Breast tissue reconstruction

Segmentation and 3D reconstruction of human breast tissue

Noah Hoogenboom
13467271

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
N. Awasthi

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

Semester 1, 2023-2024

Abstract

This thesis explores the realm of biomedical research, aiming to elevate our understanding of tissue structure and function. In contrast to traditional two-dimensional medical imaging, the study delves into the transformation of existing 2D images into insightful three-dimensional (3D) representations. The main objective is to establish a flexible and robust pipeline, encompassing crucial components like image alignment, cell recognition, and 3D-reconstruction. The pipeline is nearly complete, lacking only the finalization of a GeoJSON to XML conversion. Furthermore, rigorous testing and analysis of various alignment methods showed that the alignment approach pioneered by Kiemen et al., 2022 performed best. The imminent completion of the pipeline holds great promise for extensive applications in the fields of biology and medicine, representing a significant advancement in biomedical imaging practices.

Contents

Chapter 1

Introduction

1.1 Background

Biomedical research plays a crucial role in unraveling the complexities of tissue structure and functionality. This exploration enhances our comprehension of different tissue characteristics and could significantly contribute to medical advancements, notably in the area of disease detection and treatment (Open Medscience, 2023). However, contemporary medical imaging predominantly provides two-dimensional representations. While these images offer insights into the diverse cell types, they unfortunately lack information about the three-dimensional (3D) structure. In order to better comprehend the interlinked functionalities and gain insight in the properties of tissue, a thorough understanding of the tissue's 3D structure is advantageous (Silén et al., 2008). This thesis is dedicated to research the possibilities of using technology to create these 3D-images using the current 2D-images. This aligns with the broader trend of leveraging artificial intelligence in biomedical research for enhanced diagnostic accuracy.

By enhancing the accuracy and comprehensiveness of methodologies, the refined frameworks seek to bridge the gap between research outcomes and clinical implications. This marks a significant step towards practical applications, potentially improving diagnostic and treatment strategies for diseases. A noticeable effort in these developments was the research paper "CODA: quantitative 3D reconstruction of large tissues at cellular resolution" by Kiemen et al., 2022. Here they explored the possible methods that can be deployed for the 3D-reconstruction of the pancreas. Additionally, multiple contributions address specific sub problems within this domain. For instance, Schmidt et al., 2018 investigate cell recognition through polygonal techniques, while Lotz et al., 2017 explore diverse methodologies for image alignment. In essence, this thesis aims to bring all these elements together into one streamlined pipeline.

1.2 Thesis Goals

The primary goal of this thesis is to establish the foundation for a pipeline that integrates image alignment, cell recognition, and 3D tissue reconstruction. The intention is to create a dynamic structure that exhibits both inclusivity and flexibility for wider applications. The CODA paper serves as a key reference, providing a starting point for exploration.

By using the principles introduced by CODA as the basis and extending them, the objective is to advance the alignment and analysis of medical images. Unlike CODA, which focuses on pancreatic research, this thesis addresses the alignment and analysis of the mammary gland. Despite the progress in various components of breast tissue analysis, this exploration reveals unexplored territories. In contrast to existing methodologies that focus on isolated aspects, the refined frameworks presented in this research aim to integrate 3D imaging, deep learning, and alignment techniques. The higher end-goal is to provide a holistic understanding of the biology of breast tissue.

Beyond the technical pursuits, this thesis aspires to make meaningful contributions to the broader field of biomedical research and healthcare. The envisioned pipeline has the potential to significantly impact disease diagnostics, treatment planning and a more thorough understanding of diseases and tissue.

Chapter 2

Theoretical Foundation

This chapter explains the terms and methodologies employed throughout this thesis, divided into four key sections. The first section discusses the type of imaging used. In Section 2.2, the focus is on explicating definitions and methods for achieving precise image alignment, establishing a foundational basis for subsequent analyses. Section 2.3 provides the definitions and formulas used to examine the different alignment methodologies. The chapter concludes with explaining the investigated neural network.

2.1 Hematoxylin and Eosin Staining

Hematoxylin and eosin stain, commonly referred to as H&E stain, is a tissue staining technique widely applied in histology. Specifically, in the examination of a biopsy suspected of cancer, the histological section will likely undergo H&E staining (National Cancer Institute, 2023; University of Michigan Medical School, n.d.).

In this staining procedure, a biopsy sample of the tissue is obtained. This biopsy is sliced in thin sections of 5 μm in preparation for staining, where each section receiving a specific type of stain. In the case of H&E staining, hematoxylin imparts a purplish-blue color to cell nuclei, whereas eosin imparts a pink coloration to the collagen, enzymes, glycoproteins and cytoplasm. Various structures present diverse shades and combinations of these colors, see Figure 2.1. Consequently, pathologists can readily distinguish between the nuclear and cytoplasmic components of cells. The overall color patterns resulting from the stain unveil the general layout and distribution of cells, offering a overview of the tissue sample's structure.

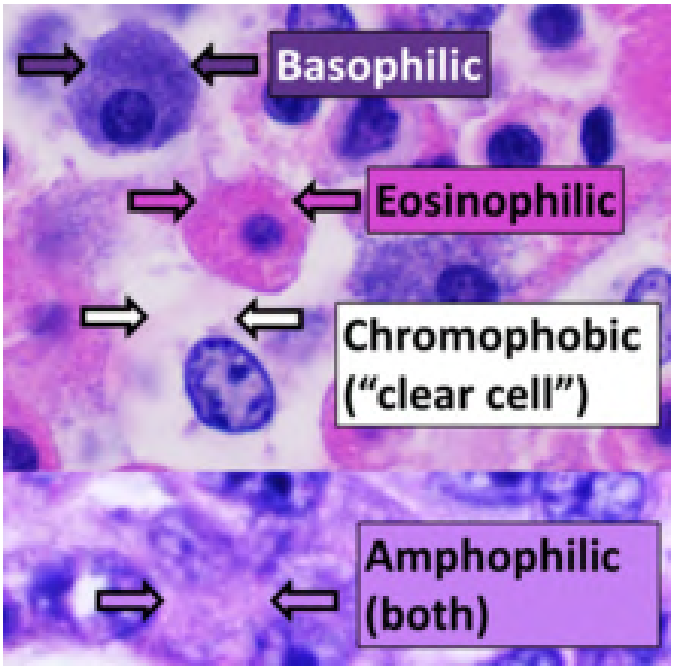


Figure 2.1: H&E stained tissue. Basophilic, stains anionic structures such as DNA in the cell nucleus and ribosomes; Eosinophilic: stains cell cytoplasm, collagen, and muscle fibers; Chromophobic, areas without any staining; Amphophilic, a combination of basophilic and eosinophilic.

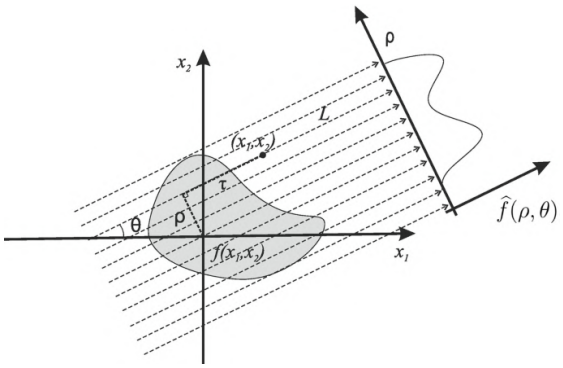


Figure 2.2: Applied Radon Transformation.

2.2 Image Alignment

The initial challenge in this thesis arose from distortions observed between images, occurring at both local and global scales. Addressing these distortions, particularly through image alignment, is the subject of Section 3.3. This section outlines the internal structure of the alignment process, providing insights into the diverse methods applied to rectify image distortions.

2.2.1 Radon Transformation

The Radon transform is a mathematical technique employed in image processing and medical imaging to analyze and represent the internal structures of objects. Mathematically, the Radon transform R of a function $f(x, y)$ is defined as an integral along lines in the plane (x, y). It is expressed as:

$$\begin{aligned} Rf(\theta,\rho) &= \int_{-\infty}^{\infty} f(x(z), y(z))\, dz \\ &= \int_{-\infty}^{\infty} f\left((z\sin(\theta) + \rho\cos(\theta)), (-z\cos(\theta) + \rho\sin(\theta))\right)\, dz \end{aligned} \tag{2.1}$$

Here, θ represents the angle of rotation, and ρ is the distance parameter and z is the parameter along the line of integration. The Radon transform calculates the total signal along various lines through an object, providing insights into how imaging signals are attenuated as they traverse the object from different angles and distances. Figure 2.2 illustrates an example of this transformation.

2.2.2 Particle image velocimetry

Particle image velocimetry (PIV) is an optical technique utilized for analyzing and quantifying the flow patterns or motion of particles. In the context of image sequences, PIV is applied to utilize the information from identified outliers or irregularities. The method employs intensity normalization to highlight these outliers. These distinctive features are identified in both images, allowing for the extrapolation of the movement of outliers between frames. Subsequently, these movements are generalized across the entire image, providing insights into the overall displacement of the entire tissue.

2.2.3 Log Polar Transformation

The Log Polar transformation is a mathematical technique employed in image processing, particularly for image registration and pattern recognition tasks. This transformation allows for the representation of images in polar coordinates with a logarithmic scale, offering advantages in scale and rotational invariance. The process begins by converting Cartesian coordinates (x, y) to polar coordinates (r, θ) using the following equations:

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \theta &= \arctan\left(\frac{y}{x}\right) \end{aligned} \tag{2.2}$$

The Log Polar Transformation then introduces a logarithmic scaling to the radial axis, altering the transformed radial coordinate r to: $r' = \log(1 + r)$. The angular coordinate θ remains unchanged ($\theta' = \theta$). The logarithmic scaling of the radial axis provides a compressed representation that emphasizes the central region while reducing the impact of the outer regions, contributing to robustness against scale variations. Figure 2.3 visually illustrates the log-polar transformation.

2.2.4 Bilinear Interpolation

Bilinear interpolation is a fundamental method in image processing, widely used for estimating pixel values at non-grid locations within an image. This technique provides a smooth and continuous approximation by considering the weighted average of the nearest neighboring pixels.

The process involves determining the contribution of each surrounding pixel based on the relative distances in both the horizontal and vertical directions. Mathematically, the interpolated pixel value P can be expressed as (for clarification see

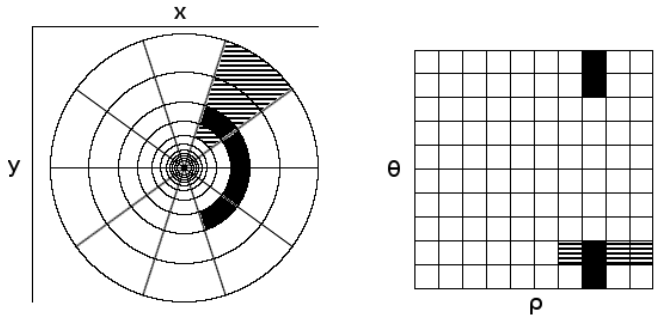


Figure 2.3: Visualization of log-polar transformation.

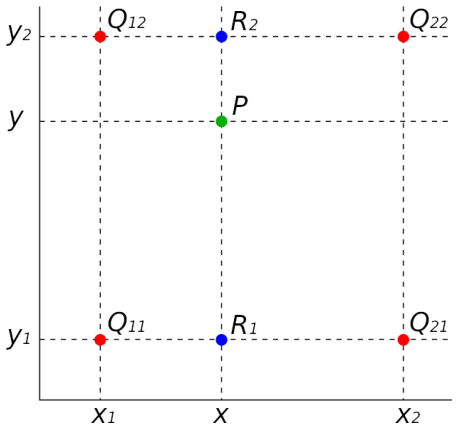


Figure 2.4: Example of bilinear interpolation.

Figure 2.4):

$$P(x,y) = (1-\alpha)\cdot(1-\beta)\cdot Q_{11} + \alpha\cdot(1-\beta)\cdot Q_{21} + (1-\alpha)\cdot\beta\cdot Q_{12} + \alpha\cdot\beta\cdot Q_{22} \quad (2.3)$$

where: $\alpha = x - x_1$, $\beta = y - y_1$ and $Q_{11}, Q_{21}, Q_{12}, Q_{22}$ are the pixel values of the surrounding grid points. The coefficients $(1-\alpha)\cdot(1-\beta)$, $\alpha\cdot(1-\beta)$, $(1-\alpha)\cdot\beta$, and $\alpha\cdot\beta$ represent the contributions from the corresponding pixels.

2.2.5 Scale-invariant Feature Transform

The Scale-Invariant Feature Transform (SIFT) is an algorithm designed to detect, describe, and match local features in images. Known for its robustness to various transformations, including scaling, rotation, and changes in illumination, SIFT is a good tool for using image alignment.

The first step of the SIFT algorithm involves the detection of key points within each image. To find these key points, the algorithm constructs a pyramid of blurred

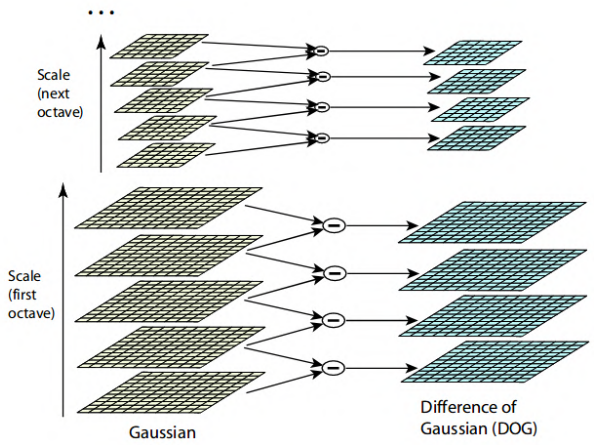


Figure 2.5: Gaussian pyramid formed by applying the Gaussian filter.

images, forming the Gaussian pyramid, to represent the image at different scales (see Figure 2.5).

The Difference of Gaussians (DoG) is computed by subtracting adjacent levels of the Gaussian pyramid, enhancing the presence of features at different scales. Local extrema in scale-space are identified by comparing each pixel in the DoG pyramid with its 26 neighbors in the current scale, as well as the corresponding pixels in the scales above and below. Detected key points are refined by eliminating points with low contrast or poorly localized extrema. This is achieved by comparing the intensity at the extrema with a threshold and discarding key points that do not meet certain criteria. The algorithm applies interpolation to accurately determine the location of key points between the sampled pixels in the DoG pyramid.

For each key point, the algorithm assigns a dominant orientation based on the local gradient directions in the image. This orientation is added to the descriptor vector. Using these vectors and the Euclidean distance, corresponding points are located in different images. An example of this output is illustrated in Figure 2.6.

To enhance the reliability of matches, the SIFT method can be further improved using the Random Sample Consensus (RANSAC) algorithm. RANSAC works by randomly selecting a minimal subset of matches (e.g., four pairs of key points) and estimating the transformation based on this subset. The estimated transformation is then applied to all other key points. The algorithm counts the number of inliers, i.e., key points that agree with the estimated transformation within a certain tolerance. This process is repeated for a predefined number of iterations, and the transformation with the maximum number of inliers is selected as the final transformation model.

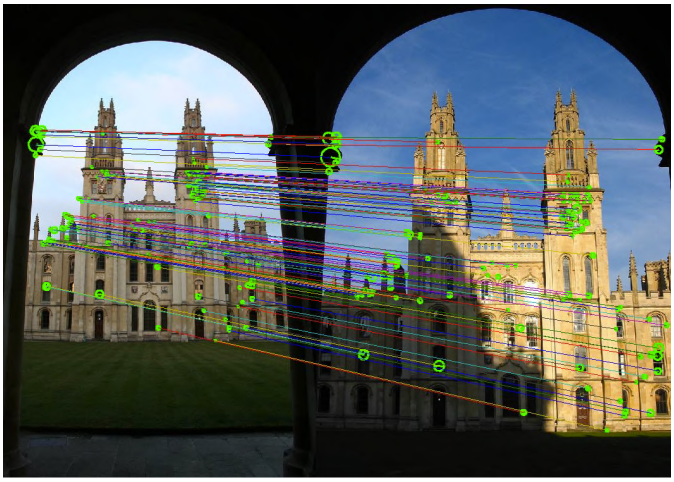


Figure 2.6: Example output of the SIFT algorithm.

2.3 Metrics

2.3.1 Structural Similarity

The Structural Similarity Index Measure (SSIM) is a perception-based method used for assessing the perceived quality of television or the similarity between images. Unlike mean square error (MSE), SSIM takes into account the inter-dependencies between pixels, particularly when they are spatially close. This consideration incorporates elements such as luminance masking, contrast masking, and internal structure, contributing to a more perceptually meaningful assessment.

Mathematically, the SSIM index is expressed as:

$$\text{SSIM}(x, y) = l(x, y)^{\alpha} \cdot c(x, y)^{\beta} \cdot s(x, y)^{\gamma} \tag{2.4}$$

Here, α, β, and γ are parameters controlling the relative importance of each component. The individual components $l(x, y)$, $c(x, y)$, and $s(x, y)$ are defined as follows:

$$\begin{aligned} l(x, y) &= \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \\ c(x, y) &= \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \\ s(x, y) &= \frac{2\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \end{aligned} \tag{2.5}$$

In these equations, μ_x, μ_y represent the means of the compared images, σ_x, σ_y are the standard deviations, σ_{xy} is the covariance, and c_1, c_2, c_3 are small constants

to avoid division by zero. These components capture essential aspects of luminance, contrast, and structure, providing a comprehensive evaluation of structural similarity.

2.3.2 Root Mean Squared Error

Root Mean Squared Error (RMSE) is a commonly used metric for evaluating the difference between two continuous variables, providing a measure of the average magnitude of the errors. In the context of image comparison, RMSE quantifies the difference in pixel values between two images.

Mathematically, RMSE is defined as:

$$\text{RMSE}(x, y) = \sqrt{\frac{1}{N * M * C} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{C} (x_{ijk} - y_{ijk})^2} \tag{2.6}$$

Here, x and y represent the pixel values of the corresponding images at each position (i, j, k). The summation loop extends over the dimensions of the images (N, M) and then over each color channel (C). In most cases C will be equal to 3 (rgb-images). The formula calculates the square root of the average squared differences between corresponding pixels. A lower RMSE values indicate better similarity, while higher values suggest greater dissimilarity. However, it is important to note that RMSE does not consider perceptual aspects or local structural information, making it sensitive to outliers.

2.3.3 Cross-Correlation

Cross-correlation is a mathematical operation that quantifies the similarity between two signals or sequences concerning their respective shifts. It measures the degree to which one signal resembles another when one is shifted over time or space. The cross-correlation R between discrete signals f and g is defined as:

$$R(m) = \sum_{n=-\infty}^{\infty} f(n) \cdot g(n - m) \tag{2.7}$$

Here, m is the displacement between the two signals. This operation essentially computes the inner product of the two sequences at each possible shift. The result, $R(m)$, provides a measure of similarity as a function of displacement. In the context of images, each signal represents pixel intensities in a 2D space. The cross-correlation function $R(m)$ calculates the similarity between two images by shifting one image over the other and measuring the similarity at each displacement

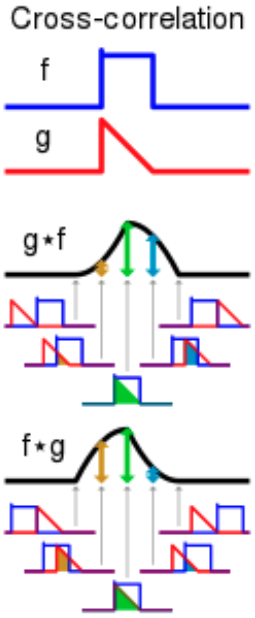


Figure 2.7: Cross-correlation of two functions.

m. This operation can be used to compute the normalized cross-correlation (NCC) for images A and B:

$$\text{NCC}(A, B) = \frac{\sum_x \sum_y [A(x, y) \cdot B(x - m, y - n)]}{\sqrt{\sum_x \sum_y A^2(x, y) \cdot \sum_x \sum_y B^2(x, y)}} \tag{2.8}$$

Here, $A(x, y)$ and $B(x, y)$ represent pixel values at coordinates (x, y) in images A and B respectively, and m and n are the displacements. Due to normalization, NCC varies between -1 and 1, where 1 indicates a perfect match.

2.3.4 Cosine Similarity

Cosine Similarity is a metric used to quantify the similarity between two vectors in a multidimensional space. Widely employed in various fields, including natural language processing and information retrieval, it is particularly valuable in measuring the similarity of documents, images, or other data represented as vectors. Mathematically, the cosine similarity between two vectors, A and B, is defined as:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{2.9}$$

Here, $A \cdot B$ represents the dot product of vectors A and B, and $\|A\|$ and $\|B\|$ denote the Euclidean norms of vectors A and B respectively.

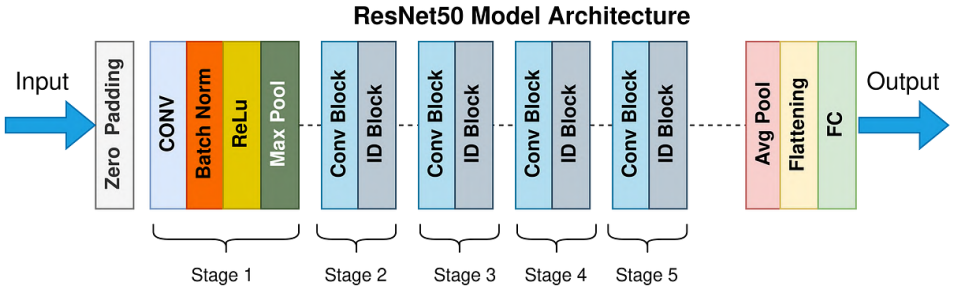


Figure 2.8: Architecture of ResNet50.

Cosine Similarity yields a value in the range of [-1, 1], where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates complete dissimilarity. In the context of image alignment and comparison, Cosine Similarity can be employed to assess the similarity between feature vectors extracted from images, providing a quantitative measure of their resemblance.

2.4 Neural Network

For this thesis only ResNet50 is investigated. ResNet stands for Residual Network and is a specific type of convolutional neural network (CNN) introduced in the paper "Deep Residual Learning for Image Recognition" by He et al., 2016. The key innovation in ResNet is the introduction of residual blocks. Traditional deep neural networks faced difficulties in training as the depth increased. The vanishing gradient problem led to the degradation of network performance with the addition of more layers. ResNet tackles this problem by introducing skip connections or shortcuts that allow the gradient to bypass certain layers during training. These shortcuts are called residual blocks and enable the model to learn residual functions, making it easier to train very deep networks. ResNet50 is a specific type of ResNet containing 50 distinctive layers. The general structure of ResNet50 is shown in Figure 2.8

Chapter 3

Pipeline Architecture

This section provides a overview of the pipeline architecture, illustrated schematically in Figure 3.1. Commencing with a qptiff dataset, the initial steps involve downsizing and conversion to tiff images. Subsequently, alignment procedures are applied to ensure spatial consistency. Following this, a neural network, trained on annotated areas and their corresponding images, is employed for precise cell detection. The culmination of this pipeline is the generation of a 3D image based on the identified cells.

3.1 Data

The dataset provided by the Vlaams Instituut voor Biotechnologie (VIB) comprises two distinct datasets, each corresponding to different biopsy samples. Each biopsy is sectioned into slices with a thickness of 5×10^{-6} m. These slices undergo staining with four different types of staining materials, consistently applied in the same sequence. Consequently, there exists a spatial interval of 20×10^{-6} m between consecutive H&E stained slices. Notably, each biopsy consists of 100 H&E stained images. The image format employed is qptiff, and individual image sizes range from 900 to 1300 megabytes.

An illustrative example of these images is presented in Figure 3.2. Within each image, six distinct classes are discernible: adipocytes (fat cells), blood vessels, air bubbles under the cover slip (gray areas), trash (dark dots), epithelium (protective cells) and stroma. In Figure 3.3 are visual representations of these classes depicted.

Given the thinness of the slices and the imprecision of the imaging apparatus, slight movements in the slices occur, see Figure 3.4. Consequently, image alignment becomes imperative to ensure accurate analysis and interpretation of the acquired data.

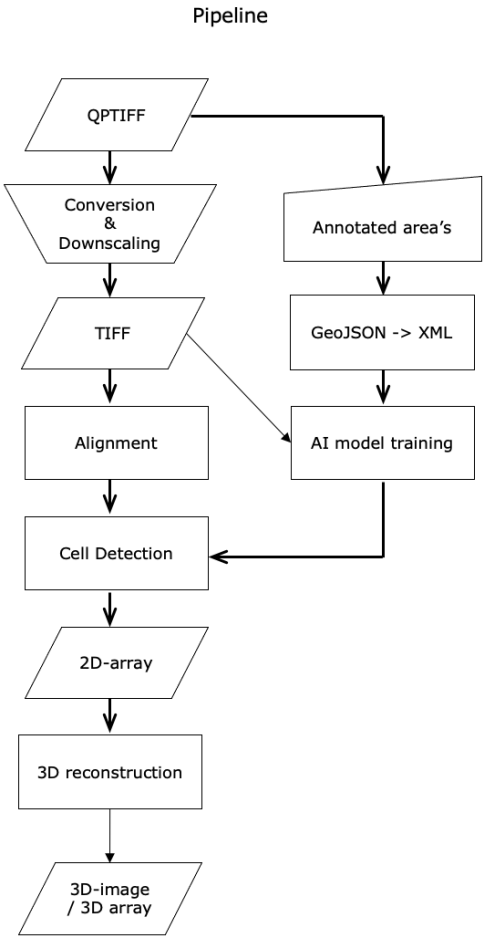


Figure 3.1: A schematic overview of the pipeline. For additional information regarding symbols see Appendix A.

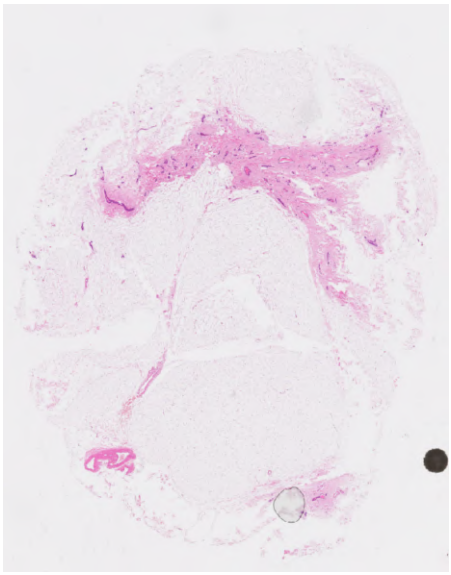
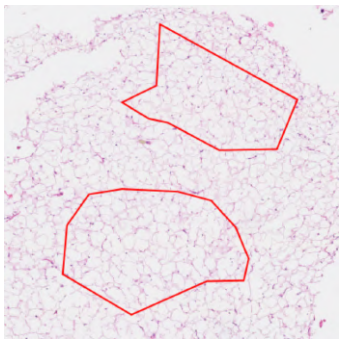
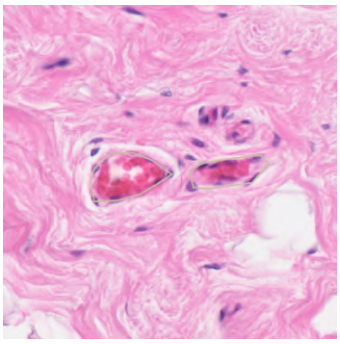


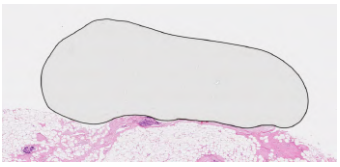
Figure 3.2: Example of qptiff image.



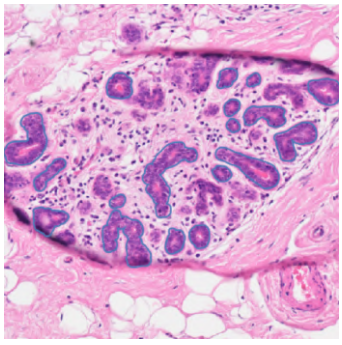
(a) Adipocytes



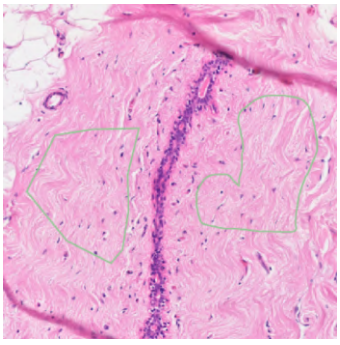
(b) Bloodvessel



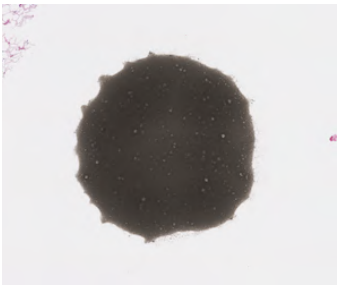
(c) Air bubble



(d) Epithelium



(e) Stroma



(f) Trash

Figure 3.3: The different important and distinguishable areas in H&E images.

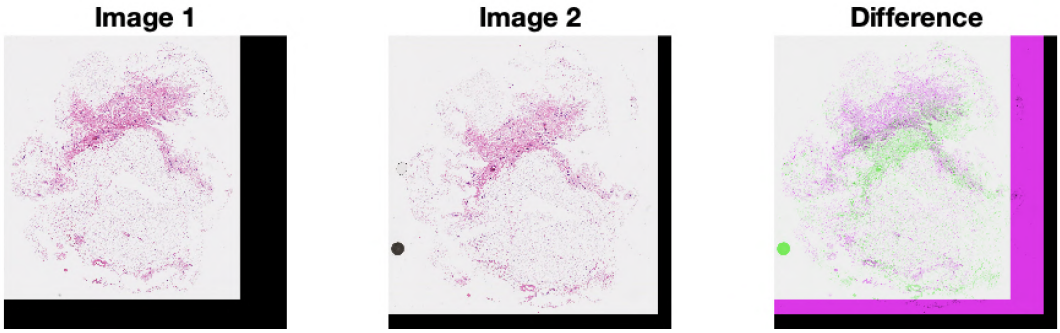


Figure 3.4: The difference between two consecutive images.

3.2 Image Conversion

The initial step in the pipeline involves converting and downscaling the images. This is imperative as the output images are in the qptiff format, which is incompatible with Matlab and only supported by a limited number of Python packages. Additionally, each qptiff image is too large for complex computations. The pipeline code specifically demands images in tif (tiff), png, or jpeg formats. Given that the tif format supports lossless compression, ensuring preservation of all image information, it is chosen for testing and pipeline development.

Two distinct methodologies can be employed for the conversion and downscaling process. The first option utilizes the pyvips package in Python, facilitating a direct conversion from qptiff to tif. This method requires specifying the desired output dimensions as a variable. The second option involves reading the image in QuPath, offering an image export functionality where both the output type and down sampling factor can be chosen.

The first method provides the advantage of automating the conversion process through a separate pipeline that iterates over all images. However, a notable drawback is the distortion introduced among the images due to the fact that each image has different dimensions but a similar tissue area size. Furthermore, when training the neural network, a file using original image coordinates is needed. Downsizing with standardized dimensions complicates the conversion of coordinates, as the scaling factor depends on the proportion between the original and new dimensions. Because the original images all vary in dimensions the coordinate conversion becomes more difficult.

The second method does not encounter these problems as the relative tissue area size remains constant. Additionally, downscaling becomes easier since division by a constant factor is all that is needed. However, a disadvantage of the second method is that each image requires manual conversion. For this thesis the second method is used due to it's convenience regarding the coordinate conversion.

3.3 Image Alignment

The image alignment process is a critical component in the pipeline, aiming to ensure spatial consistency among the acquired images. This section delineates two primary aspects of the image alignment process: the methodology derived from CODA by Kiemen et al., 2022 (Section 3.3.1), and additional implementations and methods employed to refine the alignment (Section 3.3.2).

3.3.1 CODA-based Alignment

The initial step in the CODA-based alignment involves the creation of a binary tissue mask. This process commences by subtracting a constant fill value from the image, followed by filtering based on mean absolute differences. Subsequently, small and low-variation regions are eliminated, and morphological operations are applied to enhance and connect identified tissue regions, culminating in the generation of a binary tissue mask (see Figure 3.5). This mask assumes a crucial role in quantifying the percentage of tissue within each registration region of interest (ROI).

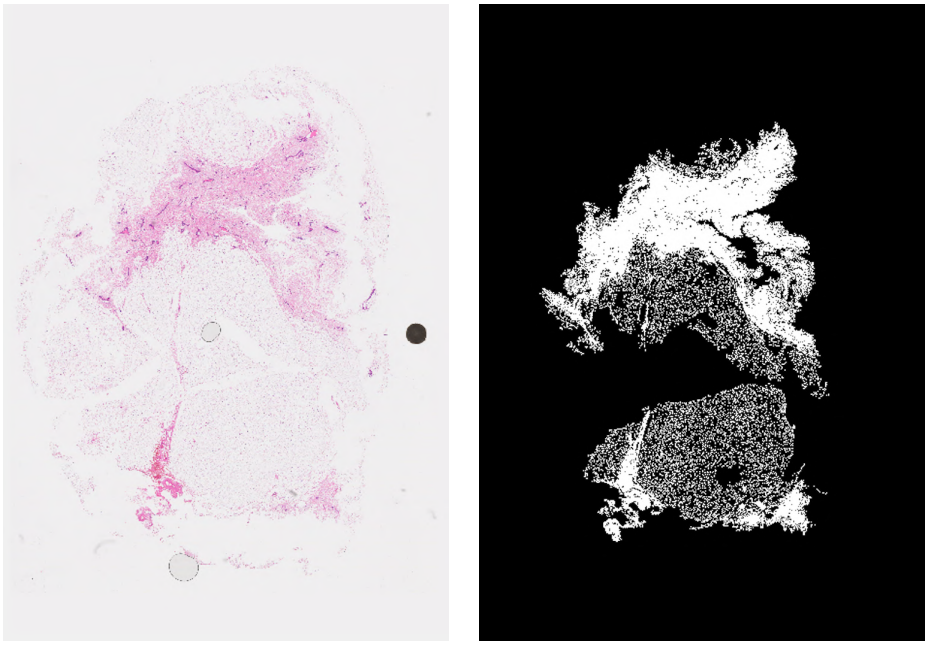
Following the creation of the tissue mask, the images undergo further processing. This includes padding addition, conversion to a grayscale image, and the application of a Gaussian filter. These preprocessing steps are essential for creating a suitable input for the subsequent alignment stages.

The image alignment process comprises of two distinct steps: global alignment and local alignment. In global alignment, the entire image undergoes rotation and translation to achieve optimal spatial correspondence. The rotational orientation is determined by comparing the radon transformation of both images at different angles. Once the correct orientation is established, image translation is calculated using particle image velocimetry and optimization via cross-correlation.

For local alignment, the images are segmented into a grid, and each grid cell is treated as a separate ROI. Cost functions are created for each ROI, representing the overlay with warped images. These cost functions are then optimized using gradient descent, refining the local alignment based on distinctive features within each ROI. The resulting local, rigid registration fields are interpolated to create a final grid.

3.3.2 Additional Implementations and Methods

In addition to the CODA-based alignment, supplementary implementations and methods are incorporated to further refine or improve the alignment process. Due to the results outlined in Chapter 4 the implementations are limited or in addition to the global alignment process.



(a) Normal tissue image (b) Tissue mask of image

Figure 3.5: Tissue area used for local registration.

To optimize the pipeline created by CODA, an alternative technique has been explored, involving the transformation of images using the log-polar coordinate system. Leveraging this coordinate transformation, an attempt has been made to optimize the rotation alignment, exploring potential improvements over the traditional method. This alternative methodology is applied after the global translation process discussed in previous section. For this optimization the log-polar transformation is applied on the images, after which a optimal rotation of determined by comparing the pixel values of both images.

Additionally, a separate pipeline has been created to evaluate the Scale-Invariant Feature Transform (SIFT) algorithm for image alignment. This is completely separate of the process explained in Section 3.3.1. Within this, key point matching is performed using the SIFT algorithm. Based on these key points the transformation matrix is computed to align the images. Further refinement steps, such as RANSAC, are applied to enhance the accuracy of the alignment.

In order to evaluate these different methodologies a file devoted to the evaluation is also added to the pipeline. This file includes an in-depth analysis of the aligned images using the following metrics: Structural Similarity Index (SSIM), Cosine Similarity, Root Mean Square Error (RMSE), and Summation over Subtracted Images. Each of these metrics are in depth explained in Section 2.3. These

metrics provide a comprehensive understanding of the performance of each alignment method, aiding in the selection of the most suitable technique for the given dataset.

3.4 Cell Recognition

The cell recognition takes place after the image alignment. This process has two different approaches: an automated algorithm based on standard values extracted from pre-processed images and a trained neural network.

3.4.1 Automated Algorithm

Within the pipeline, there is a capability to execute a form of cell recognition utilizing a standard algorithm introduced by Kiemen et al., 2022. The fundamental concept of this algorithm involves the identification of small sections of stroma or analogous cell types. Consequently, this yields a limited set of detected cells not limited to stroma cells. These detected cells can be employed to establish an initial set of points or, as undertaken in this thesis, to assess the visualization of plotted images.

3.4.2 Deep learning

Beyond the discussed cell recognition method, the pipeline extends its capabilities to include cell detection through advanced deep learning methodologies. The chosen architectural framework for this purpose is ResNet50, a convolutional neural network renowned for its efficacy in image recognition tasks. However, to deploy this model effectively, a crucial prerequisite is a comprehensive training phase.

The training process mandates the availability of an XML file containing specific coordinates delineating annotated areas. Given that the images are stored in the qptiff file format, the annotations are exclusively performed using QuPath. Upon exporting, QuPath utilizes the GeoJSON format to encode this crucial spatial information.

To address the XML file requirement, an initiative was undertaken to develop a JavaScript implementation capable of translating GeoJSON annotations into the essential XML format. The current state of this script successfully achieves a transformation into the general XML structure (see Figure 3.6). However, the intricacies lie in the subsequent conversion of this XML file into a MATLAB struct array with a highly specific architecture. Further refinement is needed in the conversion script from GeoJSON to XML to seamlessly adapt to MATLAB's struct array requirements.

Upon resolution of this detail, the remaining code for training and detection is expected to seamlessly integrate. The incorporation of deep learning techniques within the pipeline holds the promise of significantly augmenting cell detection accuracy, thereby broadening the methodology's applicability. Future endeavors will be directed towards fine-tuning the GeoJSON to XML conversion process, paving the way for the effective utilization of ResNet50 in cell detection for qptiff images.

3.5 3D-reconstruction

After cell detection, a 3D-image can be generated based on the coordinates of the detected cells. These convert into a matrix which then in term is converted into a 3D-structure. This reconstruction however does require a large amount of input points. For testing this code I used the points detected by the algorithm described in Section 3.4.1. However the amount of points is too small for a good image, but Figure 3.7 shows that the code is function provided it has correct input.

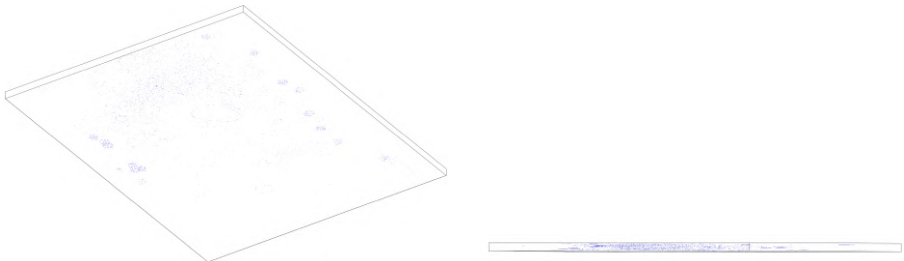
```
{"type":"FeatureCollection","features":[{"type":"Feature","id":"ffe64744-8ba6-4a94-9e2c-334bbe073678","geometry":{"type":"Polygon","coordinates":
[[[31540, 11886],[31536.5, 11887],[31532, 11887],[31525, 11889],[31523.33, 11890],[31520, 11890],[31513, 11892],[31512.55, 11892.27],[31510,
11893],[31505, 11896],[31504, 11897],[31502, 11898],[31501.18, 11898.66],[31500, 11899],[31497.28, 11900.64],[31497.27, 11900.64],[31496, 11901],
[31491, 11904],[31490.5, 11904.5],[31488, 11906],[31485, 11908],[31484, 11909],[31481, 11911],[31476, 11915],[31472, 11919],[31471, 11920],[31469.
5, 11922],[31468, 11923],[31466.81, 11924.48],[31465, 11925],[31460, 11928],[31458, 11929],[31457, 11930],[31456, 11930.8],[31454, 11932],[31453.
4, 11932.6],[31452, 11933],[31447, 11936],[31444, 11939],[31442, 11941],[31439, 11943],[31437.67, 11944.67],[31437, 11945],[31435, 11946],[31434,
11946.8],[31432, 11948],[31430, 11950],[31428.33, 11951.33],[31427, 11952],[31426, 11952.8],[31424, 11954],[31423, 11955],[31415, 11960],[31410,
11964],[31404, 11970],[31400, 11974],[31398, 11977],[31395, 11980],[31394, 11981],[31393, 11982],[31388, 11987],[31386, 11989],[31385, 11990],
[31384, 11991],[31383, 11992],[31382, 11993],[31379, 11995],[31375, 12000],[31373, 12002],[31371.8, 12004],[31371, 12005],[31366, 12010],[31362,
12015],[31361.5, 12016.5],[31361, 12017],[31359.8, 12019],[31359, 12020],[31358, 12022],[31357.75, 12022.75],[31357, 12024],[31355.91, 12027.82],
[31354, 12031],[31352.73, 12035.45],[31350, 12040],[31348, 12042],[31347, 12043],[31345.5, 12045.5],[31345, 12046],[31342, 12051],[31341.33, 12051.
83],[31341, 12052],[31340, 12052.8],[31338, 12054],[31336, 12055],[31335, 12055.8],[31333, 12057],[31332, 12058],[31330.75, 12058.75],[31330,
12059],[31329, 12059.8],[31327, 12061],[31325, 12063],[31320, 12067],[31318.86, 12068.71],[31316, 12071],[31314, 12074],[31313, 12075],[31312.77,
12075.38],[31312, 12076],[31310, 12079],[31309, 12080],[31306, 12085],[31305.63, 12086.28],[31305.5, 12086.5],[31305, 12087],[31302, 12092],
[31300, 12099],[31300, 12099.33],[31299, 12100],[31297.3, 12102.13],[31293, 12105],[31291.67, 12106.67],[31290, 12108],[31288, 12111],[31283,
12116],[31280, 12119],[31278.5, 12121.5],[31278, 12122],[31275, 12127],[31274.33, 12129.33],[31274, 12130],[31273.75, 12130.75],[31273, 12132],
[31271, 12139],[31271, 12142],[31271, 12143],[31271, 12144],[31271, 12145],[31271, 12149],[31271, 12154],[31271, 12156],[31271, 12157],[31271,
12161],[31271.19, 12161.68],[31269, 12163],[31267.5, 12164.5],[31265, 12166],[31263.5, 12167.5],[31261, 12169],[31259, 12170],[31257, 12171],
[31256.72, 12171.22],[31254, 12172],[31249, 12175],[31248, 12176],[31245, 12178],[31243.52, 12179.85],[31243, 12180],[31238, 12183],[31231, 12190],
[31230.09, 12190.55],[31225, 12192],[31220, 12195],[31218, 12196],[31216.67, 12197],[31216.25, 12197.25],[31214, 12198],[31212.63, 12198.82],
[31212, 12199],[31210.75, 12199.75],[31210, 12200],[31208, 12201],[31206, 12202],[31205, 12202.8],[31203, 12204],[31201, 12206],[31198.78, 12207.
78],[31198, 12208],[31196.63, 12208.82],[31196, 12209],[31194.75, 12209.75],[31194, 12210],[31190, 12212],[31188, 12213],[31183, 12216],[31181,
12218],[31180, 12219],[31178, 12221],[31176, 12223],[31173, 12226],[31171, 12228],[31170, 12229],[31168, 12232],[31167, 12233],[31164, 12238],
```


(a) GeoJSON


```
<?xml version="1.0"?>
<Annotations MicronsPerPixel="0.250000">
  <Annotation Id="1" Name="BALT hyperplasia" ReadOnly="0">
    <Attributes>
      <Attribute Name="Description" Id="0" Value=""/>
    </Attributes>
    <Regions>
      <RegionAttributeHeaders>
        <AttributeHeader Id="9999"/>
        <AttributeHeader Id="9999"/>
        <AttributeHeader Id="9999"/>
      </RegionAttributeHeaders>
      <Region Id="123">
        <Attributes/>
        <Vertices>
          <Vertex X="3942.5" Y="1485.75"/>
          <Vertex X="3942.0625" Y="1485.875"/>
          <Vertex X="3941.5" Y="1485.875"/>
          <Vertex X="3940.625" Y="1486.125"/>
          <Vertex X="3940.41625" Y="1486.25"/>
          <Vertex X="3940" Y="1486.25"/>
          <Vertex X="3939.125" Y="1486.5"/>
          <Vertex X="3939.06875" Y="1486.53375"/>
          <Vertex X="3938.75" Y="1486.625"/>
          <Vertex X="3938.125" Y="1487"/>
          <Vertex X="3938" Y="1487.125"/>
          <Vertex X="3937.75" Y="1487.25"/>
```


(b) XML

Figure 3.6: The difference between GeoJSON and XML.



(a) 3D top view

(b) 3D side view

Figure 3.7: 3D-image based on points from Section 3.4.1.

Chapter 4

Results

4.1 Image Alignment

For the image alignment 4 different methods were tested, all of which are explained in Section 3.3. For 3 methods the results of the alignment are shown in Table 4.1. It is clearly visible that the local registration yield the best alignment. However, looking at the computational time the global alignment has the best performance.

4.2 Scale Invariant Feature Transformation

The application of the Scale-Invariant Feature Transform (SIFT) algorithm did not yield satisfactory results in the context of this study. While the algorithm successfully identified various key points within individual images (see Figure 4.1, the subsequent establishment of key point connections between two images did not produce the desired outcome, as illustrated in Figure 4.2(a). Even with the implementation of the Random Sample Consensus (RANSAC) algorithm, depicted in Figure 4.2(b), the transformation results remained nonsensical. Consequently, alignment metrics were not applied to the output of the SIFT algorithm due to the lack of meaningful transformations.

Performance of alignment methods					
Alignment method / Metric	SSIM	RMSE	Normalized subtraction	COS SIM	Computational time (s)
No registration	0.728	68.651	0.052	0.928	-
Global registration	0.854	16.804	0.012	0.997	**338**
Log polar	0.840	18.623	0.014	0.996	375
Local registration	**0.856**	**16.617**	**0.09**	**0.998**	442
Average	0.820	30.173	0.042	0.998	385

Table 4.1: Performance metrics of the image alignment.

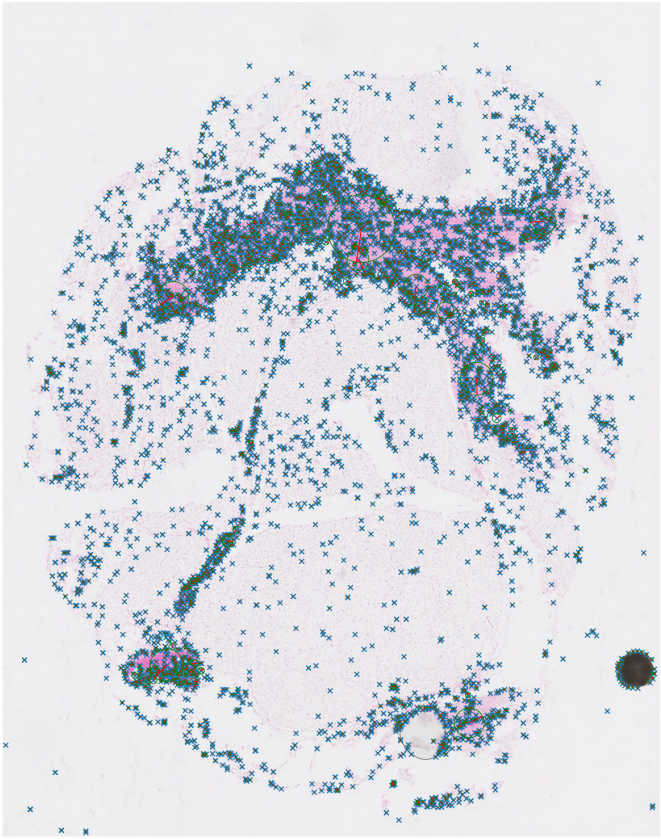
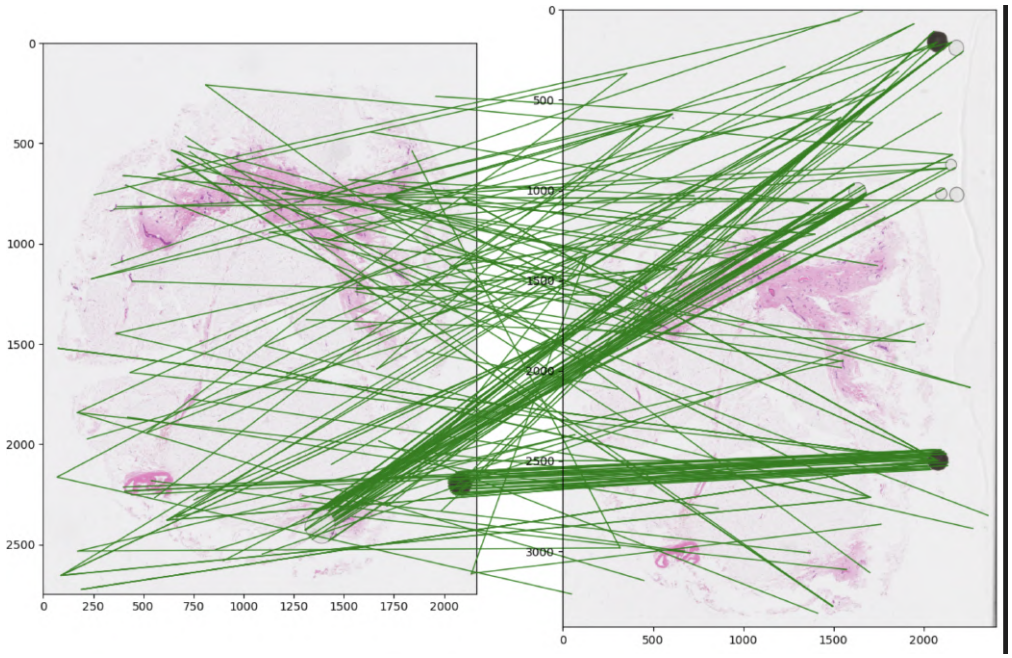
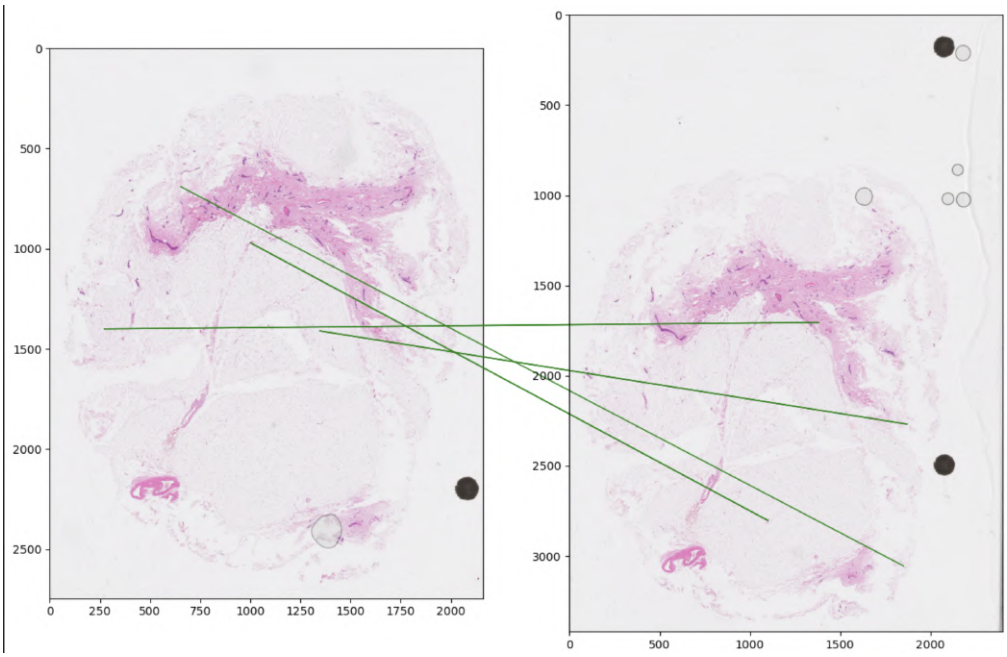


Figure 4.1: Detected key points within an image.



(a) Key points pairs without RANSAC



(b) Key points pairs with RANSAC

Figure 4.2: Key point connections between images with and without RANSAC.

Chapter 5

Discussion

5.1 Analysis

The primary goal of this thesis was to develop a robust pipeline for the analysis of biomedical images by creating insightful 3D images. The pipeline involved a series of steps including image conversion, alignment, and cell recognition, culminating in the construction of 3D images. In almost all of these steps the pipeline is operational. The only remaining refinement lies in the specific conversion process from GeoJSON to XML, and once completed, the pipeline is expected to be fully functional.

The evaluation of various image alignment methods provides insights into their performance and implications for the pipeline. Notably, the SIFT algorithm exhibited wrong results, likely attributed to the limited dissimilarity between images in terms of RGB values and distinctive features. The algorithm relies on these distinctions for effective key point matching, leading to poor connections when such differences are lacking. As a result, the SIFT algorithm may not be suitable for datasets with inherently low image dissimilarity.

Furthermore, although the local registration method demonstrates superior alignment, the improvement with respect to the global alignment is not substantial. This can be caused by the inherent characteristics of the dataset, where the average size of a cell is 5 μm, while consecutive images are 20 μm apart. The local alignment primarily addresses small corrections necessary for individual cells, which may not significantly contribute to the overall alignment process. Considering the larger computational time required by the local alignment method, its application might not provide substantial value compared to the global alignment. Consequently, the decision was made to exclusively optimize the global alignment, as it aligns more effectively with the dataset characteristics and strikes an optimal balance between accuracy and computational efficiency.

5.2 Future work

Future work for this research involves several key steps. Firstly, the immediate priority is to finalize the pipeline. While the pipeline has been substantially developed and exhibits promising functionalities, the GeoJSON to XML conversion requires refinement in order to integrate with MATLAB's struct array requirements. Once this aspect is successfully addressed, the pipeline can be considered fully operational.

Subsequently, attention should be directed towards optimizing the deep learning component of the pipeline. Currently, the pipeline incorporates ResNet50 for cell detection, but further enhancements can be achieved by exploring advanced architectures such as Region-based Convolutional Neural Networks (RCNN) or U-shaped Fully Convolutional Networks (FCN). These architectures have been deployed successfully in pixel wise segmentation tasks, as demonstrated by studies such as Zhang and Chi, 2020 and Sambyal et al., 2020.

Finally, an important future exploration involves deploying the operational pipeline on multiple tissue images to evaluate its performance across diverse datasets. This step is crucial for assessing the generalizability and adaptability of the pipeline to varying biomedical contexts. Additionally, it allows for testing the practical implications of the pipeline and assessing its potential additional value to the field of biomedical image analysis.

Chapter 6

Conclusion

This thesis aimed to develop a robust pipeline for biomedical image analysis. The focus was on setting up the pipeline using Kiemen et al., 2022's code as a foundation and enhancing it for better results. Though not fully operational, significant progress has been made in establishing a comprehensive setup.

Exploring different image alignment methods highlighted the efficacy of global alignment, especially given the dataset's characteristics. Local registration, providing marginally better alignment, was considered less significant due to cell spatial separation. Attempts to enhance global image alignments, such as SIFT and log polar transformations, yielded less favorable results compared to the foundational approach.

Deploying the operational pipeline on multiple tissue images is pivotal for assessing its performance across diverse datasets. This broader testing validates functionality and evaluates adaptability to various biomedical imaging scenarios. Importantly, it provides an opportunity to examine practical implications and assess potential additional value in biomedical image analysis.

Bibliography

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kiemen, A. L., Braxton, A. M., Grahn, M. P., Han, K. S., Babu, J. M., Reichel, R., Jiang, A. C., Kim, B., Hsu, J., Amoa, F., et al. (2022). Coda: Quantitative 3d reconstruction of large tissues at cellular resolution. *Nature Methods*, *19*(11), 1490–1499.

Lotz, J. M., Hoffmann, F., Lotz, J., Heldmann, S., Trede, D., Oetjen, J., Becker, M., Ernst, G., Maas, P., Alexandrov, T., et al. (2017). Integration of 3d multimodal imaging data of a head and neck cancer and advanced feature recognition. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, *1865*(7), 946–956.

National Cancer Institute. (2023, January). How cancer is diagnosed. https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis

Open Medscience. (2023, April). A dive into the world of 3d medical imaging. https://openmedscience.com/a-dive-into-the-world-of-3d-medical-imaging/

Sambyal, N., Saini, P., Syal, R., & Gupta, V. (2020). Modified u-net architecture for semantic segmentation of diabetic retinopathy images. *Biocybernetics and Biomedical Engineering*, *40*(3), 1094–1109.

Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018). Cell detection with star-convex polygons. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, 265–273.

Silén, C., Wirell, S., Kvist, J., Nylander, E., & Smedby, O. (2008). Advanced 3d visualization in student-centred medical education. *Medical Teacher*, *30*(5). https://doi.org/10.1080/01421590801932228

University of Michigan Medical School. (n.d.). Introduction to histology stains. https://histology.medicine.umich.edu/resources/introduction-histology-stains

Zhang, Y., & Chi, M. (2020). Mask-r-fcn: A deep fusion network for semantic segmentation. *IEEE Access*, *8*, 155753–155765.

Appendix A

Symbol overview