Microscopy Image Processing for the Human Eye

Zeyue Xia, Mohamad Nadim Barakat, Serri Matula, Zijun Hui, Dr. John Stravakrakis

***Abstract*— Vivo confocal microscopy allows scientists to better understand eye health and systemic diseases. Microneuromas could play a role, however, monitoring their growth from a mosaic of images is error prone and time consuming. We used automated image stitching as a solution; focusing on accuracy and computational speed of three different feature detection algorithms: SIFT, SURF and ORB. The results illustrated that SURF was computationally efficient with our data. Future investigation is to create a global solution that can replace the need for manual image stitching in this application.**

I. INTRODUCTION

The Heidelberg Retinal Tomograph [1], HRT, (Figure 1) produces hundreds of 400x400 micron area images that each capture a small section of the eye's posterior segment. It is used for examining the nerves in the eye that allows for easier diagnosis of glaucomas - damages to the optic nerve [2]. To map the eye, however, the small images must be stitched together and requires hours of manual work from the researcher. Many of these images are not perfectly matched and discarded, resulting in gaps of the stitched output (Figure 2). While we were unable to create an algorithm that acts globally, we created a model on how automated image stitching would take place locally. Experimentally, we evaluated three popular feature detection algorithms on the basis of their speed (by computational time) and accuracy (by image similarity) after applying this method on two locally stitched images.

Fig. 1: Heidelberg Retinal Tomograph

II. METHODOLOGY: IMAGE STITCHING MODEL

Image stitching can be achieved by 1) direct techniques, or 2) feature based techniques [3]. Direct technique does not work when images differ in scale and rotation. It is also a complex computational task because of the need to compare each individual pixel densities of different images before matching overlapping regions.

School of Computer Science, University of Sydney
May 11, 2020

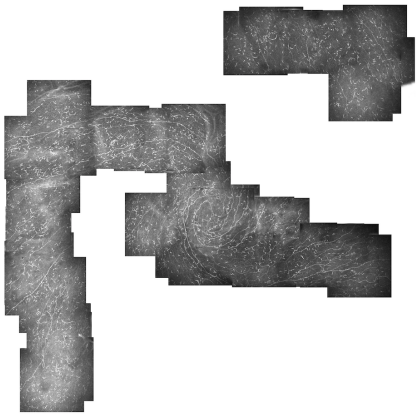


Fig. 2: Manually stitched image of eye's posterior segment

Feature-based techniques detect features in images and match them together. Features are mathematical representations of key areas which include colour, texture, shape, edges, lines, corners, etc. It is appropriate in our experiment as feature matching is unaffected by rotation and other transformations. The steps of the feature-based technique [4] are shown in Figure 3 and are explained in the series of steps A to E that include example outputs.

The manually stitched image (Figure 4) consists of 79 layers and has all been numbered. We tested three different scenarios and the following samples were chosen: 1) Layers 69 and 70 due to its high area of overlapping regions. 2) Layers 76 and 77 which have major overlaps but consist of some noise and will test the rigor of the algorithms, and 3) layers 71 and 73 where image stitching may not work as effectively or is not even possible due to the small portion of overlapping regions. The methodology will include examples of each process using layers 69 and 70. The other outputs can be found in the results. Furthermore, different layers (23/24) will also be evaluated with a similarity index however will not be shown as images.

A. Input Images

Our images are in greyscale and many of them are differently scaled, have affine transformations or are rotated in different ways. Acquire two local images and check for size parameters (height and width in pixels) ensuring they are the same size.

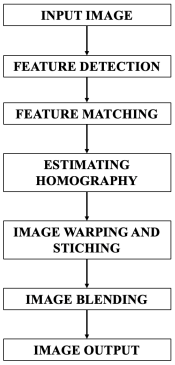


Fig. 3: Model showing the stages of image stitching using feature-based techniques

B. Feature Detection

The three algorithms used for this experiment, to pick out key points that represent a feature, are SIFT, SURF and ORB.

SIFT (Scale-Invariant Feature Transform) [5] detects key points on an image by finding the differences in Gaussian. It is able to pick our key points using colour gradients and finding the greatest contrast between them. It is popular due to its robustness, however, the slight disadvantage is that it exerts a high computational cost for it's detection [6]. SIFT is an incredible candidate in this application due to it's remarkable robustness and the fact that our images have great contrast between the white spots of the neurons and the darker surroundings of the cornea. Additionally, the downside in SIFT is negligible given the fact that the images are small in size and do not appear to have many key points.

SURF (Speeded-Up Robust Features) [7] blurs it's images using a box filter then uses a "blob detector" to repeatedly sample key pixel groups of interest. It is faster, requires less computation and storage space compared to SIFT compromising for its lower accuracy in detecting features.

ORB (Oriented Fast and Rotated Brief) [8] uses a centroid detector which recognises points which have high contrast with other points. It is a combination of FAST keypoint detector and visual descriptor BRIEF. It is the fastest in detecting key features but requires less computational power than SIFT or SURF[6]. However, it has the inability to detect features of images of different sizes, has poorer ability to filter out noise and is not very robust, especially when images have undergone a rotation[6], making it a key difficulty with recommending ORB for this application as many images have different orientations.

For a more detailed explanation on how each algorithm achieves feature detection refer to Appendix 1.

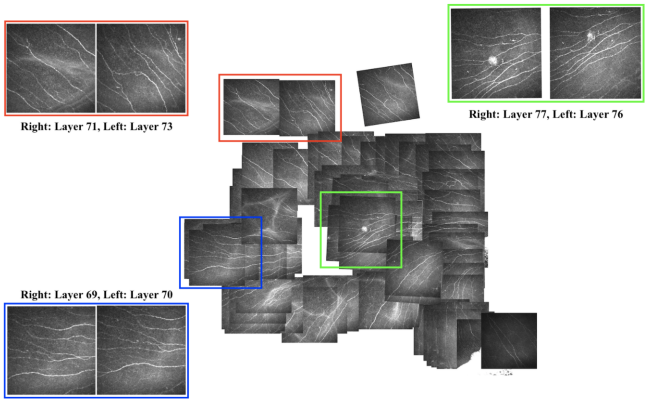
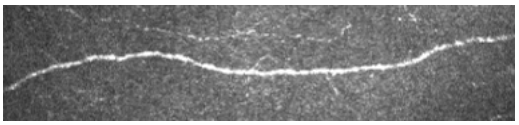
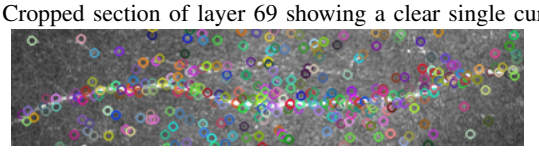


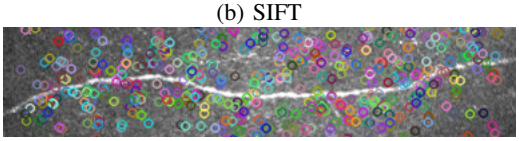
Fig. 4: Location of the layers on the full collage



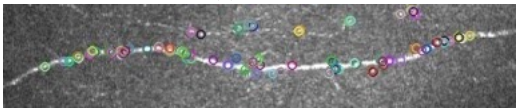
(a) Cropped section of layer 69 showing a clear single curve



(b) SIFT



(c) SURF



(d) ORB

Fig. 5: Figure 5a shows a segment of the original image and Figure 5b-d show the same segment with circles that represent a feature detected by each algorithm. These key points can be used to match features together.

C. Feature Matching

Once all the features are detected in the two images, use feature matching algorithms to find corresponding key point descriptors in two similar datasets and create linear regression lines matching them together [14]. Brute-Force matcher [9] (Appendix 2) was used and the results can be seen in Figures 6 which shows a series of lines that the computer registers as matched key points. There are some incorrect matches and the next stage accounts for them.

D. Estimating Homography

Before aligning the images together, we need to find a statistical model of the lines matching the key points together. A perfect match between the points detected is rare because the images taken are often distorted through transformation. We used RANSAC (Appendix 3) which is a statistical model used to separate features detected into 'inliers'- i.e, feature points that match the model - and 'outliers'- i.e, feature points that do not match the model. RANSAC is robust to noise and is accurate in circumstances where the number of outliers is less than inliers - even in circumstances where outliers appear in very high frequencies (Figure 6). The

matched lines are compared and the one with the greatest number of inliers is computed (Figure 7a,7b,7c).

Once the best regression line is chosen, the homography matrix (Figure 8), H, is calculated and used to translate one image onto the other. H is a 3x3 matrix of the vector that was determined to be the best matching line by RANSAC.

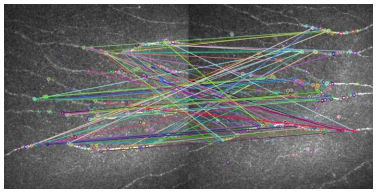
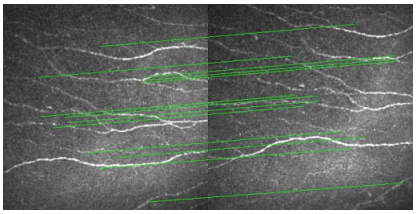
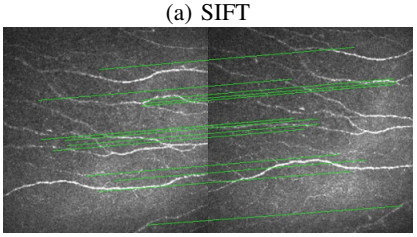


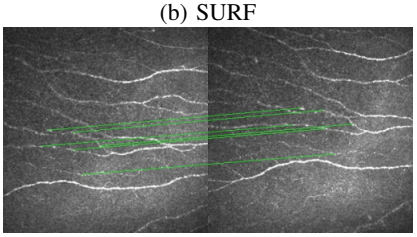
Fig. 6: Brute force matching on keypoints detected by ORB in layers 69 and 70 without RANSAC



(a) SIFT



(b) SURF



(c) ORB

Fig. 7: Feature matching with RANSAC applied on layers 69 and 70 using different keypoint detectors.

```
Estimated homography :
 [[ 9.95298670e-01 -1.84781001e-02  9.09910007e+01]
 [ 1.18587567e-02  9.70745761e-01 -3.79341415e+01]
 [ 1.12371823e-04 -1.27760010e-04  1.00000000e+00]]
```

Fig. 8: Example homography output: Using ORB, Brute-force matcher and RANSAC

E. Image stitching

Using H, we can stitch the image together by using Opencv's WarpPerspective function which is a geometric image transformation[10][14]. One image is translated onto the other and results in a combined image (Figure 9).

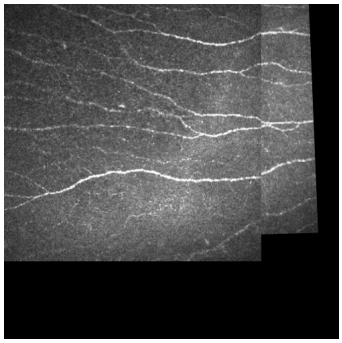


Fig. 9: Warped image of layer 69 and 70 using ORB

F. Image blending

Further steps can be taken to blend the images together, however this was not implemented on our images. This is because part of the evaluation is to compare the stitched image to the ground truth, in this case it is the collage of all the layers that has been manually stitched. The ground truth images have not been blended and the same is implemented to ours, making the comparison a fair test. Blending the scans, however, could be a future investigation as it would allow us to see the microneuromas much more clearly and with less noise and interruptions. Current literature of blending strategies include feathering, gradient domain blending and image pyramid blending.

III. EVALUATION METHOD

With these results, evaluations must be performed to test for their speed and accuracy. Speed was measured by finding the computation time in seconds. Computation time was calculated by importing the Time module in Python and recording the difference between start and end time[12]. All programs were run on one MacBookAir6,2 (OS 10.15.4, Dual-Core Intel Core i5, 1.3 GHz, 4 GB) to obtain comparable data.

The chosen evaluation method of accuracy was to compare the computed stitched image to the ground truth image (manually stitched image) via a pixel by pixel similarity index. To objectively compare the images, a quantitative approach was taken. Mean squared error (MSE) is a simple mathematical method to calculate the degree of similarity between images[11]. MSE is given by the following formula where x and y are the images, N is the number of pixels and xi, yi, are the ith values in each respective image:

$$\mathrm{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2.$$

The higher the MSE, the less similar the images. Identical images have an MSE of zero.

While simple, MSE does have its drawbacks. Given two identical images, a small change in the one image might result in a large MSE even though the perceived change to the human eye is negligible. To supplement, the structural similarity (SSIM) index was adopted [15][17].

$$S(\mathrm{x},\mathrm{y}) = l(\mathrm{x},\mathrm{y}) \cdot c(\mathrm{x},\mathrm{y}) \cdot s(\mathrm{x},\mathrm{y}) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right) \cdot \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right),$$

The formula above details the methodology of SSIM. Given two images x and y, SSIM measures the similarity of the pixel brightness $l(x, y)$, the similarity of the pixel contrasts $c(x, y)$, and the similarity of the structures $s(x, y)$. Also, x and y give the means while x and y give the standard deviations. Each C represents a small constant. Two identical images would have a SSIM of one and the most dissimilar images would have a SSIM of negative one [11].

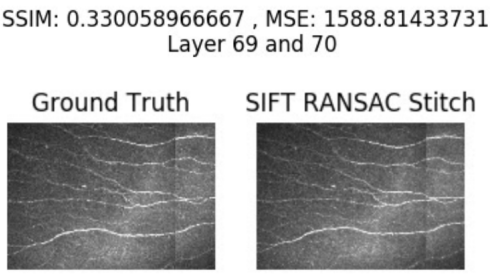


Fig. 10: Comparison of ground truth and stitched image (using SIFT)

IV. Results

As mentioned earlier, a lower MSE and higher SSIM indicate greater similarity. Computation time was the fastest with ORB, followed by SURF then SIFT in all cases, except layers 71 and 73 as they could not be matched together (Appendix 4). However, ORB had the largest MSE and smallest SSIM which illustrates that the stitched image was the least similar to the ground truth, when compared to SIFT and SURF. As for accuracy between SIFT and SURF, the similarity index depends on the different scenarios. Layers 69/70 and layers 76/77 are more similar to the ground truth when SIFT is being used. However, SURF is more accurate on layers 23/24 as well as faster than SIFT in all cases.

Layers	SIFT	ORB	SURF
69 and 70	0.471	0.211^a	0.411
76 and 77	0.661	0.311^{ab}	0.571
71 and 73	N/A	N/A	N/A
24 and 23	0.746	0.177^a	0.573

TABLE I: Comparison of Computation Time (seconds)

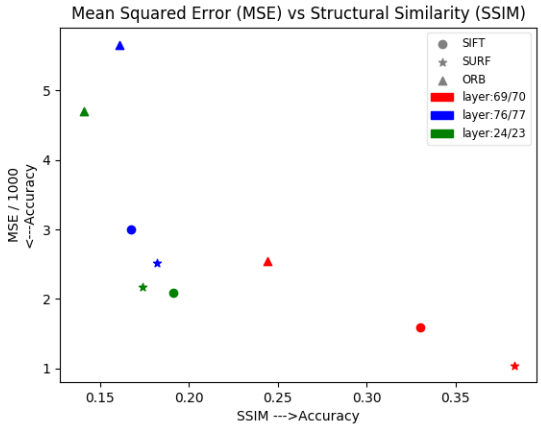
Layers	SIFT	ORB	SURF
69 and 70	1588.814	2550.391	1034.197^a
76 and 77	3000.835	5658.131^b	2512.979^a
71 and 73	N/A	N/A	N/A
24 and 23	2086.145^a	4702.831	2166.108

TABLE II: Evaluation of ground truth and output stitched image using MSE similarity index

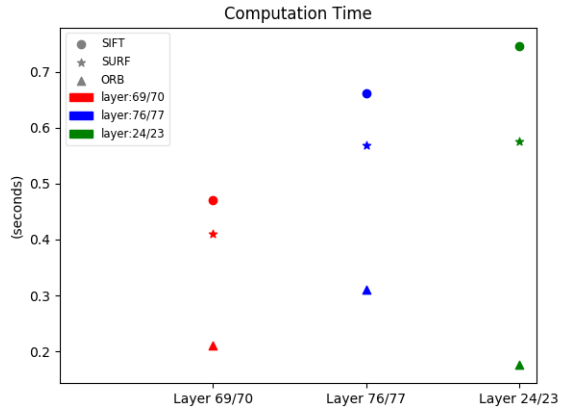
Layers	SIFT	ORB	SURF
69 and 70	0.33006	0.24434	0.38324^a
76 and 77	0.16768	0.16102^b	0.18194^a
71 and 73	N/A	N/A	N/A
24 and 23	0.19126^a	0.14116	0.17416

TABLE III: Evaluation of ground truth and output stitched image using SSIM similarity index

a: Represent best value
b: Partially stitched together
N/A: Stitching was not possible because not enough matching keypoints were detected (Appendix 4)



Graph I. Comparison of accuracy using different feature detectors



Graph II. Comparison of computation time using different feature detectors

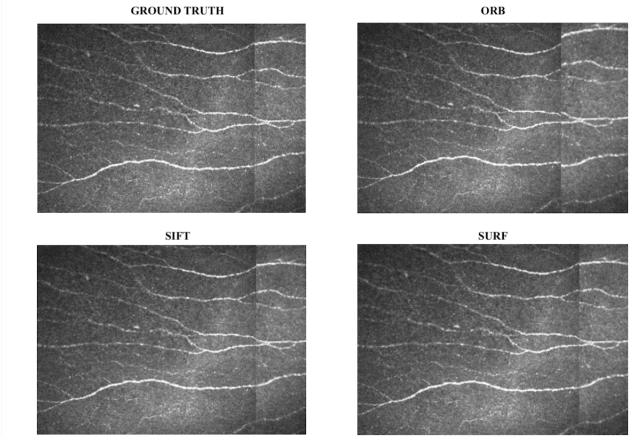


Fig. 11: Stitched image comparison to ground truth for layers 69 and 70

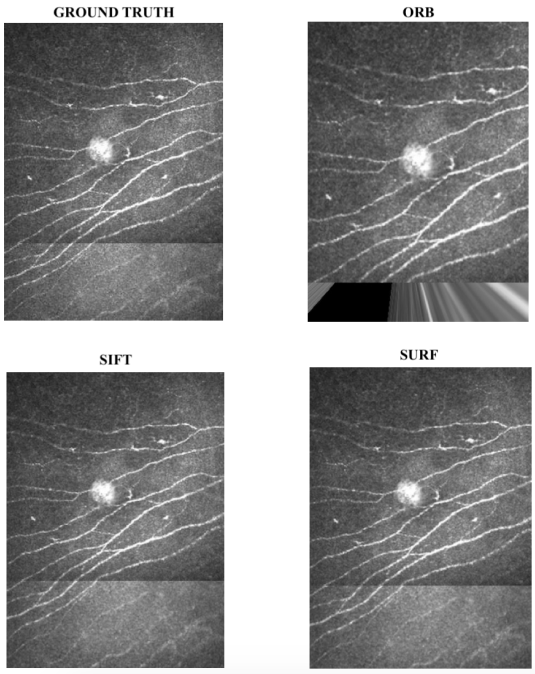


Fig. 12: Stitched image comparison to ground truth for layers 76 and 77

V. DISCUSSION

We assessed the accuracy of our automatic image stitching methodology against manually stitched images and in many cases obtained similar results. However, our pioneering attempt to automate the manual image stitching process is limited to stitching images locally. Therefore one area of further investigation is using the techniques found in this paper and applying it globally. That process would consist of expanding the methodology proposed in our paper to stitch a multitude of images and create a panorama of many images instead of one with only two.

A potential limitation in our methodology presents itself when extracting the manually stitched images. These images have been extracted from photoshop [16] and it is unknown whether issues with the extraction have contributed to the pixel difference; issues such as cropping, rotation, shading. Furthermore, issues with image blending and other strange unidentified issues may interfere with the pixel difference (Figure 11 and 12, "Ground Truth").

Future work could also focus on locating the where the differences are located between an automatic and manually stitched image. Using MSE to find the degree of similarity fails to give us an idea of what and where the actual differences are located. It would indeed be useful for optimization purposes to find the precise differences in transformations in order to more accurately apply this methodology in the biomedical field. Finally, image blending techniques which would create a less noticeable seam between each image could be implemented, especially if a global solution is created, as the resulting image would be more aesthetic.

VI. CONCLUSION

Image stitching has multifarious applications and to implement this into vivo confocal microscopy in the field of ophthalmology gives rise to considerable significance. The automatic process of stitching images taken from the Heidelberg Retina Tomograph would save considerable amounts of time and manual work from the researchers. It would allow clinicians to be able to examine growth and development of microneuromas in the eye's posterior segment, making it easier to diagnose certain eye diseases e.g. glucomas or dry eye syndrome. We found SURF to be the best feature detector and therefore our recommendation for this application as it had a fast computation time without compromising accuracy as is the case with ORB, and had a high level of accuracy without being too computation heavy like SIFT. A potential next step is applying what we found when working with local images and upscaling this application globally.

ACKNOWLEDGEMENTS

This paper would not be possible without the generous support, detailed feedback and incisive criticisms of Dr. John Stravakrakis and Zijun Hui.

Dr. Stravakrakis had always provided a clear direction through his clear expectations of what should be done while lavishly providing us with suggestions on how to improve this report. Dr. Stavrakakis was also extensively involved with the editing of this document, providing us with feedback for our initially poor citations, teaching us how to embellish our sentences to become both concise yet as informative as it could be. But perhaps the most important thanks we as a team would like to provide John is providing us with an interesting research question, on a subject backed with a plethora of literature and wealth of content to add to this report; such an experience has sparked in us an appreciation and even a passion in computer science research.

Zijun Hui has always been available to us, guided us step by step through leading weekly discussions and meetings and was a wellspring of expertise for our experimental journey with openCV. Zijun's leadership was instrumental for our team's solidarity and morale. Helping us learn how to break down an unfamiliar subject-we were never daunted by such a difficult project; his patience when answering our relentless

and sporadic questions was a testament to his remarkable attitude and reliability; his non-judgmental outlook and optimism when sometimes the group has struggled to meet his deadlines made him not only approachable but inspirational as what a leader should do in times of struggle. But out of the innumerable aspects we must acknowledge Zijun for, it is his expertise with computer science that made us feel comfortable with traversing into the difficult but rewarding journey into computer science and provide the experimental data today-for Zijun answered all of our questions, helped us with syntax errors we could not catch, and introduced us to tools that we will be able to now use for the rest of our careers. We as a team want to thank Zijun for his exceptional model leadership, and generosity with his time.

REFERENCES

[1] "Glaucoma.", Healthdirect.gov.au, 2020. [online] Available at: https://www.healthdirect.gov.au/glaucoma [Accessed 1-May-2020].

[2] O. Stachs, R. F. Guthoff, and S. Aumann, "In Vivo Confocal Scanning Laser Microscopy", High Resolution Imaging in Microscopy and Ophthalmology: New Frontiers in Biomedical Optics, Cham (CH): Springer, 2019.

[3] S. Arya, "A Review on Image Stitching and its different methods" International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 5, pp.299-303, 2015

[4] D. Vaghela, K. Naina, " A Review of Image Mosaicing Techniques", International Journal of Advance Research in Computer Science and Management Studies, vol. 2, no. 3, pp. 1-6, 2014

[5] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004. Available: https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94.

[6] E. Karami, S. Prasad and M. Shehata, "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Image", Faculty of Engineering and Applied Sciences, Memorial University, Canada, November, 2015

[7] "Introduction to SURF (Speeded-Up Robust Features) — OpenCV-Python Tutorials 1 documentation", Opencv-python-tutroals.readthedocs.io, 2020. [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html. [Accessed: 13- May-2020].

[8] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", 2011 International Conference on Computer Vision, 2011. Available: https://www.researchgate.net/publication/221111151_ORB_an_efficient_alternative_to_SIFT_or_SURF. [Accessed 13 May 2020].

[9] "Feature matching" - OpenCV 2.4, Docs.opencv.org, 2020. [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html [Accessed: 1-May-2020]

[10] "Geometric Image Transformations — OpenCV 2.4.13.7 documentation", Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html.[Accessed: 13- May-2020].

[11] Zhou Wang and Bovik, A., 2009. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. IEEE Signal Processing Magazine, [online] 26(1), pp.98-117. Available at: https://ece.uwaterloo.ca/ z70wang/publications/SPM09.pdf [Accessed 14 May 2020].

[12] Docs.python.org. 2020. Time — Time Access And Conversions — Python 3.8.3Rc1 Documentation. [online] Available at: https://docs.python.org/3/library/time.html [Accessed 9 May 2020].

[13] D. Tyagi, Introduction to ORB (Oriented FAST and Rotated BRIEF), Data Breach, 2019. [Online]. Available: https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf. [Accessed: 1-May-2020]

[14] Rosebrock, A., 2016. Opencv Panorama Stitching - Pyimagesearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/ [Accessed 15 April 2020].

[15] Rosebrock, A., 2014. How-To: Python Compare Two Images - Pyimagesearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/ [Accessed 5 May 2020].

[16] Photopea, 2020. Advanced Image Editor, ver. 4.7. [Online]. Available at : https://www.photopea.com/ [Accessed 6 May 2020]

[17] S. van der Walt et al., "scikit-image: image processing in Python," PeerJ, vol. 2, p. e453, Jun. 2014.

APPENDIX

1) SIFT, SURF, ORB

SIFT was proposed by David Lowe in 1999 as a method of resolving the problems regarding affine transformation, scale changes, and image rotation in feature matching. It detects key points by finding differences in Gaussian which is a technique used to find points on the image that have the darkest of lightest hue, called the local extrema. After the image is blurred using a Gaussian convolution - a filter that drastically reduces the resolution image - the colour gradient is used to find points with the greatest contrast with the rest of the image while eliminating points that cannot be reliably used for feature detection i.e. filtering out noise. [6] SIFT is the most popular feature detection algorithm for its robustness; robustness being the ability to detect key features despite the contortions of the image such as. A disadvantage of SIFT is that it exerts a high computational cost for it's detection; the increased robustness, the ability to capture multiple key points and complexity in its algorithm means that it requires the most processing power and is one of the slowest image detection algorithms. SIFT is not recommended to be used for images with lots of key points, or are large.

SURF blurs it's images using a box filter to find the differences in Gaussian [6]. The image is iteratively blurred and uses a "blob detector" to repeatedly sample key pixel blobs (rather than points with SIFT) of interest. The pixel blobs are then used to find the contrast between the gradients in order to detect the interest points. The advantage of SURF iteratively sampling lies in its ability to use blob detectors of different filter size, which could increase the speed of the feature detection if the contrasts of the image is very strong. This iterative method also has advantages in finding features in images with different sizes. SURF however does not filter noise as well as the other two feature detection algorithms, and in many applications do not appear to detect as many features as SIFT. The reduction in computational cost and storage also appears minimal for most applications relative to SIFT [7].

ORB, developed in 2011, is a combination of the FAST keypoint detector and the visual descriptor BRIEF. FAST (Features from Accelerated and Segments Test) takes an intensity threshold of the central pixel and compares it with its circular neighbours. If the intensity passes a certain threshold it is considered to be a key point. FAST does not have an orientational component, so ORB was developed to resolve this issue. BRIEF (Binary Robust independent elementary feature) takes the key points detected by FAST and combines them together to form larger features that are represented by feature vectors [13]. In ORB, centroid

detectors are used and it detects points which have high contrast with other points [6].

2) Brute-force matcher

Brute-force matcher is an algorithm that uses the basis of Euclidean distance to evaluate the matches between the two key points. It first stacks the images next to each other horizontally and draws the Euclidean distance, which is a straight line between two points in a Euclidean space. In this instance, the Euclidean distance of every keypoint in image A is calculated with every keypoint in image, showing the best matches. This algorithm is considered slow as it checks all the matches of every feature [Mahendran, 2013].

3) RANSAC

RANSAC randomly picks points in subsets, and by random chance, reduces the number of outliers picked. After a model, i.e, a linear regression line, is created from those randomly picked points. The final step is called the verification step: by using a threshold of that linear regression line, we count the number of points within the threshold-these becomes the inlier. We continuously refit the model by resampling points many times until we find the most number of inliers within the threshold. RANSAC however performs badly if the number of outliers is greater than the number of inliers and is also an inefficient method in terms of computer cost and processing because it is an iterative process that continuously generates models until it finds an optimal solution. The number of data points increases its inaccuracy. And because this is a random sampling technique, it is reliant on chance, and with data with a huge number of points an optimal solution may not be accurate as well. Another issue with RANSAC is in data where there may be more than one best model, and in such data, it cannot find either. These limitations are however mitigated by the fact that our dataset appears small, and there is only one optimal way of fitting images together. RANSAC's major drawback, the computational time, however is reduced by trying to remove outliers because it would produce a smaller number of data sets.

There are basically two ways of achieving this: a method called guided sampling that guides which subsets to use rather than randomly selecting samples and a method called partial evaluation, where it gets rid of verification steps that are not promising. Unfortunately many of these improved RANSAC algorithms are still in their development phase, and therefore are difficult to find open source codes for these algorithms. Alternatives to RANSAC include the least squares method and Hough transform, though they have major drawbacks compared to RANSAC. The least squares method is very simple requiring very little computational cost but is incredibly sensitive to significant outliers-something that is abundant in our images. The Hough transform detects and indexes simple lines and sometimes curves, but is not applicable to our images since neurons are not easily simplified to simple lines and curves.

4) Ratio Test

One such method to improve accuracy in matches, is a ratio test proposed by Lowe [5]. By his methodology, a given keypoint is only considered accurate, if the distance ratio between its two nearest matches is below a certain value. Given a keypoint A, the distance between A and the first nearest keypoint A1, as well as the distance between A and its second nearest keypoint A2 are measured and divided by each other. If the quotient value is below a set ratio threshold, A is considered an accurate keypoint. Based on Lowe's results, the methodology used in this paper set the ratio to 0.75.