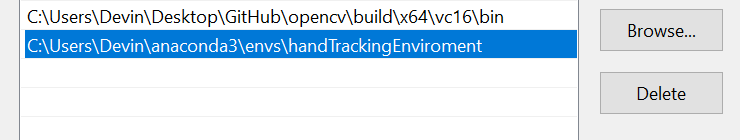
Demo Setup Notes

1. Camera\_Calibration Repo: <https://github.com/Blanchard-lab/Camera_Calibration>
   1. Sanity Check Open Camera Script: <https://github.com/Blanchard-lab/Camera_Calibration/blob/main/azureOverlay/checkCameraUtil.py>
2. Visual Studio Install
   1. Make sure the C++ compiler for visual studio is installed
      1. A screenshot of a computer

         Description automatically generated
3. Install OpenCV
   1. <https://github.com/opencv/opencv>
   2. <https://github.com/opencv/opencv/releases/tag/4.5.5>
   3. Place the “opencv” folder one directory above the “Camera\_Calibration” folder (the build is setup to look for it there)
   4. Add opencv to the path variable (example from my configuration): “C:\Users\Devin\Desktop\GitHub\opencv\build\x64\vc16\bin”
4. CONDA
   1. Make sure conda is installed and the path to the exe is added to the path:
      1. A blue and white box with black text

         Description automatically generated
   2. Create a virtual environment using the “handTrackingEnvironment.yaml” (In the repository)
5. Linking Python
   1. Set the include and libs to the directory of the python libs relative to the virtual environment you are using
      1. 
   2. Add the virtual environment location to the path variable
      1. 
   3. If python##\_d.lib doesn’t exist copy python##.lib and rename it
6. Running the Build
   1. if conda doesn't work on your account in powershell (specifically if activate isn't a valid command) try calling "conda init powershell" and run activate again
   2. open a powershell at "C:\GitHub\Camera\_Calibration\offline\_processor\build\bin\Release"
   3. activate hand tracking enviroment "conda activate C:\ProgramData\anaconda3\envs\handTrackingEnvironment"
   4. run ".\offline\_processor.exe"
7. Linking NUMPY
   1. Add the path to numpy from the virtual environment to the includes:
      1. C:\Users\vanderh\Anaconda3\envs\handTrackingEnvironment\Lib\site-packages\numpy\core\include