Kinect-Samples/tree/master/opencv-kinfu-samples.
2. Download and extract opencv-4.1.0 from https://opencv.org/releases.
    a. Copy opencv\build\include* to the opencv-kinfu-samples\extern\opencv-4.1.0\include* (Create extern directory and subdirectories accordingly).
3. Download and extract opencv_contrib-4.1.0 from https://github.com/opencv/opencv_contrib/releases.
    a. Copy opencv_contrib-4.1.0\modules\rgbd\include* to extern\opencv_contrib-4.1.0\modules\rgbd\include*
    (Create opencv_contrib-4.1.0 directory and subdirectories accordingly.)
    b. Copy opencv_contrib-4.1.0\modules\viz\include* to extern\opencv_contrib-4.1\modules\viz\include*
4. Install VTK using the below instructions. They are based off https://vtk.org/Wiki/VTK/Building/Windows with a few modifications.
    a. Download and extract VTK-8.2.0 from https://vtk.org/download.
    b. Download CMake if not already installed https://cmake.org/download.
    c. Create a build folder for VTK beside the extracted VTK folder.
    d. Run CMake using VTK-8.2.0 as the source code folder and VTKBuild as the binaries folder.

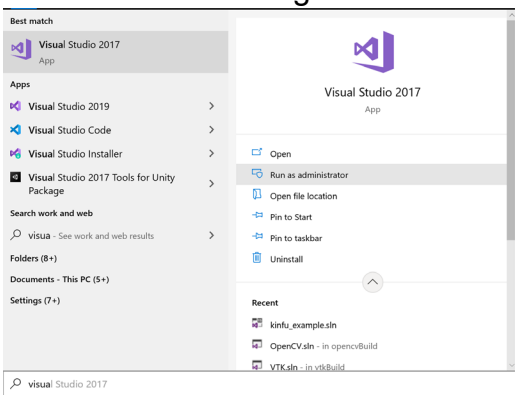


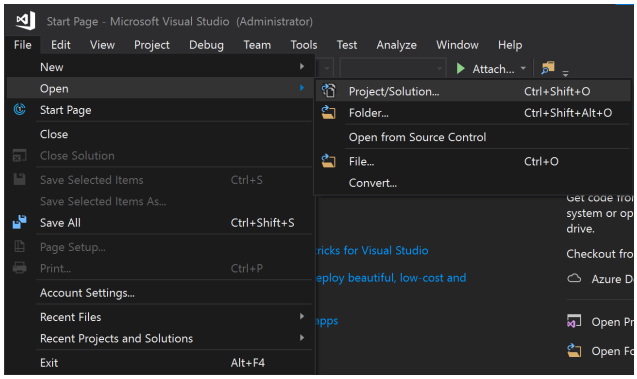
    e. Press Configure and choose Visual Studios 2017 and x64 as the generator. The default configuration will be used so press Configure until all settings turn from white to red.



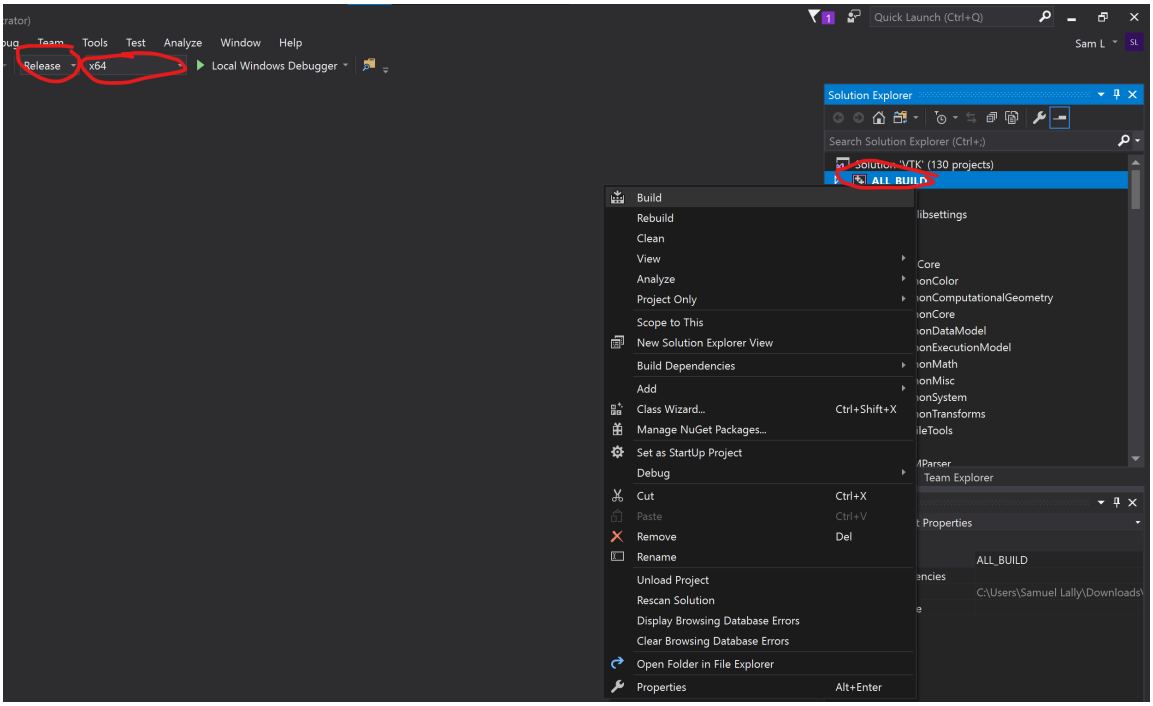
    f. Once the configurations have turned white press Generate to build VTK.

g. Open Visual Studio 2017 as administrator and then open VTK.sln from inside the VTKBuild folder using it.





h. Make sure the build mode is set to Release and x64 and then right click ALL_BUILD and select Build. This should take 10-20 minutes.



i. Once the ALL_BUILD is complete right click INSTALL and build it as well. VTK should now be built and installed.

j. OPTIONAL: Read step 6 from the vtk wiki to add VTK to the system PATH if wanted.

5. Install OpenCV using the below instructions. They are based off https://github.com/opencv/opencv_contrib with a few modifications.
   a. Create a build folder for OpenCV beside the extracted OpenCV folder.
   b. Run CMake using OpenCV/sources as the source code folder and OpenCVBuild as the binaries folder.
   c. Press Configure and use the same generators as VTK, Visual Studios 2017 and x64
   d. Change the following configurations. The Search bar can be used to find them.
      i. Set WITH_VTK to true.
      ii. Set VTK_DIR to the VTKBuild folder.
      iii. Set OPENCV_ENABLE_NONFREE to true.
      iv. Set OPENCV_EXTRA_MODULES_PATH to the opencv_contrib-4.1.0\modules folder.
   e. Press configure to switch all the configurations to white and then hit generate.

    f. Open Visual Studio 2017 as administrator and open OpenCV.sln from inside the OpenCVBuild folder.
    g. Make sure the build mode is set to Release and x64 and then right click ALL_BUILD and select Build. This should take 10-20 minutes.
6. Go to OpenCVBuild/lib/Release and copy the following library folders to opencv-kinfu-samples/extern/lib/Release.
   (Create lib folder and subdirectories accordingly)
    a. opencv_calib3d410.lib
    b. opencv_core410.lib
    c. opencv_highgui410.lib
    d. opencv_imgproc410.lib
    e. opencv_rgbd410.lib
    f. opencv_viz410.lib
7. Open kinfu_example.vcxproj from the opencv-kinfu-samples folder with a text editor. Find <AdditionalDependencies> </AdditionalDependencies> and replace it with

   <AdditionalDependencies>%(AdditionalDependencies);opencv_core410.lib;opencv_calib3d410.lib;opencv_rgbd410.lib;opencv_highgui410.lib;opencv_viz410.lib;opencv_imgproc410.lib;</AdditionalDependencies>
8. Open kinfu_example.sln from the opencv-kinfu-samples folder with Visual Studios 2017
    a. Open main.cpp and uncomment HAVE_OPENCV
    b. Right click kinfu_example on the right and select build.
9. kinfu_example.exe will now be in opencv-kinfu-samples/x64/Release but the OpenCV and VTK dll files have to be added
    a. Copy the dlls from VTKBUILD/bin/Release to opencv-kinfu-samples/x64/Release
    b. Copy the dlls from OpenCVBUILD/bin/Release to opencv-kinfu-samples/x64/Release

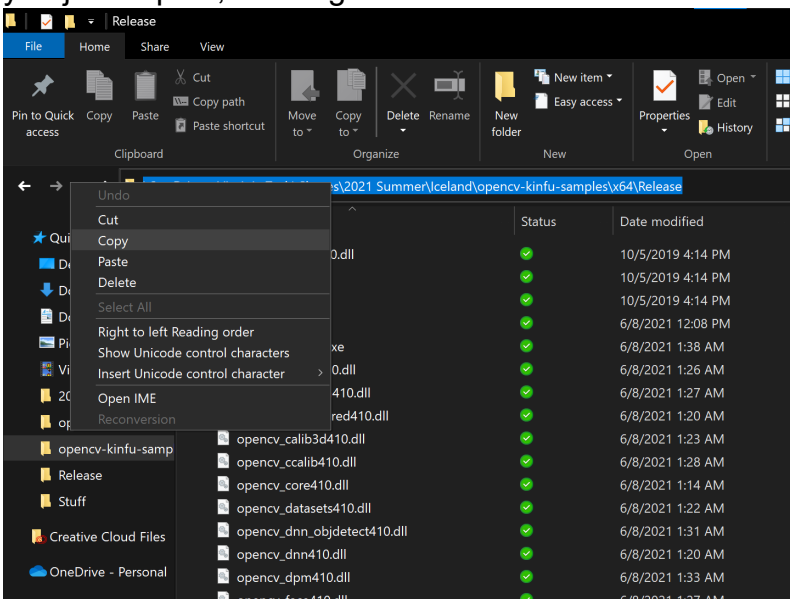
Installation from GitHub Build

1. Download the Github build from https://github.com/slally17/kinfu.
2. Open kinfu_examples.sln.
3. Install the k4a package.
    a. Right click References on the right and select Manage Nuget Packages
    b. If you get a dropdown box about missing packages click that to install k4a.
    c. If not, click Microsoft.Azure.Kinect.Sensor and click install.
    d. NOTE: The program is tested on version 1.3.0, updating to the latest version has not been tested.
4. Build the solution in Release x64 mode.

Running kinfu

1. Open powershell or command prompt.

2. Copy the file path to the .exe build and run **cd FILEPATH**, replacing FILEPATH with the path you just copied, to navigate to the folder.



3. Run **kinfu_examples.exe** to start the program and also to bring up the syntax and controls.
    a. The optional mode variable allows the Kinect's field of view to be changed.
4. Pressing **w** will save the current point cloud as kf_output.ply within the exe folder. Pressing w multiple times will overwrite previous point clouds so rename or move them to another folder to save them.
5. NOTE: If you click off of powershell/cmd and **q** wont work to quit, press **ctrl-c** to quit the program.

Converting Point Cloud to Mesh

1. Download/open Meshlab and open the ply file with it.
2. Apply Filters > Normals, Curvatures, and Orientation > Compute Normals for Point Sets
3. OPTIONAL: Apply Filters > Point Set > Point Cloud Simplification
4. Filters > Remeshing, Simplification, and Reconstruction > Surface Reconstruction: Ball Pivoting
5. Export file as obj or file type of choice.