# Hyper_stitch Demo

Tom Selby, tas72@cam.ac.uk

15/02/2025

After working through this demo you should be able to use the hyper_stitch GUI to stitch many contiguous high dimensional microscopy datasets together. All source code can be found on Github.

## INSTALLATION

1. Install the relevant python packages. Hyper_stitch is largely built upon OpenCV and Kornia, pip installing kornia should provide both of these and all other requirements.

```
pip install kornia
```

2. Download the code and data from Github and then unzip the files.

```
Download
```

3. In terminal navigate to the newly downloaded directory.

```
cd .../Downloads/hyper_stitch-main
```

4. There should be a copy of the following files located there:

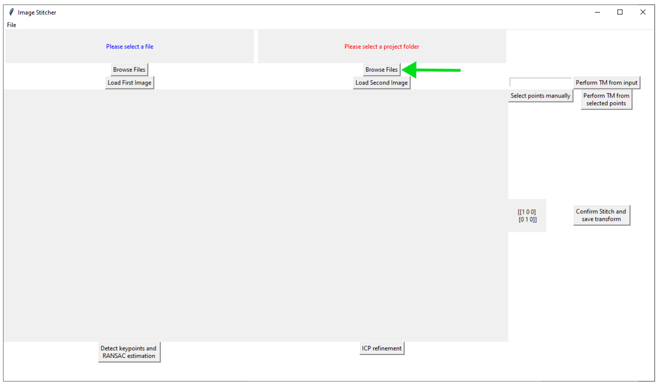
```
gui.py
korina_env.yml
LICENSE
README.md
stitching.py
transform_list
utils.py
```

5. Run `gui.py` with python

```
python gui.py
```

## DEMO

1. First select a project folder, this is where the ‘`transform_list`’ will be updated and saved as you continually stitch images together. Select, you should see the path appear in red in the top right.



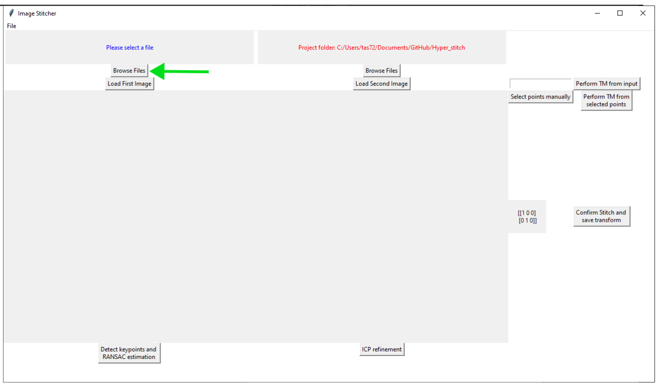
Please select a file

Please select a project folder

```
[[1 0 0]
 [0 1 0]]
```

2. Load the images, we use virtual bright field images as a proxy for the 4D-STEM dataset which are saved as two `.npy` files in the `/data` directory.

(i) Select ‘`Browse Files`’ on the left. This will open a popup, from here navigate and select one of the images located in `/data`. You should see the path appear in blue in the top left.

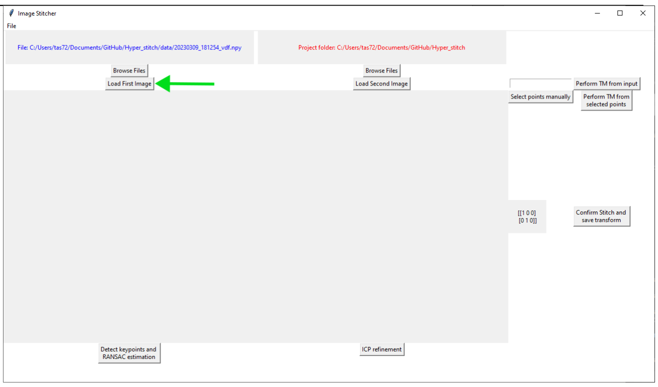


Please select a file

Project folder: C:/Users/tas72/Documents/GitHub/Hyper_stitch

```
[[1 0 0]
 [0 1 0]]
```

(ii) Select ‘`Load First Image`’ to load the image at the selected path.

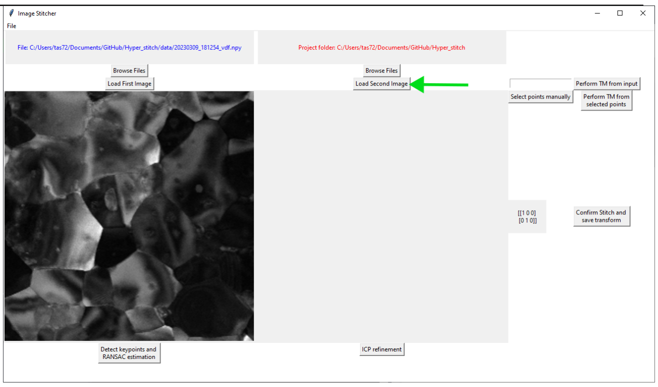


File: C:/Users/tas72/Documents/GitHub/Hyper_stitch/data/20230309_181254_vdf.npy

Project folder: C:/Users/tas72/Documents/GitHub/Hyper_stitch

```
[[1 0 0]
 [0 1 0]]
```

(iii) Repeat this process but now loading the second image with the ‘`Load Second Image`’ button.

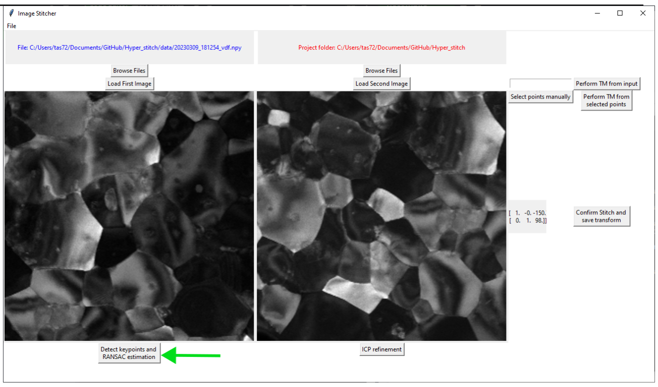


File: C:/Users/tas72/Documents/GitHub/Hyper_stitch/data/20230309_181254_vdf.npy

Project folder: C:/Users/tas72/Documents/GitHub/Hyper_stitch

```
[[1 0 0]
 [0 1 0]]
```

3. Now use the ‘`Detect keypoints and RANSAC estimation`’ to map one image onto the other. Once this has been performed two windows will appear sequentially where you can inspect the stitching. An affine transform will now be displayed to the right of the images.

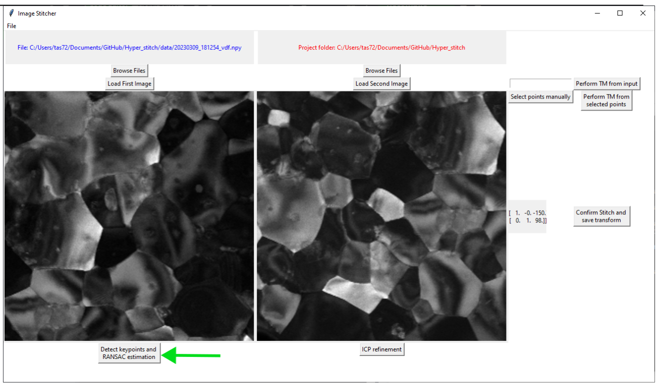


File: C:/Users/tas72/Documents/GitHub/Hyper_stitch/data/20230309_181254_vdf.npy

Project folder: C:/Users/tas72/Documents/GitHub/Hyper_stitch

```
[  1.  -0. -150.
[  0.   1.   98.]]
```

4. If the stitching needs to be improved one way to do this is to use an iterative closest point (ICP) refinement. To do this press the ‘`ICP refinement`’ button in the bottom right. Note with the current implementation this can be very computationally costly.

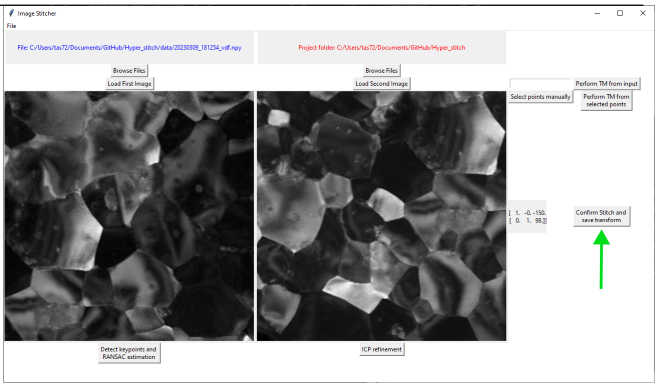


File: C:/Users/tas72/Documents/GitHub/Hyper_stitch/data/20230309_181254_vdf.npy

Project folder: C:/Users/tas72/Documents/GitHub/Hyper_stitch

```
[ 1.  -0. -150.
[ 0.   1.   98.]]
```

5. Once you are happy with the stitching confirm this with the ‘`Confirm Stitch and save transform`’ button. **This will delete the original images and save the stitched version in their place. We therefore recommend to make a copy of all unstitched raw images before confirming the stitch.** The `transform_list` will also be saved in the project folder as a ‘`transform_list`’ file. This is a record of all affine transforms and which images they are applied to. **Again this will overwrite any old ‘`transform_list`’ file in the same directory. We therefore recommend loading in any previous ‘`transform_list`’ file by navigating to `File -> Load transform file` when the GUI is first started.**

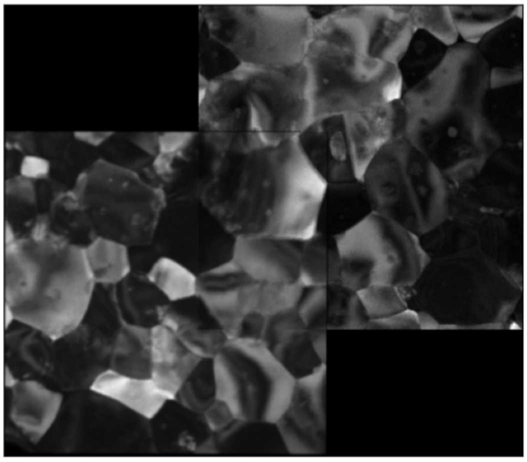


File: C:/Users/tas72/Documents/GitHub/Hyper_stitch/data/20230309_181254_vdf.npy

Project folder: C:/Users/tas72/Documents/GitHub/Hyper_stitch

```
[ 1.  -0. -150.
[ 0.   1.   98.]]
```

6. The stitch is now complete and you can inspect the results. To load the 'transform_list' file in python use pandas, pd.read_pickle(path_to_file). The transform_list can then be used to calculate where each pixel from each image ends up in the stitched dataset. This can then be used to calculate the position of pixels once spatially correlated to another microscopy technique.



## ADDITIONAL FEATURES AND SETTINGS