**Pattern Match Study Procedure**

**General\*:**

1. Take multiple pictures of the area to pattern match.
2. Use Images imported from PIL in python to crop the templates.
3. Use LabVIEW’s pattern match function to pattern match all these images.
4. Use python to make a scatterplot of template length vs. efficiency, time, standard deviation, mean, etc.

\*The “general” versions of these programs do not exist, refer to the programs used to complete this study on the Fiducial Marks to write programs that suit your needs.

**For Fiducial Marks (cropped only in y):**

1. Take pictures of each corner (in order or C, B, A, D, # of pictures is 1, # or loops can be varied)
   1. Find the corner locations with ContinuousCameraStageV2.vi.
   2. Use visuallyInspectPositionListResults2.vi (under Utils -> Standalone) to take pictures of each corner (as many as desired).
2. Use AutomaticTemplateMaker.py (in git, the Fiducial folder) to cut the templates in y.
   1. Start from the largest image (which you can crop manually) and decide the interval (pixel difference between each image). This will write templates of varying length.
3. Run these images through Pattern\_Matching\_size\_study\_length.vi (under Iria -> ATLAS17 -> Template Size Study). The results are in the folder where the corner images folder is.
   1. Input for VI:
      1. Image folder: Images should be the order in which they were taken, i.e. in loops (from step 1) where no two consecutive images are for the same tooling pin
      2. Template folder: templates should all vary in size by a consistent number of pixels (the interval). See my previous study for the naming system (in my folder).
   2. Note: the number of images/coordinates per loop is hard coded.
4. Run these files through TemplateSizeAnalysis3.py (in git, the Fiducial Folder).
   1. Enter the appropriate files in the driver code, specify the Fiducial mark and whether the template varies in length or width.

**For Tooling Pins (cropped in y and x):**

1. Take pictures of each tooling pin. All the right pins should be in one folder, all the left pins should be in another.
   1. Use visuallyInspectPositionListResults2.vi (under Utils -> Standalone) (# of pictures is 1, # or loops can be varied).
2. Use AutomaticTemplateMaker-Length.py or AutomaticTemplateMaker-Width.py to cut the templates in length (y direction) or width (x direction), respectively.
   1. Start from the largest image (which you can crop manually) and decide the interval (pixel difference between each image). This will write templates of varying length.
3. Run these images through Pattern\_Matching\_size\_study\_length.vi or Pattern\_Matching\_size\_study\_width.vi (under Iria -> Tooling Pin). The results are in the folder where the corner images folder is.
   1. Input for VI:
      1. Image folder: Images should be the order in which they were taken, i.e. in loops (from step 1) where no two consecutive images are for the same tooling pin
      2. Template folder: templates should all vary in size by a consistent number of pixels (the interval). They should be named as widthInPixels + “x” + lengthInPixels + “.png”.
      3. REMEMBER to change pixel to micron
   2. Note: the number of images/coordinates per loop is hard coded.
4. Run these files through ToolingPinPMStudy2.py (in git, the Tooling Pin folder).
   1. Enter the appropriate files in the driver code, specify whether the template varies in length or width.