|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Revision** | **Date** | **Name** | **Change Description** |
| 01 | 07/12/2021 | Yang Hyo Kim | New Document |

1. Purpose
   1. Code explanation for calibration step c.
2. Scope
   1. Engineering/Manufacturing
3. Responsibility
   1. Engineering and Manufacturing is responsible in keeping this document updated
   2. Procedure must be followed by Engineering/Manufacturing
4. References
   1. List any References

# Development environment

1. OS (Windows 10)
2. Python package (Anaconda3-2020.07-Windows-x86\_64, <https://www.anaconda.com/products/individual>)
3. IDE (Spyder 4.1.4)
4. Python 3.8.3
5. GUI (wxPython 4.1.0 msw (phoenix) wxWidgets 3.1.4, <https://wxpython.org/>)
6. Plotting (matplotlib 3.2.2, <https://matplotlib.org/>)
7. exe freezing (PyInstaller 4.0, <https://www.pyinstaller.org/>)

# Source code location

D:\Optical Biosystems\Regular work\RND\2020-07-29 Calibration 2\Python codes\2021-02-03 GUI

# Manual location

\\ma2files\Production systems\04\_Documentation\Calibration\PROC-10XX\_01, SV Calibration step C.pdf

Delivered exe file location  
\\ma2files\Production systems\09\_Production Tools\Python\SV Calibration

# Program flow overview

## [SVCalib\_20210203.py]

|  |
| --- |
| GUI implementation using wxPython  **runBtnClick()**  -> checkLsgd(), *Check lsgd file existence* -> pathSaoReconExe(), *Check SaoRecon exe file existence* -> threading.Thread(target=self.runData).start(), *prevent GUI freezing* -> runData()  **runData()**  multiprocessing.Process(target=FullCalibrationStep2M1), *multiple-core for speedup* -> FullCalibrationStep2M1() |

## [CalibLib.py]

|  |
| --- |
| Conversion of existing Matlab codes into Python  - FullCalibrationStep2M1.m  **FullCalibrationStep2M1()**Find the Pitch/Orientation generating the best contrast for single beads.  -> findLoc(), *Find Isolated Single Bead* -> reconPitchOrient(), *Find Pitch/Orientation* -> showPitchOrient(), *Show plots of contrast and save them into a png file* -> findMaxPitchOrient(), *Find the best Pitch/Orientation* -> reconMaxPitchOrient(), *Recon with the best Pitch/Orientation Parameters* -> finalResult(), *Output results and clean up*  **findLoc()** FIND ISOLATED SINGLE BEAD.  -> LsgdToImageStack(), *Return an image stack of 12 images from a lsgd file* -> EstimateBackgroundNoise(), *Return estimated black background intensity value from the camera* -> SelectIsolatedTargets(), *Return an Nx2 array of isolated single bead location from an input image* |

## **[BeadCalibLib.py]**

|  |
| --- |
| Conversion of existing Matlab codes into Python  - LsgdToImageStack.m - EstimateBackgroundNoise.m - SelectIsolatedTargets.m  **LsgdToImageStack()** Return an image stack of 12 images from a lsgd file  **EstimateBackgroundNoise()** Return estimated black background intensity value from the camera  **SelectIsolatedTargets()** Return an Nx2 (N number of targets and (row,col) location) array of isolated single bead location from an input image |