UNIVERSITY OF COPENHAGEN

Detection of pyramidal neurons and analysis of their characteristics from 3D microscopy images.

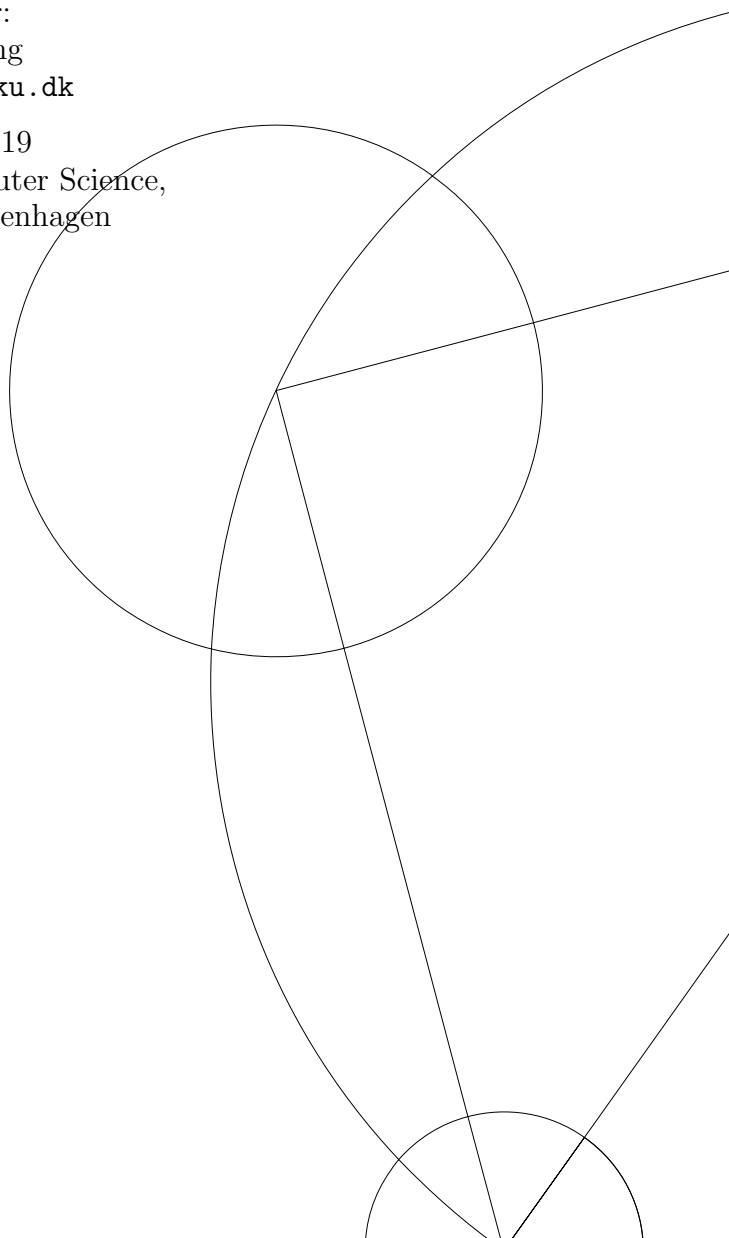
MSc Thesis in Computer Science

Cecilie Novak
bqk662@alumni.ku.dk

Supervisor:
Jon Sporring
sporring@di.ku.dk

March 4, 2019

Department of Computer Science,
University of Copenhagen



1 Abstract

Pyramidal neurons are neurons in the brain that are named for their appearance that resembles a pyramid structure. The arrangement of pyramidal neurons in the brain is important to neuro-biologist, but it is a slow and time consuming task to determine their arrangement, so neuro-biologists need faster tools.
The intent of this master thesis project will therefore be to implement a program that can automatically detect neurons in 3D microscopy images, annotate detected neurons and estimate the direction of pyramidal neurons. Pyramidal neurons are named by their appearance in 2D image sections and we therefore do not know if they look like actual pyramids in the 3D environment that they belong to. The second attempt of this master thesis project will therefore be to investigate how accurate this perception of pyramidal neurons is.
Two methods for shape analysis will be presented and experimentally applied to 3D objects that represent pyramidal neurons. The two methods are principle component analysis and spherical harmonics analysis. The former is a linear method and the latter is a non-linear method. We aim to determine which of the two methods is best for doing shape analysis of pyramidal neurons.
The master thesis project resulted in a program that can perform semi-automatic registration of image sections, automatic detection of neurons, only manual annotation, automatic shape analysis using either of the two methods and do simple statistics on pyramidal neurons.
The shape analysis revealed that principal component analysis is the best for doing shape analysis of neurons, even though neither of the methods seem good for pyramidal neurons.

Contents

1 Abstract **2**

2 Introduction **5**
2.1 Neurons in the brain . . . 5
2.1.1 Pyramidal neurons . . . 6
2.1.2 Astrocyte neurons . . . 7
2.2 Tissue samples and image data . . . 7
2.3 Hardware and software . . . 9

3 Preliminary work: Image registration **10**
3.1 Type of image transformation . . . 10
3.2 Optimization algorithms and metrics . . . 10
3.3 Chosen image registration scheme . . . 11
3.3.1 Coarse registration . . . 11
3.3.2 Fine registration . . . 12

4 Segmentation of neurons **13**
4.1 Data observations . . . 14
4.2 Chosen segmentation scheme . . . 14
4.2.1 Preprocessing . . . 14
4.2.2 Computation of pixel features . . . 15
4.2.3 Choice of threshold values . . . 16
4.2.4 Postprocessing . . . 18
4.2.5 Segmentation examples . . . 19

5 Annotation of neurons **21**
5.1 Theory based annotation . . . 21
5.2 Annotation based on classification methods . . . 21
5.3 Chosen annotation method . . . 22
5.4 Frequency analysis . . . 22
5.4.1 Spherical harmonics analysis . . . 23
5.4.1.1 Spherical coordinates . . . 23
5.4.1.2 Spherical harmonics . . . 23
5.4.1.3 Decomposition into frequency components . . . 24
5.4.1.4 Reconstruction of data observations . . . 24
5.4.2 Pros and cons of the method . . . 25
5.5 Rotation invariant shape descriptors from spherical harmonics . . . 25
5.6 Choosing an optimal number of classes . . . 27
5.6.1 Applying minimum description length to the k-means method . . . 28
5.7 Annotation experiment: K-means classification of rotation invariant shape descriptors . . . 29
5.7.1 Hypothesis . . . 30
5.7.2 Experimental setup . . . 30
5.7.3 Results . . . 30
5.7.3.1 Optimal number of classes for neurons . . . 31
5.7.3.2 Ability of the descriptor to separate neurons . . . 34
5.7.4 Conclusion on annotation experiment . . . 36

6 Shape analysis of pyramidal neurons **36**
6.1 Appropriate shape representation . . . 37
6.1.1 Pincus' and Theriot's method . . . 37
6.1.2 Extension of Pincus' and Theriot's method . . . 38
6.1.2.1 Uniform distribution of points on the unit sphere . . . 39
6.1.2.2 Alignment of neurons . . . 41
6.1.2.3 Computation of radial distances . . . 42
6.2 Principle component analysis of pyramidal neurons . . . 43
6.2.1 Principle component analysis . . . 43
6.2.1.1 The covariance matrix . . . 43
6.2.1.2 Decomposition into principle components . . . 44
6.2.1.3 Reconstruction of data observations . . . 44
6.2.1.4 Pros and cons of the method . . . 45
6.2.2 Applying principle components analysis to pyramidal neurons . . . 45
6.2.2.1 Explaination of shape variation . . . 45
6.2.2.2 Visualization of principle components . . . 46
6.2.2.3 Pyramidal neuron reconstruction . . . 48
6.3 Spherical harmonics analysis of pyramidal neurons . . . 49
6.3.1 Visualization of frequency components . . . 50
6.3.2 Pyramidal neuron reconstruction . . . 51
6.4 Comparison of the shape analysis methods . . . 52

7 Direction of pyramidal neurons **52**
7.1 Finding the apex . . . 53
7.2 Computing the direction . . . 54
7.3 Observed pyramidal neuron directions . . . 54

8 Statistics on pyramidal neurons **55**
8.1 Dimensions . . . 56
8.2 Volume and surface area . . . 58

9 Discussion on improvements and further work **59**

10 Conclusion **60**

11 References **61**

12 Appendix **63**
12.1 A1: Thesis contract . . . 63
12.2 A2: Code . . . 65

2 Introduction

One important challenge in neuro-biology today is that of determining the arrangement of neurons in the brain, since the arrangement might play an important role when it comes to various psychiatric and neurogenerative diseases such as Alzheimer's disease, epilepsy, schizophrenia and dementia. Another reason for the importance of this challenging task is its relation to the study and proving of the theory of minicolumnarity in the nervous system of the brain, as described in the introductions of [1] [2]. For those unfamiliar with the theory of minicolumnarity, the theory regards small vertical columns of neurons as elementary units of the nervous system of the brain.

It is a slow and time consuming task to manually determine the arrangements of neurons from the 3D microscopy images that depicts the tissue samples, as is it to manually annotate the different kinds of neurons that appear in the images. Even though it is slow and time consuming to do it manually, the neuro-biologists are often forced to do so, due to the lack of tools for automatic detection, annotation and direction estimation. Programs for automatic detection, annotation and direction estimation of neurons from such images are therefore needed. This thesis project will therefore attempt to implement a program for automatic detection, annotation and direction estimation of neurons that can aid the neuro-biologists in their research.

Due to the lack of automatic tools, neurons are often characterized and described based on the observations that neuro-biologists do when examining the tissue samples through microscopes. This has given rise to the naming of a certain type of neurons called pyramidal neurons, since their shape resemble that of a pyramid to the neuro-biologists. It has, however, never been investigated if these neurons actually have the shape of a pyramid or just a shape that reminds the neuro-biologists of a pyramid. Due to these circumstances, it will also be attempted to investigate how well this empirically based way of describing this type of neuron is by performing shape analysis of the pyramidal neurons detected using the implemented program. As pyramidal shapes are presumably not linearly expressable, the performance of the usually popular linear method principle component analysis (PCA) will be compared to the not so well-known nonlinear method spherical harmonics analysis (SHA), which is essentially frequency analysis in 3D.

It will be assumed that the reader of this thesis have a basic understanding of statistics, data analysis and modelling corresponding to that of a bachelor student in Computer Science. Additionally it will be assumed that the reader have basic knowledge of signal and image processing corresponding to having taken introductory courses on the subject.

2.1 Neurons in the brain

In the brain we have different types of neurons. They all have a cell body, which is called the soma. The soma consists of a cell core and a cell border and it varies in size and shape depending on the type of neuron. Besides the soma, all neurons have tree-like structures of thread-like fibers that are used for recieving neuronal signals from nearby neurons and also a tree-like structure of thread-like fibers for transmitting neuronal signals to nearby neurons. These are called the dendrites and the axon, respectively. Both the dendrites and the axon branches several times after emanating from the soma. Both the dendrites and the axon are too fine and small to be visible to us through microscopes, even when using the best currently available microscopes. [14] [12] [13]. See figure 1 and 2 for visual examples of neuron anatomies.

Neurons in the brain mostly belong to one of two dominant families of neurons, the family of excitatory neurons or the family of inhibitory neurons. Neurons are sorted into these families based on what substance they use as neurotransmitter and thus how they stimulate other neurons around them. Neurons that increase stimulation of other neurons around them

belong to the family of excitatory neurons, while neurons that decrease stimulation of other neurons around them belong to the family of inhibitory neurons [14] [13].

Neurons stimulate each other in order to promote or inhibit neuronal signals in the nervous system. They do so by releasing neurotransmitter substances using their axon. When a neurotransmitter substance is released from the axon of a neuron, nearby neurons that react to its neurotransmitter substance absorb it through their dendrites and thus become affected. When a neuron has absorbed sufficient amounts of the neurotransmitter substance that it produces itself, a biochemical electric response is caused inside the neuron. This respons is called its action potential. The action potential runs through the dendrites to the soma, further down the axon, causing the neuron to release its own neurotransmitter substance and thus promoting neuronal signals. Neurons do not only absorb neurotransmitter substances that they produce themselves, excitatory neurons also absorb the neurotransmitter substances from inhibitory neurons and vice versa. The absorbtion of a neurotransmitter substance that the neuron does not produce itself will delay the firing of action potentials, thus inhibiting neuronal signals. [13].

Two types of neurons are relevant to know more about in connection with this thesis project. Those are pyramidal and stellate neurons. Information about pyramidal and stellate neurons can be found in the following subsections.

2.1.1 Pyramidal neurons

Pyramidal neurons belong to the family of excitatory neurons. They are found in the parts of the brain called the cerebral cortex, hippocampus and amygdala [14] [12]. The cerebral cortex, hippocampus and amygdala are responsible for the different kinds of memory (long-term and short-term memory), for attention abilities, for awareness, decision making and conciousness and for emotional response. These are in general advanced cognitive abillities. Pyramidal neurons comprise about $\frac{2}{3}$ of all neurons in these brain areas and this type of neuron thus contribute to important functions of the brain and play an important role when it comes to advanced cognitive abillities [14] [13].

Pyramidal neurons can be distinguished from other neurons by the shape of their soma, which represents rounded pyramids. Their dendrites are usually divided into two groups named after the part of the soma that they emanate from. These groups are the apical dendrites that emanates from the apex of the soma, and the basal dendrites that emanates from the base of the soma [12]. For pyramidal neurons the axon is much longer than the dendrites and it usually emanates from the base of the pyramidal formed soma. Please see figure 1 for visual details of the anatomy of pyramidal neurons [12].

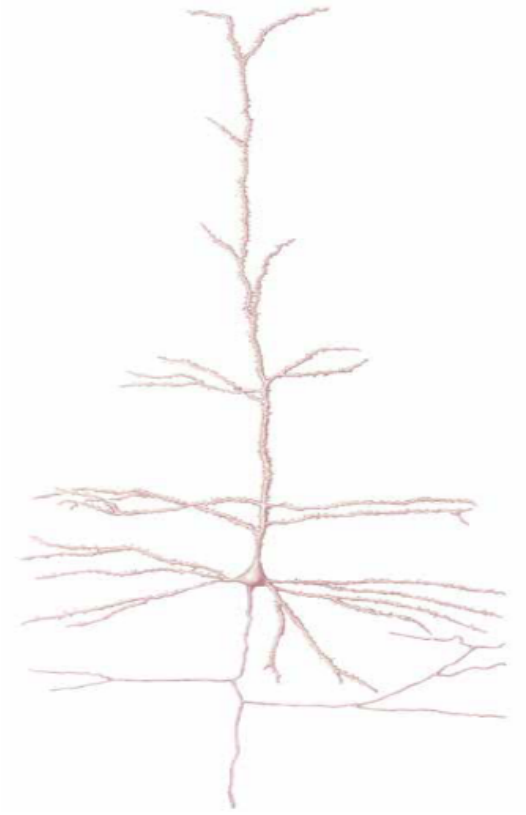


Figure 1: *Example illustration of a pyramidal neuron. Here, the apical dendrites are those pointing upwards, the basal dendrites are those that points to the sides, while the axon points downwards from the soma that can be seen in the middle.*

Pyramidal neurons come in various sizes. In the tissue samples available for the thesis, the soma of pyramidal neurons have a diameter of $13.45 \pm 0.24\mu m$. The average height of the soma of pyramidal neurons are about $20\mu m$. The diameter of the dendrites typically lies in the range from half a μm to several μm. The length of the dendrites are usually several hundreds μm long for pyramidal neurons and when branching they often measures several cm. The axon of pyramidal neurons that are much longer than the dendrites, can reach a length of many cm [12].

As mentioned, pyramidal neurons belong to the family of excitatory neurons and thus increase the stimulation of nearby neurons. They do so using the neurotransmitter substance glutamate, which is the most common neurotransmitter substance for excitatory neurons in the brain. In addition to using and responding to glutamate, they also respond to GABA, which is used by inhibitory neurons. [14] [13].

2.1.2 Astrocyte neurons

Astrocyte neurons can belong to both the family of excitatory neurons and the family of inhibitory neurons, but those found in the available tissue samples will mostly belong to the family of inhibitory neurons. Astrocyte neurons are found in several areas of the brain, including the frontal cortex, the cerebral cortex, hippocampus and hypothalamus. It is not well defined what proportion the astrocyte neurons comprise in these brains areas [15].

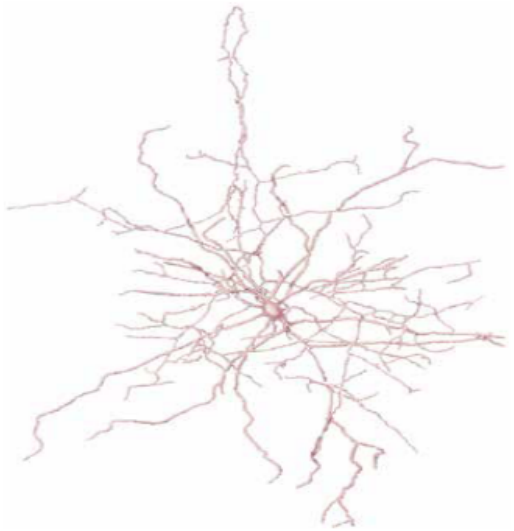


Figure 2: *Example illustration of an astrocyte neuron. The dendrites and the axon are of approximately the same length. Both emanate in arbitrary direction from the spherical soma that can be seen in the middle.*

Astrocyte neurons are characterized by having spherical somas and being considerably smaller than pyramidal neurons, as they have an average soma diameter of $5 - 8\mu m$ [15]. Their dendrites and axon are of approimately equal length. These features distinguishes them from pyramidal neurons.

As the astrocyte neurons relevant to the thesis, belong to the family of inhibitory neurons, they decrease the stimulation of nearby neurons. They do so by releasing the neurotransmitter substance called ATP [15]. Astrocyte neurons belonging to the class of excitatory neurons releases the neurotransmitter substance glutamate, like pyramidal neurons do [15].

2.2 Tissue samples and image data

The tissue samples will have their origins from 40 danish brains in total. Ten out of 40 brains were considered healthy, while the remaining 30 brains came from people diagnozed with one of three diseases, depression, Schizophrenia or Alzheimer's disease. Each disease were represented by tissue samples from 10 brains. It is unclear what age, gender and potential disease is associated to each of the brain samples.

All sections of tissues are almost $0.5\mu m$ thick and every tissue sample will be represented by 1000-2000 sections each. Tissue samples might shrink a bit during acquisition and conservation processes, but they do not differ substantially in size.

The tissue samples have been stained prior to the image acquisition, in order to enhance

the contrast of the images. During a staining process, the sample tissues are put in liquid dye solutions for a duration of time. Different dye solutions requires different staining durations and yields different looking results. It has been decided to use Toluidine dye for the tissue samples. Its optimal staining duration is 20 minutes. See figure 3 for examples of stained neurons and figure 4 for an example section of tissue samples.

The image data available for the thesis will consist of regular optical images of unaligned consecutive sections of tissue samples taken through a microscope that enlarges the samples by 40 times. This results in a pixel size of 0.5 x 0.5 μm for all sections. Each section will be photographed in high resolution - in the approximate size of 10.000-14.000 pixels in width and 12.000-16.000 pixels in height - and saved in TIFF format, yielding image files of ∼300-400 Mb for each section.

Since each tissue sample will be represented by 1000-2000 sections, the full data set will be of the approximate size of 12 Tb. Not all of the data set will be available during the thesis project, as the data set is currently being generated. However, but there will be enough data available for implementing and testing the program and for doing statistics on the characteristics of the pyramidal neuron cells. The program will be executed on as much data as time permits.

For testing and development of the program, 50 sections available from the start of the thesis has been used.

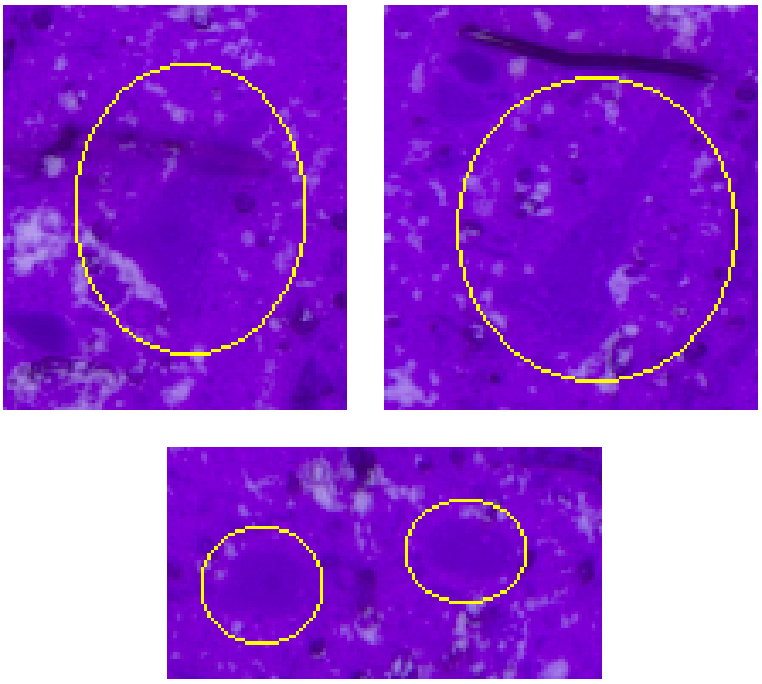


Figure 3: *Image clippings showing examples of stained neurons. The examples were obtained using Toluidine dye and a staining duration of 20 minutes. The two image clippings above displays one pyramidal neuron each, while the image clipping below display two astrocyte neurons. The neurons are marked by rings.*

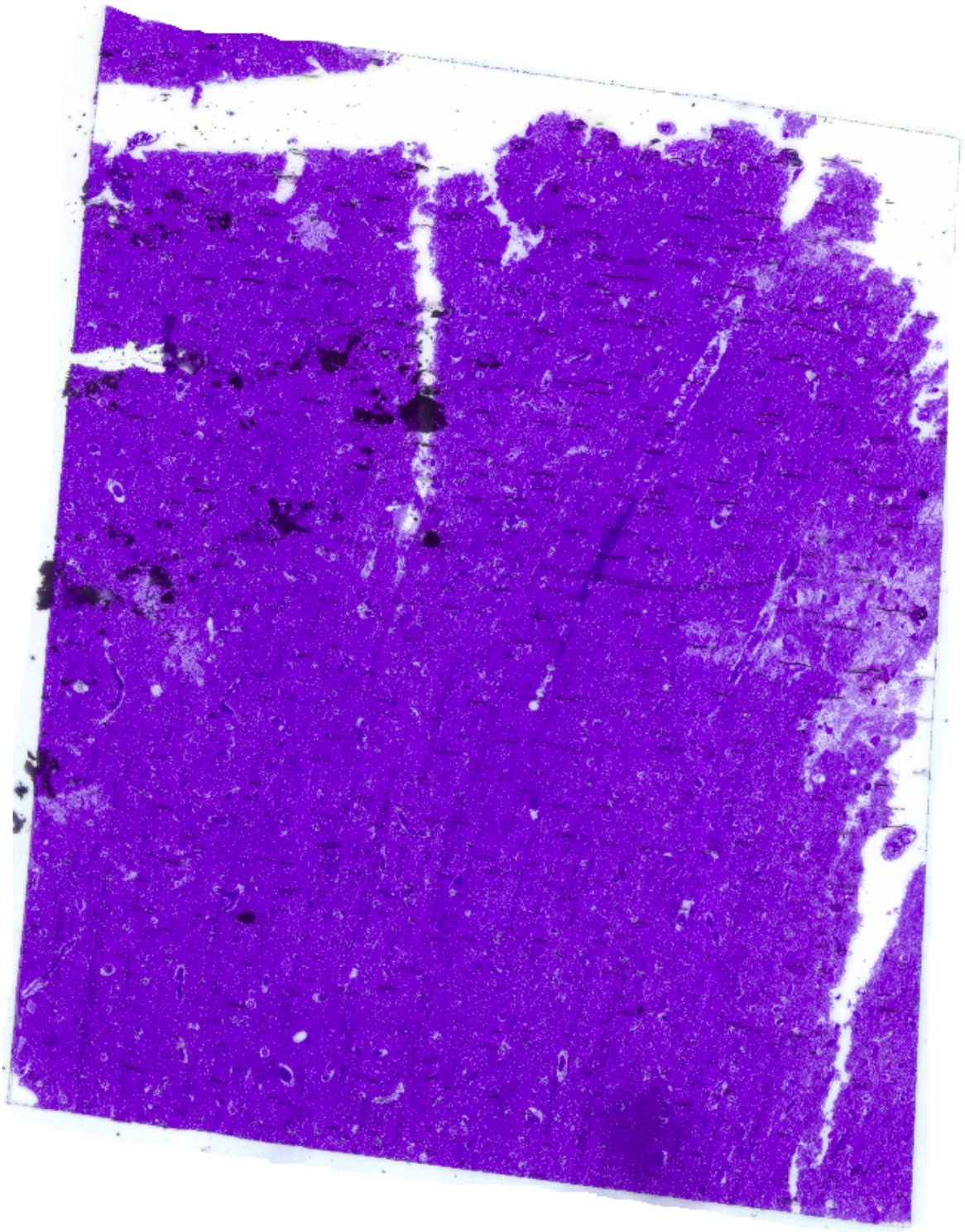


Figure 4: *Example section from a tissue sample. The section is clearly not aligned to anything and contains artefacts from the preprocessing of the tissues. For example, there are tears originating from the staining process and dark spots originating from the conservation process.*

2.3 Hardware and software

The program will be implemented in Matlab 9.4.0.813654 also known as Matlab R2018a and it will be made for TIFF images. The SLURM computing cluster available at DIKU image section was used for execution of the program. This computing cluster contains

three partitions, all having different architectures available. The used partition had a single computing node and 8x8 AMD cores, as well as 131826392 kB RAM. [11].

3 Preliminary work: Image registration

The sections posing the available image data are not aligned, so they need to be registered, meaning that they should be correctly placed in a common coordinate system through image dependent image transformations.

Most registration algorithms make use of optimization, as optimization methods are often quite precise and not as sensitive to noise in the image data, compared to other methods. This advantage comes also comes with a disadvantage, since optimization methods might convergence towards local extrema instead of a global extremum. This requires the image data to be nearly aligned, before we can use optimization techniques, in order to reach the most optimal solution. Therefore, coarse registration using other methods might be needed beforehand.

Another option for registering images would be automatic computation of landmark correspondences, but this approach is sensitive to noise in the image data and requires clearly distinct landmarks to be available, which is not the case for most microscope images.

Other possibilities exist, but the two methods already mentioned are the most prominent methods within the field. It is clear that the landmark correspondence approach is not likely to yield good results, so in order to choose a proper registration scheme for the image data, we will examine the image data and properties of optimization methods.

3.1 Type of image transformation

All tissue sections have been subjected to enlarging using the same magnification factor, but they may not be placed in the same angle or position. As the shape of the neurons are needed for shape analysis later on, the shapes of the neurons must not be altered by the registration process. Together, this implies that the image transformations need to be rigid, as rotation and translation might be needed for changing the angle and position, but must not be affine, as affine image transformations include shearing, reflection and scaling, which could alter the shape features. The proper type of image transformations for the image data are thus rigid image transformations, which covers only translation and rotation.

3.2 Optimization algorithms and metrics

The question of which optimization-based image registration scheme to use, is often decided based on the nature of the images, i.e. if the images are monomodal or multimodal. For monomodal image data, the images are acquired from the same image modality and thus the intensities in one image matches the intensities in the other image. This is not the case for multimodal images, where the same object might be represented with varying image intensities.

The image data available for this thesis cannot be regarded as monomodal image data, despite that only a single image modality was used to acquire the images. This is due to the staining and conservation processes that the tissue sections went through. Staining and conservation processes are difficult to control to such degree that there will be no colour difference from section to section. Thus an image registration scheme for multimodal images should be used for the available image data.

An optimization-based image registration scheme for multimodal images would be to maximize the mutual information of the images. Mutual information (MI) is a metric based on probability theory that measures how well a random variable describes another random

variable. The mutual information metric is given by

$$\mathrm{MI}(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\cdot\log\left(\frac{p(x,y)}{p(x)\cdot p(y)}\right) \tag{1}$$

for any two discrete random variables (a continous version does also exist). In the equation above $p(x,y)$ is the joint probability mass function, while $p(x)$ and $p(y)$ are the marginal probability mass functions for X and Y. Mutual information can also be expressed using the joint and marginal entropies of the random variables.

The mutual information metric is available in Matlab. In theory the mutual information metric can be combined with any maximization algorithm, but in Matlab the available maximization algorithm for image registration is an evolutionary algorithm called "One-plus-one evolutionary". According to the Matlab documentation, an evolutionary algorithm is an algorithm that tries to find a set of parameters for an image transformation that produces the best possible registration by iteratively pertubing or mutating the parameter set, saving the better set of parameters, whenever a better one is found.

The above described algorithm and metric will be used for the fine registration.

3.3 Chosen image registration scheme

By inspection of the sections, it could be seen that the sections are far from the sought alignment. Often the image positions differ by several hundreds of pixels and the angles differ substantially, too. The sections will need to be coarsely registered before attempting to use any optimization-based registration algorithm, as it will not be likely to converge to the sought extremum for all the sections, if any of them. The chosen registration scheme will thus consist of a semi-automatic coarse registration before doing a fine registration using an optimzation-based registration scheme. The registration scheme are further detailed in the following subsections.

3.3.1 Coarse registration

The coarse registration process will be based on estimation of the rigid transformation from landmarks. A rigid transformation from point $p_1 = (x_1, y_1)$ into point $p_2 = (x_2, y_2)$ are given by the equation

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = R\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2}$$

where R is a rotation matrix for counterclockwise rotation of the angle θ about the origin and t is a translation vector for translation along the x- and y-axis.
From the system of equations we can compute an estimation of the rigid transformation by fitting a function of θ, t_x and t_y to sets of landmarks or other point correspondences. Such fitting is often done by optimizing various residual functions.

The equation system has three degrees of freedom and therefore we need at least two pairs of points to estimate the transformation. Microscope images often lack clearly distinct landmarks and therefore automatic landmark detection is likely to yield poor results. It has therefore been decided that the user of the program must find and mark the same two landmarks through all sections using the computer mouse.

An example of the coarse registration of an image pair can be seen in figure 5. The figure displays the images on top of each other by placing them in different colour bands (green and purple respectively). Areas of the registered images that are equal in intensities becomes light grey, while areas of differing intensities becomes either green or purple.

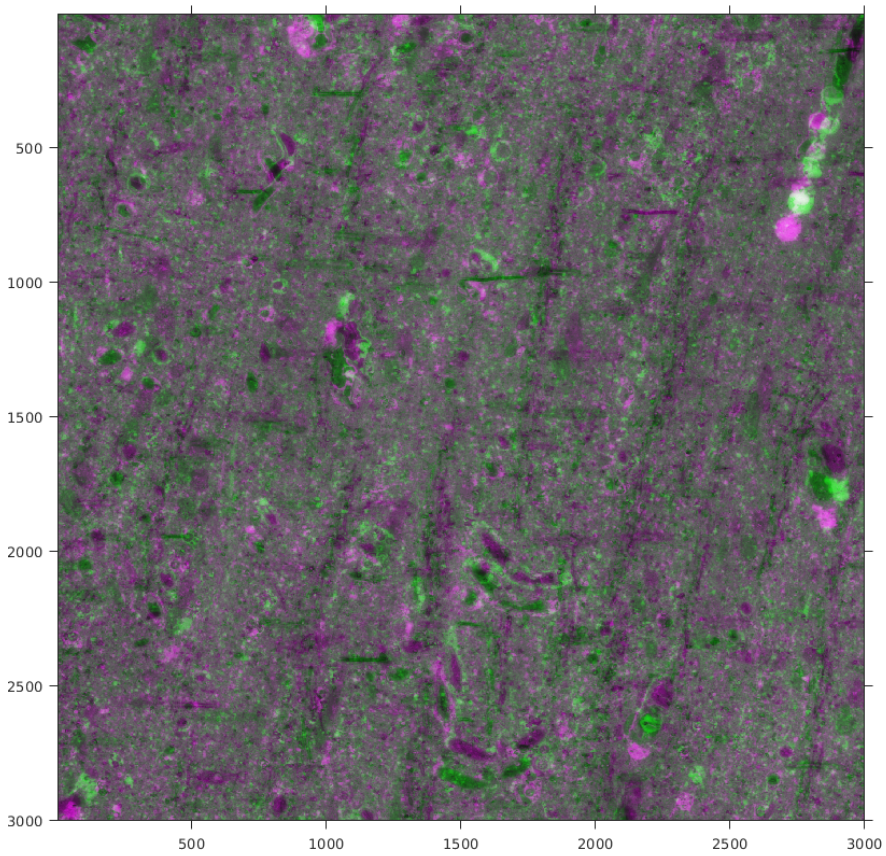


Figure 5: *Example of coarse registration of an image pair. One image is shown in green colour band and the other in purple colour band. Grey areas indicate that images are very similar and coloured areas that the images differ from each other.*

3.3.2 Fine registration

The coarse registration of the sections are not sufficiently precise and further needs to be finely registered. During the fine registration, the sections will be registered to their consecutive images.

The fine registration will be done using the built-in image registration function `imregister` from Matlab. The image registration function must be provided with a metric and an optimizer, as it can be used for registering both monomodal and multimodal image data.

The choice of metric and optimizer algorithm, are as described in section 3.2. The optimizer available in Matlab comes with default settings, but four parameters can be set to improve the registration results. These parameters are the maximal number of iterations, the initial search radius in parameter space, the minimum size of the search radius in parameter space (epsilon) and the growth factor that defines how quickly the search radius increases.

To ensure that the optimizer converges correctly without overstepping or diverging, the parameter setting were set to

- Maximal number of iterations: 500,
- Initial search radius: 0.001,
- Epsilon: 0.00001,
- Growth factor: 1.01.

Based on inspection of the 50 test images, these parameter settings seems to work well. An example of the fine registration of an image pair can be seen in figure 6. The registration

are displayed in the same manner as in figure 5.

The registration shown in figure 6 is much more precise than the registration shown in figure 5. Almost no parts of the layered images are blurred, which indicates a precise registration.

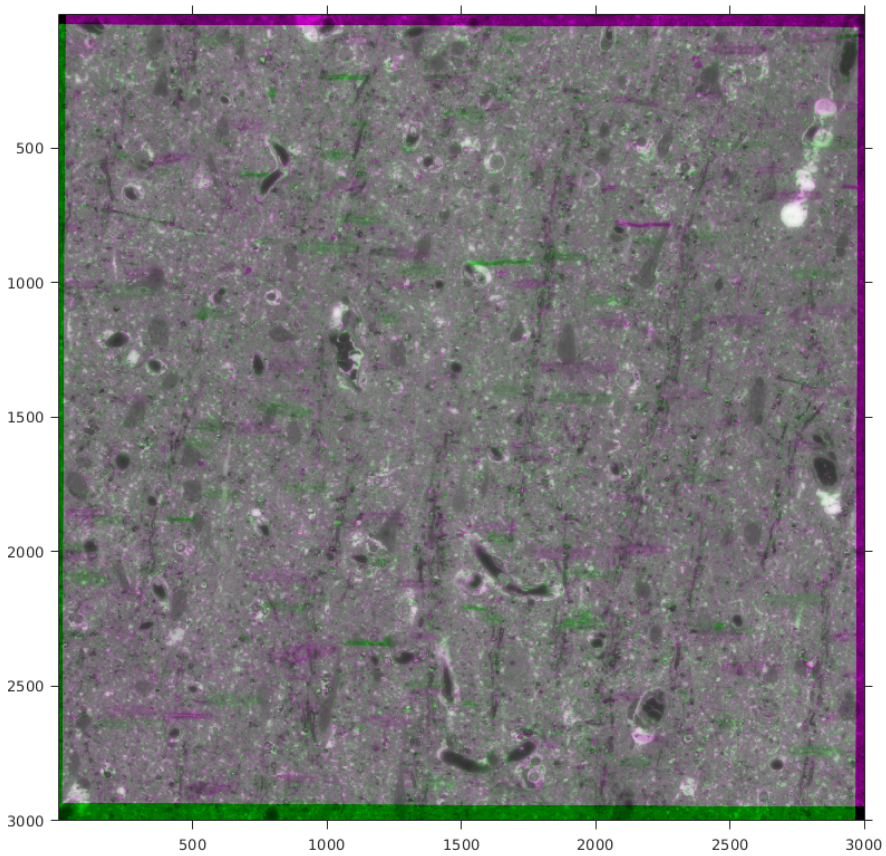


Figure 6: *Finely registered image pair. One image is shown in green colour band and the other in purple colour band. Grey areas indicate that images are very similar and coloured areas that the images differ from each other. The coloured lines are artefacts from the image aquisition.*

4 Segmentation of neurons

Out of all the pixels in a section, only the rather small amount of pixels that potentially image neurons are of interest to us, thus the sections need to be segmented. Segmentation refers to the division of an image into perceptually meaningful regions which, in the case of the available image data, will be regions of neurons and regions of non-neurons.

In order to do the segmentation, we need to define some criterion on a pixel level to distinguish between the two regions. As with the choice of image transformation, these criterions will be based on data observations.

There exists several methods for segmentation, including very sophisticated methods that for example can be based on neural networks or other computationally heavy constructions. However, due to the vast amount of image data that the program will need to segment, a simple and fast segmentation method is needed. The focus will therefore be on finding a segmentation scheme that works in addition to being fast to perform.

4.1 Data observations

From the examples of stained neurons in figure 3, it can be observed that the soma of the neurons appear to be more consistent in their colour, as well as having a darker colour than the surrounding tissues. In addition, the neurons in figure 3 do not exhibhit any clear contours. These observations imply that

- the average pixel intensity of local pixel neighborhoods inside the neurons will be less than the average pixel intensity of local neighborhoods outside the neurons.
- the pixel intensities of local pixel neighborhoods inside the neurons will fluctuate less than the pixel intensities of local neighborhoods outside the neurons.
- segmentation based on edge-detection will likely not provide any good segmentation result, due to the lack of clear neuron contours.

The implications points towards ordinary segmentation methods like threshold segmentation or clustering based on pixel features.

Out of threshold segmentation and clustering based on pixel features, threshold segmentation is the fastest and most simple. It however comes with the disadvantage that thresholds must be decided and this must often be done manually, which can be hard to do properly.

Clustering based on pixel features do not have the same problem, as the distinction between clusters are decided by the algorithms, but it is often computationally heavy and requires that clusters are clearly separable. In addition to this, the number of clusters cannot easily be decided either.

4.2 Chosen segmentation scheme

Due to the need for a fast and simple segmentation method, it has been decided to use threshold segmentation.

To reflect the data observations described in the previous section, the threshold segmentation will consist of application of multipe thresholds, applied to the local average pixel intensity and local variance of pixel intensities.

The sections will be preprocessed before the segmentation to improve stability of neuron detection and the segmentation will be postprocessed in order to refine the segmentation too.

The needed preprocessing will consist of a slight smoothing of the sections and an intensity adjustment to let all sections have the same average intensity. The smoothing will be done in order to eliminate digitization artefacts and the intensity adjustment ensures that the same thresholds apply to all sections.

The segmentation will be postprocessed in order to eliminate segments that do not represent neurons as well. This will be necessary as threshold segmentation is a simple method that does not account for the shape or size of a segment.

4.2.1 Preprocessing

To eliminate digitization artefacts originating from the image aquisition, the sections will be sligthly smoothed before segmentation will be performed. The smoothing has been done by convolving the image section with a gaussian kernel. Any smoothed image section are thus mathematically defined as

$$I_{smoothed} \;=\; G(\sigma) \;**\; I \tag{3}$$

where $**$ denotes the operation of convolution, I denotes the image section and $G(\sigma)$ denotes the gaussian kernel of standard deviation σ.

As only digitization artefacts should be eliminated, a small kernel of standard deviation

$\sigma = 1$ was used for the smoothing. To retain the image size, the image sections have been padded with zeros prior to the convolution.

If a single set of thresholds are to be applied to the sections, the sections must have comparable levels of pixel intensities. This is not the case for the available image data, since it cannot be regarded as monomodal image data. The intensity levels of the sections must thus be adjusted to match a certain average pixel intensity.

To match a certain intensity level without altering information in a sections, the intensity level must be changed in such a way that the difference between pixel intensities within the section is not affected.

An easy way to acheive such alteration is to add the constant

$$c = T - E(I) \tag{4}$$

to the section I. T is the desired average intensity and $E(I)$ is the actual average intensity of the section.

As the issue of varying intensity levels was discovered after deciding on the threshold values, the intensity level adjustment will be carried out to match an average pixel intensity of 104, as the section that the threshold values were based upon had this average intensity.

4.2.2 Computation of pixel features

To do threshold segmentation of a section, the pixel features of local average intensity and local variance of pixel intensities must be computed for the section.

The computation of the local average intensity can be done efficiently by convolving the section with a mean kernel. Thus, the local average intensities of a section are mathematically given by

$$I_{local\_average} \;=\; M(size) \;**\; I \tag{5}$$

where $**$ denotes the operation of convolution, I is the section and $M(size)$ is a mean kernel of size $size$.

Computation of the local variance of pixel intensities can also be done efficiently by use of convolution. To see this we look at the definition of variance

$$\begin{aligned} \mathrm{Var}(X) &= E((X - E(X)^2)) \\ &= E(X^2 - 2XE(X) + E(X)^2) \\ &= E(X^2) - 2E(X)\,E(X) + E(X)^2 \\ &= E(X^2) - E(X)^2. \end{aligned} \tag{6}$$

By the definition we can compute the variance as the average of squared pixel intensities minus the squared average of pixel intensities. Thus, we can compute the local variance of pixel intensities as

$$I_{local\_variance} \;=\; M(size) \;**\; I^2 \;-\; (I_{local\_average})^2 \tag{7}$$

where $**$ still denotes the operation of convolution, I^2 is the squared pixel intensities of section I and $M(size)$ is stil the mean kernel of size $size$.

The computation of the pixel features involves choosing a value for the kernel size parameter $size$. The value of this parameter was set to 11x11 pixels, since

- the smallest neurons we want to segment is no larger than $5 - 8\mu m$, corresponding to 10-16 pixels.
- we need to retain as much of the neurons as possible to get proper shape analysis results.

- The speed of computation decreases when the kernel size increase, especially if the kernel size is defined by even numbers.
- Fast computation of the pixel features are needed.

The neighborhood size of 11x11 pixels seemed to work better than smaller neighborhood sizes and the above mentioned reasons do not justify a bigger neighborhood sizes.

4.2.3 Choice of threshold values

The choice of the threshold values was based on examinations of neurons from a single section along with inspections of the histograms of pixel features for the same section.

The pixel features were computed for small parts of the section that lay inside neurons. This yielded local average intensities of approximately 85 and local pixel intensity

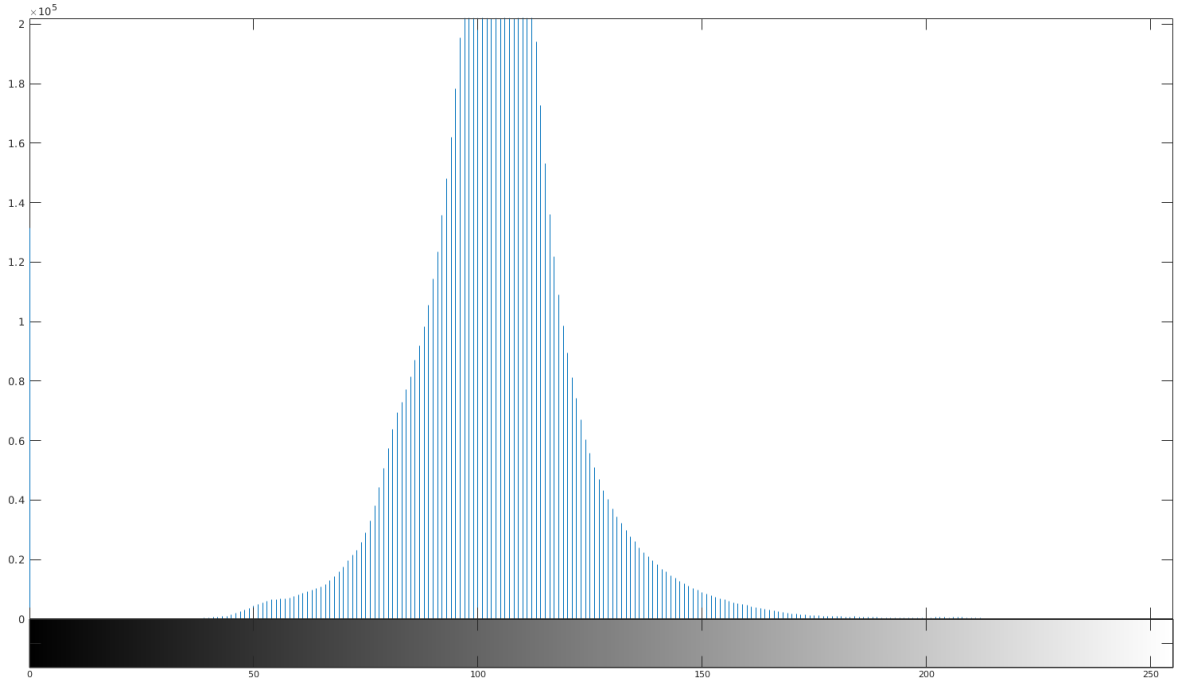


Figure 7: *Histogram of pixel intensities in a single section. The histogram seems to consist of two distributions that have been mixed together, a little one with a peak around the intensity 85 and a big one with a peak around 100-110. The single, thin peak at intensity 0 comes from the registration.*

variances of less than 7. These results seems to agree well with the image histogram in figure 7, which shows signs of two distributions, one we will assume to reflect the neurons and one we will assume to reflect the surrounding tissues. The same signs can be found in the histogram of the local average intensity displayed in figure 8, although they are less visible.

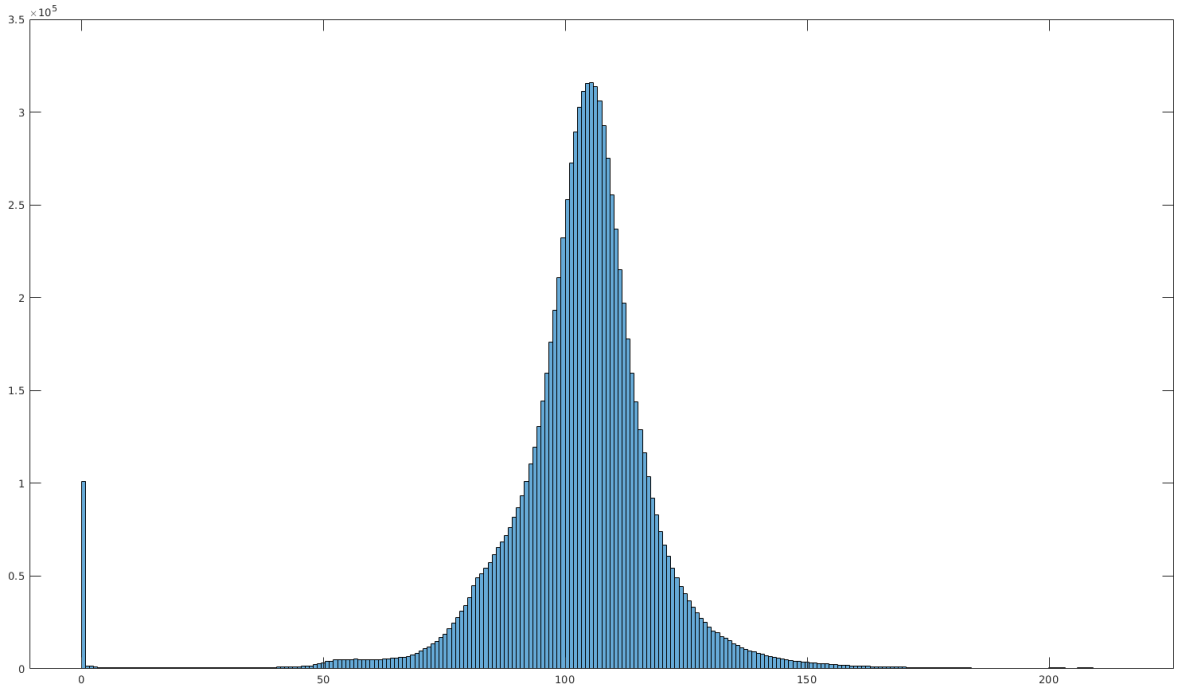


Figure 8: *Histogram of local average intensities. The histogram is very similar to the intensity histogram in figure 7. Again, the single, thin peak around intensity 0, comes from the registration.*

When inspecting the histogram of the local variance of pixel intensities in figure 9 however, there seem to be no clear distinction between neurons and the surrounding tissues. Thus it seems that only the local average intensity will be truly useful to distinguish between pixels imaging neurons and pixels imaging the surrounding tissues.

Even as the local variance of pixel intensities seems to not be useful for distinguishing between the two kinds of neurons when examining the histogram in figure 9, it seemed to provide a small improvement to the segmentation during experiments. The small improvements was with respect to the edges of the neurons.

The local variance of pixel intensities inside neurons suggest a pixel should be allowed to deviate from the suggested local average intensity of 85 by approximately 7. Thus, the final thresholds are

- A lower threshold on the local average intensity of 78.
- An upper threshold on the local average intensity of 92.
- An upper threshold on the local variance of pixel intensities of 7.

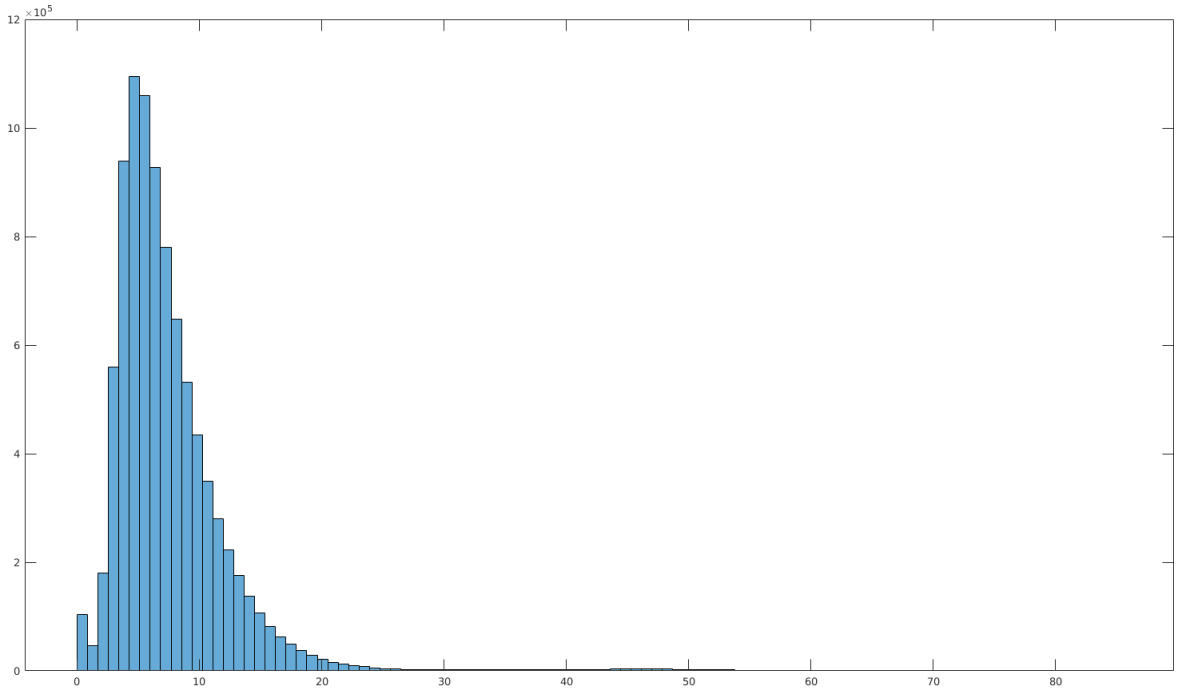


Figure 9: *Histogram of local variances of pixel intensities. There is no signs of clear distinction between pixels imaging neurons and pixels imaging surrounding tissues.*

4.2.4 Postprocessing

The segmentation needs to be postprocessed after doing the threshold segmentation for a number of reasons

- Cell cores of the neurons have a significantly smaller local average intensity than the cell border and are thus wrongly classified as surrounding tissues.
- Threshold segmentation can miss edges of the neurons when the cell core is visible, as it is based on local pixel neighborhoods. This results in gaps in the cell borders.
- The segments produced by threshold segmentation might have ragged edges due to noise in the sections.
- The segmentation may contain faulty segments that are not neurons, which can clutter the analysis results later on.

Correction of these problems will partly have to be performed on the individual sections and partly on the 3D image that the sections constitutes.

The first step of the postprocessing will consist of closing of the gaps in the cell borders. This can be achieved by morphological closing. For the closing of the gaps in the cell borders, morphological closing using a disc-formed structuring element of radius 11, was applied.

When the cell borders are closed, visible cell cores can be detected as background that cannot be connected to background pixels on the edges of the section. This constitute the second step of the postprocessing.

Ragged edges of the segments will have to be corrected in 3-dimensional space, as it will not be clear from the individual sections, when an edge is ragged or not. Smoothing of ragged edges can be done using morphological opening in 3-dimensional space. This constitute the third step of the postprocessing. As morphological opening alters the shape of the segments, a small spherical structuring element of radius 5 was used. This also takes car of small noise elements in the segmentation.

The last part of the postprocessing will be removal of faulty segments, which requires a specification of what faulty segments are. From theory we know that the segments should be either spherical or pyramidal. Thus, we can sift the segments by inspecting their surface to volume ratio.

For a sphere the surface-to-volume ratio r_{sphere} is a function of the radius r given by

$$r_{sphere} = \frac{4\pi r^2}{\frac{4}{3}\pi r^3} = \frac{3}{r}. \tag{8}$$

It is difficult to sift a segmentation according to a function. We thus use the ratio

$$r_{sphere'} = \frac{(4\pi r^2)^{\frac{3}{2}}}{\frac{4}{3}\pi r^3} = \frac{8\pi^{\frac{3}{2}} r^3}{\frac{4}{3}\pi r^3} = 6\pi^{\frac{1}{2}} \simeq 10.635 \tag{9}$$

instead. Spherical neurons might not be perfectly spherical and so, we allow $r_{sphere'}$ to deviate with ~ 1.5. Thus, segments for which we have that $r_{sphere'} \in]9, 12[$ will be allowed in the segmentation.

The literature about pyramidal neurons mentions a diameter, which indicates that a pyramidal neuron can be modelled as a square pyramid. The surface-to-volume ratio of a square pyramid is a function of the base length and the height of the pyramid given by

$$r_{pyramid} = \frac{a^2 + 2a\sqrt{\frac{a^2}{4} + h^2}}{a^2\frac{h}{3}} = \frac{3\left(a + \sqrt{a^2 + 4h^2}\right)}{ah} \tag{10}$$

where a is the base length and h is the height of the pyramid. $r_{pyramid}$ is more complicated then r_{sphere} and cannot be reduced further. We know that pyramidal neurons have an approximate diameter of $13.45 \pm 0.24\mu m$ and an approximate height of $20\mu m$. Using this knowledge, we get that $r_{pyramid} \in [0.79, 0.81]$.

The interval describing the surface-to-volume ratio of a pyramidal neuron is based on the simplifying assumption that the pyramidal neurons are ordinary pyramidal objects, but according to the theory, they should be round pyramids. Therefore we allow segments to deviate with ~ 0.4. Segments for which we have that $r_{pyramid} < 1.2$ will therefore be allowed in the segmentation. Segments that fulfill neither of the two criteria will be removed from the segmentation.

4.2.5 Segmentation examples

Figure 10 and figure 11 presented in this section, shows examples of segmentation results in both 2D and 3D. The first example consists of a single 3D segment, while the second example consist of a segmented section.

The 3D segment in figure 10 represents a pyramidal neuron that seems to have the expected size, when comparing its dimensions to the size information provided in section 2.1, except when looking a the z-axis, where the segment is only half of the expected size. When looking at the concerned sections, there are no signs that the neuron might have been divided into separate segments and so it will not be assumed that the size difference is a result of poor segmentation.

Figure 11 shows an example of a segmented section from the set of sections that was used for development and testing. It displays segments that are clearly part of pyramidal neurons, as well as parts that likely belong to astrocytal neurons. Some of the neurons in the section have been poorly segmented, due to other section content lying on top of the neurons, which the segmentation scheme does not handle. There is also signs that the holes from the cell cores have not been completely closed.

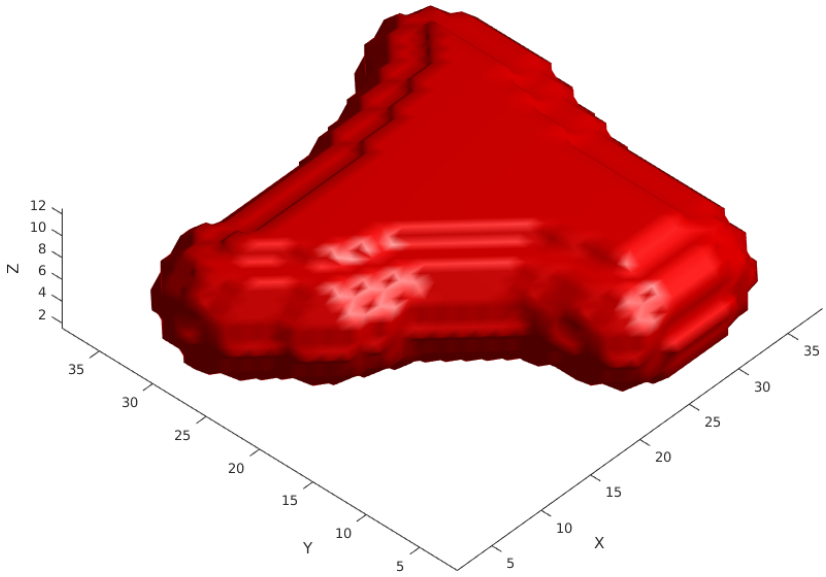


Figure 10: *Example of 3D segment representing a pyramidal neuron. The neuron seems to be of the expected dimensions, except when looking at the z-axis.*

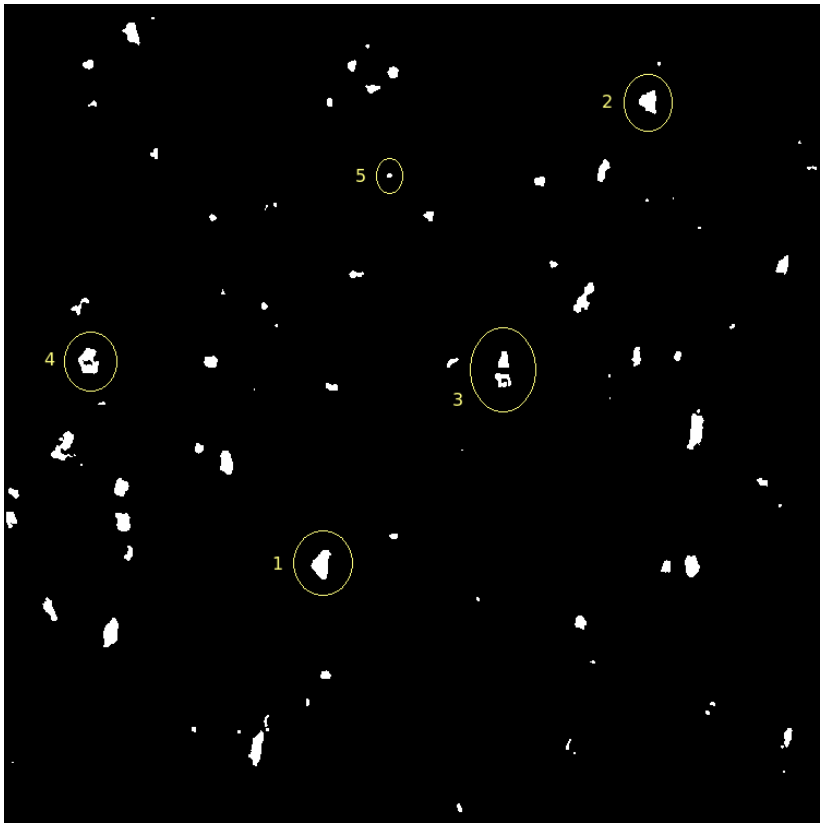


Figure 11: *Example of a segmented section. Neuron 1 and 2 are neurons that must clearly be part of pyramidal neurons. Neuron 3 is a pyramidal neuron that has been poorly segmented and thus been divided into two separate segments, while neuron 4 shows sign of its cell core not being properly closed. Neuron 5 is likely an astrocytal neuron.*

5 Annotation of neurons

The segments are not sorted by shape during the segmentation process. This means that the segments can be either astrocytal neurons or pyramidal neurons. The segments therefore need to be annotated as either of the two types of neuron, for any meaningful shape analysis of either the astrocytal neurons or the pyramidal neurons to be performed.

Exploitation of theoretical characteristics of the concerned types of neurons and computation and classification of distinct shape descriptors could possibly both provide for a basis for an annotation method. Both possibilities will be considered with respect to the astrocytal and pyramidal neurons.

5.1 Theory based annotation

In theory the two types of neurons differs in size. This is in addition to the astrocytal neurons being represented as spherical segments, while the pyramidal neurons being represented as segments of rounded pyramidal shapes. See section 2.1 for shape details of either type of neurons.

The theoretical characteristics provides us with the knowledge that

- The two types of neurons ought to divide into two clusters with respect to their size.
- Their principle axes differs in the sense that an astrocytal neuron have principle axes of roughly the same length, while a pyramidal neuron have a single principle axis that is significantly longer than its remaining two principle axes.

It should thus, in theory, be possible to distinguish the two types of neurons alone by looking at their sizes or the lengths of their principle axes.

In practice we cannot rely on these shape descriptions to distinguish the two types of neurons, as the shape descriptions are based on the observations and assumptions made by neuro-biologists as explained in the introduction.

The approach also requires us to choose some threshold for the length difference between the principle axes. Such threshold cannot be reliably set as they would need to depend on the unconfirmed shape descriptions from the neuro-biologists. Theory based annotation therefore does not seem like a good choice with respect to neurons.

5.2 Annotation based on classification methods

By computing distinctive shape descriptors it becomes possible to do annotation using unsupervised classification methods such as k-means clustering. This approach of annotation comes with the advantages of

- not basing the annotation on potentially unreliable assumptions,
- not having to choose thresholds as algorithms such as k-means uses automatic separation criterions,
- being able to account for noise segments, as we are not limited to assume all segments to be of neuron classes. The possibility of easily adding noise classes to the classification models exist.

We are however presented with two new problems: The problem of choosing a number of possible classes for such descriptor and the problem of deciding on a distinctive shape descriptor.

With respect to the former problem, we know that there ought to be two classes of neurons, but we do not know how potential noise segments distributes themselves through

the descriptor space. There is therefore no easy answer to the problem of how many classes should be modelled during annotation. The number of classes will need to be *objectively* chosen. An automatic determination of the number of classes will therefore be required. Optimization based on model selection principles like Occam's razor or formalization of it, such the minimum description length, can provide for such automatic determination.

With respect to the latter problem, computation of shape descriptors usually involves normalizing shapes for translation, rotation and scale. However, Kazdhan, Funkhouser and Rusinkiewicz state that

> *"...methods for rotation normalization are less robust and hamper the performance of many descriptors."*

in [5]. According to Kazdhan, Funkhouser and Rusinkiewicz, rotation normalization are therefore not well suited for computing descriptors. They propose to use a rotation invariant shape descriptor based on frequency analysis using spherical harmonics instead. Their results of using such shape descriptor seems promising.

Classification based annotation seemingly only poses some problems that can be overcome using proper shape descriptors and algorithms implementing certain principles for choosing the optimal number of classes. It will therefore be preferable to theory based annotation of the astrocytal and pyramidal neurons.

5.3 Chosen annotation method

As stated in the previous section, annotation based on classification methods must be preferable to theory based annotation of the neurons. According to Kazdhan, Funkhouser and Rusinkiewicz [5], it must also be preferable to use a shape descriptor that does not need rotation normalization of shapes to be performed before computation of the shape descriptors.

The rotation invariant shape descriptor that Kazdhan, Funkhouser and Rusinkiewicz propose to use, is based on frequency analysis however. As the distinctive characteristics of the shape of the two types of neurons have been observed in coordinate space rather than frequency space, it is questionable if the two neurons can be sufficiently separated using their rotation invariant shape descriptor.

This doubt contrasts Kazdhan's, Funkhouser's and Rusinkiewicz's promising results from their article and it has therefore been decided to conduct an annotation experiment using their shape descriptor. If their shape descriptor can succesfully separate the two types of neurons, annotation will be performed using classification of it. In the case that it cannot separate the two types of neurons succesfully, the annotation will be performed using manual annotation.

Classification of the shape descriptor will be performed using the k-means clustering method. The number of classes will objectively be decided upon using optimization based on the minimum description length principle. The optimization process and Kazdhan's, Funkhouser's and Rusinkiewicz's rotation invariant shape descriptor will be presented in later sections and a short introduction to frequency analysis will therefore be given in next section.

5.4 Frequency analysis

Frequency analysis is a tool used for a range of purposes, including audio analysis, molecule modelling and shape analysis. It is the study of how functions can be represented and approximated by sums of oscillatory components.

Oscillatory components are functions that oscillates with a certain spatial frequency. They can be defined for spaces of arbitrary dimension. Examples of oscillating functions

would be the sine and cosine functions.

The central activities in frequency analysis are the decomposition of functions into (possibly infinite) weighted sums of oscillatory components and the reconstruction of functions as linear combinations of oscillatory components.

If a set of oscillatory components are chosen such that they are orthogonal to each other, they can form a basis for other functions in the same dimension. In theory this enables complete reconstruction, just like in the case of principle components analysis. However, unlike principle components analysis, frequency analysis is not limited to linearly expressable problems, as oscillatory components are not linear of nature.

5.4.1 Spherical harmonics analysis

Spherical harmonics analysis is a frequency analysis method that can be regarded as the natural extension of the well-known Fourier analysis for 2D problems [16]. Where Fourier analysis makes use of trigonometric functions defined on the perimeter of the circle as the basis, spherical harmonics analysis makes use of spherical harmonics for the basis.

5.4.1.1 Spherical coordinates

The definitions for spherical harmonics are given using coordinates defined in the spherical coordinate system. For the thesis, the mathematical convention for this coordinate system will be used [9].

This means that a point in 3D space can be given by the three numbers, the radial distance to the origin, the azimuthal angle and the polar angle [9].

The radial distance is usually defined as the euclidian distance between the origin and the point, while the azimuthal angle θ lies in the range $[0, 2 \cdot \pi]$ and the polar angle ϕ lies in the range $[0, \pi]$. See figure 12 for visual details.

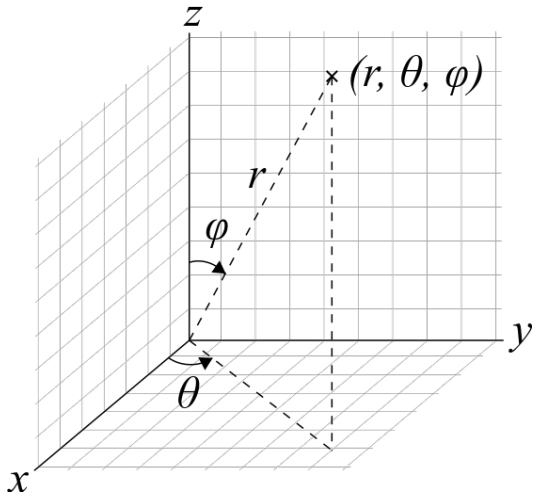


Figure 12: *The spherical coordinate system as defined by the mathematical convention.*

5.4.1.2 Spherical harmonics

Spherical harmonics are special functions defined on the surface of the sphere. They form an infinite set of continous and complex polynomial functions that are complete on the surface of the sphere [16]. Spherical harmonics are in general orthogonal, but the subset of real spherical harmonics are also orthonormal [16]. The real spherical harmonics thus form a natural basis for spherical functions in 3D, i.e. closed surfaces.
Several conventions for defining the spherical harmonics exist depending on the field they are used within. The general definition of real spherical harmonics (*on the unit sphere*) are the definition originating from the convention used in the acoustics field and also the definition that will be used during the thesis [16]. It is given as

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} \, P_l^m(cos(\phi)) \, e^{im\theta} \tag{11}$$

where the square root is a normalization factor, the frequency l and the order m are integers that fullfills the condition $|m| \leq l$ and P_l^m are the associated legendre polynomial [17] defined

as

$$P_l^m(x) = \frac{(-1)^m}{2^l\, l!}\,(1-x^2)^{\frac{m}{2}}\,\frac{d^{l+m}}{dx^{l+m}}(x^2-1)^l, \quad 0 \leq m \leq l. \tag{12}$$

The range that the associated legendre polynomials are defined for, can be extended to the range $-l \leq m \leq l$ by multiplying by a proportionality constant c_l^m [17]. Thus we have that

$$P_l^{-m}(x) = c_l^m\; P_l^m(x), \quad c_l^m = (-1)^m\,\frac{(l-m)!}{(l+m)!}, \tag{13}$$

which is convenient, as $P_l^{-m}(x)$ are often needed when doing frequency analysis.

Note that the definition in equation 11 changes depending on the normalization in use, as several different normalization terms for spherical harmonics are used in the scientific community. The normalization term in equation 11 corresponds to the general definition and the one used in acoustics.

5.4.1.3 Decomposition into frequency components

Any continous spherical function f can be represented as a linear combination of spherical harmonics

$$f(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} a_l^m\; Y_l^m(\theta,\phi) \tag{14}$$

where a_l^m is a unique constant given as

$$a_l^m = \int_0^{\pi}\int_0^{2\pi} Y_l^m(\theta,\phi)^*\; f(\theta,\phi)\; sin(\phi)\; d\theta\; d\phi, \tag{15}$$

where $*$ denote complex conjugation [16]. It should be noted that the a_l^m constant defined above are for functions defined on the unit sphere. When the defining sphere is not the unit sphere, the constant needs to be normalized according to the surface area of the sphere, in order to retain real spherical harmonics.

To do decomposition of a spherical function f in practice, discrete versions of the above equations are needed. By substituting the integrals for sums, we can get the following discrete approximations

$$f(\theta,\phi) \simeq \sum_{l=0}^{\infty}\sum_{m=-l}^{l} a_l^m\; Y_l^m(\theta,\phi) \tag{16}$$

and

$$a_l^m \simeq \sum_{0}^{\pi}\sum_{0}^{2\pi} Y_l^m(\theta,\phi)^*\; f(\theta,\phi)\cdot sin(\phi)\;\Delta\theta\;\Delta\phi \tag{17}$$

where $d\theta$ and $d\phi$ still equals the difference in coordinates, now as the forward distance between sample points of f. The intersection of the spherical function f and a chosen number of concentric spheres must then be computed. The discrete approximations can then be applied to the intersections to obtain an approximated decomposition of f. The approximation error will scale with the size difference of the concentric spheres.

5.4.1.4 Reconstruction of data observations

As indicated by equation 14 the observed spherical function f can be reconstructed by a linear combination of the spherical harmonics and the a_l^m constants obtained from the decomposition of f.

The a_l^m constants are analogs to the Fourier coefficients and are often called either spherical coefficients or Laplace coefficients [16]. Just as Fourier coefficients, the spherical coefficients can be sorted according to their frequency l, where the coefficients of lower frequencies

describes the more coarse structures of the observed function f and the coefficients of higher frequencies describes the detailed structures of the observed function f.

The observed function f can be reconstructed up to a desired precision by leaving out coefficients of higher frequency and in theory f can be completely reconstructed [16]. In practice it can only be reconstructed up to the precision allowed by the used decomposition, as data loss occurs during that phase.

5.4.2 Pros and cons of the method

As frequency analysis in general produces nonlinear bases, this type of analysis is not limited to problems of certain polynomial degree like methods such as principle components analysis are. This advantage comes at the cost of more complex computations together with data loss due to the sampling and discretization processes described in the previous two sections. Spherical harmonics analysis might also be more sensitive to numerical precision errors than methods such as principle components analysis, as more complex terms are used in the frequency analysis equations.

5.5 Rotation invariant shape descriptors from spherical harmonics

The rotation invariant descriptor that Kazdhan, Funkhouser and Rusinkiewicz defines in [5] is a grid descriptor based on spherical harmonics.

Kazdhan, Funkhouser and Rusinkiewicz claims that for a spherical function, the L2-norm of a frequency component will not change when the spherical function is rotated. They therefore conclude that the vector

$$SH(f(\theta,\phi)) = [\ ||f_{l_0}(\theta,\phi)||, ||f_{l_1}(\theta,\phi)||, \cdots, ||f_{l_n}(\theta,\phi)||\ ] \tag{18}$$

will be rotation invariant for the spherical function $f(\theta,\phi)$. f_{l_0} to f_{l_n} denotes the frequency components of frequency l given by the equation

$$f_l = \sum_{m=-l}^{l} a_l^m \cdot Y_l^m(\theta,\phi) \tag{19}$$

while $||\cdot||$ denotes the L2-norm. The frequency components can be obtained from the spherical harmonics decomposition of $f(\theta,\phi)$. The parameters θ and ϕ of the spherical function and the frequency components, are defined as in section 5.4.1.1

As shapes in general are non-spherical functions, Kazdhan, Funkhouser and Rusinkiewicz propose to create rotation invariant shape representations by computing $SH(f(\theta,\phi))$ for a set of spherical functions, obtained by intersecting the shapes with concentric spheres.

They thus derive the 2-dimensional rotation invariant shape descriptor

$$SHS(s) = \begin{bmatrix} SH(f_{r_0}(\theta,\phi)) \\ SH(f_{r_1}(\theta,\phi)) \\ \cdots \\ SH(f_{r_n}(\theta,\phi)) \end{bmatrix}. \tag{20}$$

where $f_r(\theta,\phi)$ denotes the spherical function obtained by intersection of the shape with the concentric sphere of radius r.

The rotation invariant shape descriptor $SHS(s)$ is of a quite big dimension. It has a grid size of $2 \cdot l + 1 x r_n$. It will therefore require *a lot* of computations to do k-means clustering using this descriptor.

As the sum of rotation invariant measures is also a rotation invariant measure, the descriptor $SHS(s)$ can be made smaller by summing over the radii. This lets us define the

new rotation invariant descriptor

$$SHS_{small}(s) = \sum_{r=r_0}^{r_n} SH(f_r(\theta, \phi)) \tag{21}$$

that is more suitable for clustering purposes. As $SHS_{small}(s)$ is more suitable for clustering purposes, this will be the descriptor to be used for the experiment.

The descriptor has been implemented using the built-in Matlab function `legendre` for computing the associated Legendre polynomials required for equation 19. As this Matlab function can only compute Legendre polynomials in the range $[0, l]$, equation 13 has been applied to extend the supported range to $[-l, l]$.

In order to implement the intersection of the shapes with concentric spheres, the radial distance of the sample points that constitute the shapes, had to be rounded to a certain precision. The rounding results in a precision error in the implemented descriptor. The extent of the precision error has been investigated by defining the precision error as a function of the rotation angle. The precision error are thus defined as

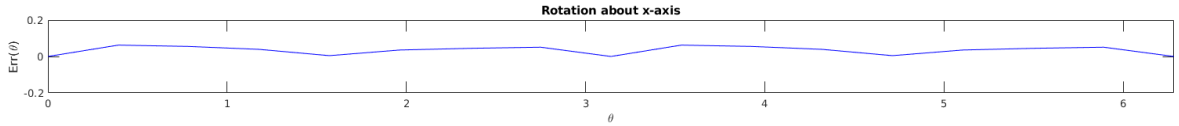
$$E(\theta) = ||SHS_{small}(s) - SHS_{small}(R(s, \theta))|| \tag{22}$$

where $R(s, \theta)$ denotes the rotation of the shape s by θ degrees around a single axis.

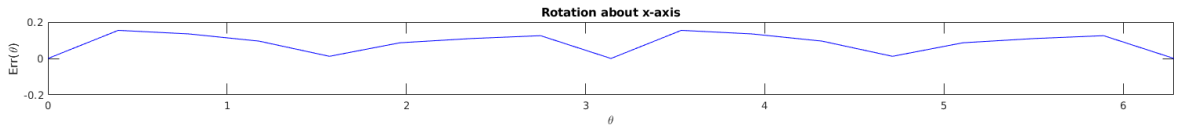
Figure 13, 14 and 15 shows plots of $E(\theta)$ for different numbers of radii and frequencies around all three axes. The plots have been made using a carefully selected pyramidal neuron.

The precision error seems to increase when the number of frequencies increases, as well as decrease when the number of radii increases. This makes good sense, as the increased number of radii makes the rounding factor smaller, while the increased number of frequencies includes more details of the shapes.

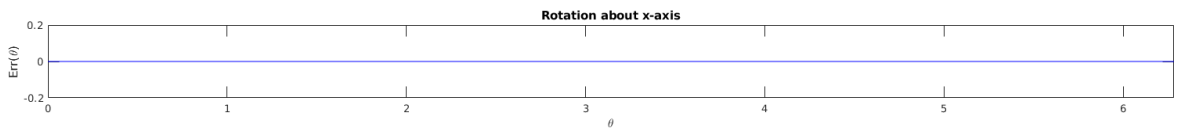
The implemented rotation invariant shape descriptor ensures that the distance between two descriptors for the exact same shape with different orientations, lies at most 0.2 from each other in the descriptor space. When considering that the descriptor is a vector of dimension $2l + 1$, an error below 0.2 seems quite small and it will thus likely not affect the result of any clustering method significantly.



(a) $E(\theta)$ *for 10 frequencies and 10 radii.*

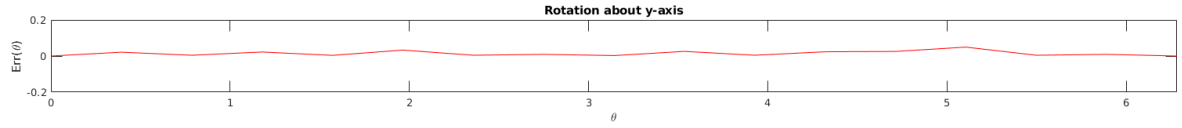


(b) $E(\theta)$ *for 50 frequencies and 10 radii.*

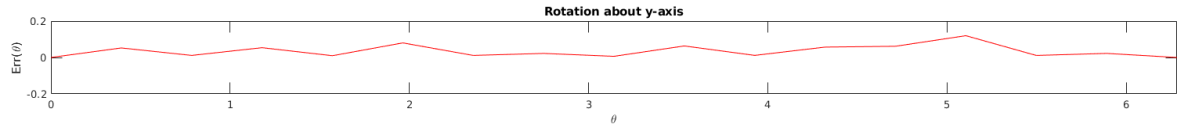


(c) $E(\theta)$ *for 10 frequencies and 50 radii.*

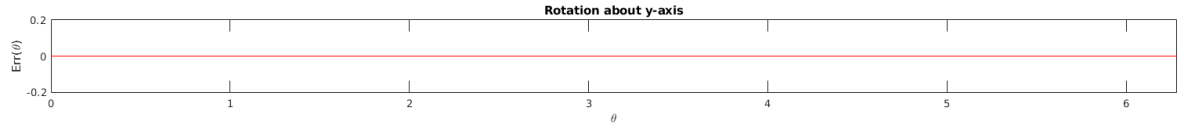
Figure 13: *Error in descriptor computation when rotating around the x-axis in steps of* $\frac{\pi}{8}$. *The peaks in the error plots are caused by a combination of the rounding of sample points and numerical precision errors from the rotation of the sample points.*



(a) $E(\theta)$ *for 10 frequencies and 10 radii.*

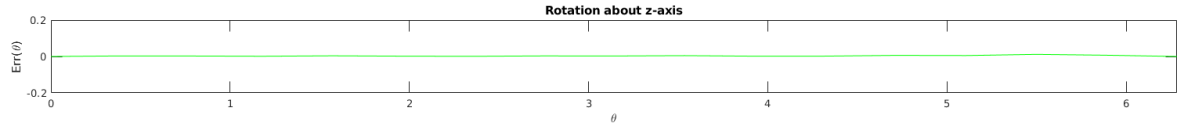


(b) $E(\theta)$ *for 50 frequencies and 10 radii.*

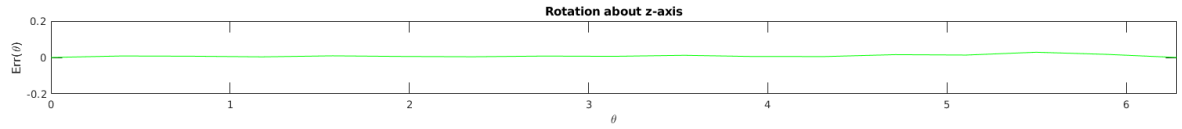


(c) $E(\theta)$ *for 10 frequencies and 50 radii.*

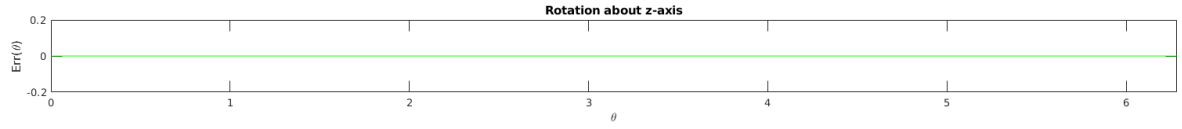
Figure 14: *Error in descriptor computation when rotating around the y-axis in steps of* $\frac{\pi}{8}$*. The peaks in the error plots are caused by a combination of the rounding of sample points and numerical precision errors from the rotation of the sample points.*



(a) $E(\theta)$ *for 10 frequencies and 10 radii.*



(b) $E(\theta)$ *for 50 frequencies and 10 radii.*



(c) $E(\theta)$ *for 10 frequencies and 50 radii.*

Figure 15: *Error in descriptor computation when rotating around the z-axis in steps of* $\frac{\pi}{8}$*. The peaks in the error plots are caused by a combination of the rounding of sample points and numerical precision errors from the rotation of the sample points.*

5.6 Choosing an optimal number of classes

Ordinary optimization cannot be used for optimizing the number-of-clusters parameter k in the k-means method, since the error measure in the k-means method used to assign a cluster to the data points are trivially optimized, when the number of clusters are equal to or greater than the number of data points. The minimum description length principle will therefore be applied to the problem in order to choose an optimal value for the number-of-clusters

parameter k of the k-means algorithm.

The minimum description length principle is about introducing regularization to an optimization process by penalizing any choice of model by the description length of the data given that model. The description length of the encoding of the data given the model will therefore be added as a penalty to the function that will be optimized to avoid trivial optimization. In general the function to be optimized will be the error function of the chosen type of model.
The minimum description length L_{min} is defined as

$$L_{min}(D) = \min_{M \in \mathcal{MS}} L(M) + L(D|M) \tag{23}$$

where $L(M)$ denotes the description length of the encoding of model M and $L(D|M)$ denotes the description length of the encoding of the data set D given model M, while $\mathcal{MS}$ denotes the set of possible models [**?**].

Grünwald defines the description length $L(M)$ of the model M by the equation

$$L(M) = \frac{K}{2}\log(N) + c_k \tag{24}$$

where K is a model dependent variable describing the number of characters needed to describe the model, N is the number of data points and c_k is a constant that depends on the number of classes k, but not on the number of data points N [**?**].

The description length $L(D|M)$ of the encoding of a data set D given a model M has Grünwald defined by the negative log-likelihood for the specific data from the given model [**?**]. We therefore have that

$$L(D|M) = -\log(P(D|M)). \tag{25}$$

Under the assumption that all data points are drawn from their distribution independently, we also have that

$$P(D|M) = \prod_{i=1}^{N} P(x_i|M) \tag{26}$$

where D denotes the data set, N denotes the number of data points and M is the chosen model.

5.6.1 Applying minimum description length to the k-means method

A k-means clustering model consists of k cluster centroids and an assignment of all data points to the nearest cluster. The error function of such a model would thus consist of the total euclidean distance between the data points and their assigned clusters. Thus the error function of a k-means clustering model are defined as

$$E_{k-means}(k) = \sum_{x \in D} ||x - k_x|| \tag{27}$$

where x is an individual data observation and k_x is its assigned cluster centroid. The desired regularized error function will therefore be

$$E_{k-means-reg}(k) = \sum_{x \in D} ||x - k_x|| \ + \ L_{min}(D). \tag{28}$$

For the k-means clustering model, we need a single character to describe the number of clusters, d characters per cluster to describe the coordinates of the clusters centroids and d characters per cluster to describe their variances in each dimension of the clusters. We therefore define K as

$$K = 1 + 2kd. \tag{29}$$

The constant c_k regulates how prone the optimization process will be to divide clusters into subclusters. As the litterature did not describe how to set the value of the constant, it has been set experimentally to

$$c_k = 2^k. \tag{30}$$

The definition in equation 25 requires that some assumptions on what type of distribution the clusters originates from, to be made. We will therefore assume that the clusters we seek to model using the k-means clustering method, originates from independent multivariate normal distributions. The probability of drawing a single point from a multivariate normal distribution are given by its probability density function

$$P(x) = \frac{1}{\sqrt{2\pi^d|\Sigma|}}\, e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \quad x \in \mathbb{R}^d \tag{31}$$

where Σ denotes the covariance matrix and $|\cdot|$ denotes the determinant. This equation can only be applied, when the covariance matrix is full rank, otherwise the inverse of the covariance matrix does not exist.

In the case where the covariance matrix is not full rank, the inverse of the covariance matrix can be substituted for its pseudo-inverse matrix, while the determinant of the covariance matrix can be substituted for its pseudo-determinant. The pseudo-determinant of a matrix equals the product of all its non-zero eigenvalues. When the covariance matrix is either not full rank or sufficiently ill-conditioned [1], the equation becomes

$$P(x) = \frac{1}{\sqrt{2\pi|\Sigma|^\dagger}}\, e^{-\frac{1}{2}(x-\mu)^T\Sigma^\dagger(x-\mu)}, \quad x \in \mathbb{R}^d \tag{32}$$

where $(\cdot)^\dagger$ denotes the pseudo-inverse of a matrix and $|\cdot|^\dagger$ denotes the pseudo-determinant of a matrix.

For the k-means model class $\mathcal{MS}_k$ we will therefore have that

$$\begin{aligned} L_{min}(D) &= \min_{M_k \in \mathcal{MS}_k} L(M_k) + L(D|M_k) \\ &= \frac{K}{2}\log(N) + c_k - \log\left(P(D|M)\right) \\ &= \frac{1+2kd}{2}\log(N) + 20\cdot 2^k - \log\left(\prod_{i=1}^{N} P(x_i|M_k)\right). \end{aligned} \tag{33}$$

where the probability $P(x|M_k)$ of drawing the data point x_i from its multivariate normal distribution using the k-means model M_k is given as

$$P(x|M_k) = \begin{cases} \frac{1}{\sqrt{2\pi^d|\Sigma_k|}}\, e^{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)} & \text{if } \operatorname{rank}(\Sigma_k) = d \\ \frac{1}{\sqrt{2\pi|\Sigma_k|^\dagger}}\, e^{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^\dagger(x-\mu_k)} & \text{otherwise} \end{cases}. \tag{34}$$

where $x \in \mathbb{R}^d$.

5.7 Annotation experiment: K-means classification of rotation invariant shape descriptors

The k-means clustering method have a high computational complexity of $O(n^{kd+1})$. This makes the k-means clustering method ill suited for classification of high dimensional shape

[1] A matrix is said to be ill-conditioned when the ratio between its largest and smallest eigenvalue is large. Computation of the inverse or determinant of such matrix may lead to numerical instability, due to the limited numerical precision of computers.

descriptors.

Fortunately the lowest frequencies in a frequency decomposition represents the coarsest parts of a signal. Therefore a few of the lowest frequencies from a frequency decomposition of the neurons should be sufficient to represent either spherical ground structure or pyramidal ground structure. The ground structure should in theory meet our need of classifying the neurons.

The hypothesis for the annotation experiment will be based on the above information.

5.7.1 Hypothesis

The hypothesis for the annotation experiment will be that k-means classification of the rotation invariant shape descriptor SHS_{small} presented in equation 21 can provide for annotation of astrocytal and pyramidal neurons. The two types of neurons ought to be well separable in frequency space, therefore a rotation invariant shape descriptor computed using at most three frequencies will be sufficient for the classification and annotation of the neurons.

5.7.2 Experimental setup

The annotation experiment will be conducted under an experimental set up where

- The number of neurons to be classified are 561.
- The neurons to be classified originates from the 50 test images.
- Classification of the neurons will be performed using k-means clustering of rotation invariant shape descriptors.
- Optimization based on the minimum description length principle will be used to decide upon the number of classes k.
- The rotation invariant shape descriptor to be used will be the SHS_{small} descriptor presented in equation 21.
- The rotation invariant shape descriptors will be computed using $1 \leq f \leq 3$ frequencies and therefore have dimension $d = 2 \cdot f + 1$.
- The rotation invariant shape descriptors will be computed using $r = 50$ radii.

The argument for performing classification using k-means clustering will be its simplicity and ease of comprehension.

Arguments that justify the parameter setting for f and r have already been given. Furthermore the error plots of figure 13, 14 and 15 indicates that the parameter setting is reasonable.

Arguments that justify the use of optimization based on the minimum description length principle for deciding on the parameter setting for k have been given as well.

5.7.3 Results

We will focus the evaluation of the results of the annotation experiment on the two aspects

- Will the application of the minimum description length principle to the optimzation of the number of classes k actually yield the optimal number of classes for the neurons.
- Can the rotation invariant descriptor actually provide for separation of astrocytal and pyramidal neurons in descriptor space.

5.7.3.1 Optimal number of classes for neurons

The results of optimization of the number of classes k yields almost the same results for rotation invariant shape descriptors of the dimension $d = [3, 5, 7]$.

When optimizing for the number of classes when $f = 1$ and $d = 2f+1 = 3$ for the rotation invariant shape descriptor, the optimal number of classes was revealed to be 9 classes. The total euclidean distance error for the k-means model becomes ~ 5 when using this number of classes for the rotation invariant shape descriptor.

When optimizing for the number of classes when $2 \leq f \leq 3$ and descriptor dimension of $d = 2f + 1 = 5$ or $d = 2f + 1 = 7$ for the rotation invariant shape descriptor, the optimal number of classes was revealed to be 10 and 11 classes. The total euclidean distance error for the k-means model becomes ~ 8 when using this number of classes for the rotation invariant shape descriptor.

Figure 16 to 18 visualize the optimization processes by plotting the total euclidean distance error $E_{k-means}(k)$, the minimum description length regularization $L_{min}(D)$ and the regularized error function for the k-means clustering method $E_{k-means-reg}(k)$ as functions of the number of classes k.

The plots of $E_{k-means}(k)$, $L_{min}(D)$ and $E_{k-means-reg}(k)$ show that rotation invariant shape descriptors of the dimensions 3, 5 and 7 overall exhibit the same general tendency. Increasing the number of classes k when we have $k < 10$ improves the classification model significantly, while increasing the number of classes k when $k > 11$ will only improve the classification model insignificantly according to the regularization scheme. The optimal number of classes k therefore seem to be stable around 10-11 classes.

Recalling that frequency f represent a certain oscillation on a sphere surface, it makes sense that the rotation invariant shape descriptor of dimension $d > 3$ yield a slightly higher value for the optimal number of classes. The descriptors of dimension $d > 3$ likely just catch more shape information and are therefore able to distinguish two similar classes.

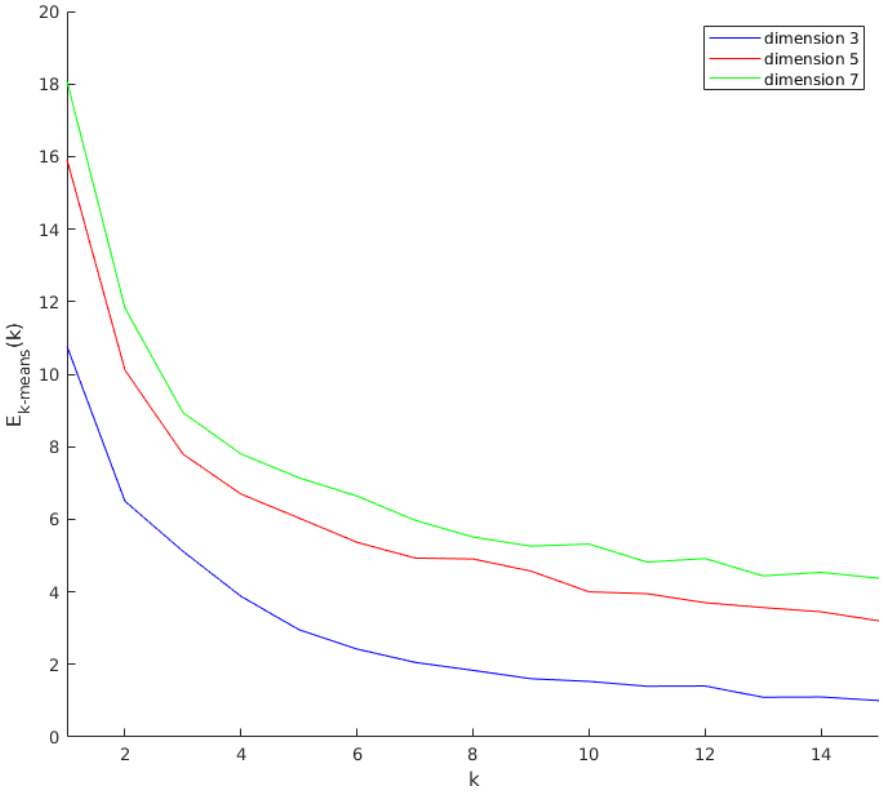


Figure 16: *Error function plot from k-means classification of neurons using rotation invariant shape descriptors. The total error seem to increase with the descriptor dimension d and decrease with the number of klusters k, as ecpected.*

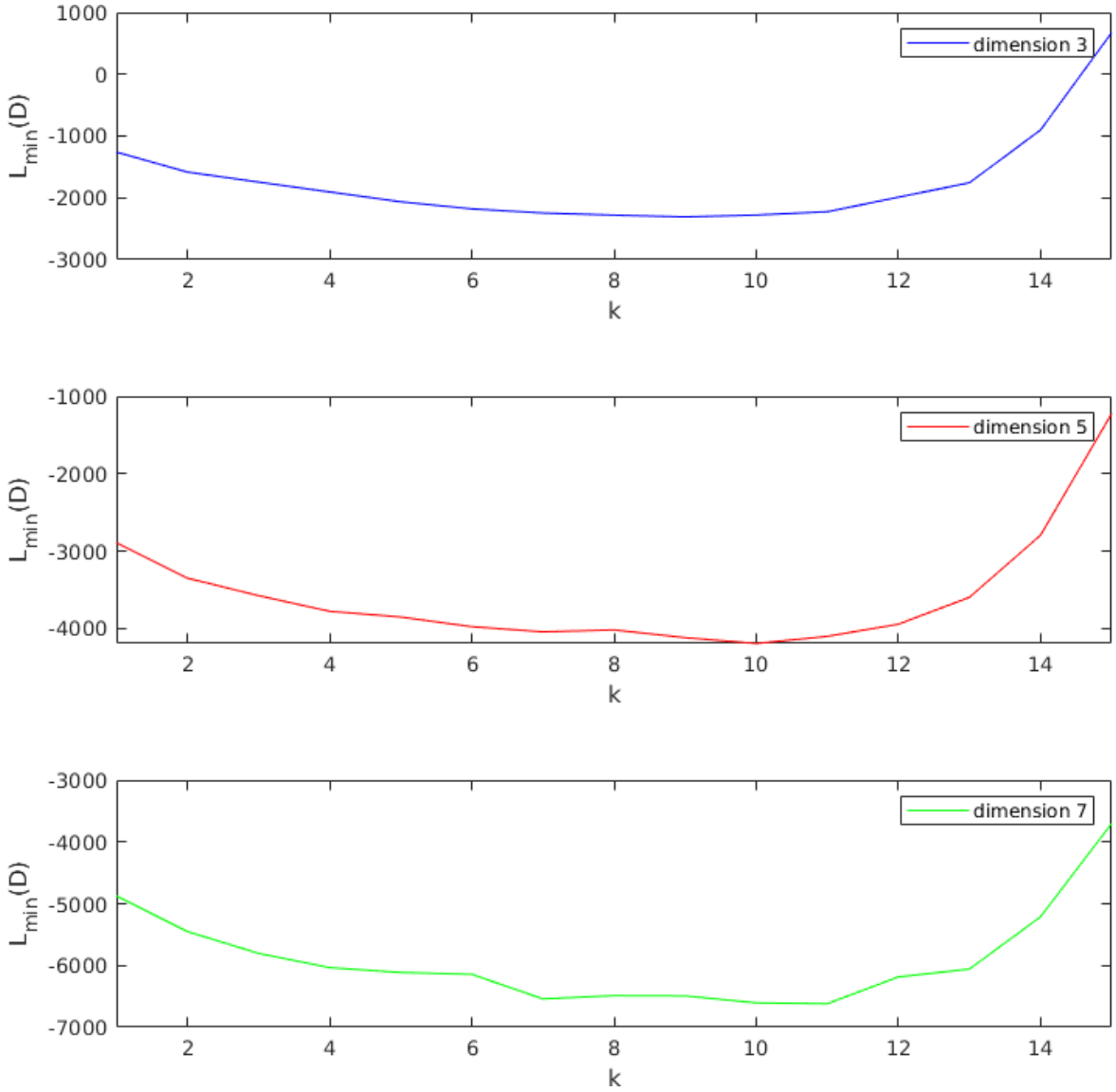


Figure 17: *Plot of minimum description length regularization of k-means model for neurons. The regularization functions seem to have their minimum around 9,10 and 11classes for rotation invariant shape descriptors of dimensions 3,5 and 7.*

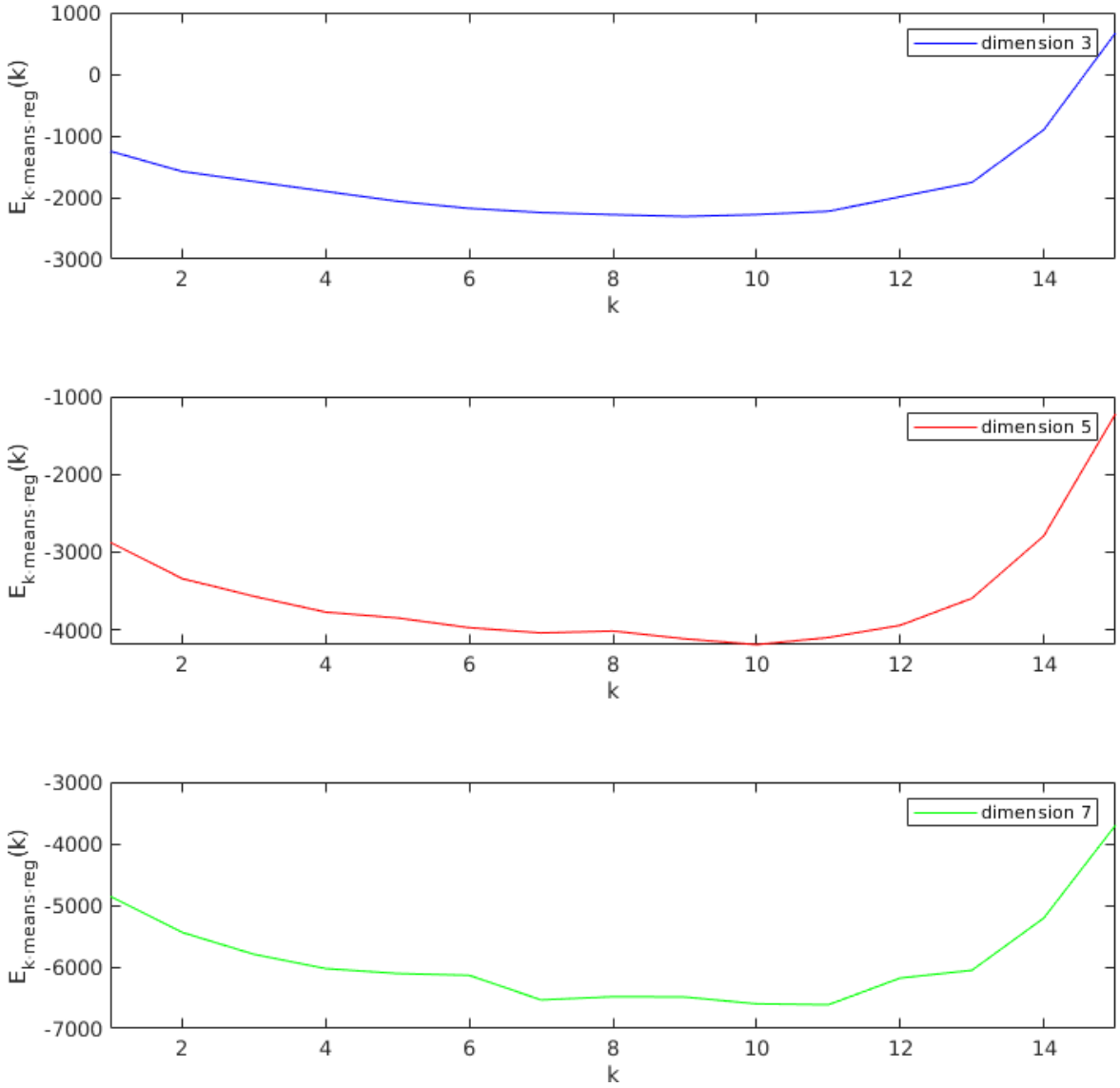


Figure 18: *Plot of regularized minimum description length regularization of k-means model when adapted to a k-means model for neurons. The regularization functions seem to have their minimum around 9,10 and 11 classes for rotation invariant shape descriptors of dimensions 3,5 and 7.*

5.7.3.2 Ability of the descriptor to separate neurons

When using the information from previous section to perform k-means classification of the rotation invariant shape descriptor of dimension $d = 3$, we obtain the 11 classes of neurons seen in in figure 19. The division into the 11 classes seems reasonable, as most classes seem to span an equal area of the descriptor space.
High shape variety for each class has been confirmed by manual inspection of the classes. All 11 classes seem to represent a mix of various neurons and the 11 classes can therefore not provide a basis for annotation of astrocytal and pyramidal neurons.

Manual annotation was to be performed if k-means classification proved unsuccesfull in separating the astrocytal and pyramidal neurons. The 3 classes resulting from manual annotation can be seen in figure 20.

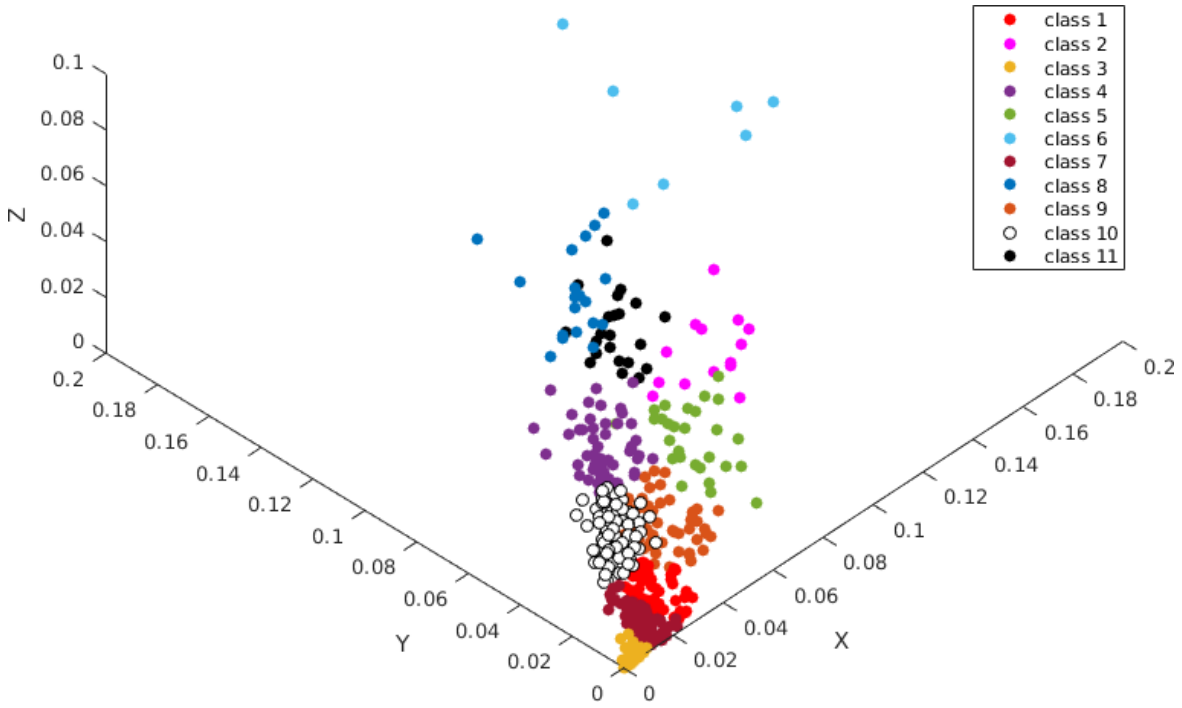


Figure 19: *Plot of the 11 classes of neurons that was suggested by optimization applying the minimum description length principle. The 11 classes seem to have approximately same size.*

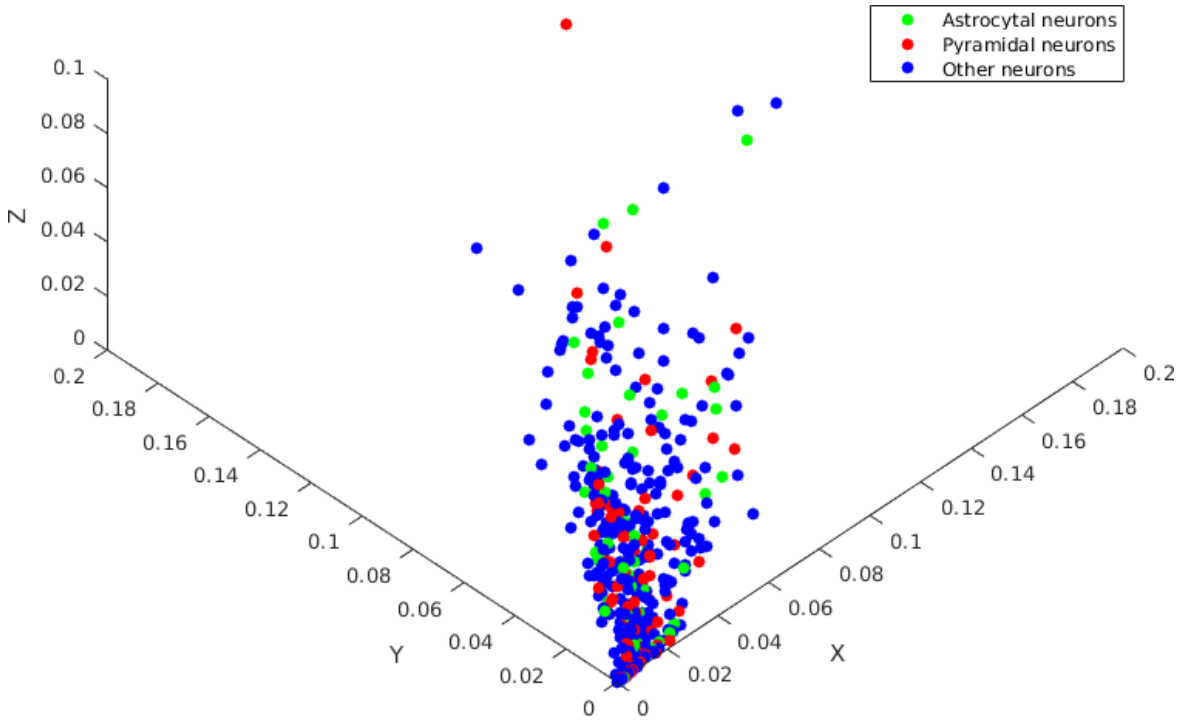


Figure 20: *Plot of manual annotation of the rotation invariant shape descriptors. The plot reveal that the neuron data are not separable in descriptor space, since the pyramidal neuron class, astrocyte class and other neurons class are all mixed up.*

It is clear that the rotation invariant shape descriptor SHS_{small} is not able to separate the neurons in a sufficient manner, when comparing the manual annotation to the k-means classification of the descriptor. Astrocytal and pyramidal neurons are all spread out through the 11 classes from the k-means classification. It is therefore very difficult to obtain a correct k-means classification for the SHS_{small} descriptor using a reasonable number of classes.

Possible explanations to why the rotation invariant shape descriptor SHS_{small} is not able to separate the neurons could be that

- The distinctive characteristics needed for discriminating between the three (or more) possible classes of neurons are not described by the first three frequency components.
- The summation over the radii in equation 21 results in a descriptor that is too simple to catch a sufficient amount of relevant shape information.
- The neurons are simply not separable using frequency analysis.

The first explanation seem unlikely. The first few frequency components represent the largest oscillations. They therefore ought to capture the largest distinctive characteristics of an object.

The second explanation seem much more likely in comparison to the first explanation. The rotation invariant shape descriptor SHS_{small} is a simplification of the rotation invariant shape descriptor that Kazdhan, Funkhouser and Rusinkiewicz propose to use. Oversimplifying a shape descriptor can result in an insufficient description of the relevant shape information.

If the original rotation invariant shape descriptor SHS cannot separate the neurons either, the third explanation seems likely considering the neuron classes from figure 20. To determine if this is actually the case, the experiment will have to be conducted again using the descriptor SHS.

5.7.4 Conclusion on annotation experiment

We will conclude that k-means classification of the rotation invariant shape descriptor SHS_{small} presented in equation 21 cannot provide for annotation of astrocytal and pyramidal neurons. The descriptor shows a poor ability to separate the neurons, when it is computed using $1 \leq f \leq 3$ frequency components and $r = 50$ radii.

6 Shape analysis of pyramidal neurons

Pincus and Theriot describe and compare different combinations of shape representations and methods for cell shape analysis in [3]. The aim of their study was to determine to what degree the different combinations provide quantitative measures of morphology that are biologically meaningful and human interpretable, in addition to being robust towards different types of cells. Here, morphology refers to the form and structure of an organism when any misalignment have been accounted for.

Their conclusion on the study was that combinations using principle components analysis in general provided better measure than combinations using other analysis methods, while also concluding that the shape representation that seems to be most appropriate when working with cells, would be either polygonal outlines or signed distance maps, due to their ability to capture subpixel shape differences.

As their results suggest that principle components analysis will be the most appropriate method for shape analysis, this method will be attempted and compared to the presumably better working method of spherical harmonics analysis.

6.1 Appropriate shape representation

Based on their results Pincus and Theriot state that they

> *"... do not recommend the further statistical analysis of shapes represented as binary masks because that representation over-emphasizes what humans perceive as small shape differences."*

Here, binary masks refer to binary images [5]. The over-emphasizing of small shape differences would be due to binary representations being inable to encode smooth changes of shapes. Shape analysis should thus be performed on shapes encoded using other representations than binary representations, if meaningful results of shape analysis are to be achieved.

As segmentation provides with segments encoded using binary representations, it seems to be necessary to compute more suitable shape representations from the segmentation in order to do proper shape analysis of the segmented neurons.

As Pincus' and Theriot's results suggest that the most appropriate shape representation for neurons would be either polygonal outlines or signed distance maps, it has been decided to follow that suggestion and use polygonal outlines for encoding the shape of the neurons[3].

Pincus and Theriot only describes a method for computing polygonal outlines for 2D shapes. Their computation method are thus not directly applicable to the pyramidal neurons that are 3D shapes. It will therefore be necessary to extend their method for computing polygonal outlines to 3D.

The extension of their method for computing polygonal outlines, which will from now on be referred to as Pincus' and Theriot's method, will involve substitution of outlines for surfaces, which will complicate sampling of a shape. The original method will be briefly described in section 6.1.1, while the extension will be presented in detail in section 6.1.2.

6.1.1 Pincus' and Theriot's method

Pincus' and theriot's method for computing polygonal outlines involves two steps. The first step is the extraction of the contour of the object from a binary image. The second step is uniform sampling of the extracted contour[3].

Pincus and Theriot use a classical contour-extraction method to obtain the contour of the cells. Such a method computes all points in an image that are midway between the inside and outside of an object. For binary images, the contour will consist of all points having an intensity value of 0.5. They do so in order to avoid encoding of digitization artefacts that makes otherwise smooth shapes ragged [3].

In order to avoid biased shape representation, Pincus and Theriot chooses to do uniform sampling of the extracted contour. They do this by sampling the contour with equal distance between the sample points [3].

Pincus and Theriot achieves the equal distance between the sample points by fitting a natural cubic spline to the contour, thus constructing a parametric plane curve, which they sample equi-distanced in terms of the arcparameter. They then iteratively adjust the sample points to obtain even spacing.

Pincus' and Theriot's method has been implemented as an initial step towards the extension of the method to 3D. The provided implementation for the method however, differs from the method described above. The differences consists of

- Replacement of the contour-extraction method by contour-extraction using the built-in Matlab function `contourc`.
- Doing uniform sampling of the contour using a simpler method. The simpler method consists of computing the total length of the contour L, then taking steps of size $\frac{L}{(N-1)}$ along the contour to obtain the sample points.

Interpolation are used for steps in between two contour points. Figure 21 shows an example of applying Pincus' and Theriot's method to a pyramidal neuron from a single section.

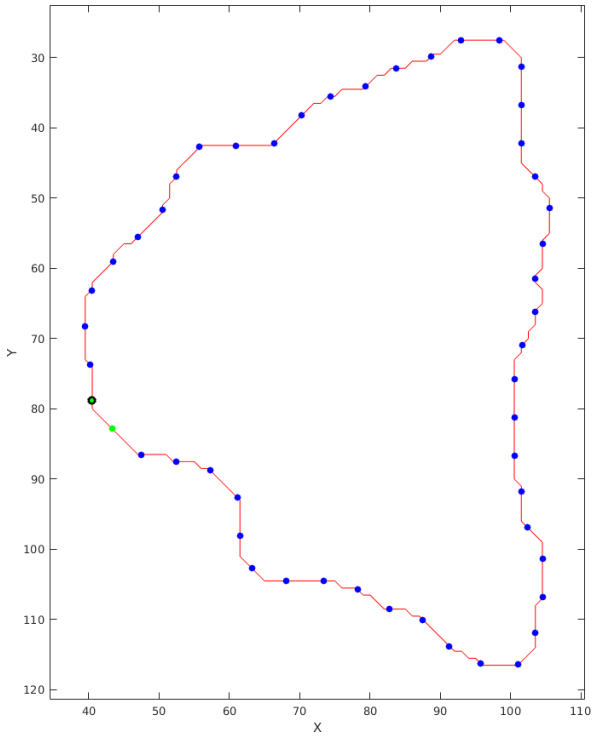


Figure 21: *Example of a 2D shape representation for a single pyramidal neuron. The sample points are placed on top of the outline that are shown in red. The first two sample points are green, while the remaining sample points are blue. The start point is marked by a black ring.*

6.1.2 Extension of Pincus' and Theriot's method

The outlines must be substituted for surfaces to extend Pincus' and Theriot's method. Uniform sampling of surfaces can be a computationally complex problem however. The substitution of the outlines for surfaces would require sophisticated sampling methods. Examples could be simulation methods or surface fitting followed by mathematical distribution of sample points.

Their method is most properly extended by uniform sampling of the surface, so that the sample points have an equal, empty area around them, but other sampling schemes can possibly provide for unbiased shape representations too. As long as any extension of their method satisfies the two conditions that

- a surface of a 3D object is computed as the midway between the inside and outside of the object,
- sampling of a surface does not introduce bias to the shape representation,

the extension will be equally valid, not matter how the extension handles the sampling of a surface.

The first condition corresponds to using the isosurface of an object. The midway between inside and outside for binary images still corresponds to all points having an intensity value of 0.5 and so that will be the value of the sought isosurface. As there exists methods for computing isosurfaces, this condition can easily be met.

The second condition of not introducing bias to the shape representation during sampling of the surface are somewhat more difficult to meet. Pincus and Theriot equate uniform

sampling resulting in equi-area sample points and the avoidance of introducing bias to a shape representation, but we will loosen this equation a bit.

Recall that Pincus and Theriot achieve their uniform sampling by doing equi-distanced sampling in terms of the arcparameter of a cubic spline that was fit to the outline. The sample points are then adjusted in an iterative manner.

If adopting this approach, we can substitute the arcparameter for directions in 3D space. We will therefore do uniform distribution with respect to the directions from the center of mass of the neurons to the sample points of the surfaces.

The azimuthal and polar angle of a spherical coordinate can together define a direction in 3D space. Uniform distribution of directions can therefore be achieved by doing uniform distribution of sample points on a sphere in spherical coordinates, where we will omit the radial distance of the spherical coordinates. Sample points that lie on the surface of the neurons can then be found by computing a radial distance for each sample direction. We will use the radial distance, at which a direction vector intersect the neuron surface.

The adjustment of the positions of the sample points makes uniform sampling of surfaces a computationally complex matter. It is however needed to ensure uniform distribution of the sample points with respect to surface area instead of sample point directions. Omission of the adjustment would therefore be at the expense of not achieving truly uniform distribution of the sample points, although the omission provides for a closed-form solution to the sampling problem. Closed-form solutions are considerably easier to implement and therefore desirable, so we will omit the adjustment of the sample points.

We therefore introduce a systematic bias to the shape representation in exchange for ease-of-implementation and ease-of-computation. One can however argue that the sample points are uniformly distributed in a weaker sense, than what Pincus and Theriot achieve, as the sample point are uniformly distrbuted with respect to direction.

6.1.2.1 Uniform distribution of points on the unit sphere

If the distribution of sample points on the unit sphere should be uniform, it will not suffice to do uniform distribution in the θ-ϕ plane, as this will cluster the sample points around the poles on the unit sphere. This happens as the area of a differential surface element on the unit sphere is given as

$$dA = \sin(\phi)\ d\theta\ d\phi, \tag{35}$$

The amount of surface area expressed by dA thus decreases when $\sin(\phi) \to 0$, which happens near the poles where $\phi = 0$ and $\phi = \pi$.

Sampling of the unit sphere must thus be done according to the probability density function $f(\theta, \phi)$ that maps to the uniform distribution on the unit sphere. The probability density function must be constant, since it should be uniform. As the surface area of the unit sphere is 4π, the probability of sampling the point (θ, ϕ) must therefore be given by

$$p(\theta, \phi) = \frac{1}{4\pi}. \tag{36}$$

We must then have that the probability of sampling a point (θ, ϕ) inside the differential surface element is given as

$$p((\theta, \phi) \in dA) = p(\theta, \phi)\ dA = \frac{\sin(\phi)}{4\pi}\ d\theta\ d\phi. \tag{37}$$

From the definition of a probability density function we have that

$$p((\theta, \phi) \in dA) = \int_{d\theta, d\phi} f(\theta, \phi)\ d\theta\ d\phi \tag{38}$$

and thus that the desired probability density function that maps to the uniform distribution on the unit sphere must be

$$f(\theta, \phi) = \frac{\sin(\phi)}{4\pi}. \tag{39}$$

The probability density function scales with the individual variables according to their marginal probability density function. These are given as

$$\begin{aligned} f(\theta) &= \int_0^{\pi} \frac{\sin(\phi)}{4\pi}\, d\phi = \frac{1}{2\pi} \\ f(\phi) &= \int_0^{2\pi} \frac{\sin(\phi)}{4\pi}\, d\theta = \frac{\sin(\phi)}{2} \end{aligned} \tag{40}$$

and shows that $f(\theta)$ is constant and $f(\phi)$ scales with the sine function. The joint probability density function thus does not scale uniformly with ϕ. To sample ϕ correctly, we sample ϕ using inverse transform sampling, where sampling is done according to the inverse cummulative distribution function $F^{-1}(\phi)$.

The cummulative distribution function $F(\phi)$ is given as

$$F(\phi) = \int_0^{\phi} f(\phi)\, d\phi = \frac{1}{2}(1 - cos(\phi)) \tag{41}$$

and it has the inverse

$$F^{-1}(u) = \cos^{-1}(1 - 2u), \quad u \in [0, 1]. \tag{42}$$

To sample ϕ uniformly on the sphere we thus sample u uniformly, then applies $F^{-1}(u)$ the u samples.

Remaining is only the uniform sampling of θ. θ can be sampled using golden ratios of whole turns around the sphere [**?**]. As the golden ratio is the most irregular number we know, this approach will not lead to clustering of sample points. The sequence of golden ratios will not converge towards a regular pattern.

Figure 22 shows an example of uniform distribution of 300 sample points on the unit sphere using the described method.

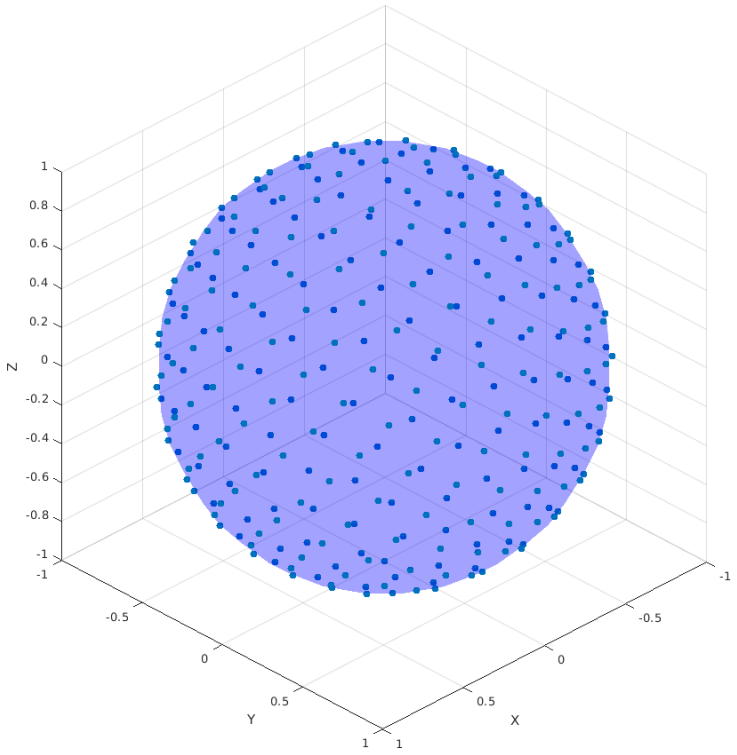


Figure 22: *Example of uniform distribution of 300 sample points on the unit sphere. All sample points have the same empty area around it. Note that the sphere is partially transparent.*

6.1.2.2 Alignment of neurons

In order to make the shape representation of a neuron comparable to shape representations of other neurons, we must ensure alignment of the neurons before computing any radial distance. We therefore ought to center and scale the neurons, as well as rotate them so that they point in the same direction before computing any radial distance.

We center a neuron on the origin using equation 45 to compute a radial distance r. We scale sample points by dividing each coordinate of the sample points by the variance of the coordinates of the neuron surface points.

Rotation of the neuron still remains. Our uniform distribution of sample points on the unit sphere are systematic and will therefore produce the same sample points in the same order at any time, given that we use a fixed number of sample points. We can therefore implement rotation alignment of a neuron by aligning the coordinate system of the sample points to the principle axes of the neuron.

The currently implemented rotation alignment is not optimal. Alignment of objects using their principal axes can sometimes yield unexpected results. Principle axes do not always match the human perception of what the principle axes of an object ought to be. Another problem that affect the reliability of the rotation alignment is that a set of principal axes provide for orientation information of an object only, not for direction information of an object. We therefore risc that objects having directions may point in opposite directions even though their principle axes match each other.

Rotation alignment based on principle axes are however easy to implement. Two coordinate systems A and B, both given by their orthonormal basis, are related to each other by the equation

$$RA = B \tag{43}$$

where R is the rotation matrix that relate the two coordinate systems to each other. Due to orthonomality we then have that

$$R = BA^{-1} = BA^{T}. \tag{44}$$

The principle axes forms such orthonormal basis and it is therefore straight forward to implement the rotation alignment using equation 44. To partly correct the orientation problem of principle axis alignment, we change the sign of the first principle axis to minimize the euclidean distance to the direction of a pyramidal neuron. Please see section 7 for information about directions of pyramidal neurons.

6.1.2.3 Computation of radial distances

To complete sampling of a neuron surface, we need to determine a radial distance for each sample point direction. The vector representing a sample point direction (θ, ϕ) will intersect the surface of a neuron at least once and possibly multiple times, if the surface of the neuron has high curvature. We therefore need to choose what to do, if a sample point direction intersect a neuron surface more than once.

Some shape information will be lost regardless of what we choose to do. Choosing the smallest obtainable radial distance will result in possible loss of information about the length of a neuron, while using the largest obtainable radial distance will result in possible loss of of information about the curvature of the neuron surface. The length of a pyramidal neuron is important for recognizing pyramidal neurons and represent an important morphological trait. Curvature of the neuron surface is not as distinctive characteristics and therefore less important to the human perception of pyramidal neurons than information about the length. We therefore choose to use the largest obtainable radial distance for each sample direction.

The radial distance r belonging to a sample point direction is given by the equation

$$r = ||S(\theta, \phi) - M||. \tag{45}$$

where M denotes the center of mass, $S(\theta, \phi)$ denotes the intersection of the sample point direction (θ, ϕ) and the neuron surface, while $||\cdot||$ denotes taking the euclidean distance. A method for determining the intersection $S(\theta, \phi)$ will therefore be needed.

A simple method for determining the intersection $S(\theta, \phi)$ of a sample point direction (θ, ϕ) and the neuron surface will be to follow the direction vector $-||S(\theta, \phi) - M||$ from the outside of the neuron surface. We therefore initially let

$$r = \max_{v \in V} ||v|| \tag{46}$$

for all sample point directions. The variable v represent a vertex from the set V of vertices on the neuron surface, which has been defined to be the isosurface of value 0.5 of the neuron. We then iteratively decrease the radial distance r for all sample point directions by a certain precision untill we reach the neuron surface. The decreament of the radial distance r by a certain precision in each iteration introduce a precision error. We use a precision of 0.1 during the thesis.

When the intersection of the sample point direction (θ, ϕ) and the neuron surface have been reached, we can determine the sample point in spherical coordinates as

$$p = (\theta, \phi, r). \tag{47}$$

A sample points will therefore need to be converted to cartesian coordinates after the sampling procedure has ended. Figure 23 contains an example of a pyramidal neuron shape representation.

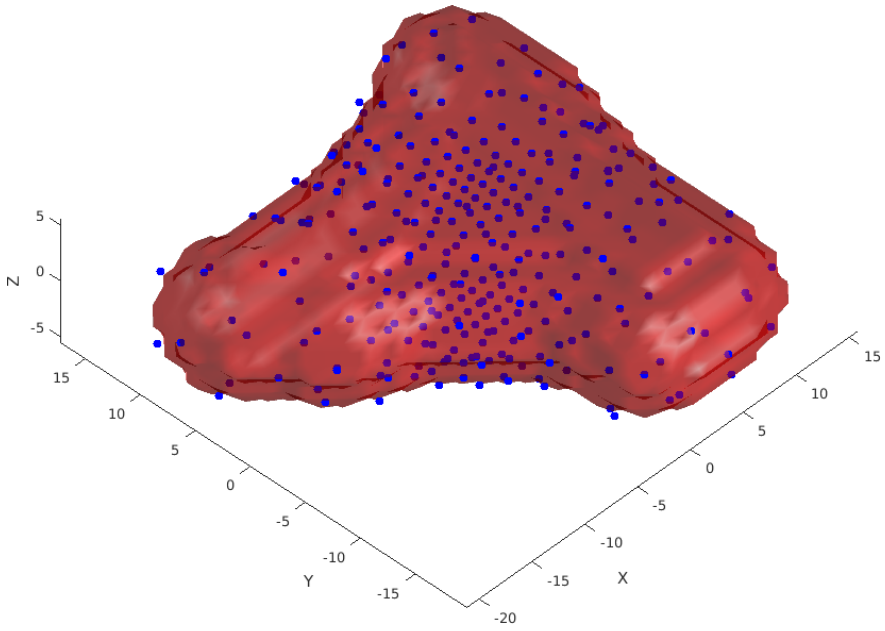


Figure 23: *Example of pyramidal neuron and the sample points that constitute its shape representation. The sampling has been made using the 300 sample directions seen in figure 22. Note that the pyramidal neuron is partially transparent.*

6.2 Principle component analysis of pyramidal neurons

6.2.1 Principle component analysis

Principal components analysis (PCA) is a statistical method for analyzing variation in observations. It relates possibly correlated data observations to uncorrelated orthogonal components through an orthogonal transformation, thus decomposing the observations into a common linear basis. Due to this, all data observations can in theory be completely reconstructed by linear combinations of the principal components obtained through the analysis, although in practice, this can only be done up to numerical precision.

Principal components analysis can be performed using different computational methods. Usually eigenvalue decomposition or singular value decomposition are used for the decomposition step, while the covariance matrix or the correlation matrix are used for representing the variations of the observations. Only the case of eigenvalue decomposition of the covariance matrix will be covered in the following subsections.

6.2.1.1 The covariance matrix

The covariance matrix is a matrix describing empirically observed variance in data. The matrix holds the variance for each observed variable in its diagonal and the covariances between all pairs of variables in the remaining entries. This matrix has at most $min(n, m)$ non-zero eigenvalues, where n is the number of observations and m is the number of variables.

As all variance in the data can be explained by the components with non-zero eigenvalues and the matrix has at most $min(n, m)$ non-zero eigenvalues, the analysis can safely be done using the smallest obtainable covariance matrix without loss of information and saving computations. It only requires the proper conversion under reconstruction of data observations.

The covariance matrix can be computed from the observed data by the procedure:

1. Create data matrix X by placing all data observations in its rows. TThe data matrix should be of dimension nxm with n observations of m observed varaibles.

2. Compute the mean observation $\overline{X}$ from the data matrix X. The mean observation should be a row vector (of dimension $1xm$).

3. Subtract the mean observation from all observations (all rows in the data matrix X) to obtain the matrix X'.

4. Compute the covariance matrix as either $\Sigma_0 = X'^T X'$ or $\Sigma_1 = X' X'^T$, depending on whether there are more data observations than variables for each data observation. If there are more observations than variables, use the first formula otherwise use the second formula.

6.2.1.2 Decomposition into principle components

The decomposition of matrices can, as mentioned, be done using different computational methods. For this thesis project, eigenvalue decomposition has been used, even though singular value decomposition was an option too.

Eigenvalue decomposition is a factorization method in linear algebra for factorizing any **diagonizable** matrix A. The matrix is factored into the matrices Q and Λ that respectively contains the eigenvectors and eigenvalues of A. Under the assumption that A is indeed diagonizable, the matrices are related by the equation

$$A = Q\Lambda Q^{-1}. \tag{48}$$

If A is not diagonizable, the invertible matrix Q does not exist and the matrices instead relates to each other by the equation

$$AQ = Q\Lambda. \tag{49}$$

The eigenvectors of any matrix that explains the variance in a data set can be regarded as principle components of the data set, as they form a linear basis for the matrix. The corresponding eigenvalues will form a measure for how much of the variance can be explained by the individual principle components. Thus PCA can be performed by the procedure:

1. Compute the covariance matrix Σ.

2. Do eigenvalue decomposition of the covariance matrix Σ.

3. Obtain the principle components as the eigenvectors of the covariance matrix found in the columns of the matrix Q, sorted in descending order according to their eigenvalues.

4. Obtain the amount of variance explained by the individual components as their corresponding eigenvalues (in descending order).

5. Return the principal components and the variance explained by the individual components. Return also mean observation for data reconstruction.

6.2.1.3 Reconstruction of data observations

A model for the observations in the data set in terms of the principle components would be

$$X = \bar{X}_d + P_k Z \tag{50}$$

where $P_k = [p_1, p_2, \cdots, p_k]$ are the first k principle components, k is a parameter of the model that defines how much of the variability from the data set the model should include and $\bar{X}_d$ is a matrix containg d rows of the mean observation $\bar{X}$. The matrix Z is defined as

$$Z = P_k^T (X - \bar{X}_d) \tag{51}$$

where the matrix $\bar{X}_d$ enable us to do row-wise subtraction of the mean observation. The model can be used for different purposes, including modelling of new observations and generation of artificial observations. To retain plausibility of artifical observations, the elements of $(X - \bar{X}_d)$ should not vary by more than three standard deviations of each column.

6.2.1.4 Pros and cons of the method

The drawbacks of principal component analysis come from its linear nature. Due to the Principal component analysis computing a *linear basis*, the analysis is only well-suited for linearly expressable problems. When using eigenvalue decomposition, the covariance matrix computed from the data set must also be diagonizable, thus placing a restriction on the data set. In addition to this, linear methods are sensitive to outliers in the data observations, which can give rise to precision problems when using noisy data. However, the method also has its advantages, as it can be easily computed and can be used for data sets of arbitrary dimensions.

6.2.2 Applying principle components analysis to pyramidal neurons

Application of principle components analysis to the pyramidal neurons have been done using the extension of Pincus' and Theriot's method to produce shape representations consisting of 100 sample points (yielding shape representations of dimension 1x300). This section will present the results of the principle components analysis, while focusing on the method's ability to model pyramidal neurons. We will specifically look at

- How well the method explains the shape variation of pyramidal neurons.
- What shape characteristics the first few principle components represent.
- How well the method can reconstruct pyramidal neurons.

6.2.2.1 Explaination of shape variation

Figure 24 visualize how much of the variation in the pyramidal neuron data has been captured by the 97 principle components. The plot reveal that approximately 10 principle components

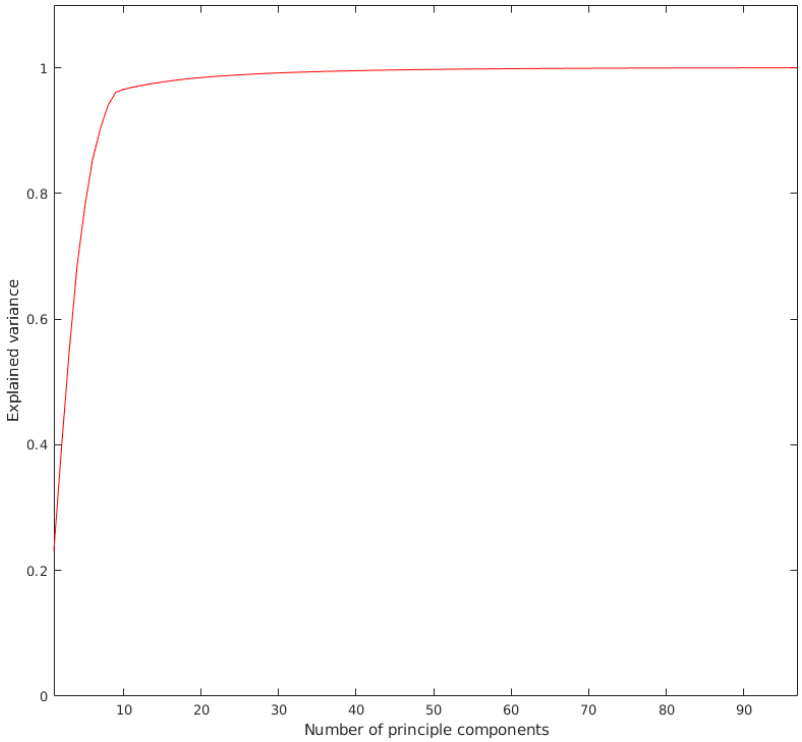


Figure 24: *Amount of variance in the pyramidal neuron data that are explained by the first principle components. The plot reveal that we can explain about 60% of the variance by using the first 5 principle component, while we need 9 principle components to explain 95% of the variance.*

are necessary for capturing more than 95% of the variation in the pyramidal neuron data. The first 10 principle components are therefore sufficient to model more than 95% of the shape variation for the pyramidal neurons.

It seems like a low number of principle components considering that there are 97 principle components for these data. The low number of ncessary principle components indicate that we can build a simple model for pyramidal neuron shapes without losing much information.

We therefore conclude that a principle component model can explain the shape variation of pyramidal neurons well.

6.2.2.2 Visualization of principle components

Figure 25-27 visualize the first three principle components and how they affect the shape of pyramidal neurons. Each figure compare the mean shape of the pyramidal neurons to shapes, where the first, second and third principle components have been added to or subtracted from the mean shape to different degrees.

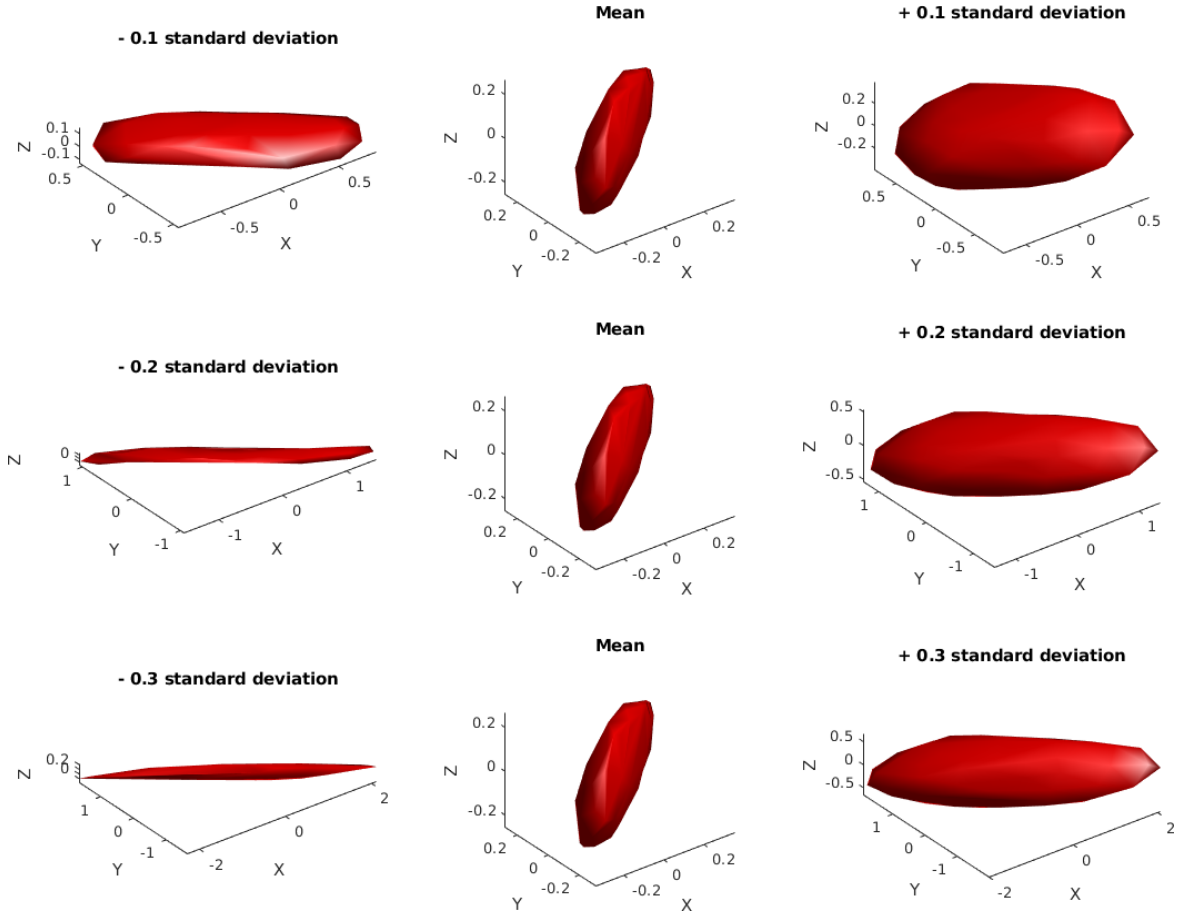


Figure 25: *Visualization of the first principle mode. The first principle mode seem to represent height of the neuron together with volumen along the first principle axis.*

Figure 25 visualize the first principle component using the pyramidal neuron from figure 10.

Both addition and subtraction of the first principle component seem to add height to the pyramidal neuron. Addition of the first principle component seem to make the pyramidal neuron thicker along its primary axis, while subtraction of the first principle component seem to make the pyramidal neuron more narrow along its primary axis. The sharpness of the apex of a pyramidal neuron also seem to be affected by the first principle component as a consequence of representing the thickness or narrowness along the primary axis of a pyramidal neuron.

The first principle component therefore seem to represent several characteristics of pyramidal neurons. The characteristics represented by the first principle component seem to be

- Height of a pyramidal neuron.
- Volume along the primary axis of a pyramidal neuron.
- Sharpness of the apex of a pyramidal neuron.

Figure 26 visualize the second principle component using the pyramidal neuron from figure 10. Addition of the second principle component seem to add orientation to the base of a pyramidal neuron, as well as adding length along the secondary axis of the pyramidal neuron. Addition of the second principle component also seem to narrow the base of a pyramidal neuron. Subtraction of the second principle component seem to affect the orientation of the base of a pyramidal neuron too, as well as thickening the base and decrease the base lenth.

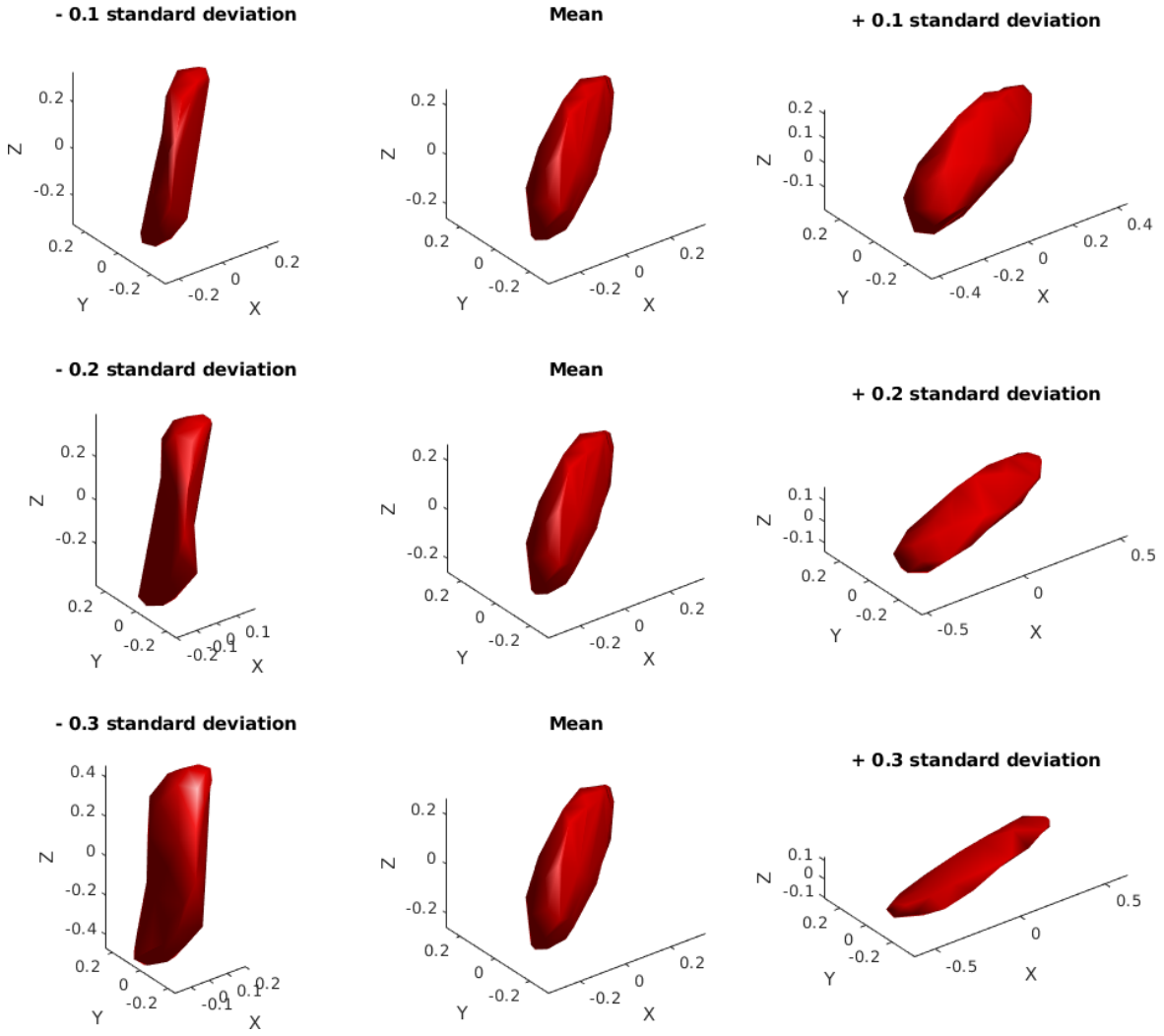


Figure 26: *Visualization of the second principle mode. The second principle mode seem to represent length and width of the base of the neuron together with volumen along the base.*

The second principle component therefore seem to affect several characteristics of pyramidal neurons. The characteristics affected by the second principle component seem to be

- Length and width of the base of a pyramidal neuron.

- Volume of the base of a pyramidal neuron, as a consequence of affecting the base dimension.

Figure 27 visualize the third principle component using the pyramidal neuron from figure 10. The third principle component seem to affect similar characteristics to the second principle component.

The length and width of the base however seem to have changed to the length and width of some other side of a pyramidal neuron than the base. This could be an indication that the alignment of the pyramidal neurons are not sufficient along the secondary or tertiary axis of the pyramidal neurons, as suspected. Please see discussion on alignment in section 6.1.2.2 for details.

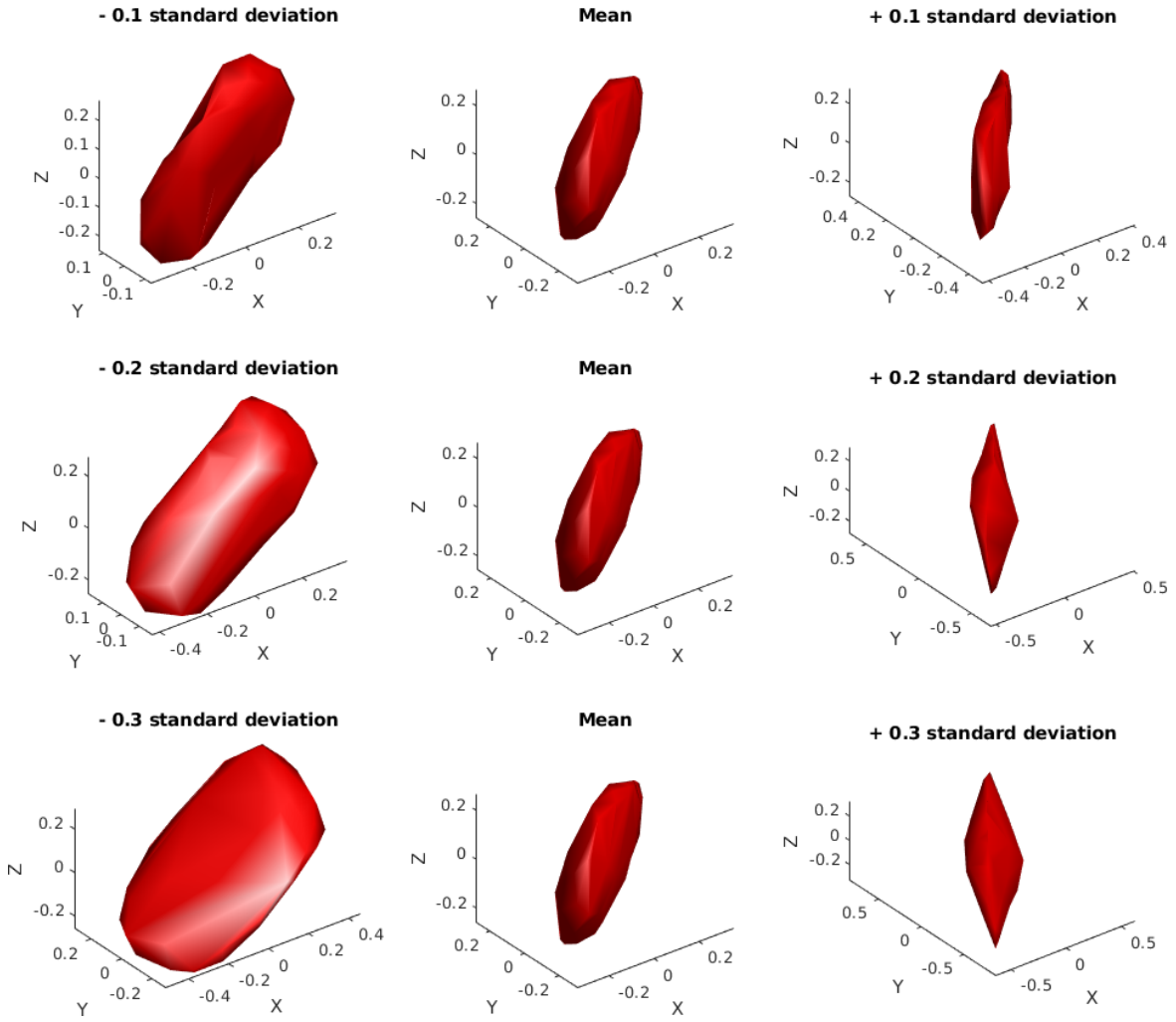


Figure 27: *Visualization of the third principle mode. The third principle mode seem to represent length and width of some other side than the base of the neuron together with volumen along this side.*

6.2.2.3 Pyramidal neuron reconstruction

Reconstruction of the pyramidal neuron from figure 10 has been performed and illustrated in figure 28 using 10, 20 and 50 principle components.

The reconstruction reveal that 10 principle components are not sufficient to model a pyramidal ground structure. The reconstruction of the pyramidal neuron look more like an egg than a pyramid, even though the first 10 principle components ought to be sufficient to explain more than 95% of the shape variation of the pyramidal neurons.

The egg-like reconstruction of the pyramidal neuron suggest that pyramidal ground structure are explained by the less than 5% of shape variation that was left out by only using 10 principle components.

Using 20 principle components for the reconstruction, we begin to see a sign of pyramidal ground structure in the reconstruction of the pyramidal neuron.

We have to use approximately 50 principle components for the reconstruction to see a clear pyramidal ground structure in the reconstruction of the pyramidal neuron.

If a model cannot reconstruct the ground structure of a class of shapes using 95% of shape variation in the class, the model cannot be said to reconstruct that shape class very well. We therefore conclude that principle components analysis is not well-suited for shape analysis of pyramidal neurons.

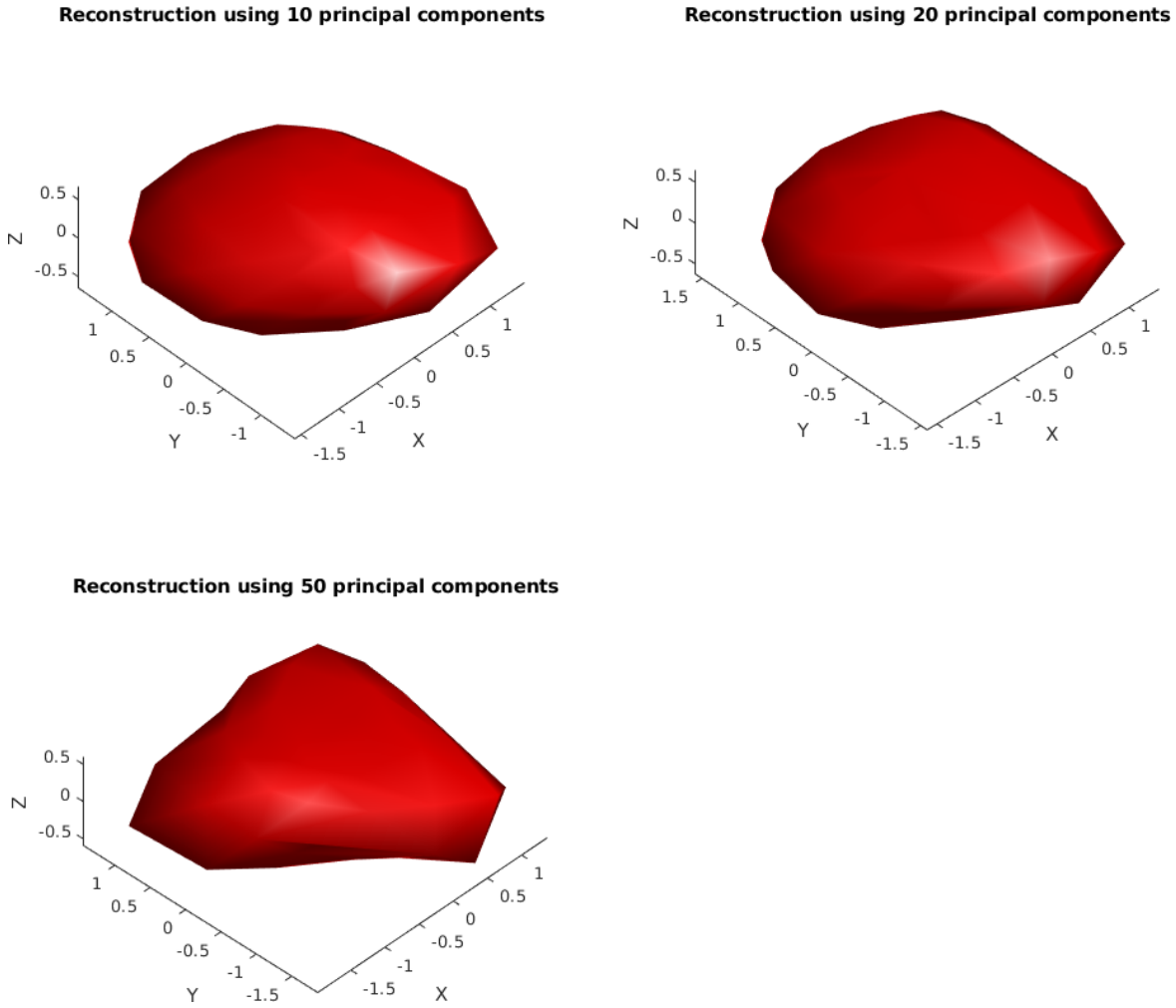


Figure 28: *Reconstruction of the pyramidal neuron from figure 10 using 10, 20 and 50 principle components. Using less than 20 principle components for the reconstruction, the pyramidal ground structure become barely visible. At least 50 principle components are needed for a distinct reconstruction.*

6.3 Spherical harmonics analysis of pyramidal neurons

Application of spherical harmonics analysis to the pyramidal neurons have been done to the same shape representations as we performed principle components analysis on. By using the same shape representations we make a comparison of the performance of the two models possible. This section will present the results of the spherical harmonics analysis, while

focusing on the method's ability to model pyramidal neurons. We will look at

- What shape characteristics the first few principle components represent.
- How well the method can reconstruct pyramidal neurons.

6.3.1 Visualization of frequency components

Figure 29-31 visualize the first three frequency components and how they affect the shape of pyramidal neurons. For the visualization of the first frequency components we have again made use of the neuron from figure 10.

Due to spherical harmonics analysis only utilizing addition of frequency components, we do not need to examine how the frequency components scale. It will suffice to visualize the frequency components as they are.

All three frequency components seem to represent narrow and oval volumes that stretch from base to apex of a pyramidal neuron. This seems odd considering that the volume of a pyramidal neuron scatter around the base. One would think that the first three frequency components would then stretch in at least two directions.

It seems reasonable that the frequency components are quite small. A complex shape requires a lot of frequency components to be reconstructed to a recognizable degree and a lot of individual frequency components might contribute to the size of the shape.

The visualization of the three frequency components possibly explain why the rotation invariant shape descriptor SHS_{small} failed in distinguishing the pyramidal neurons from the astrocyte neurons. The shape information that the first three frequency components represent simply seem to similar to the shape of the astrocyte neurons.

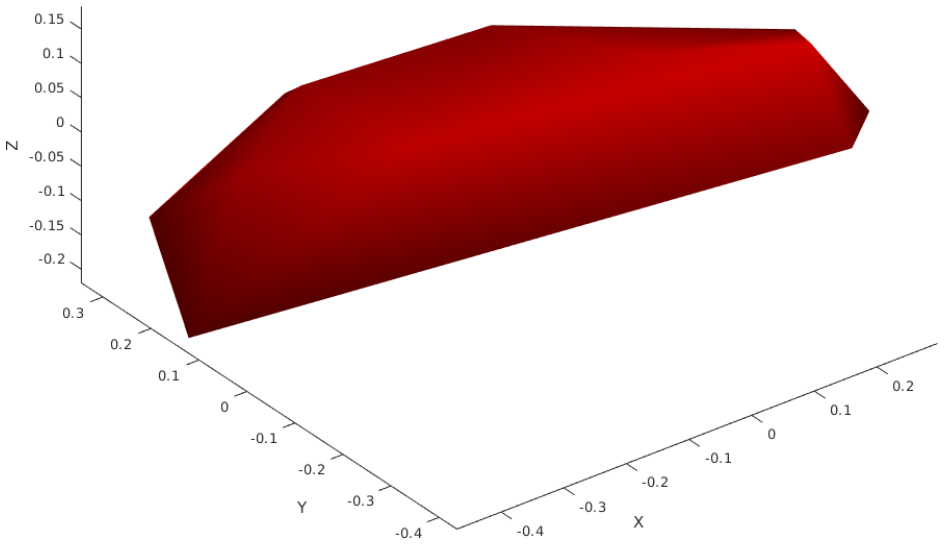


Figure 29: *Visualization of first frequency component. The component seem to represent some volume along some apex-to-base axis.*

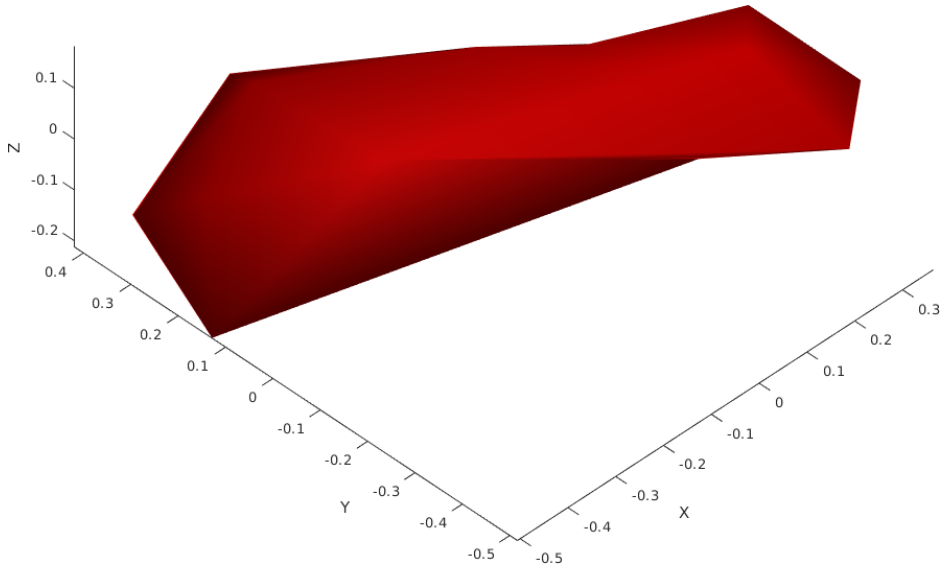


Figure 30: *Visualization of second frequency component. The component seem to represent some volume along some apex-to-base axis.*

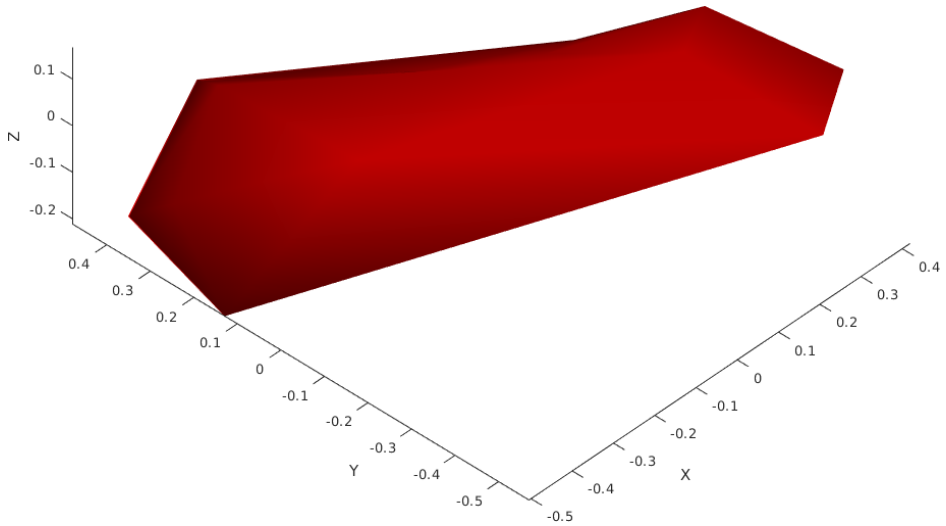


Figure 31: *Visualization of third frequency component. The component seem to represent some volume along some apex-to-base axis.*

6.3.2 Pyramidal neuron reconstruction

Reconstruction of the pyramidal neuron from figure 10 has been performed and illustrated in figure 28 using one, two and three frequency components. The first three frequency components do not seem to vary the shape of the pyramidal neuron much and the reconstruction does not look like a pyramidal neuron. This may simply be due to using an insufficient number of frequency components for the reconstruction.

The odd looking frequency components together with the poor reconstruction results, gives rise to the suspicion that the spherical harmonics analysis implementation might contain a fault however. It would not come as a surprise, since there were other troubles with this implementation.

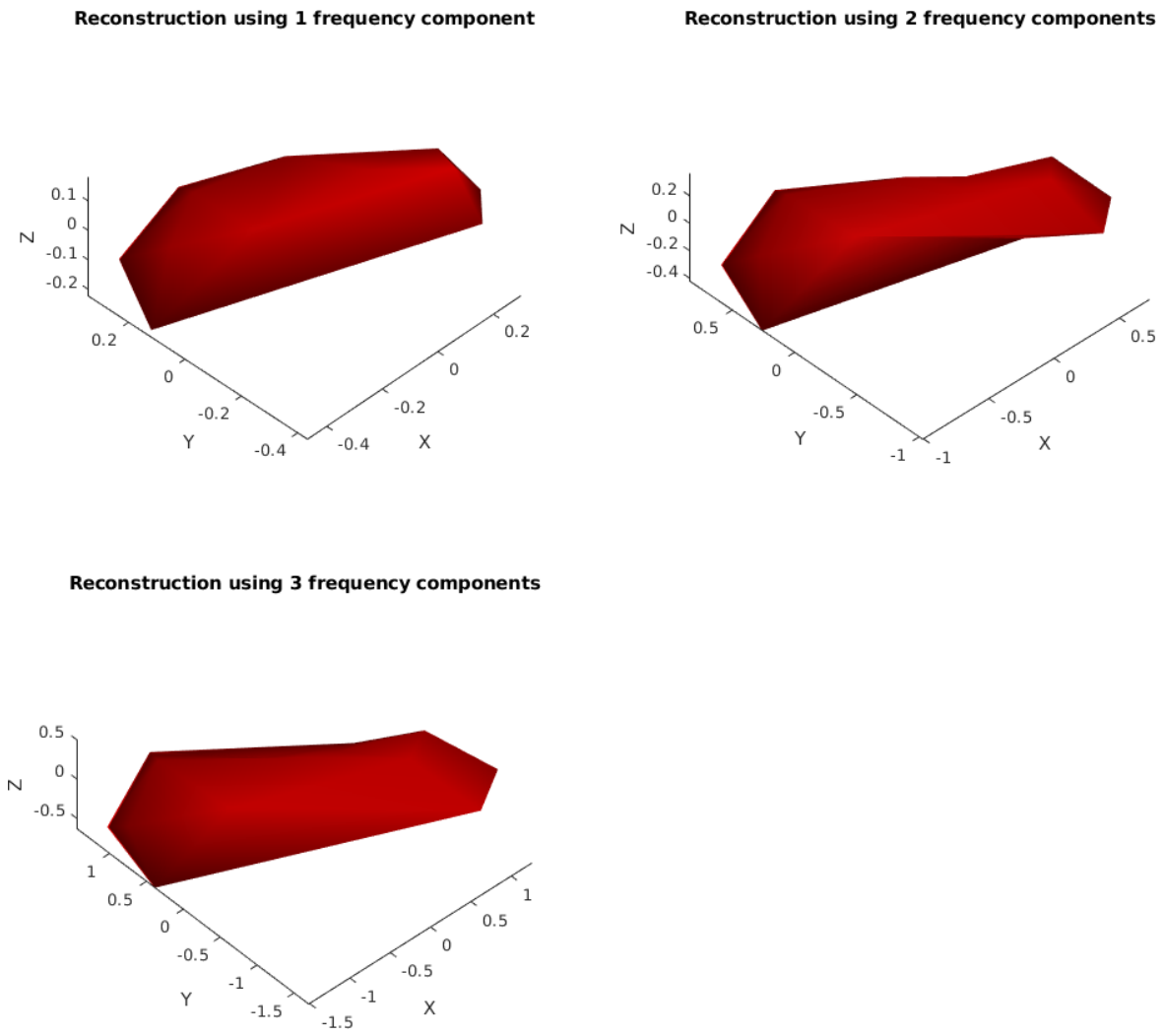


Figure 32: *Reconstruction of pyramidal neuron from figure 10 using 1, 2 and 3 frequency components. The reconstruction are far from good and look much like the corresponding frequency components.*

6.4 Comparison of the shape analysis methods

Under the assumption that the implementation of the spherical harmonics analysis is correct, the principle components analysis method seem to perform the best.

The principle components seem to be easier to relate to actual shape characteristics than frequency components. The principle components analysis seem to provide for more accurate and more recognizable reconstructions than spherical harmonics analysis does.

Neither method seems to be able to model pyramidal neurons using only a few of their respective components.

7 Direction of pyramidal neurons

Computation of the direction of an object highly depends on the definition of direction. Several definitions of direction could be possible to use with respect to the pyramidal neurons. Three possible definitions for pyramidal neurons that are both intuitive and easy to compute would be

- the direction of the first principle axis.

- the direction of the vector from the cell core to the apex.
- the direction of the vector from the center of mass to the apex.

The three definitions above might result in different direction vectors and they have their respective advantages and disadvantages. Even so, they are all equally valid in the context of neuron direction.

The first definition relies only on principle components analysis of the voxel positions that constitute the neuron. Principle components analysis are already to be implemented for the shape analysis task, so use of the first definition would not require additional work to be done. Computation of directions of pyramidal neurons would therefore become an easy task using the first definition.

However, the first definition suffers from the fact that our perception of primary axes of an object, is not always equal to the actual principle axes of that same object. Using the first principle axis to define the direction of pyramidal neurons might thus not yield the desired result.

The second definition matches our perception of the primary axes of the pyramidal neurons better. The cell core theoretically ought to be in the middle of the soma and the apex of a pyramid structure conceptually lies at the top of the structure. The second definition thus theoretically matches the upward direction.

The second definition unfortunately relies on the theoretical placement of the cell core. As neurons must be assumed to be flexible biological structures that might deform slightly over time, the assumption could introduce precision errors to the direction vector. Segmentation errors on the edge of the soma could affect the relative position of the cell core too. This makes precision errors even more likely to occur. In addition, usage of the second definition would also require additional steps to be taken during segmentation, so that cell cores will be detected.

The third definition conceptuallly matches our perception of the primary axes of the pyramidal neurons as well. The third definition does not depend on the theoretical placement of the cell core however and results in a more robust direction vector than the second definition. It is however, also prone to producing precision errors as a result of segmentation errors on the edge of the soma.

Based on the discussion above, the third definition seem to produce the most reliable directions. The definition seem to be the most robust definition as well as matching our perception of direction of pyramidal neurons the best. Therefore it has been decided to go with the third definition of pyramidal neuron direction during the thesis.

7.1 Finding the apex

The diameter of a pyramidal neuron roughly equals two third of its height. Thus it seems reasonable to assume that the first principle axis of a pyramidal neuron must have an orientation parallel to some vector from the base of the pyramidal neuron to its apex. Under this assumption, the first principle axis can be used to compute the apex position of a pyramidal neuron, even if the first principle axis does not match our perception of pyramidal neuron direction directly.

The assumption about the first principle axis orientation results in the voxels of a pyramidal neuron being distributed unevenly along the first principle axis. Around the apex there will simply be fewer voxels than around the base. The apex should therefore be possible to locate by projecting all voxels to the first principle axis. When projecting the voxels to the first principle axis, the voxels that constitute the base will tend to cluster more tightly than the voxels constituting the apex. The range of the projected voxels can therefore be divided into two halves, one of them holding fewer voxels than the other. The apex can then be located in the smaller half, as the voxel farthest away from the bigger half of projected voxels.

A simple way to implement the described localization of the apex would be to initially align the first principal axis of the pyramidal neuron to the x-axis using PCA alignment. This will reduce the projection step to a matter of setting the y- and z-coordinates to 0. However, as only the x-coordinates are needed for deciding on the apex loation, the change of y- and z-coordinate values are in fact not necessary to perform. After alignment, the range in which the x-coordinates lies can be divided into the two equal halves and it can be decided which of them has more voxels by a simple count. Depending on the count, the apex location will be the original coordinates of either the voxel that has the maximal or minimal x-coordinate value after the projection to the first principle axis was performed.

7.2 Computing the direction

When its apex has been located, the direction D of a pyramidal neuron can be computed. The direction will be given as

$$D = A - M \tag{52}$$

where A and M denotes the apex and the center of mass of the pyramidal neuron, respectively. Figure 33 shows an example of pyramidal neuron direction and the relations between the center of mass, the apex and the first principle axis.

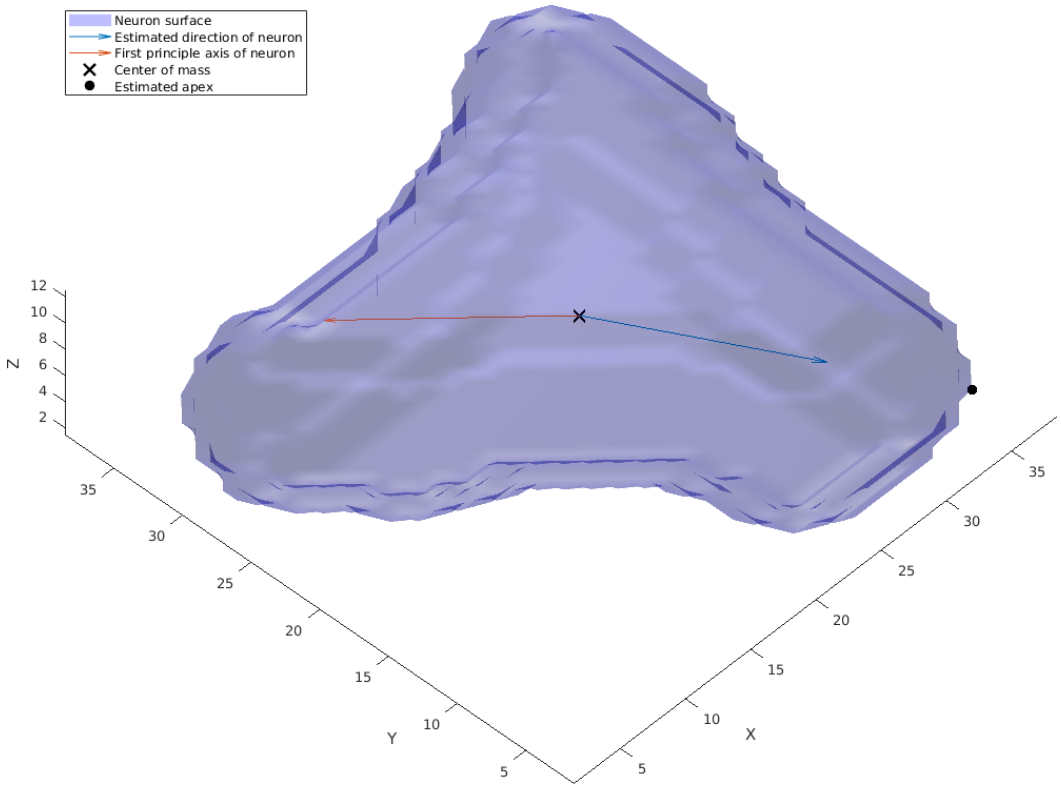


Figure 33: *Relation between estimation parameters for direction estimation. The apex are marked by a black dot, while the direction of the pyramidal neuron are visualized by the blue arrow. The red arrow correspond to the first principle axis and point in the wrong direction. The black cross mark the center of mass.*

7.3 Observed pyramidal neuron directions

Figure 34 contain a plot of estimated directions of pyramidal neurons. Directions of pyramidal neurons seemingly tend to cluster around 2-4 certain directions. The plot seem to reveal two pairs of clusters that pairwise contain directions opposite to each other.

To verify the indication of clustering in the plot, some statistical analysis of the distribution of the pyramidal neuron directions will be necessary. Entropy analysis on the sphere can provide for such statistical analysis of the pyramidal neuron directions.

Under the assumption that the indication of clustering in the plot is indeed correct, we have an indication of a certain arrangement of pyramidal neurons in the brain tissue too. If an arrangement exist, we will not observe random pyramidal neuron directions. On the contrary, if no arrangement exist, we must observe random pyramidal neuron directions.

Statistical analysis of the distribution of the pyramidal neuron directions are therefore not just necessary to confirm the clustering tendency, but also necessary to confirm if any arrangement of the pyramidal neurons in the brain tissue exist.

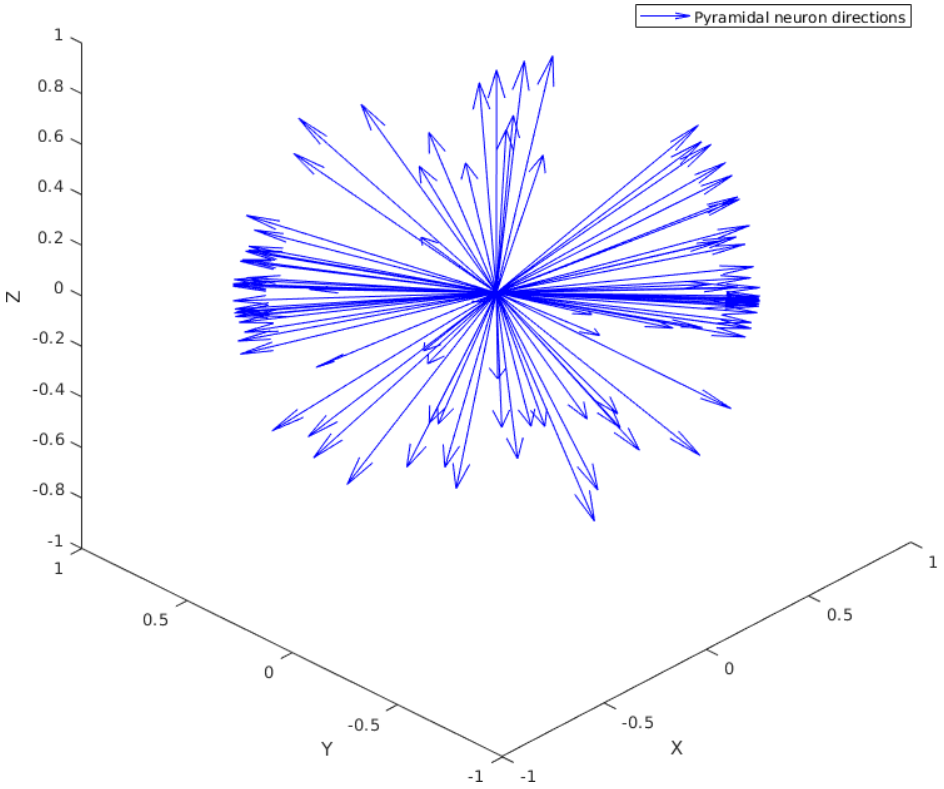


Figure 34: *Directions of pyramidal neurons displayed as arrows starting in the origin. The directions seemingly have a tendency to cluster around two to four certain directions opposite to each other.*

8 Statistics on pyramidal neurons

Statistics are useful for investigating the characteristics of pyramidal neurons.

If we want to know how accurate our theoretical knowledge of characteristics of pyramidal neurons is, we can compare what we know to statistics such as the mean, the standard deviation, the variance and the median observation.

It would be even better, if we can determine some kind of distribution for the various characteristics of pyramidal neurons. We would then be able to actually say how likely it is to find a pyramidal neuron of some specific characteristics.

We will base the statistics in this section on the 97 pyramidal neurons from the test images. It should be kept in mind that this is not a lot of pyramidal neurons to do statistical analysis upon. Any statistical result given in this section must therefore be confirmed using more pyramidal neurons later on.

8.1 Dimensions

Table 1 summarize statstics on the dimensions of pyramidal neurons. We define the dimensions of a pyramidal neuron to be the length of its principal axes.

	Mean	**Median**	**Standard deviation**	**Variance**
First principle axis	17,9974 μm	15,5931 μm	8,0760 μm	130,4430 μm
Second principle axis	11,2136 μm	10,1255 μm	4,4448 μm	39,5127 μm
Third principle axis	5,9397 μm	5,7799 μm	0,6635 μm	0,8804 μm

Table 1: *Statistics on dimensions of pyramidal neurons. We use the length of the first three principle axes as dimensions.*

The length of the first principal axis should be around $\sim 20\mu m$ in average, since the first principal axis should equal the height of a pyramidal neuron. We observe that the average length of the first principle axis is around $\sim 18\mu m$. When considering that the segmentation likely miss some of the neuron border, the average length seem to be consistent with the theoretical height of pyramidal neurons.

The standard deviation and variance of the length of the first principal axis seem high, while the median observation is close to the average length of the principal axis. Together this indicates that we either have some outlying observations with respect to the length of the first principle axis or have a distribution with a single tail.

The statistics indicate the same tendency to match the theoretical value well for the second principal axis. The length of this principle axis should equal the diameter of a pyramidal neuron and should therefore be around $\sim 13.5\mu m$. We observe that the average length of the second principle axis is around $\sim 11\mu m$. There are indications of outlying observations or a distribution with a single tail for the second principle axis as well.

Under the assumption that we could model pyramidal neurons as square pyramids, this principle axis ought to be of the same length that the second principle axis. This is not the case however. The average length of the third principle axis seem to lie around $\sim 6\mu m$. At the same time the median observation for the length of the third principle axis is close to its average length. Its variance and standard deviation is quite small too.

What the neuro-biologists call a diameter of a pyramidal neuron could actually be a diagonal for a rectangle base. This claim is based on application of Pythagoras' theorem to the average length of the second and third principle axis. It yields an average diangonal length of

$$d_l = \sqrt{11.2136^2 + 5.9397^2} \sim 12.6895 \tag{53}$$

that would match the theoretical value of the diameter even better than the average length of the second principle axis. Manual inspection of pyramidal neurons have indicated that this could be the case as well.

Figure 35-37 display histograms of the three dimensions together with a lognormal distribution fit. First and second dimension seem to match the lognormal distribution fit fairly well. The third dimension matches its fit to lognormal distribution less well. Lognormal distribution however is the distribution that fit the histogram of the third dimension best out of known distributions.

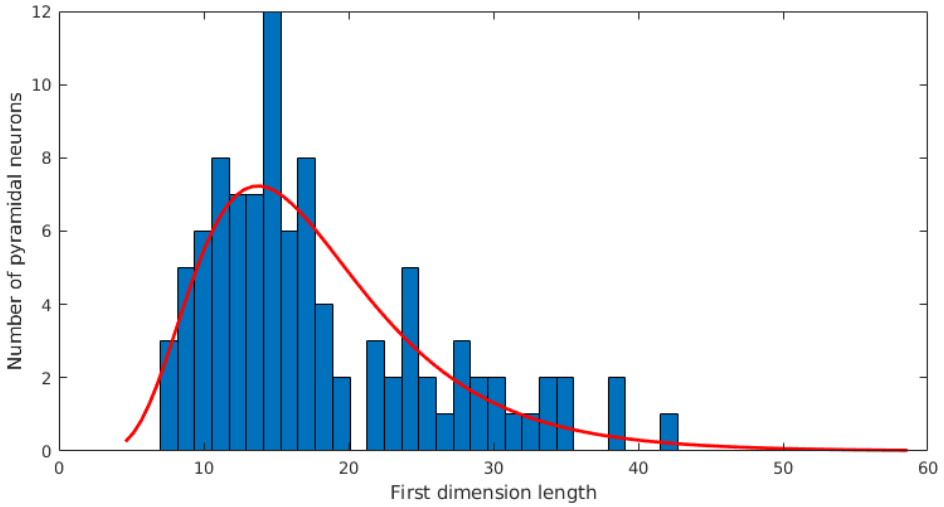


Figure 35: *Histogram that visualize the distribution of the first dimension. The histogram have been fit to a lognormal distribution, which seem to match the histogram well.*

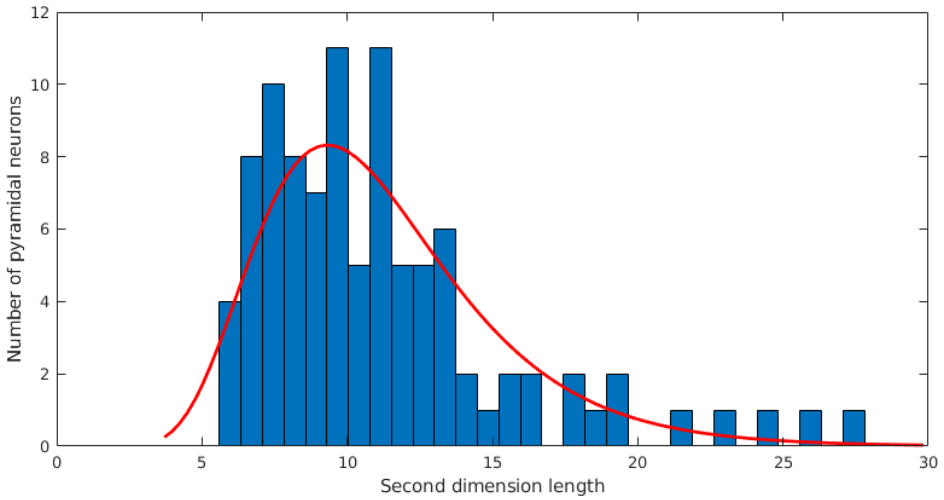


Figure 36: *Histogram that visualize the distribution of the second dimension. The histogram have been fit to a lognormal distribution, which seem to match the histogram well.*

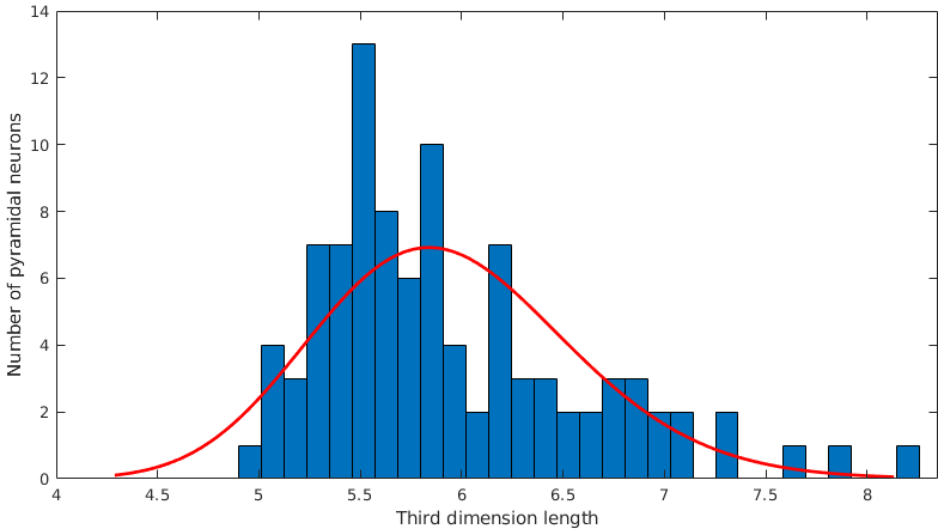


Figure 37: *Histogram that visualize the distribution of the third dimension. The histogram have been fit to a lognormal distribution, which seem to match the histogram somewhat.*

8.2 Volume and surface area

Statistics on volume and surface area of pyramidal neurons are found in table 2 that summarize these. The average volume seem to be far apart from the median volume. So does the average surface area and the median surface area. The variance and standard deviation of both metrics seem quite high too.

	Mean	**Median**	**Standard deviation**	**Variance**
Volumen	929,7875 μm^3	617,5000 μm^3	902,8500 μm^3	6521125,0000 μm^3
Surface area	548,5750 μm^2	413.3250 μm^2	397,4250 μm^2	631750,0000 μm^2

Table 2: *Statistics on volume and surface area of pyramidal neurons.*

The average volume is $\sim 300\mu m^3$ less than we would expect for a pyramidal neuron that can be modelled as a square pyramid. The average surface area of a pyramidal neuron is $\sim 200\mu m^2$ less than we would expect it to be. Considering again that some of the neuron border is likely lost during the segmentation process, it does not seem to be unreasonable average values.

Figure 38 and 39 display histograms of the volume and surface area of the pyramidal neurons together with lognormal distribution fits. The volume and variance of the pyramidal neurons seem to match lognormal distributions very well.

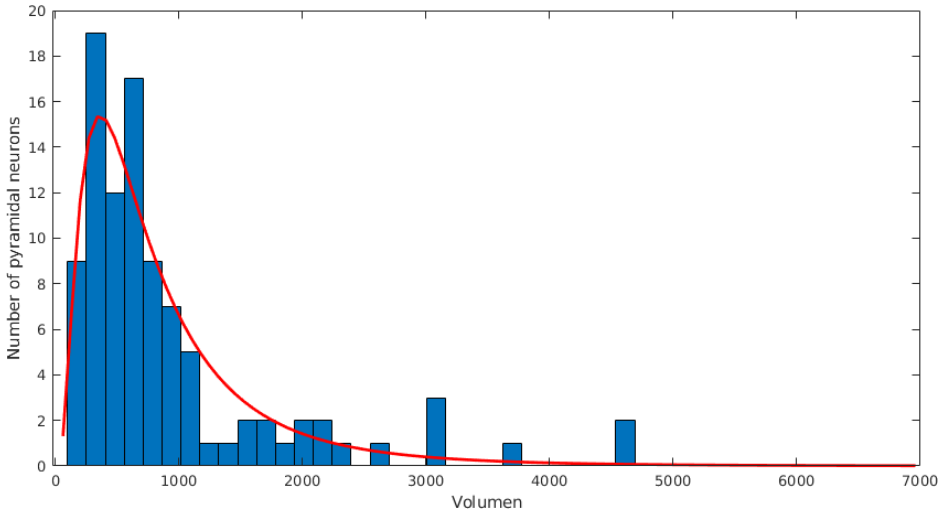


Figure 38: *Histogram on volume of pyramidal neurons. A lognormal distribution has been fit to the histogram. It seem to match the lognormal distribution quite well.*

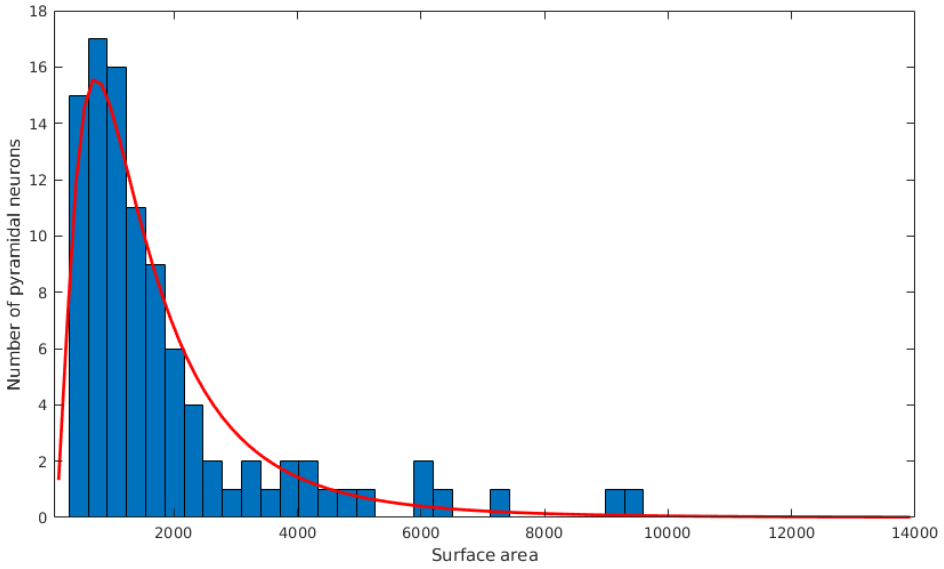


Figure 39: *Histogram on surface area of pyramidal neurons. A lognormal distribution has been fit to the histogram. It seem to match the lognormal distribution quite well.*

9 Discussion on improvements and further work

A number of challenging problems have been addressed throughout the thesis report and it is therefore clear that there is room for improvements.

We will first suggest to automate the registration completely by implementing detection of landmarks. Using the current implementation, the user of the program must spend time on selecting two landmarks for each image section. It might not sound like much, but it can become time consuming, when there is enough image sections.

The segmentation process is currently the slowest part of the program and we will therefore suggest to optimize the program by running the segmentation using parallel programming.

The annotation experiment will need to be repeated using the rotation invariant shape descriptor SHS. This will be to determine if the rotation invariant shape descriptor SHS_{small}

is too simple to catch a sufficient amount of shape information to separate the neurons. Other shape descriptors could also be used for the repetition of the experiment.

Another suggestion will be to implement proper alignment for pyramidal neurons. This will consist of computing a directional axis set to use for the alignment instead of the somewhat unreliable principle axes. An easy way to do so would be to extend the direction estimation process to three dimensions.

One of the most important improvements for the thesis work will consist of investigating or removing the bias from the extension of Pincus' and Theriot's method for computation of shape representations. It is unclear how the systematic bias that we introduce to the shape representation will affect shape analysis and it therefore seem beneficial to either implement the missing adjustment of the sample points or investigate the effect of the bias. Another important improvement to the shape analysis will be to correct potential faults in the spherical harmonics analysis.

The last suggestions will be to verify the clustering tendency and the underlying lognormal distributions for the dimensions, volume and surface area of pyramidal neurons. The verification can be done using statistical tests.

10 Conclusion

The attempt at implementing a program that can aid neuro-biologist in their research succeeded. The implemented program provide for

- semi-automatic registration of tissue samples.
- automatic detection of neurons in tissue samples.
- performing principle component analysis of neurons.
- performing spherical harmonics analysis of neurons.
- automatic estimation of the direction of pyramidal neurons.
- automatic computation of simple statistics on neurons.

The program has been designed for tissue samples that have been subjected to staining prior to image acquisition using high resolutions. Whether the spherical harmonics analysis of neurons works as intended is however unclear at the moment.

Automatic annotation of neurons has been attempted as an experiment during the thesis project. Rotation invariant shape descriptors constructed from spherical harmonics were classified using the k-means clustering method to annotate the neurons. Comparison to a manual annotation revealed that the rotation invariant shape descriptor was not able to separate the neurons. More annotation experiments will have to be conducted to determine why the rotation invariant shape descriptor cannot separate the neurons.

Principle component analysis seem to perform better than spherical harmonics analysis when applied to pyramidal neurons. It is easier to relate the principle components to the characteristics of the pyramidal neurons and principle component analysis provide for better reconstructions than spherical harmonics analysis do. Characteristics of pyramidal neurons seem to be represented by the smaller principle components however. The method therefore seem to capture characterstics of presumably nonlinear structures poorly.

Statistics on pyramidal neurons indicate that pyramidal neurons might be better modelled using a rectangle pyramid than a square pyramid. The reason is that a side length of the pyramidal neurons in general doesn't match the expected size. Statistics also indicate that theoretical estimates on pyramidal neurons sizes are reasonable. We hereby confirm the neuro-biologists description of this type of neuron to some degree.

It has been observed that directions of pyramidal neurons tend to cluster around two to four opposite directions, which indicate that the pyramidal neurons have a certain arrangement.

11 References

[1] A. H. Rafati, F. Safavimanesh, K.-A. Dorph-Petersen, J.G. Rasmussen, J. Møller and J. R. Nyengaard. *Detection and spatial characterization of minicolumnarity in the human cerebral cortex*. Journal of Microscopy, Vol. 261, Issue 1 2016: 115–126

[2] M. Jafari-Mamaghani, M. Andersson, P. Krieger. *Spatial point pattern analysis of neurons using Ripley's K-function in 3D*. Frontiers in Neuroinformatics, Vol. 4 2010, article 9.

[3] Z. Pincus, J. A. Theriot. *Comparison of quantitative methods for cell-shape analysis*. Journal of Microscopy, Vol. 227, Pt. 2 2007: 140–156

[4] A. H. Rafati, J. F. Ziegel, J. R. Nyengaard and E. B. V. Jensen. *Stereological Estimation of Particle Shape and Orientation From Volume Tensors*. Centre for Stochastic Geometry and Advanced Bioimaging, research report 2015: `https://data.math.au.dk/publications/csgb/2015/math-csgb-2015-12.pdf`.

[5] Michael Kazhdan, Thomas Funkhouser and Szymon Rusinkiewicz. *Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors*. Eurographics Symposium on Geometry Processing 2003: 156-164.

[6] L. Shen, and M. K. Chung. *Large-Scale Modeling of Parametric Surfaces using Spherical Harmonics*. Third International Symposium on 3D Data Processing, Visualization, and Transmission 2006: 294-301.

[7] Ming-Ching Chang and Benjamin B. Kimia. *Measuring 3D shape similarity by matching the medial scaffolds*. IEEE 12th International Conference on Computer Vision Workshops 2009.

[8] P. Grünwald, I. J. Myung and M. Pitt. *Advances in Minimum Description Length: Theory and Application*. MIT Press 2004.

[9] Chris Solomon and Toby Breckon. *Fundamentals of Digital Image Processing - A Practical Approach with Examples in Matlab*. Wiley Blackwell 2011.

[10] P. Jensen. *Den lille blå*. Polyteknisk Forlag 2014.

[11] `http://image.diku.dk/mediawiki/index.php/Slurm_Cluster` - 23/07-18

[12] `https://en.wikipedia.org/wiki/Pyramidal_cell` - 11/2-18

[13] `http://www.scholarpedia.org/article/Pyramidal_neuron` - 11/2-18

[14] `http://www.cell.com/current-biology/pdf/S0960-9822(11)01198-5.pdf` - 11/2-18

[15] `https://en.wikipedia.org/wiki/Astrocyte` - 14/11 - 18

[16] `https://en.wikipedia.org/wiki/Spherical_harmonics` - 02/07-18

[17] `https://en.wikipedia.org/wiki/Associated_Legendre_polynomials` - 02/07-18

[18] https://en.wikipedia.org/wiki/Mutual_information - 25/03-18

[19] http://www.scholarpedia.org/article/Mutual_information - 25/03-18

[20] http://kurser.ku.dk/course/ndaa09027u/2018-2019 - 8/8-18

12 Appendix

12.1 A1: Thesis contract

Working title

Detection of pyramidal neurons and analysis of their characteristics from 3D microscopy images.

Problem and motivation

The arrangement of pyramidal neuron cells in the brain, i.e. the placement and orientation of the neuron cells, are important to biologists for various reasons, but it is a time consuming and slow task to manually detect and annotate the cells in 3D microscopy images, which is needed for estimating the arrangement. A faster way of detecting the pyramidal neuron cells and their arrangement in such images are therefore needed.

All neuron cells has a cell body (the soma) and some threads (the dendrites) enabling them to send neural signals to other, nearby neuron cells. The dendrites are in general not visible in microscopy images, only the somas are big enough to be visible. The biologists have named and described the pyramidal neuron cells according to the shape of their soma, which is kind of formed as pyramids in contrast to other neuron cells that are more round, but it has not been investigated if this is the actual shape of the pyramidal neuron cells.

It would thus be interesting and beneficial to implement a system for the detection of the pyramidal neuron cells and for doing statistics over their characteristics in order to investigate how accurate this way of describing the cells is, and to enable the biologists to do their work faster.

Goals for the thesis project

The overall goals for this thesis project will be to

1. Implement a system that detects the pyramidal neuron cells in the 3D microscopy images of brain tissues, estimates the orientation of the pyramidal neuron cells and does various statistics on the characteristics of the pyramidal neuron cells.
2. Evaluate the statistics to determine, how accuarate the way the biologists describes the pyramidal neuron cells, is.

Learning objectives

The overall learning objectives for this thesis project will be

1. Knowledge about characteristics of pyramidal neuron cells.
2. Knowledge about what types of shape analysis is suitable for different kind of shapes.
3. Design and implement a system for detection of pyramidal neuron cells and estimating their orientation.
4. Provide statistics on the characteristics of pyramidal neuron cells.
5. Evaluate our current way of describing pyramidal neuron cells in the sense of accuracy.

Data set

The data set that is intended to be used for the thesis project will contain 3D microscopy images of tissues from 40 different brains. Each brain tissue sample will contain approximately 1000-2000 consecutive sections. In total the entire data set will be of approximately 12 TB of image data and each section will be of around 300 MB. The data set is however, still being produced.

Data available now includes both marked and unmarked microscopy images of about 50 serial images, which will contain approximately 200-300 pyramidal neuron cells, along with consecutive sections of the entire tissue sample for a single brain. This will be more than sufficient to design and test the algorithms for detecting the pyramidal neuron cells and estimating their orientation. More data may be available during the thesis work, which will be examined if time permits.

Tasks

Tasks that are to be done during the thesis project will include

1. Registration of the consecutive sections that makes up the 3D microscopy images of the brain tissues.
2. Segmentation of the registered consecutive sections that makes up the 3D microscopy images of the brain tissues.
3. Annotation of the neuron cells in the consecutive sections to distinguish pyramidal neuron cells from other neuron cells. As an experiment the annotation will be attempted by doing clustering of shape descriptors based on spherical harmonics. Manual annotation will be done, if the experiment is unsuccesful.
4. Experimental determination of whether Principal component analysis or spherical harmonics analysis are more appropriate for analyzing (assumed) pyramidal structures. As pyramidal structures are pressumably not linear, it is expected that spherical harmonics analysis will be more appropriate.
5. Do the two above mentioned analyses on the shapes of the detected pyramidal neuron cells.
6. Doing statistics involving the volume, surface area and orientation of the detected pyramidal cells.
7. Evaluating the statistics.

Status and schedule

Status for the thesis is that all code needed for task 1-2 and 4-6 are written and tested. Code for the annotation task has been written, but contains an error that are to be fixed. Registration and segmentation code has been executed for 50 serial images.

To complete task 3, the error in the code for the annotation experiment will need to be fixed. If the experiment is unsuccesful, manual annotation will need to be done instead.

To complete task 4-6, the code for shape analysis and statistics must be run at the CPU cluster available at DIKU image section. The code for has been previously run using all detected neuron cells to confirm that the code works correctly.

All parts of the report that are not about the results of task 3-6 are done. A revised and more detailed schedule for the thesis project can be found in table **??**.

Date	**Activity**
4-7/12-18	Debugging of annotation code
9-12/12-18	Execution of annotation experiment
13-15/12-18	Adding of annotation experiment results to report
17-21/12-18	Manual annotation, if annotation experiment is successful
27-30/12-18	Principle components analysis of pyramidal neuron cells
2-5/1-19	Adding of principle components analysis results to report
7-11/1-19	Spherical harmonics analysis of pyramidal neuron cells
18-21/1-19	Adding of spherical harmonics analysis results to report
22-24/1-19	Comparison of shape analysis results to answer task 4
25-28/1-19	Doing statistics by execution of statistics code
29/1-2/2-19	Adding statistics and statistics evaluation to report
3-12/2-19	Finishing report

12.2 A2: Code

Code for this thesis project is available at github:
`https://github.com/TinyPumpkin92/master_thesis`.
Note that the repository is subject to change.