Robust Automation for Connectomics
Argonne National Laboratory - Summer 2021

Marium Yousuf
Supervisor: Mark Hereld

August 2021

1 Introduction

3D reconstruction of neuronal structure facilitates an in-depth analysis of brain regions by allowing visualization of neural anatomy. The reconstruction of biological systems is desirable because it allows extraction of specific regions of interest, which can be further used to evaluate and treat any abnormalities found in the underlying structure[1]. TrakEM2 is a software that provides a mean for such reconstruction using electron microscopical images [2]. Multi-slice imaging techniques are often used to detect and explore the 3D reconstruction of anatomical brain features [1] and TrakEM2 uses such images for 3D reconstruction from serial section image data sets [2].

For this project we used Transmission Electron Microscopy (TEM) images, which are high-resolution images. These images are often used to investigate biological structures at sub-cellular level [3], allowing structural analysis at the nanometer level [4]. TEM images are the highest resolution images available yet [3] and are able to reveal largest macro-molecular assemblies and other sub-cellular structures inside cells. We use TEM tiles that scans of mouse brain slices to conduct all our analysis and experiments.

The project focuses on investigating the accuracy in the alignment of these tiles, for which we need the toolchains such as stitching, segmentation, registration, etc. However, the current approaches to these toolchains and model extraction involve human intervention, which limits the processing of the high-throughput TEM images. To overcome this limitation, we are working on new efficient approaches that automate the processing of large datasets of these multi-tile multi-slice images to create 3D neuronal structure.

We use overlap of the tiles to determine how well they align. This paper describes our procedure of finding overlaps of tiles from the same image using a tile layout determined from TrakEM2. We then use these overlaps to determine how their spatial detail differs and find patterns in their disparities by selecting regions of interests that cover the overlapping space in every direction. Eventually, we compare the errors in our experiment with those from TrakEM2 that requires manual input to align and deform the images for 3D reconstruction.

The flow of this paper is as described: Section 2 elaborates on the problem and provides the background information necessary to understand the experiments conducted and the data used. Section 3 discusses the experiments and their validation techniques. Section 4 concludes the setup and includes a discussion of next steps.

Tile	Global Coordinates	
r1c1	0.0	0.0
r1c2	11081.8	-10.1
r2c1	90.4	11012.2
r2c2	11027.6	11019.8
r3c1	57.6	22067.3
r3c2	11070.7	22102.5

Table 1: Table consisting of the tile name, position in the global system, and corresponding global coordinates.

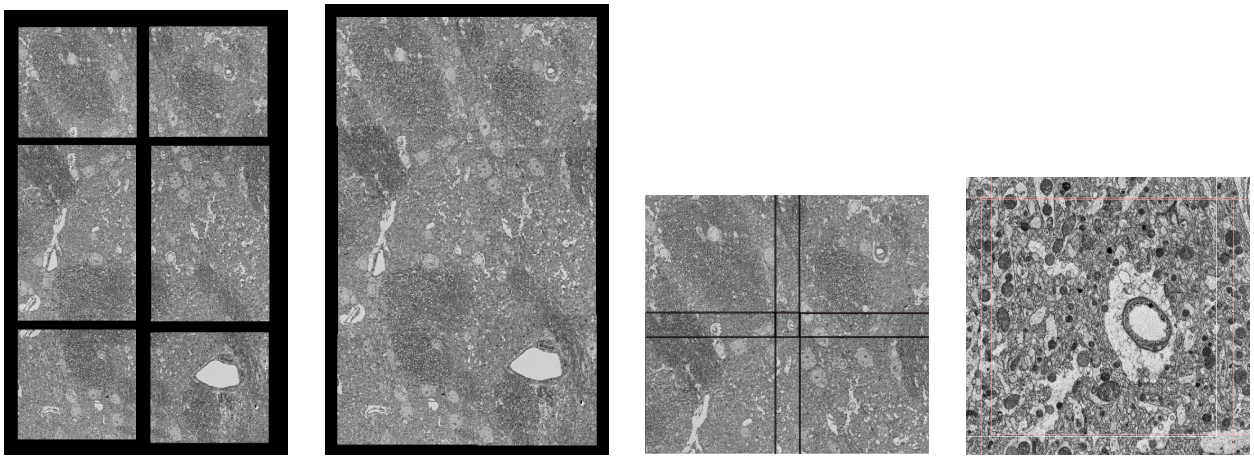
2 Methods

2.1 Background and Data

TrakEM2 [3] software allows 3D reconstruction from serial section TEM images and yields near-perfect registration and stitching of multi-tile multi-slice images. TrakEM2 provides both manual and automatic deformation correction tools for the images using linear and non-linear transformations [2]. Additionally, TrakEM2 provides access to all the transformations used on the data and so using the coordinate transforms from our data of TEM image tiles on TrakEM2, we collected the positions of the upper-left coordinates for each tile as a part of the entire image. We refer to these coordinates as *global* coordinates. These global coordinates of six tiles make up a full image with the dimension of 12288×12288 pixels.

These six tiles are processed on TrakEM2 to automatically stitch them together and then use the metadata consisting of global offsets and translation vectors to conduct our analysis. These tiles are referred to as r$ic j$, where ri refers to the ith row and cj refers to the jth column (refer to Table 1).

Using these global offsets, we determine the overlapping sections between the tiles and investigate the error in the alignment. Figure 1 shows the six tiles (a) that stitch into a single structure (b), while (c) shows the first four tiles (r1c1, r1c2, r2c1, r2c2) and the outline of their overlapping regions. Figure 1(d) shows a zoomed in overlap of the four tiles mentioned. Section 2.2 gives more information on the positioning and merging overlapping image tiles into a coherent image.



(a) (b) (c) (d)

Figure 1: Sub-figures (a)-(d) highlight the stitching and stacking functions of the TrakEM2 interface. (a) All six tiles that construct a full image. (b) Stitched image from TrakEM2 of passed-in six tiles. (c) Overlap outline of tiles (r1c1, r1c2, r2c1, r2c2). (d) Zoomed in overlap of tiles (r1c1, r1c2, r2c1, r2c2).

From the six tiles available, we focus on [r1c1, r1c2, r2c1, r2c2] and their 1259×1207 overlap (as shown in Figure 1(c)-(d)). From now onward, we refer to these tiles as $[x_0, x_1, x_2, x_3]$. The four overlapping fragments are produced by computing the upper-left, upper-right, lower-left, and lower-right global coordinates of the overlapping regions and then translating those coordinates to local points on each tile. Using the local coordinates, we crop corresponding fragments from each tile. Now, to compare each tile with the all other tiles, we compute a combination of x_i yielding six pairs $(x_0, x_1), (x_0, x_2), (x_0, x_3), (x_1, x_2), (x_1, x_3), (x_2, x_3)$, such that $p_0 = (x_0, x_1), ..., p_5 = (x_2, x_3)$ such that . The experiments in this paper focus to find and verify the minimum standard deviation in the alignment of each pair of fragments.

2.2 Band-Pass Filter

For a reliable comparison of the overlaps, we need to ensure that each fragment matches well with the other. While scanning TEM slices, it is highly unlikely to have the scans of the same section to match exactly. The

reasons for this could include rotation at some angle, different or uneven illumination, zoom-ins or zoom-outs, etc [5]. To denoise and eliminate the changes in the scans, as well as identify the overlaps accurately, we use the band-pass filtering method called the difference of Gaussian.

The band-pass filtering requires a range (or band) of signal frequencies and uses that range to attenuate the frequencies external to that range [6]. This technique enhances the key features and edges by creating two blurred versions of the same image and then subtracting the more blurred version of the original from the less blurred version. Consequently, making it possible to distinguish features from the noise as well as reduce the noise from the fragments.

The difference of Gaussian methods require values for low sigma and a high sigma and uses a Gaussian kernel to determine the range of frequencies to attain. The low sigma acts as the smaller standard deviation for the Gaussian kernel for all axes, while the high sigma acts as a larger standard deviation for the Gaussian kernel [7]. Figure 2 shows four examples of the same fragment with different sigma levels. The resulting image has high-frequency components in the original image eliminated by the low-sigma Gaussian kernel.

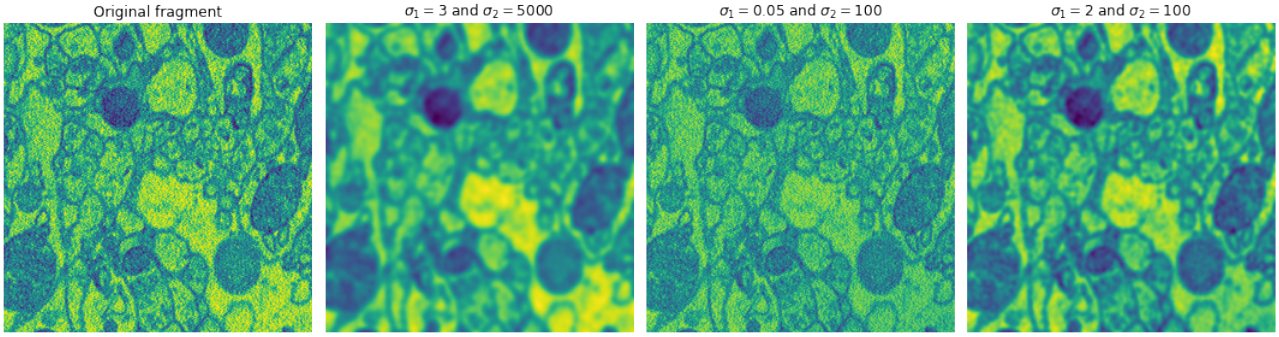
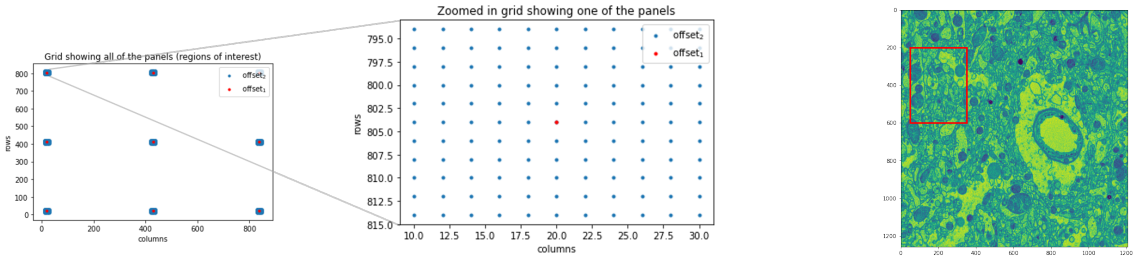


Figure 2: Four images showing how the values of low sigma and high sigma affect the Gaussian kernel in the difference of Gaussian band-pass filtering technique.

To determine the deviations in the four fragments accurately, we focus on 9 regions of interest. We produce a 3×3 grid spanning over the entire 1259×1207 fragments. For this grid, we fix certain offsets and using those fixed offsets as the middle coordinates of the grid, we compute a 11×11 grid. Figure 3(a) shows the 9 regions of interest (left) and a zoomed in grid for the upper left region (right).



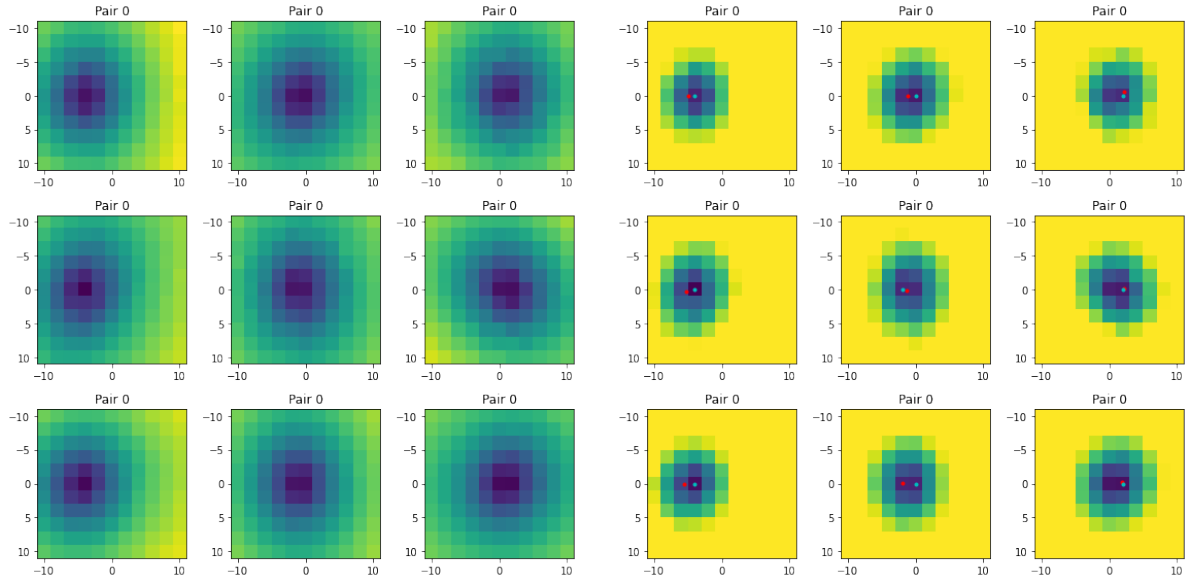
(a) (b)

Figure 3: (a) A 3×3 grid on the left shows the accumulation of points at 9 locations of the whole fragment. Each fragment is compared with the other paired fragment to determine how they differ in all these locations. The zoomed location on the right shows all the offsets; the red offset is the fixed location for one fragment while the blue offsets are for the paired fragment to scan around the first fragment. (b) A 400×300 window displayed on a 1259×1207 fragment.

2.3 Alignment and Optimization

Using the grid as above, one fragment from each fragment pair is fixed at the red offset, while the second fragment from the pair is scanned around it at all blue offsets, starting from the upper-left. This scanning involves matching up the both fragments and computing the standard deviation of the resulting image from the difference of Gaussian technique mentioned above.

We take both fragments represented as arrays and use the difference between them as the input to the difference of Gaussian technique with low sigma = 3 and high sigma = 100. The scanning window is chosen to be 400×300 and the computed standard deviation result from each red and blue offset is stored. Figure 3(b) shows how the 400×300 compares to the entire overlap.



(a) (b)

Figure 4: For pair (x_0, x_1), these figures show the comparison standard deviations for all 9 panels in the overlap of tiles r1c1 and r1c2. (a) The darkest sections indicate the minimum standard deviation. (b) The red dot is the minimum standard deviation computed using the polynomial fitting. The cyan dot is the minimum standard deviation computed using the ad-hoc computation of the standard deviation.

We use two approaches to determine the true minimum of the 2D data of standard deviations. First, we use 2D polynomial fitting by least squares of the 11×11 array of the standard deviations and use the resulting fit to evaluate a 2D polynomial. Second, we evaluate the row and column index from the array that yields the minimum standard deviation. Figure 4(a) shows the plots for one pair of fragments (out of six) and their computed standard deviations for the entire grid, while Figure 4(b) shows the points in red and cyan, with former being the minimum using the polynomial fitting and later from our ad-hoc computation of the minimum.

Considering the 9 regions of interest and 6 pairs of fragments, we investigated how each pair related to other pairs in each region. Figure 5 shows all 9 panels and for each panel it shows how the pairs correspond to each other. Since the tiles overlap with each other due to some underlying transformation, we investigated further to determine which pairs can be defined as linear combination of other pairs. Picking tile x_0 with global offset $(0, 0)$ as the tile of interest, we determined the following linear combinations:

$$p_0 = -x_1, \quad p_1 = -x_2, \quad p_2 = -x_3, \quad p_3 = x_1 - x_2, \quad p_4 = x_1 - x_3, \quad p_5 = x_2 - x_3.$$

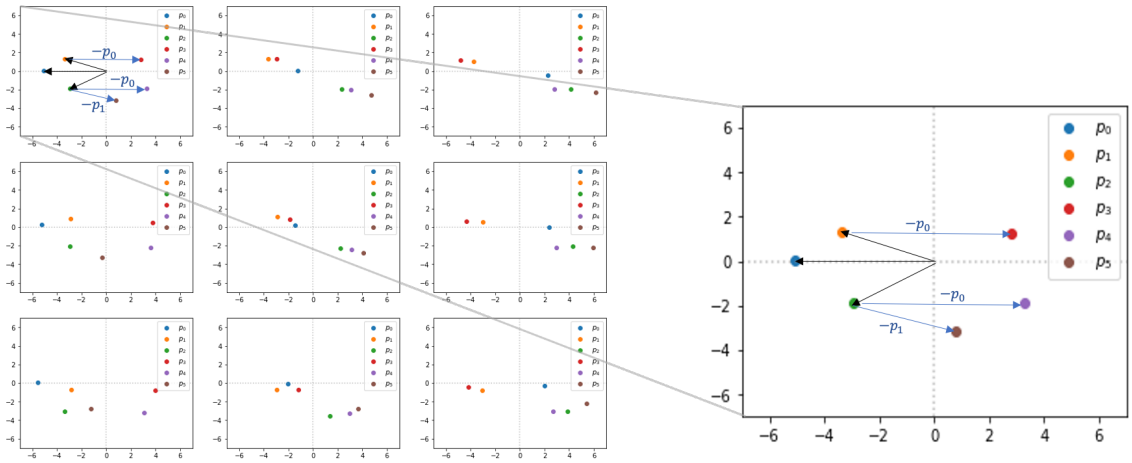
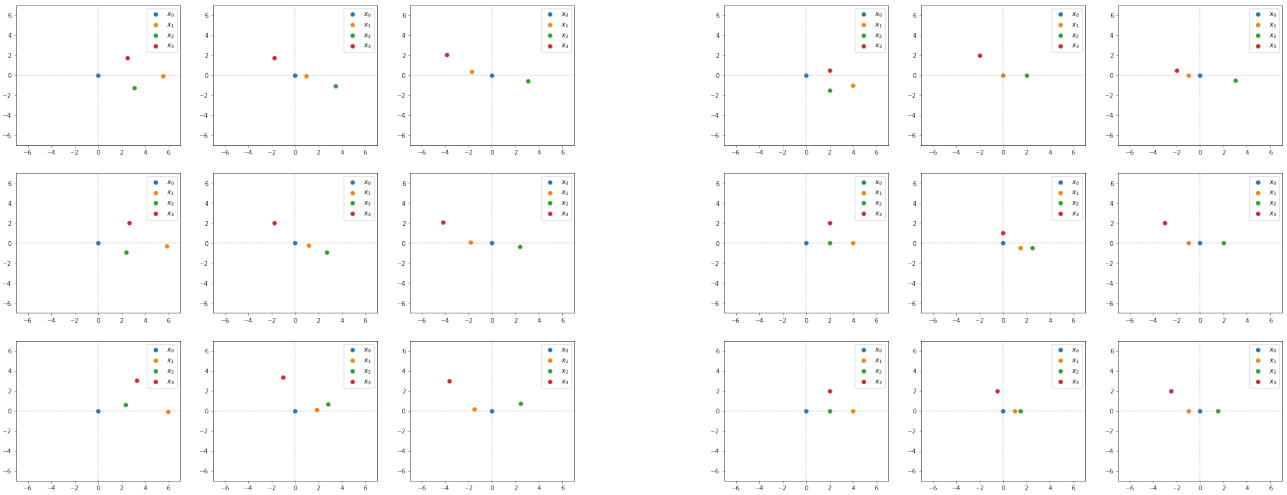


Figure 5: Six pairs $p_0, ..., p_5$ shown as linear combinations of first three tiles x_0, x_1, x_3.



(a) (b)

Figure 6: Using the coordinates of the points p_0 as shown in Figure 5 and the linear combinations above, we find the least squares solution for $X = [x_1, x_2, x_3]$. This figure shows the solutions to X with both approaches (a) polynomial fitting, and (b) ad-hoc minimum.

3 Experiments and Results

To compare how the dot plot from Figure 6(a) compares with the original tiles, we sampled out fragments from the exact locations of 9 regions of interest from the original overlaps. To do this, we filtered the sections of original fragments to binary values using a threshold. That is, if the values in the array representing the original fragment were below a certain threshold, the values were replaced with 0, otherwise 1. Figure 8 shows a comparison of the images where the one on the left is the original fragment and the one on the right is the binary image. Observing the elliptical patches in the tiles in the binary image, we fitted and ellipse to interactively collected data points from all fragments and then extracted ellipses across all four tiles for all 9 panels. These ellipses are shown in Figure 7.

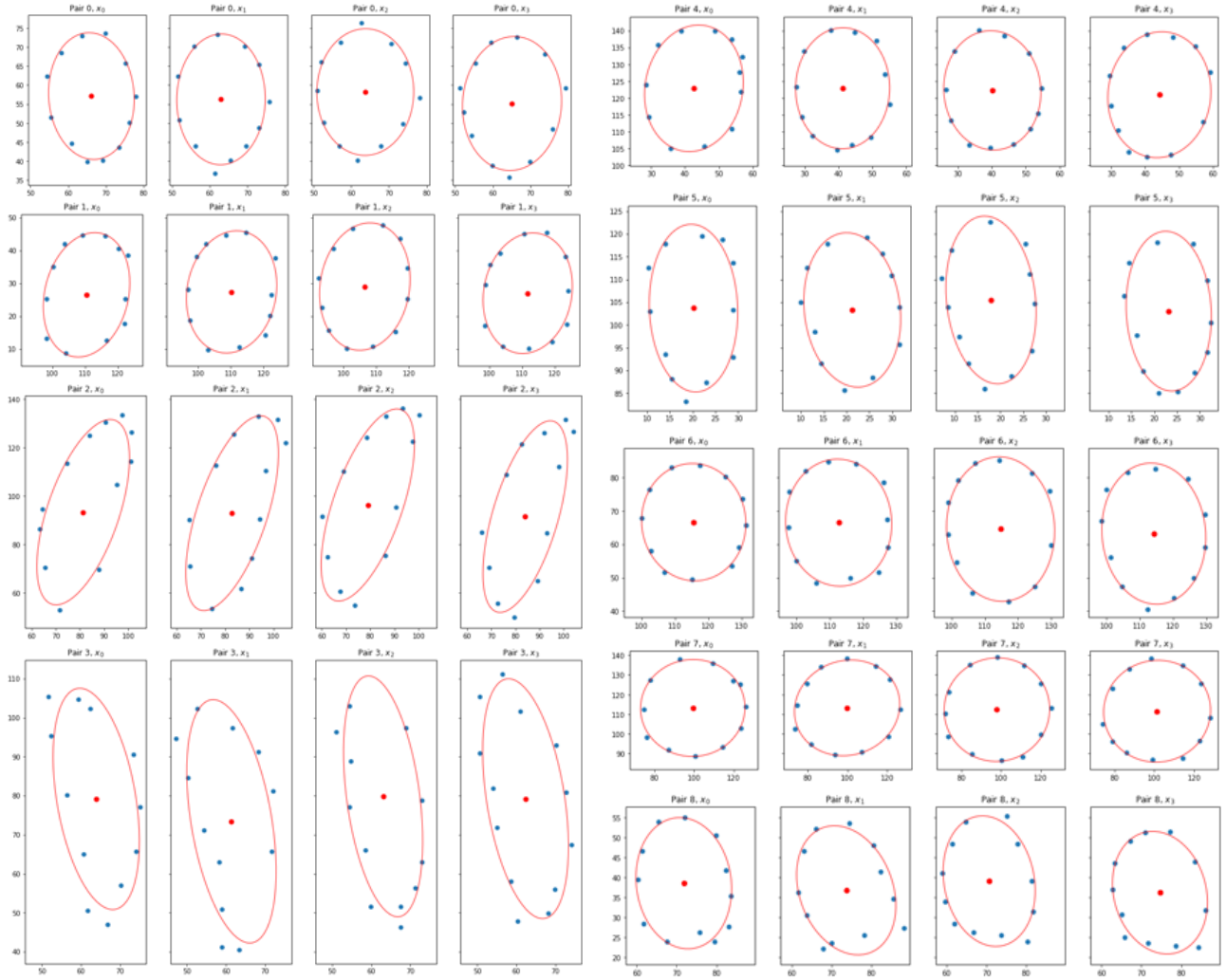


Figure 7: This figure shows all the manually selected elliptical patterns observed in all four tiles for all 9 panels.

3.1 Sensitivity Analysis

Considering the 9 panels for each of the pairs and their overlaps, we investigated the uncertainty in the parameter values derived from the center (x0, y0) of the ellipses. We did this to quantify how accurate the computed ellipses' centers are.

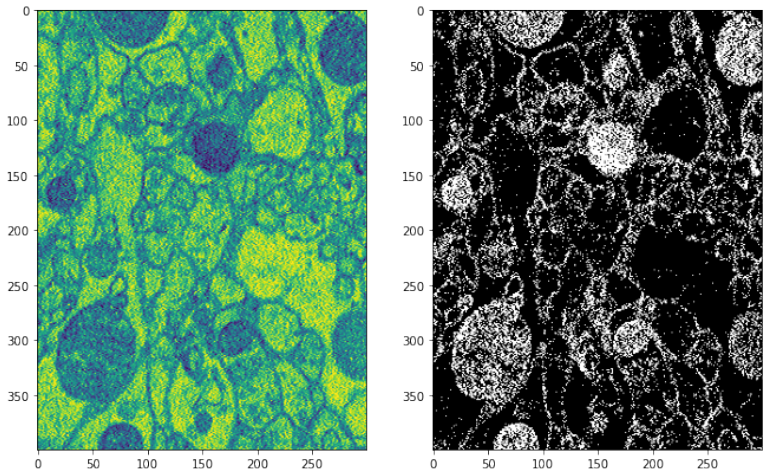
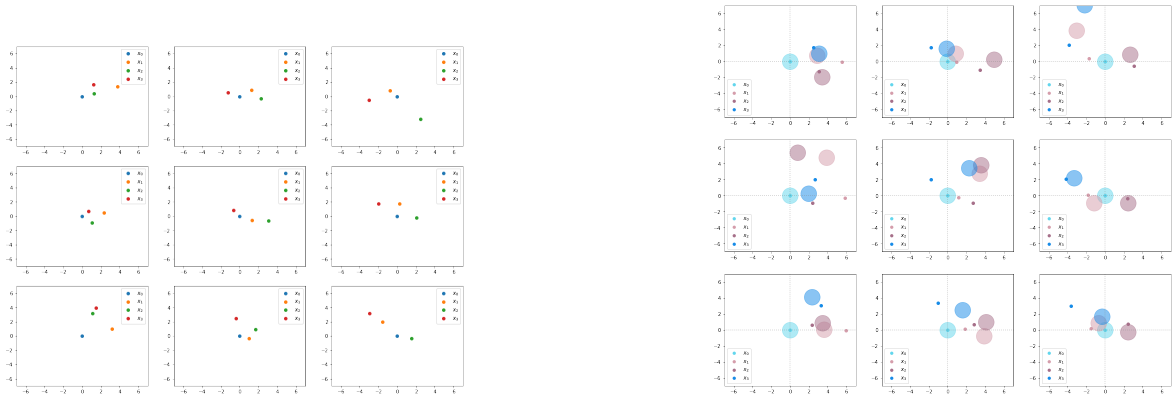


Figure 8: This figure shows a comparison of an original fragment and a filtered binary image. Binary images like these were used to extract the ellipses for all 9 panels across the four tiles.



(a) (b)

Figure 9: (a) shows a dot plot of fitted ellipse's centers. (b) shows the comparison of this dot plots with the solution found from the polynomial fitting. The dots are the solutions and the circles are the ellipse's centers after sensitivity analysis.

Computing the coordinates of the centers of the ellipses fit, we computed an average norm difference between the coordinates in Figure 6(a) and coordinates in Figure 9. The difference comes out to be approximately 1.863, which is less than the mean pixel displacement error of 2.257 from TrakEM2.

References

[1] Harvey E Cline, CL Dumoulin, HR Hart Jr, William E Lorensen, and S Ludke. 3d reconstruction of the brain from magnetic resonance images using a connectivity algorithm. *Magnetic resonance imaging*, 5(5): 345–352, 1987.

[2] Ulrich Meissner, Ewald Schröder, Dirk Scheffler, Andreas G Martin, and J Robin Harris. Formation, tem study and 3d reconstruction of the human erythrocyte peroxiredoxin-2 dodecahedral higher-order assembly. *Micron*, 38(1):29–39, 2007.

[3] Stephan Saalfeld, Albert Cardona, Volker Hartenstein, and Pavel Tomančák. As-rigid-as-possible mosaicking and serial section registration of large sstem datasets. *Bioinformatics*, 26(12):i57–i63, 2010.

[4] Heiner Friedrich, Peter M Frederik, Gijsbertus de With, and Nico AJM Sommerdijk. Imaging of self-assembled structures: interpretation of tem and cryo-tem images. *Angewandte Chemie International Edition*, 49(43): 7850–7858, 2010.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[6] scikit-image development team. Band-pass filtering by difference of gaussians, . URL `https://scikit-image.org/docs/stable/auto_examples/filters/plot_dog.html`.

[7] scikit-image development team. Difference of gaussians, . URL `https://scikit-image.org/docs/stable/api/skimage.filters.html#skimage.filters.difference_of_gaussians`.