Robust Automation for Connectomics
Argonne National Laboratory - Summer 2022

Marium Yousuf
Supervisor: Mark Hereld

August 2022

1 Introduction

Transmission Electron Microscopy (TEM) images are widely known to capture the cellular-level information of biological structures. Having an insight on how these images stack on top of each other to form a 3D structure provides a way to make organ-level observations and discoveries. Previously, we had explored how the TrakEM2 software stitches multi-tile, multi-slice images at a given configuration and what alignment error it produces. We studied 4 tiles that overlap in corners and focused on this 4-way overlap to investigate how the image points differ with respect to each other in each tile. We did this by manually choosing a handful of points in 9×9 segments of the 1200×1200 overlap that take a shape of an ellipse. We used these points to then fit an elliptical contour to retrieve an error metric comparable to that of TrakEM2.

Now, taking this further, we adapted the Scale Invariant Feature Transform (SIFT) algorithm to gather several data points instead of a few manually chosen points. We fix one tile and find matching SIFT descriptors in other tiles(s) that meet some threshold criteria. We picked our threshold to be two-fold: we limit the match in the other tile(s) to be within some window size, and we limit the norm between SIFT descriptors to be less than a set number.

Using these SIFT matches, we first studied how the difference between these matches is normally distributed as well as investigated the relationship between their difference in both x- and y-direction by using linear regression. Observing the results from these two investigations, we hypothesized that the matched image points can be explained further by optical distortion. We used a Brown-Conrady distortion model [1, 2] to test our hypothesis.

2 Data

We use SIFT algorithm on TEM mouse brain slices to generate our data in terms of image coordinates. We use these coordinates together with the metadata generated from TrakEM2 software to conduct our experiment. We used 4 tiles for our experiments, hereby referred to as T0, T1, T2, and T3, such that (T0, T1) are horizontally overlapped, (T0, T2) and (T1, T3) are vertically overlapped. We primarily focused on 3 kinds of settings: 1) taller overlap between (T0, T1), 2) wider overlap between (T0, T2), and a 4-way overlap between (T0, T1, T2, T3). Each tile image has a dimension of 12288×12288 pixels. The metadata from TrakEM2 software used is the location of each tile with respect to each other. Table 1 lists these location offsets.

Tile	Global Coordinates	
T0	0.0	0.0
T1	11081.8	-10.1
T2	90.4	11012.2
T3	11027.6	11019.8

Table 1: Tiles and their corresponding TrakEM2 location offsets.

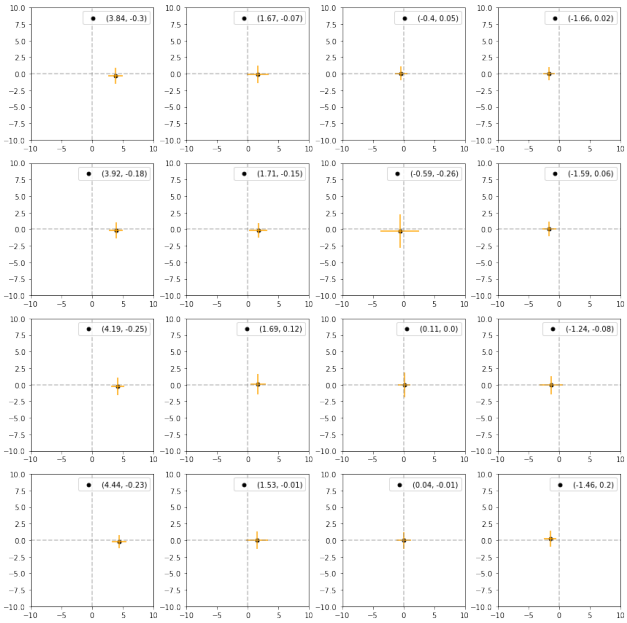
We use subset of tiles, T, and use the SIFT algorithm to find best matching coordinates for our distortion model as follows:

- find SIFT features for each tile/image independently
- fixing one tile, find a data point in the other tile(s) that holds both:
  - the norm between the keypoint descriptors of the feature from first and the other tile(s) is within a threshold t. We set t = 200 for our experiments.
  - each match found is within w pixels away from the data point of the fixed tile. We set w = 29 for the first two experiments and w = 10 for the experiment using the distortion model.

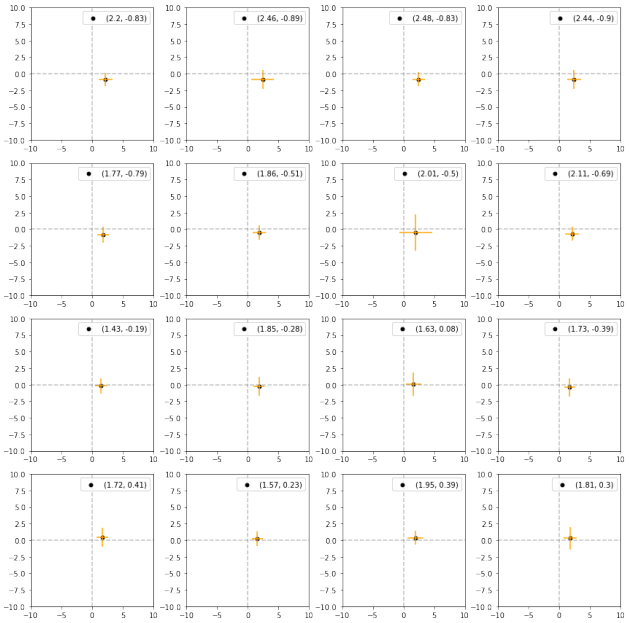
3 Initial Experiments

3.1 Pixel differences between SIFT image points and their matches

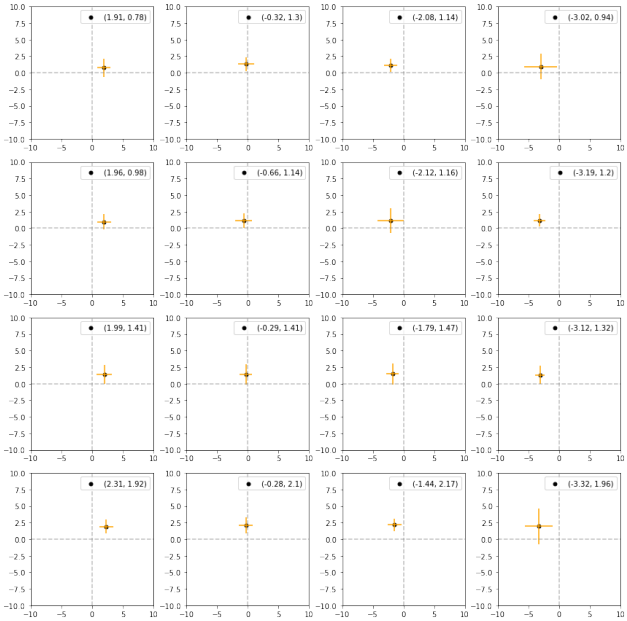
Let Δ_x and Δ_y be the difference between the SIFT coordinates from the fixed tile and their best matching SIFT coordinates from the other tile(s). We use the 4-way overlap between tiles T0, T1, T2, and T3 and their respective SIFT coordinate matches with the thresholds t = 200 and w = 59 to determine how the coordinates deviate from each other. To observe how Δ_x and Δ_y were distributed among all the matches, we conducted a pairwise experiment between all 2 sets of tiles (total 6 pairs) by distributing the 1200×1200 overlap into 16 300×300 patches and observed the Gaussian distribution separately for each patch. Figure 1 shows how Δ_x and Δ_y are distributed for each of the pairs.



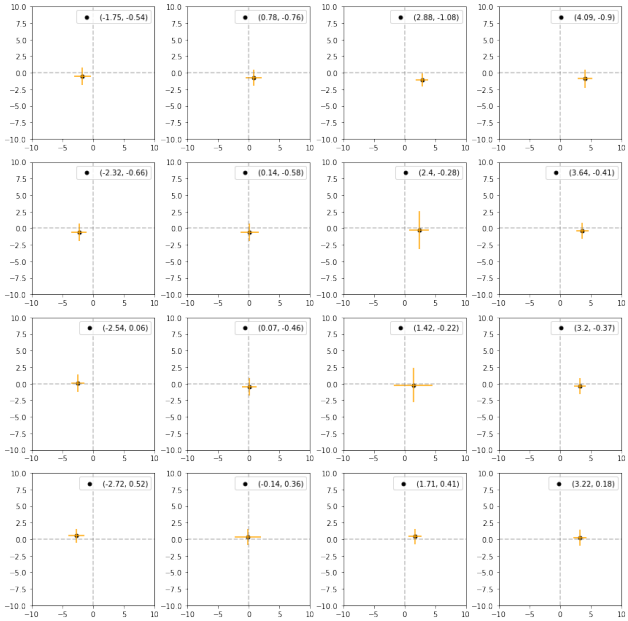
(a) T0 and T1



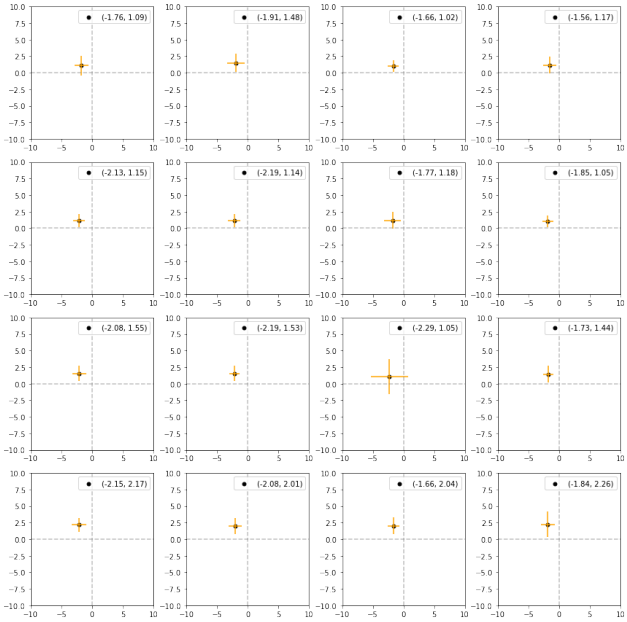
(b) T0 and T2



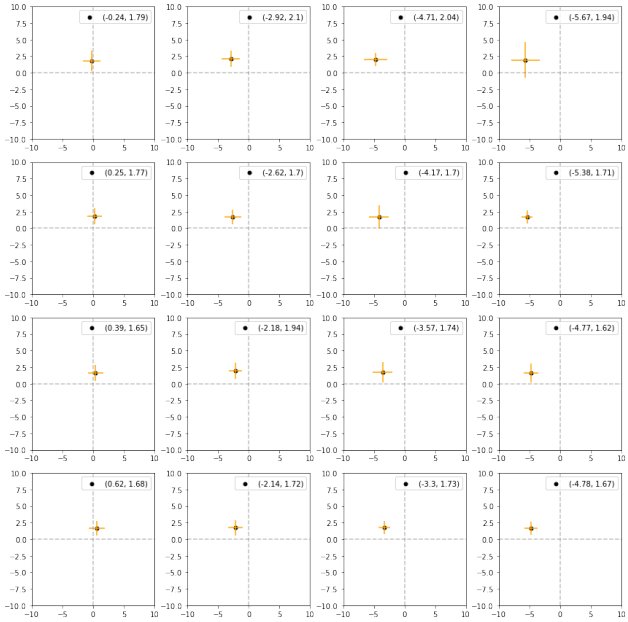
(c) T0 and T3



(d) T1 and T2



(e) T1 and T3



(f) T2 and T3

Figure 1: The dots in each plot are the mean Δ_x and Δ_y and horizontal and vertical orange lines represent the standard deviation of Δ_x and Δ_y, respectively.

3.2 Linear Regression

We also used linear regression to add to our observation of how Δ_x and Δ_y are distributed for all the matches. To do this, we combined the rows and columns to observe how the image points differ with horizontally and vertically. Table 2 shows the results for pairwise comparisons of image points for tiles T = [T0, T1, T2, T3].

Tiles	Rows	Slope	Intercept
T0T1	[0, 300)	-0.0062	4.61
	[300, 600)	-0.0065	4.73
	[600, 900)	-0.0061	4.81
	[900, 1200)	-0.0064	4.99
T0T2	[0, 300)	0.0002	2.25
	[300, 600)	0.0004	1.72
	[600, 900)	0.0003	1.48
	[900, 1200)	0.0003	1.6
T0T3	[0, 300)	-0.0057	2.53
	[300, 600)	-0.0058	2.49
	[600, 900)	-0.0057	2.59
	[900, 1200)	-0.006	2.91
T1T2	[0, 300)	0.0067	-2.54
	[300, 600)	0.0068	-3.13
	[600, 900)	0.0062	-3.15
	[900, 1200)	0.0068	-3.61
T1T3	[0, 300)	0.0004	-1.93
	[300, 600)	0.0005	-2.27
	[600, 900)	0.0003	-2.23
	[900, 1200)	0.0005	-2.21
T2T3	[0, 300)	-0.0062	0.29
	[300, 600)	-0.0063	0.8
	[600, 900)	-0.0058	0.93
	[900, 1200)	-0.0058	1.15

(a) Horizontal comparison

Tiles	Columns	Slope	Intercept
T0T1	[0, 300)	0.0001	-0.3
	[300, 600)	0.0001	-0.09
	[600, 900)	0.0001	-0.13
	[900, 1200)	0.0002	-0.05
T0T2	[0, 300)	0.0014	-1.22
	[300, 600)	0.0012	-1.1
	[600, 900)	0.0014	-1.06
	[900, 1200)	0.0013	-1.18
T0T3	[0, 300)	0.0013	0.46
	[300, 600)	0.0009	0.96
	[600, 900)	0.0012	0.78
	[900, 1200)	0.001	0.74
T1T2	[0, 300)	0.0013	-0.96
	[300, 600)	0.0012	-1.09
	[600, 900)	0.0015	-1.16
	[900, 1200)	0.0011	-1.03
T1T3	[0, 300)	0.0012	0.76
	[300, 600)	0.0006	1.17
	[600, 900)	0.001	0.74
	[900, 1200)	0.0012	0.75
T2T3	[0, 300)	0.0	1.71
	[300, 600)	-0.0004	2.09
	[600, 900)	-0.0002	1.94
	[900, 1200)	-0.0004	1.95

(b) Vertical comparison

Table 2: Table showing how the SIFT image points for tiles in column one differ horizontally and vertically. The points deviate more horizontally across the image than they do vertically.

Observing the results of linear regression on Δ_x and Δ_y in Table 2, we hypothesized that the difference between the SIFT matches can be explained by optical distortion from the transmission electron microscope that captured these image slices. To test our hypothesis, we used a first-order Brown-Conrady's method discussed further in Section 4.

4 Distortion Model

Let M be the (x, y)-coordinates for tiles in T that takes shape of $n \times |T|$, where n is the number of points that are best SIFT matches. We used M with the first-order Brown-Conrady's method to model radial distortion, which comprises the following equations for the model:

$$x_u = x_c + \frac{x_d - x_c}{1 + Kr^2} \tag{1}$$

and

$$y_u = y_c + \frac{y_d - y_c}{1 + Kr^2}, \tag{2}$$

where (x_u, y_u) are the undistorted points, (x_d, y_d) are the distorted points, (x_c, y_c) is the optical center, K is the radial coefficent, and r is the Euclidean distance between the distorted point and the optical distortion.

To find optimal K and (x_c, y_c), we use the coordinates from M by letting:

- X_i be the point we pick from a given tile,
- X_{ii} be the location of X_i in the overall 12288×12288 image,
- X_{iii} be the undistorted point in 12288×12288 computed using the Brown-Conrady's equations (1) and (2), and
- X_{iv} be the undistorted, transformed point in the overall stitch returned by TrakEM2. Since we are only considering the first 4 tiles, the TrakEM2 stitch has dimensions 24576×24576 , since each tile is 12288×12288.

We then compare X_{iv} for each tile by first computing the average, $\bar{X}_{iv}$, coordinates over the tiles and then computing the following measure that we minimize:

$$D = \sqrt{\sum_j \left(X_{iv_j} - \bar{X}_{iv_j}\right)^2}, \tag{3}$$

where $j \in T$.

4.1 Optimization

We use a direct search method, Nelder-Mead method [3], to find values for radial coefficient K and optical center (x_c, y_c) by optimizing (3) using best-matched coordinates for a given T. Here we elaborate and discuss results from two overlapping settings:

- a tall overlap between horizontally overlapping tiles T0 and T1 of size 7688×1207, and
- a wide overlap between vertically overlapping tiles T0 and T2 of size 1259×12198, and

We explore the space of $\Delta_x \in [-50, 50]$ and $\Delta_y \in [-10, 10]$, by fixing the origin of the fixed tile and moving the origin about with Δ_x and Δ_y, to observe how the radial coefficient and the optical center behave with the shifts with respect to the Δ_x and Δ_y.

4.2 Regulizer

Since the dimension for each tile is 12288×12288, we expect the optical center to be about $(6144, 6144)$, such that computing for K using (1) or (2) with average Δ_x and Δ_y computed from the total SIFT-coordinate matches in M, we get $K = -2 \times 10^{-11}$. However, when we optimize (3), we see unexpected behavior in values for K as well as (x_c, y_c). For example, K values often tend to a value very close to zero $c \times 10^{-19}$ for any c, for example, and x_c and y_c reaching negative or positive values in millions. To regularize this behavior and favor values close to the expected center, we introduced the following regularizers, R^σ such that $\sigma = \{1, 2, 4\}$:

$$R^4 = \left(\left(\frac{x_{c_i}}{6144} - 1\right)^2 + \left(\frac{y_{c_i}}{6144} - 1\right)^2\right)^2, \tag{4}$$

$$R^2 = \left(\frac{x_{c_i}}{6144} - 1\right)^2 + \left(\frac{y_{c_i}}{6144} - 1\right)^2, \tag{5}$$

and

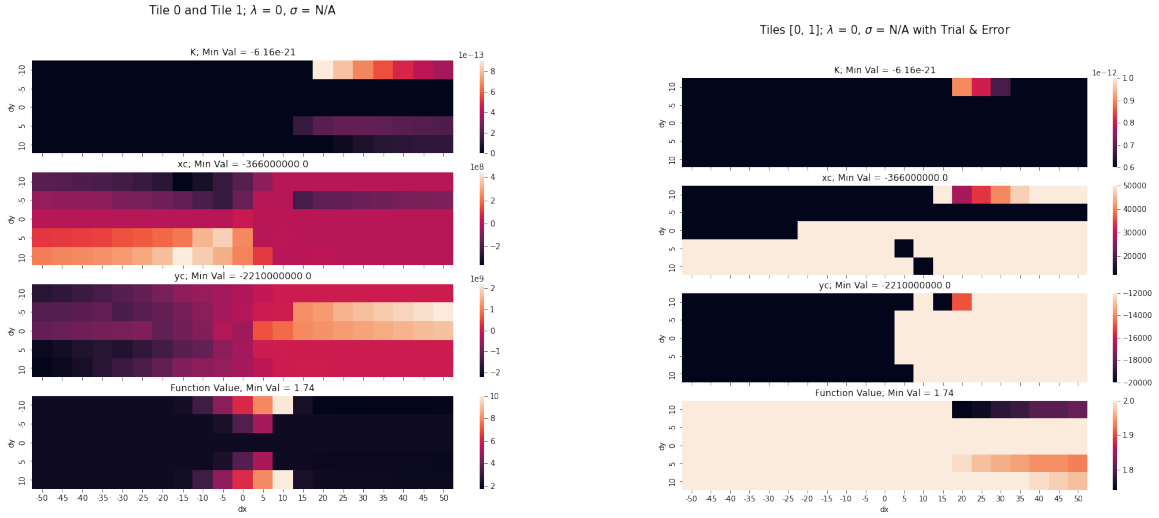
$$R^1 = \sqrt{\left(\frac{x_{c_i}}{6144} - 1\right)^2 + \left(\frac{y_{c_i}}{6144} - 1\right)^2}, \tag{6}$$

5 Analysis

5.1 Setting 1 - Tile 0 and Tile 1

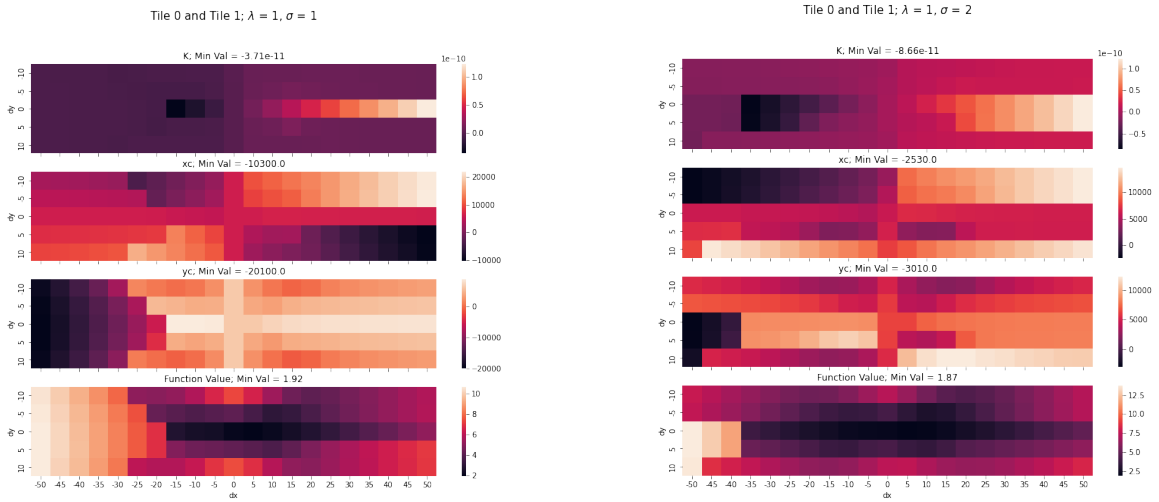
Figure 2 shows heatmaps of how K, x_c, and y_c behave without regularizer term while moving the origin offset of the second tile, T1 in this case. Looking at the bottom heatmap for optimizer function value, most values are between 2 and 3, with the highest function value being about 10. The heatmap doesn't inform well which Δ_x and Δ_y give the optimal K and (x_c, y_c). Limiting the function values to be under 2, we see that the minimum occurs at $\Delta_x = 20$ and $\Delta_y = -10$. With trial and error, we find out that at this location, $K \approx 0.9 \times 10^{-12}$, $x_c \approx 25000$, and $y_c \approx 15500$.

Figure 3 shows the behavior when regularizers are used. With R^1, the lowest value for D seems to occur when $\Delta_x = 5$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately $2 \times 10^{-}11$, 6000, and 6000, respectively. With R^2, the lowest value for D seems to occur when $\Delta_x = 10$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately $3 \times 10^{-}11$, 7000, and 6500, respectively. With R^4, the lowest value for D seems to occur when $\Delta_x = 5$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately $2 \times 10^{-}11$, 8000, and 6500, respectively.



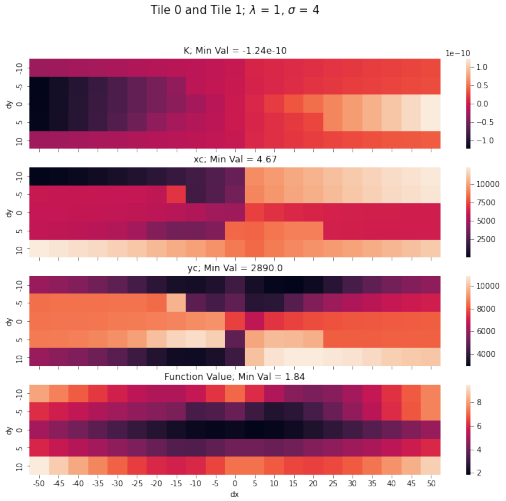
(a) (b)

Figure 2: Heatmaps showing the optimizer behavior for coordinates in T0 and their matches in T1 without using the regularizers. These heatmaps depict how values for K, x_c, and y_c change with T1's origin being shifted by Δ_x and Δ_y on the horizontal and vertical axis, respectively. Figure shows heatmaps of how K, x_c, and y_c behave without regularizer term while moving the origin offset of the second tile, T1 in this case. Looking at the bottom heatmap for optimizer function value, most values are between 2 and 3, with the highest function value being about 10. The heatmap doesn't inform well which Δ_x and Δ_y give the optimal K and (x_c, y_c). Limiting the function values to be under 2, we see that the minimum occurs at $\Delta_x = 20$ and $\Delta_y = -10$. With trial and error, we find out that at this location, $K \approx 0.9 \times 10^{-12}$, $x_c \approx 25000$, and $y_c \approx 15500$.



(a) R^1

(b) R^2

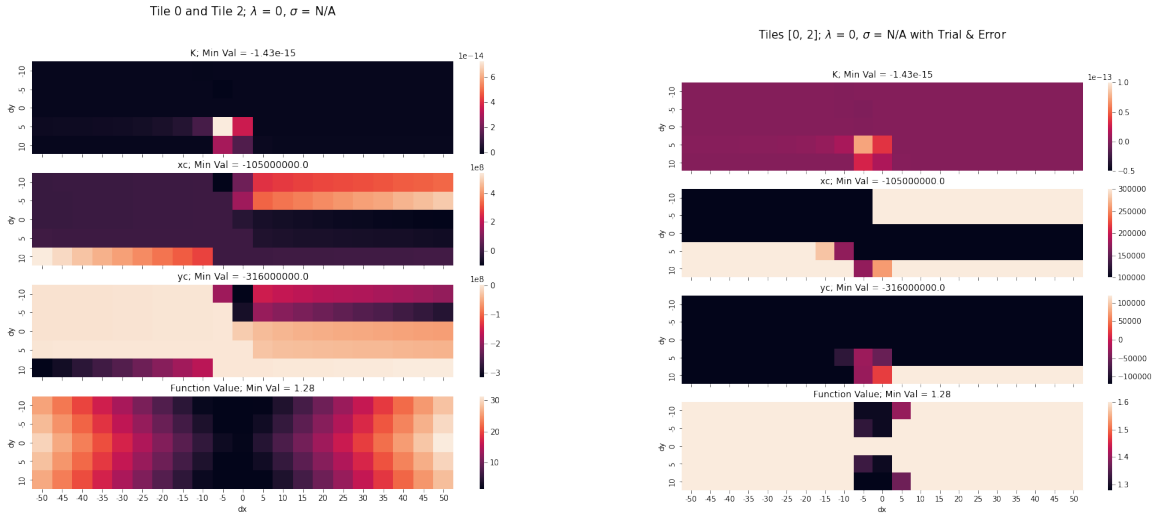


(c) R^4

Figure 3: Heatmaps showing the optimizer behavior for coordinates in T0 and their matches in T1 using the regularizers, a) R^1, b) R^2, c) R^4. These heatmaps depict how values for K, x_c, and y_c change with T1's origin being shifted by Δ_x and Δ_y on the horizontal and vertical axis, respectively. With R^1, the lowest value for D seems to occur when $\Delta_x = 5$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately 2×10^{-11}, 6000, and 6000, respectively. With R^2, the lowest value for D seems to occur when $\Delta_x = 10$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately $3 \times 10^{-}11$, 7000, and 6500, respectively. With R^4, the lowest value for D seems to occur when $\Delta_x = 5$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately $2 \times 10^{-}11$, 8000, and 6500, respectively.

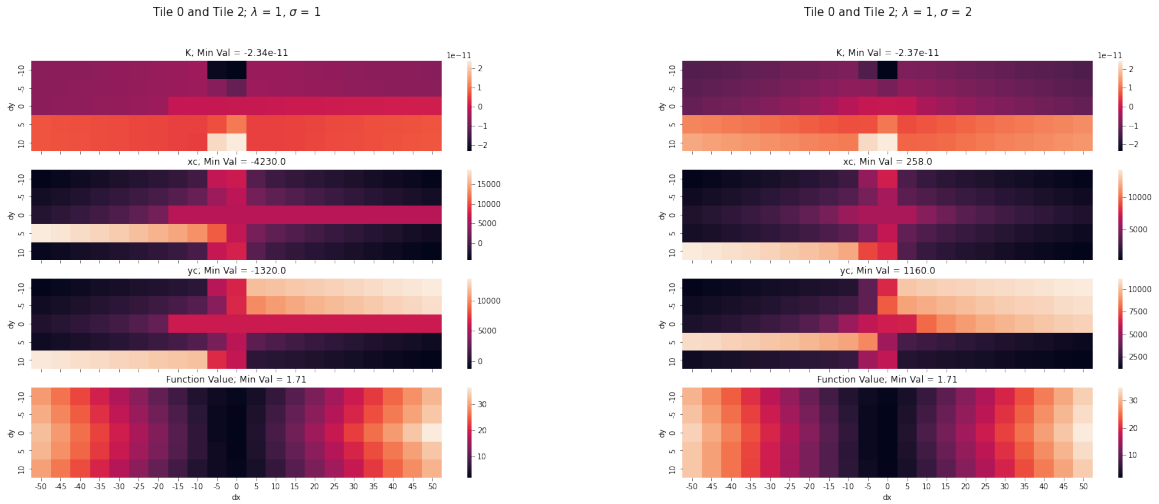
5.2 Setting 2 - Tile 0 and Tile 2

Figure 4 shows heatmaps of how K, x_c, and y_c behave without regularizer term while moving the origin offset of the second tile, T2 in this case. Looking at the bottom heatmap for optimizer function value, most values are under 8, with the highest function value going over 30. The heatmap doesn't inform well which Δ_x and Δ_y give the optimal K and (x_c, y_c). Limiting the function values to be between [1.28, 1.6], where 1.28 is the minimum value we see for all Δ_x and Δ_y, it is observed that the minimum occurs when $\Delta_x = -5$ or $\Delta_x = 0$ and $\Delta_y = \pm - 10$. With trial and error, we find out that at $\Delta_x = 0$ and $\Delta_y = -10$, $K \approx 2 \times 10^{-13}$, $x_c \approx 260000$, and $y_c \approx 25000$. At $\Delta_x = 5$ and $\Delta_y = -10$, $K \approx 4 \times 10^{-13}$, $x_c \approx 170000$, and $y_c \approx -25000$, while we don't see any reasonable values for $K, x_c, and y_c$ when $\Delta_y = 10$.



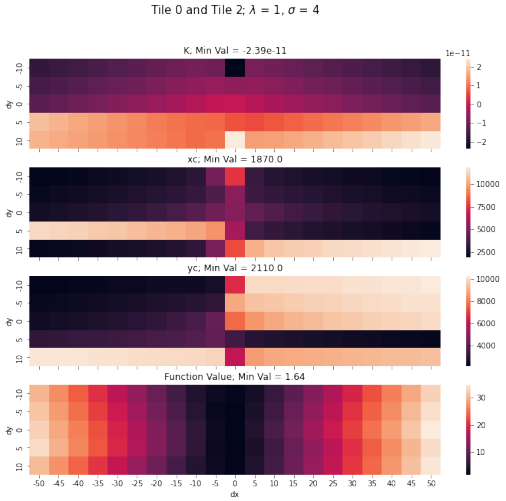
(a) (b)

Figure 4: Heatmaps showing the optimizer behavior for coordinates in T0 and their matches in T2 without using the regularizers. These heatmaps depict how values for K, x_c, and y_c change with T2's origin being shifted by Δ_x and Δ_y on the horizontal and vertical axis, respectively. Heatmaps capture how K, x_c, and y_c behave without regularizer term while moving the origin offset of the second tile, T2 in this case. Looking at the bottom heatmap for optimizer function value, most values are under 8, with the highest function value going over 30. The heatmap doesn't inform well which Δ_x and Δ_y give the optimal K and (x_c, y_c). Limiting the function values to be between [1.28, 1.6], where 1.28 is the minimum value we see for all Δ_x and Δ_y, it is observed that the minimum occurs when $\Delta_x = -5$ or $\Delta_x = 0$ and $\Delta_y = \pm - 10$. With trial and error, we find out that at $\Delta_x = 0$ and $\Delta_y = -10$, $K \approx 2 \times 10^{-13}$, $x_c \approx 260000$, and $y_c \approx 25000$. At $\Delta_x = 5$ and $\Delta_y = -10$, $K \approx 4 \times 10^{-13}$, $x_c \approx 170000$, and $y_c \approx -25000$, while we don't see any reasonable values for $K, x_c, and y_c$ when $\Delta_y = 10$.



(a)

(b)



(c)

Figure 5: Heatmaps showing the optimizer behavior for coordinates in T0 and their matches in T1 using the regularizers, a) R^1, b) R^2, c) R^4. These heatmaps depict how values for K, x_c, and y_c change with T2's origin being shifted by Δ_x and Δ_y on the horizontal and vertical axis, respectively. With R^1, the lowest value for D seems to occur when $\Delta_x = 0$ and $\Delta_y = 5$. For K, x_c, and y_c this block corresponds to approximately -1×10^{-11}, between (6000, 7000), and between (6000, 7000), respectively. With R^2, the lowest value for D seems to occur with $\Delta_x = 0$ and $\Delta_y = -5$. For K, x_c, and y_c these values correspond to approximately $3 \times 10^{-}11$, 7000, and 6500, respectively. With R^4, the lowest value for D seems to occur when $\Delta_x = 5$ and $\Delta_y = 0$. For K, x_c, and y_c these values correspond to approximately $2 \times 10^{-}11$, 8000, and 6500, respectively.

References

[1] C Brown Duane. Close-range camera calibration. *Photogramm. Eng*, 37(8):855–866, 1971.

[2] Timothy A Clarke and John G Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91):51–66, 1998.

[3] J Nelder. A. & mead, r.(1965). *Computer J*, 7:308–313, 1965.