Improving the accuracy of scroll-alignment with image registration

1. Summary

- Accurate alignment of scroll volumes scanned at different energies is an important pre-requisite for multi-spectral ink detection.
- I have implemented a python program that uses image registration to improve the current alignment transforms.
- Given a pair of volume scans and an initial estimate of their alignment transform, the program finds an affine (non-deformable) transform that maximizes the mutual information between the volumes.
- When applied to the existing transforms for Scrolls 3 and 4, the algorithm increases the mutual information, and visually improves the alignment.
- Because the image files are very large, the affine registration takes several tens of hours to complete, suggesting that deformable registration may not be feasible.
- The code and instructions for applying it to new scans are available in the project repository, https://github.com/Paul-G2/VesuviusScrollAlignment.

2. Evaluating the current accuracy

To begin, consider the Pherc1667 transform that's given in the 20231107190228-to-20231117161658.json file on ash2txt.

Applying that transform to slice 15836 of the 88 keV scan, and overlaying the result on the original slice produces Figure 1, where the original slice is in the green channel, and the transformed slice is in the red channel.

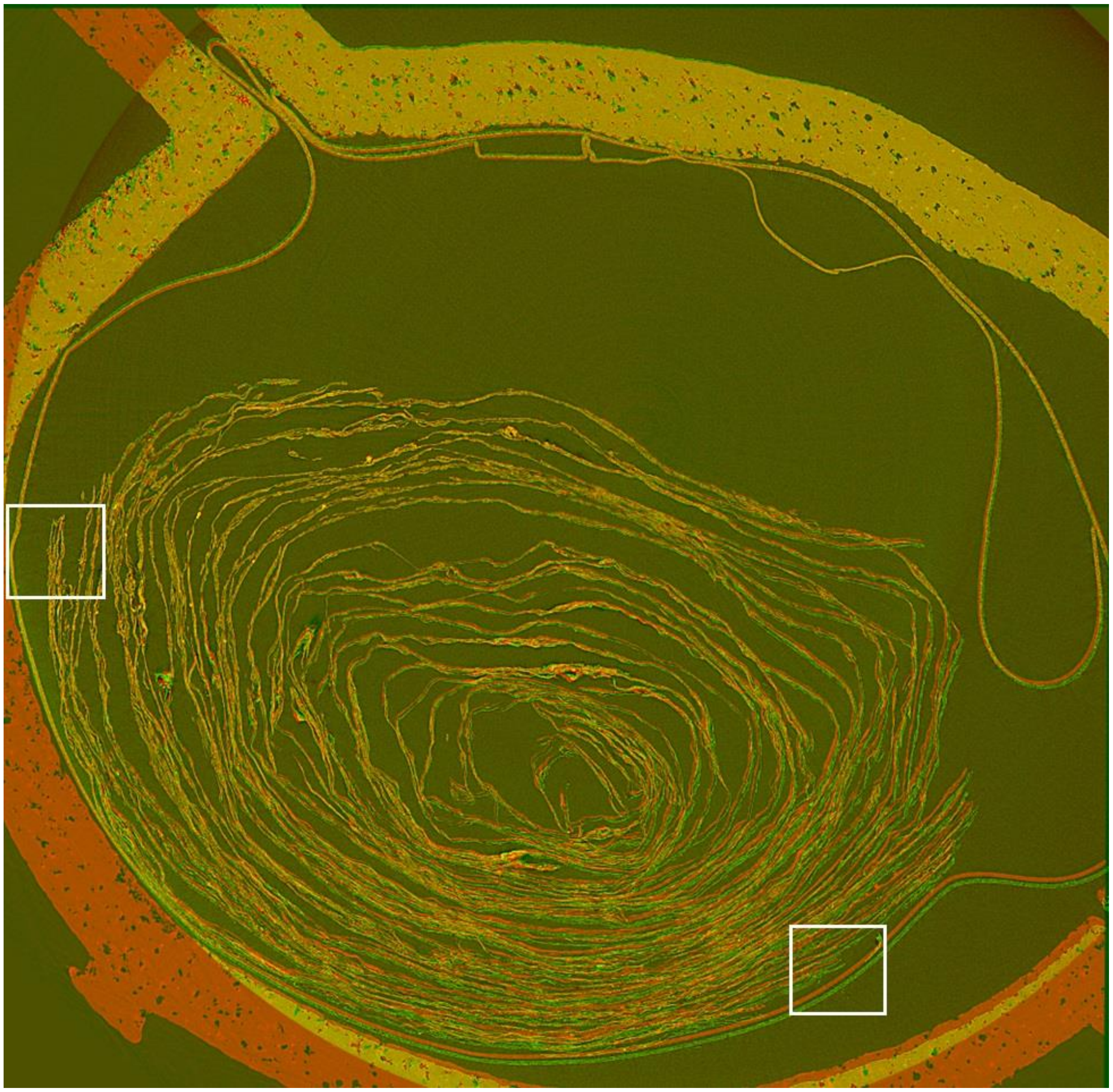

Figure 1. Slice 15836 of the 88 keV scan (green channel) overlaid with the corresponding transformed slice from the 53 keV scan (red channel). The areas in the white boxes are zoomed up in Figures 2 and 3 below.

Overall, this looks good, but zooming-in brings out some important details: In the left-hand white box, the alignment is quite accurate (see Figure 2). This is expected, because that box is centered on one of the reference points that was used to calculate the transform.

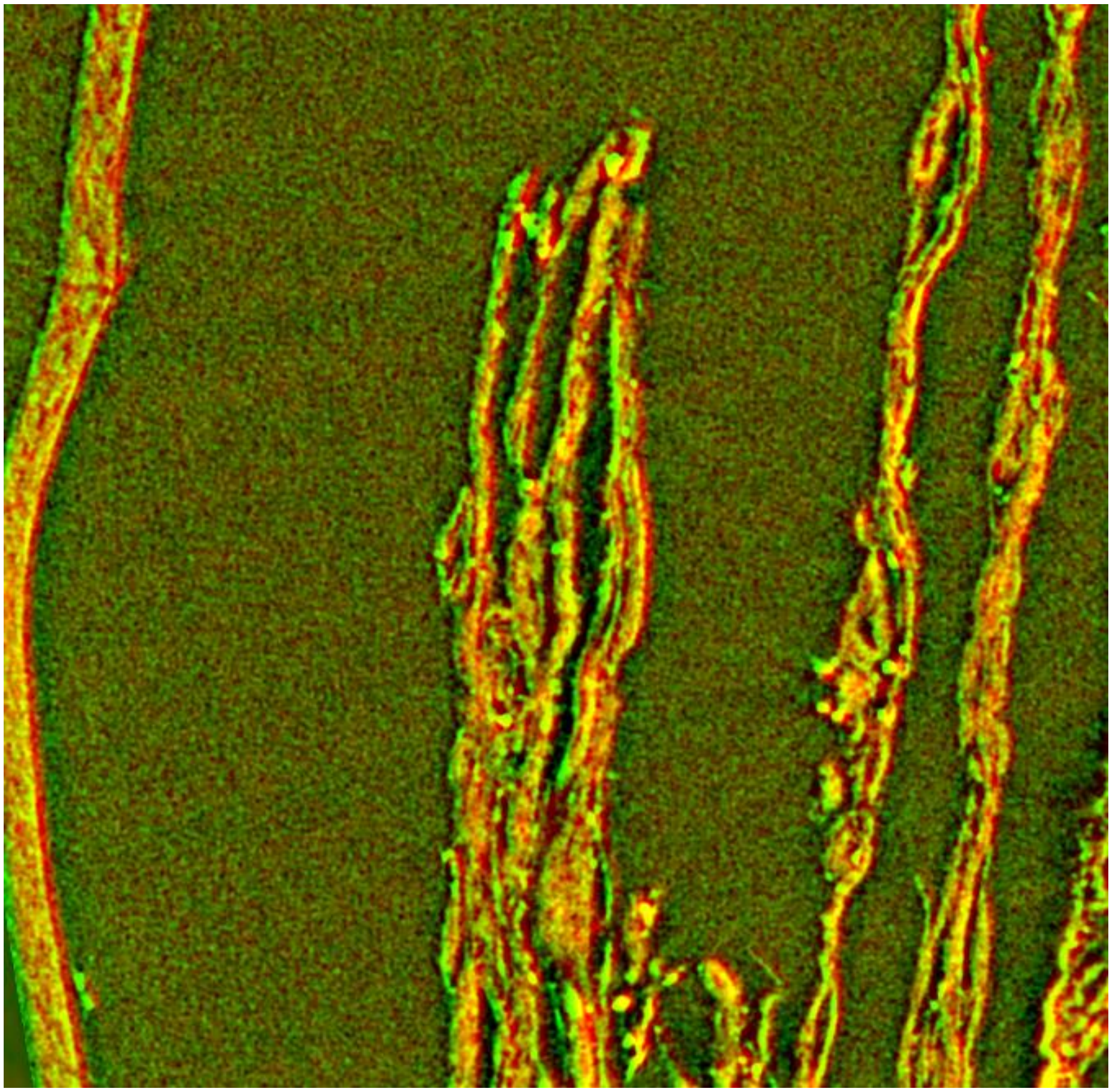

Figure 2. The alignment is good in the left-hand white box of Figure 1.

However, in lower-right white box, the alignment is noticeably off (Figure 3).

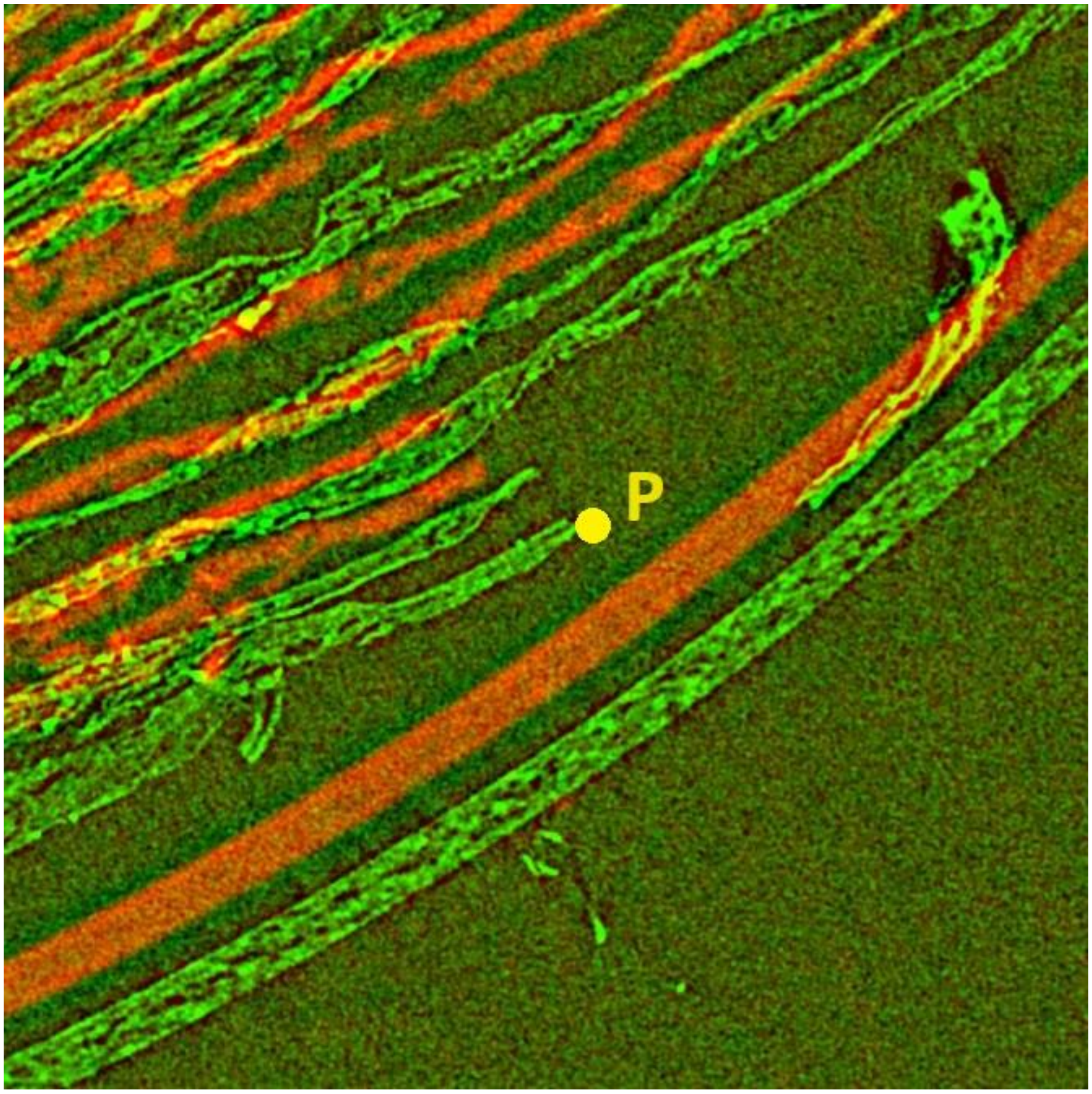

Figure 3. The alignment is not as good in the lower-right part of Figure 1.

This is a surprisingly large misalignment. Could I have calculated the aligned image incorrectly? My code can be checked, it's in the `extract_img_from_volume()` function in the `scroll_align.py` file, but there's also an easy sanity test:

The point P in Figure 3 has coordinates (6190, 7045, 15836) in the 88 keV volume. Applying the transform T maps it to P' = (2651, 2935, 6616) in the 53 keV volume. Figure 4 shows that this point is offset from the papyrus edge by about the same amount as the red-to-green offset in Figure 3. So the misalignment seems to be real, not an artifact of my alignment code.

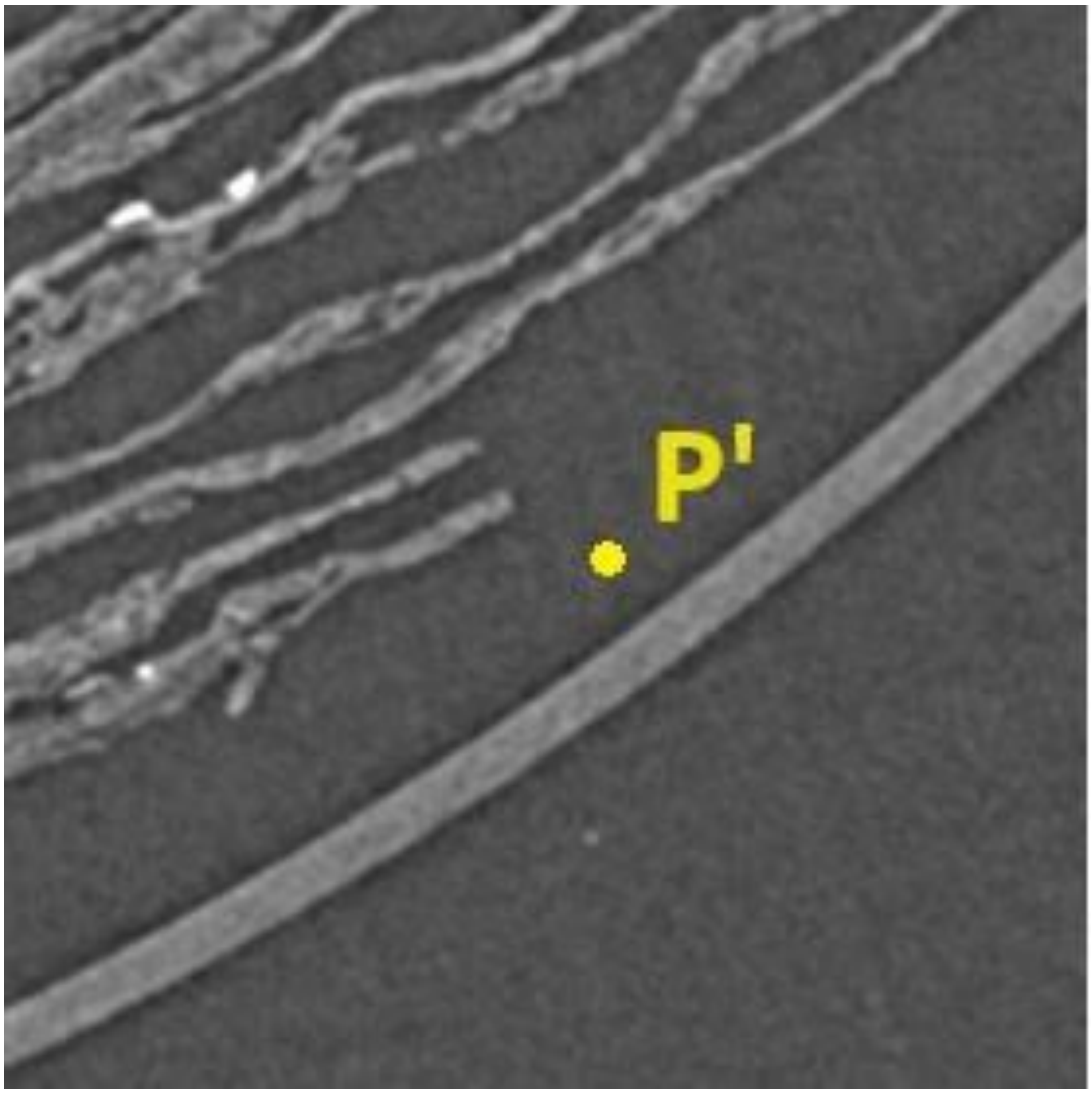

*Figure 4. The transformed point P' = T*P does not align with the papyrus edge in the 53 keV volume, even though P does so in Figure 3.*

Other similar examples can be found throughout the volume. Perhaps one or more of the reference points used to estimate T somehow moved between scans.

The transforms for Scroll 3 are much better, but they can still be improved, as described in the next section.

3. Improving the alignment accuracy

The python code, available at https://github.com/Paul-G2/VesuviusScrollAlignment, uses image-registration to improve the scroll alignment accuracy. The main features of the implementation are:

- It seeks to find the affine transformation matrix T that maximizes the mutual information between corresponding image slices in the two volumes.
- Only 10 slices (equally spaced along the volume z-axis) are used, to make the calculation tractable. (Even with this limited number of slices the maximization can take several days on an average desktop machine, because it searches a 12-dimensional space with a fairly expensive cost function.)
- The registration is done on full-resolution slices, since we are aiming for voxel or sub-voxel accuracy. However, it sometimes helps to first get an approximate transform using down-sampled slices, and the code supports this option.
- `scipy.optimize` is used to perform the maximization. Like any multi-dimensional optimizer, it can get stuck in a local optimum, so re-running from slightly different starting transforms is sometimes required to get the best result.
- A utility program, called `CheckAlignment`, is also provided, which extracts a slice of arbitrary orientation from the source and transformed-target volumes, so that the alignment can be visually inspected. This program was used to generate most of the images in this report.

Section 4 gives a snapshot of the results.

4. Results

4.a Scroll 4, 20231107190228-to-20231117161658

The mutual information score was increased from 0.22 to 0.47.

The improved transform is

```
[[ 4.14521304e-01,  1.43281857e-03, -9.78993656e-04,  7.99755617e+01],
 [-1.40450330e-03,  4.14519993e-01,  2.84106801e-03, -4.15029954e+01],
 [ 9.42907081e-04, -2.82785109e-03,  4.13311730e-01,  7.15943589e+01],
 [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  1.00000000e+00]]
```

Example result:

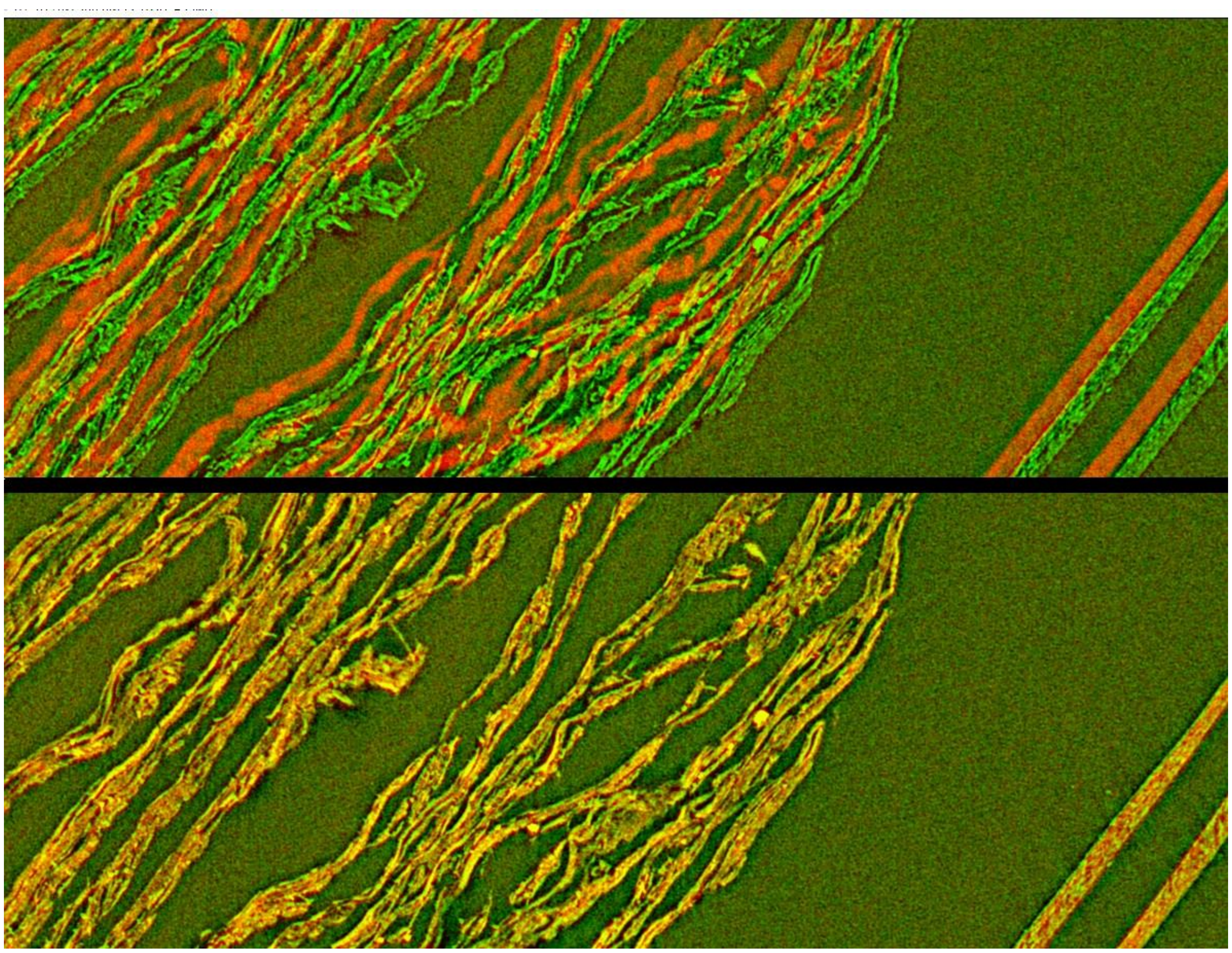

Figure 5. Poor initial alignment in a part of slice 4000 of Scroll 4 (top) is improved by the registration procedure (bottom). The source volume is drawn in green and the transformed target volume in red. The areas where they overlap appear as yellow/orange. Full images are available in the project repository.

4.b Scroll 3, 20231027191953-to-20231117143551.json

The mutual information score was increased from 0.41 to 0.53.

The improved transform is:

```
[[ 4.14653255e-01, -6.14104505e-05, -7.28389136e-05, -6.32924527e+01],
 [-1.59538720e-04,  4.14657455e-01,  5.41603057e-05, -2.53168325e+02],
 [ 2.04095988e-04, -8.22682284e-06,  4.13435755e-01,  6.04319148e+01],
 [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  1.00000000e+00]]
```

Example result:

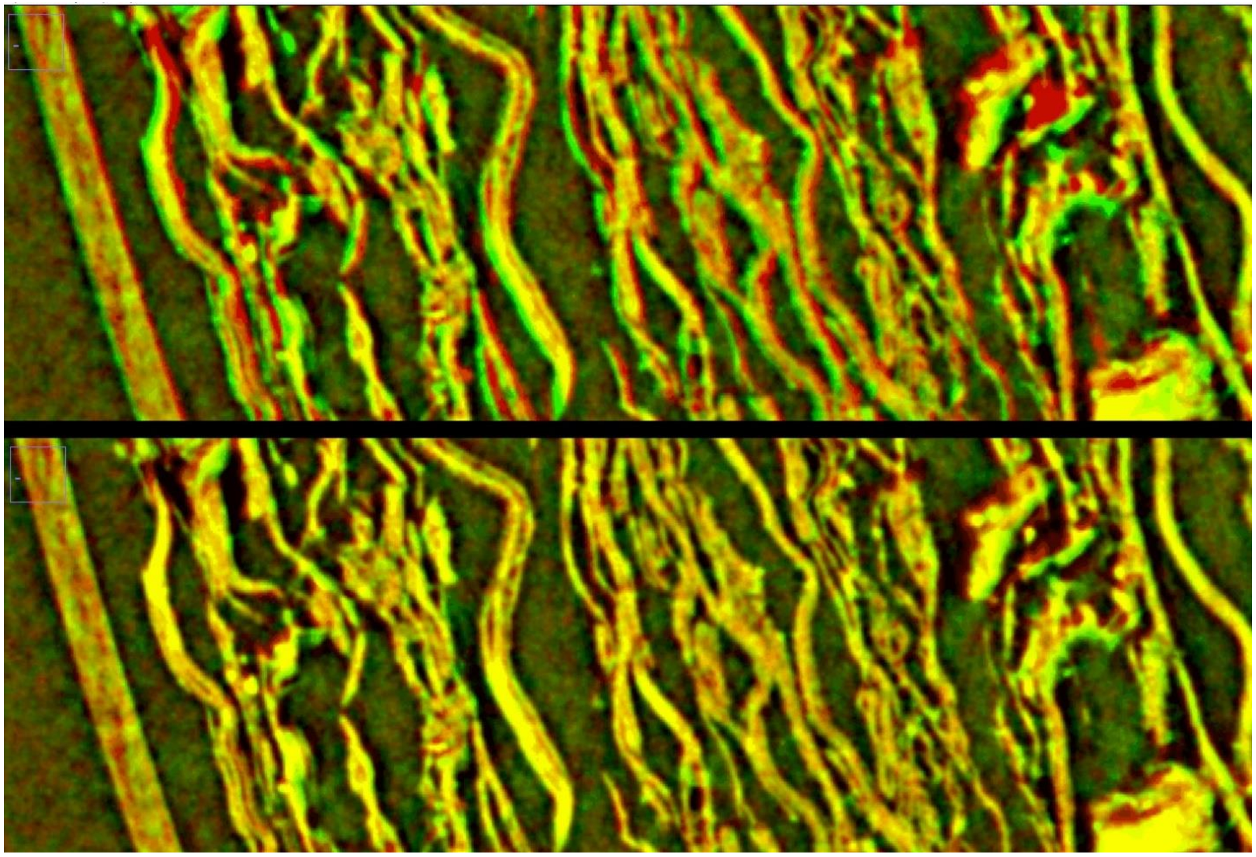

Figure 6. The fairly good initial alignment in slice 11000 of Scroll 3 (top) is improved by the registration procedure (bottom), as evidenced by the reduction in red and green fringes. The colors are explained in the Figure 5 caption. Full images are available in the project repository.

4.c Scroll 3, 20231027191953-to-20231201141544.json

The mutual information score was increased from 0.75 to 0.78.

The improved transform is

```
[[ 1.000145652e+00, -1.424050975e-05,  3.174470350e-05,  9.919033236e-01],
 [ 2.660178919e-05,  1.000302722e+00, -6.659096342e-05, -3.157709230e-01],
 [-9.715877060e-05,  1.644539341e-04,  9.999427021e-01, -8.127077228e+00],
 [ 0.000000000e+00,  0.000000000e+00,  0.000000000e+00,  1.000000000e+00]]
```

Example result:

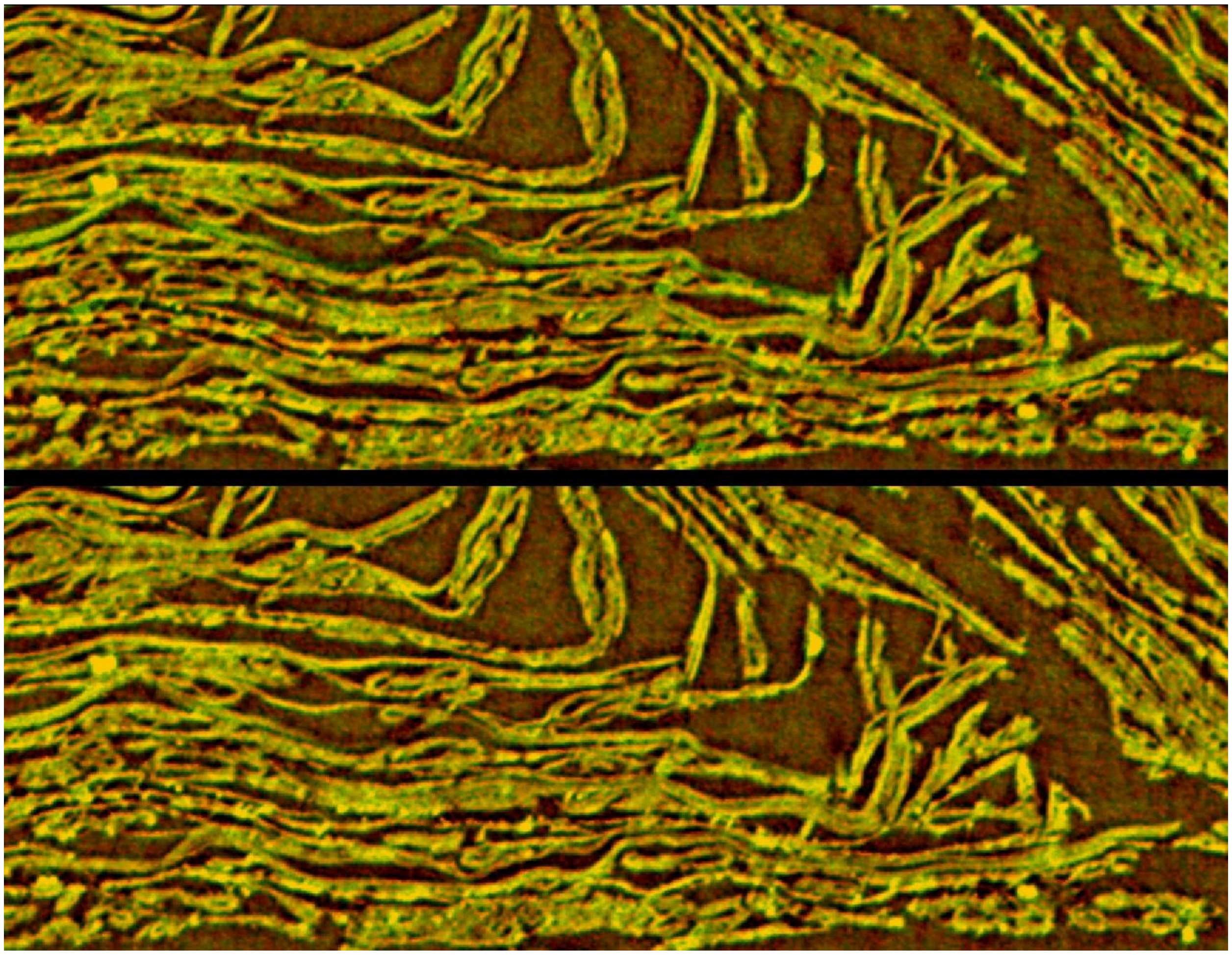

Figure 7. The initial alignment was very accurate in this case, but registration still produces a slight visible reduction in the red and green fringes, indicating improvement. The colors are explained in the Figure 5 caption. Full images are available in the project repository.