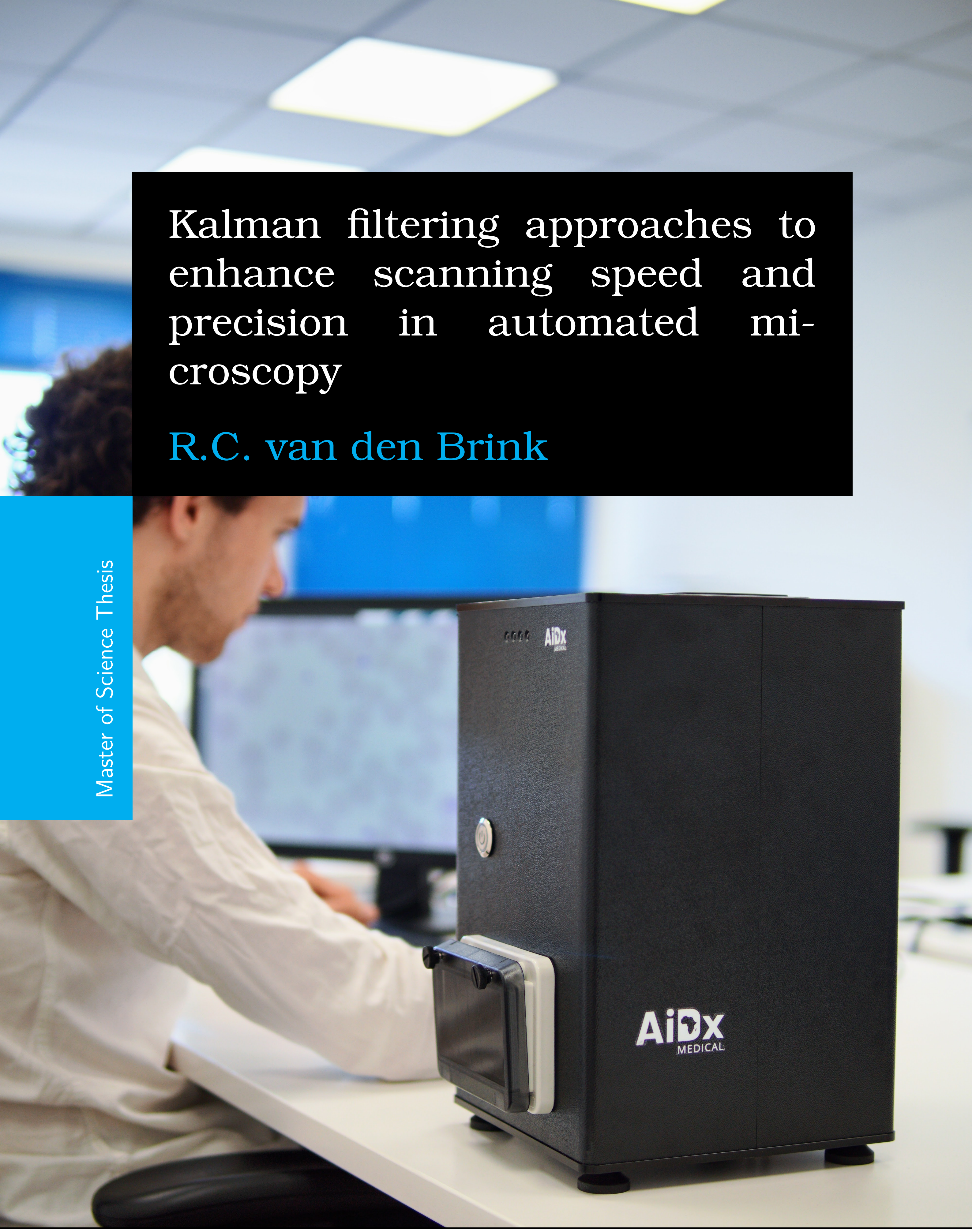
Kalman filtering approaches to enhance scanning speed and precision in automated microscopy

R.C. van den Brink

Master of Science Thesis



Delft University of Technology

Delft Center for Systems and Control

Kalman filtering approaches to enhance scanning speed and precision in automated microscopy

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

R.C. van den Brink

June 21, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by AiDx Medical bv.

Abstract

Fast and accurate diagnosis of illnesses or other health complications is not accessible in many locations around the world. Due to this, illnesses are unnecessarily left untreated. Therefore, AiDx Medical has developed a portable automated diagnostic microscope with reliable and rapid AI-assisted detection specifically for low-resource settings.

Scan speed is a key issue for the popularisation of whole slide imaging systems [1]. To address this issue a recent paper by [29] proposed a Kalman filter-based scanning algorithm. This approach eliminates the necessity for focus map generation prior to scanning and in doing so, reduces the scan time. Importantly, the proposed approach requires no additional hardware, and is more robust to noise.

In this thesis, two modifications to this work are proposed. Firstly, higher order process models are used to generate more precise estimates of the best-in-focus positions. Secondly, a two-dimensional Non-Symmetric Half Plane Kalman filter is developed to incorporate neighbouring state estimates in the prediction – an approach previously thought inapplicable for this purpose [24]. In a simulation, the new scanning algorithms are applied to scan thin smear malaria specimens and compared to state-of-the-art focus map surveying procedures.

Table of Contents

List of Figures

List of Tables

Preface

This is report is a part of my graduation project for the degree of Master of Science in Systems and Control at the Delft University of Technology. The idea of doing my thesis at AiDx Medical came after being inspired by the work of my supervisors dr. ir. Gleb Vdovin and dr. ir. Temitope Agbana. Both men have dedicated a lot of their resources and time to fighting for a better, healthier future for people in developing countries. Contributing to that mission has been an amazing opportunity for me.

At the start of my thesis I was asked to help bring AiDx's automated microscope to the next level. Our goal was to obtain in focus scans of diseases such as malaria, tuberculosis and urinary tract infections. After investigating this problem I found this issue could be split into three sub-problems, namely 1) achieving the required mechanical resolution, 2) determining the best image based focus algorithm to find the best-in-focus plane and 3) selecting a scanning algorithm with the best speed and accuracy. I am proud to be able to state that all three problems have been solved. AiDx Medical is now able to make in focus images of malaria.

As the title of this report suggest this document is centered around the last subproblem. This issue was found to have the most academic relevance and does not contain company sensitive or intellectual property issues. The work throughout this academic year has however encompassed a much broader scope. Milestones of my work that are not discussed in this report are:

- Reducing the scan time for low magnification scan systems from 30 to 4 minutes.
- Implementing an approach to circumvent the actuator backlash.
- Improving the focus metric accuracy from 26.17 μm to 0.97 μm (see appendix A)
- Implementing a Fibonacci search method to dynamically find the best-in-focus position
- High precision thermal actuator design, control and system identification development

The research has touched on topics ranging from material science to optics, and from lab work to filtering theory. This allowed me to use just about all the tools that the university has thought me throughout my time as a student. It made the experience all the more enjoyable, challenging and interesting.

Acknowledgements

First of all, I would like to thank the AiDx Medical team for allowing me to join their mission this year. It has been inspiring to see what a small group of people can achieve when they set their minds on a mission. In this case: stopping unnecessary suffering because of inaccessible diagnosis. Your passion to help those in need, combined with the technical knowhow and drive will continue to inspire me.

I would also like to thank my supervisors, Prof. dr. ir. Michel H.G. Verhaegen and Dr. ir. Gleb Vdovin for giving me the freedom to explore my interests and the opportunities I saw to both improve the performance of AiDx's device and make a scientific contribution. I was always surprised by the number of ideas you would come up with during our meetings. Thank you for pushing and inspiring me to embrace the scientific challenges in this project. I will cherish that mentality.

A special thanks goes out to my friends, family and housemates and above all Jip. Thank you for supporting me when I got stuck and enduring at times probably endless seeming jabbering about focus metrics, Kalman filtering and other interesting puzzles. Without your support I would not have come this far.

As I reflect on this journey, I am filled with optimism for the future. The experiences gained and the lessons learned alongside the AiDx Medical team have equipped me with the confidence and determination to continue making a positive impact. I am grateful beyond words for the collective support and inspiration that has propelled me forward.

Delft, University of Technology R.C. van den Brink
June 21, 2023

"Breng ieder uur een woord, een daad, die voor de wereld iets achterlaat."
— *Jacques van Marken*

Chapter 1

Introduction

Fast and accurate diagnosis of illnesses or other health complications is not accessible in many locations around the world. Due to this illnesses are unnecessarily left untreated. This leads to loss of quality of life and premature deaths on a large scale. In order to improve world wide health and well-being and reduce inequality, accessible healthcare is vital. This starts with accessible diagnosis.

Microscopy is commonly used to analyse pathology slides for diagnosis of illnesses and to better understand disease processes. Due to the limited Field of View (FoV) of a microscope several images (commonly referred to as frames) have to be surveyed in order to achieve the required sensitivity for diagnosis. The number of frames that have to be surveyed depends on the system magnification and the infection rate of the sample.

To diagnose parasitic diseases such as malaria, the microscope has to be able to resolve features with a radius of 0.7 - 0.9 μm. This requires using an objective lens with a numerical aperture of at least 0.4. Increasing the numerical aperture of the system implies an increase in the magnification of the objective lens. Additionally, the Depth of Field (DoF) decreases. When the DoF is smaller than the variations in the surface topography inherent to a particular specimen, the working distance of the objective lens (the distance between the objective lens and the specimen) has to be changed in every position to obtain an in focus image. The tracking of the specimen topography with the objective lens is commonly referred to as the autofocusing problem. Increasing the magnification additionally reduces the FoV of the system. Hence, to achieve the required sensitivity up to hundreds of frames may have to be acquired.

Currently, microscopes are still commonly operated manually by pathologists. Focusing performed manually using the focus knob of the microscope platform. This traditional slide reviewing process remains the gold standard in diagnosing. There are several issues inherent to manual operation of the microscope for the diagnosis of different illnesses. Being that it is time consuming, labour intensive, highly subjective and sensitive to human errors [3]. Therefore, AiDx Medical has developed a portable automated diagnostic microscope with reliable and rapid AI-assisted detection specifically for low-resource settings.

Acquisition of high-quality, in-focus images at high speed is a fundamental challenge in automated microscopy and a key issue for the popularisation of Whole Slide Imaging (WSI) to be realised [1] [3]. It currently takes state of the art systems 9 minutes to 2 and a half hours to scan a 15 × 15 mm area with a 40x magnification objective [15], depending on the system used. To address these issues Zhang et al. [29] proposed a Kalman filter-based method that predicts the best in-focus position of the next tile based on previous measurements and historical data. This eliminates the need to sample the focus height at positions along the sample for focus map generation prior to scanning. In doing so it reduces the scan time. Importantly, the proposed approach requires no additional hardware, and performs well on a low-precision platform [29]. The latter provides opportunities to save costs on the system hardware.

In this thesis, we propose two modifications to the work by Zhang et al. [29]. Firstly, higher order process models are used in the Kalman filter to generate more precise estimates. Secondly, an attempt is made to incorporate neighbouring state estimates in the state prediction using a two dimensional Non-Symmetric Half-Plane (NSHP) Kalman filter to improve the prediction accuracy. This method was previously disregarded for this purpose by Wang [24]. The new scanning algorithms will be applied in simulation to scan thin smear malaria specimens using serpentine scanning directions on a AiDx HR device. The methods are compared to state of the art focus map surveying procedures.

1-1 Report outline

The report is split into three chapters. Each chapter is written using a similar structure. First, one or multiple (variations of) scanning algorithms are described. Than the algorithms are used to filter the best in focus positions from the measurement data. Lastly, the obtained results are described and discussed. In summary:

- Chapter 2 describes and evaluates the state of the art of scanning algorithms for image based autofocus techniques.
- Chapter 3 describes and evaluates the Kalman filter based scanning approaches proposed by Zhang et al. [29] and Wang [24]. Additionally, other state space models and adjustments to the filter algorithm are proposed and evaluated.
- Chapter 4 describes and evaluates how the Non-Symmetric Half-Plane (NSHP) based Kalman filter can be used to incorporate neighbouring state estimates to improve the predicted state.
- The final chapter (chapter 5) summarises the findings of this thesis and proposes directions for further research.

Chapter 2

Autofocusing and scanning methods

As was described in the introduction the autofocusing systems have to track the specimen topography to create in focus images. To do so, the best-in-focus height has to be determined at a each position of the to be scanned area. Many techniques have been developed for this purpose. These techniques are commonly referred to as autofocusing approaches. The autofocus approach used in this thesis is described in the first section of this chapter. The following section describes different scanning methods that can be employed using the chosen autofocusing approach. Finally, a comparison is made between the autofocusing methods, evaluating their accuracy and the number of sampling points needed to reconstruct a height map of a malaria sample.

2-1 Autofocusing approaches

There are two types of autofocus systems; active and passive. Active systems irradiate light from a dedicated autofocus light source onto the specimen and focus based on the returned light. Passive systems determine the best-in-focus position solely based on the acquired images. This is also commonly referred to as the image contrast method. For the interested reader, an excellent overview of the the state of the art of different autofocusing techniques in 2020 is given in the literature review by Bian et al. [3].

A disadvantage of passive autofocus approaches is that the working distance (the distance from the objective lens to the sample) must be changed to detect an increase or decrease in contrast of image. Because multiple frames have to be analysed in order to determine whether the system is in focus, this approach is slower than the active approach. Passive approaches do however require fewer components and hence are relatively inexpensive [25]. As this thesis focuses on realising an economic and reliable scanning solution only passive, image based autofocus techniques that require no additional hardware will be discussed.

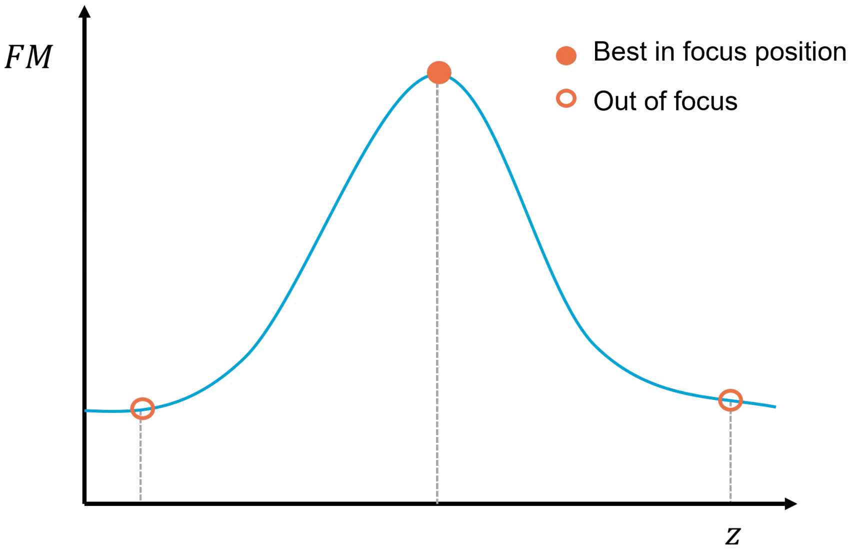


Figure 2-1: Traditional axial scanning procedure for autofocusing. A focus curve is computed by determining the focus metric values of a z-stack in a particular position.

2-2 Best-in-focus position computation

To determine the best-in-focus position of the objective lens along the z-axis at a particular coordinate, multiple images are acquired at different heights. Such a collection of images is called a z-stack. Using a focus metric, the quality of each image in the stack is computed. Plotting the focus metric values versus the position of the objective lens produces a focus curve that resembles the sketched plot in Figure 2-1. Ideally, the best-in-focus-position corresponds to the maximum of the focus curve.

Focus metrics exhibit various forms and variations, each rooted in distinct operational principles. The choice of a particular focus metric relies on factors such as the computational resources available, the optical configuration, and the unique attributes of the specimen. A comprehensive account of the focus metric selection process and the preprocessing techniques employed to achieve the findings presented in this report can be found in Appendix A.

Image fusion If the specimen under analyses contains 3D structures, such as stacked cells, that are larger than the DoF it will not be possible to select a single plane in which the entire image is in focus. To create an in focus image of all the elements of interest in one FoV image fusion techniques have been developed. In these methods all the in focus sections from multiple images along the optical axis are compiled in a single image.

Depending on the DoF of the optical system and the thickness of specimen, multi-focus fusion methods can require tens or even hundreds of partially in focus images to generate an all-in-focus image [13]. If the z-stack is undersampled, multi-focus fusion methods cannot remove the defocus blurs completely [30].

2-3 Scanning algorithms

The previous section described how the best-in-focus position of the objective lens can be determined. In this section the different methods that can be used in order to scan the area efficiently using a passive, image based autofocus technique are described and compared.

2-3-1 Autofocus on each tile

The most trivial scanning procedure is to determining the best-in-focus height by acquiring a z-stack in each position using a fixed scan range. The range along the z-axis in which the images are obtained can be based on the historical data to ensure the best-in-focus position is always included. This approach is sensitive to measurement noise. Therefore it requires a highly accurate Focus Metric (FM) which may have a larger computational burden. Additionally, the method does not make a prediction of the best-in-focus position based on previous measurements. Hence, a relatively large z-stack has to be acquired to ensure the best-in-focus position is included in the stack. This increases the scan time of the procedure.

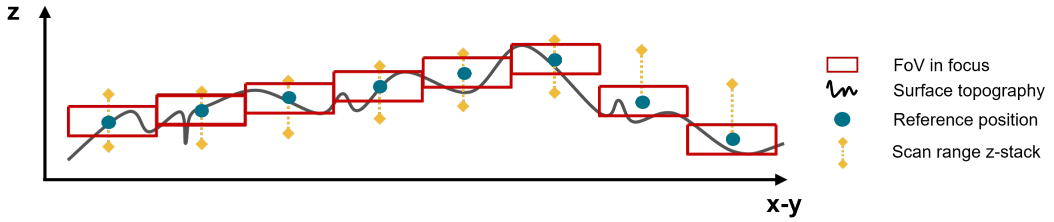


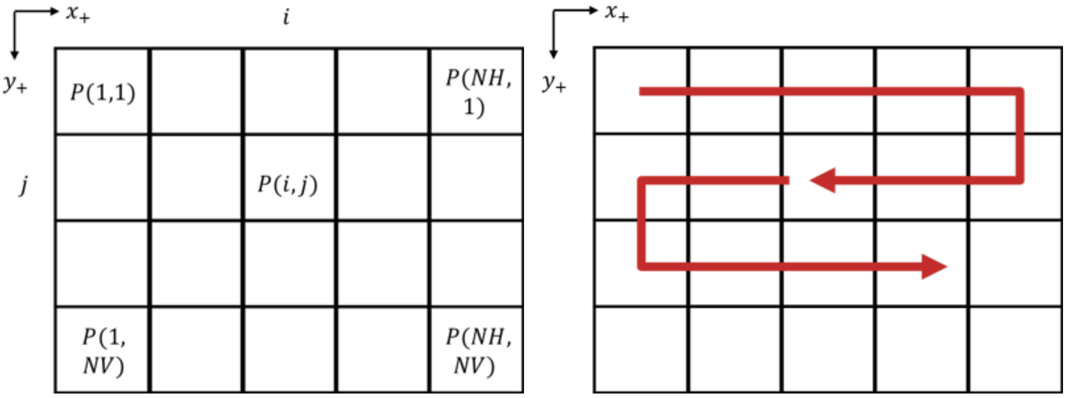
Figure 2-2: Propagation of the autofocus on each tile algorithm using across a cross section of a sample.

Scanning the grid can be performed in different ways. When performing a scan the grid coordinates and scanning direction as specified in Figure 2-3 will be used. In this thesis bidirectional scanning, also called serpentine scanning, will be used (see figure 2-3b). This scanning method is characterised by a back-and-forth or zigzag movement resembling the movement of a snake.

Focus map of a thin smear malaria sample

To investigate the topography characteristics of a thin smear malaria sample and generate simulation data a scan was made using the autofocus on each tile approach. The resulting focus map is displayed in figure 2-4. Three potential sources of topography variations are (1) the substrate supporting the specimen is not flat [18], (2) inaccuracies in the positioning of the actuators and (3) existing variations in solid tissue samples. Regarding the latter, it is given by Montalto et al. [12] that the topography variations in solid tissue samples can range from nanometres to several micrometers over a single millimetre in the x or y direction. This corresponds to what is observed in the obtained height map in figure 2-4.

The focus map shown in figure 2-4 will be used to generate test data for the simulated comparison of the scanning algorithms. The test data is generated as follows. First, it is



(a) Reference grid

(b) Serpentine scanning direction

Figure 2-3: Reference grid displaying the frame coordinates used and scanning directions used in the report.

assumed that sudden irregularities in the focus map are due to noise. Therefore, the focus map is smoothed using a Gaussian convolution filter ($\sigma = 1$). This smoothed map will function as a ground truth. The measurement data is simulated by adding normally distributed noise ($\sigma = 1.5\ \mu m$) to the measurement. The 80x80 map is split into 16 20x20 maps. This is the size that is required in order to diagnose malaria with a sensitivity that is approximately equal to that of a rapid diagnostics test.

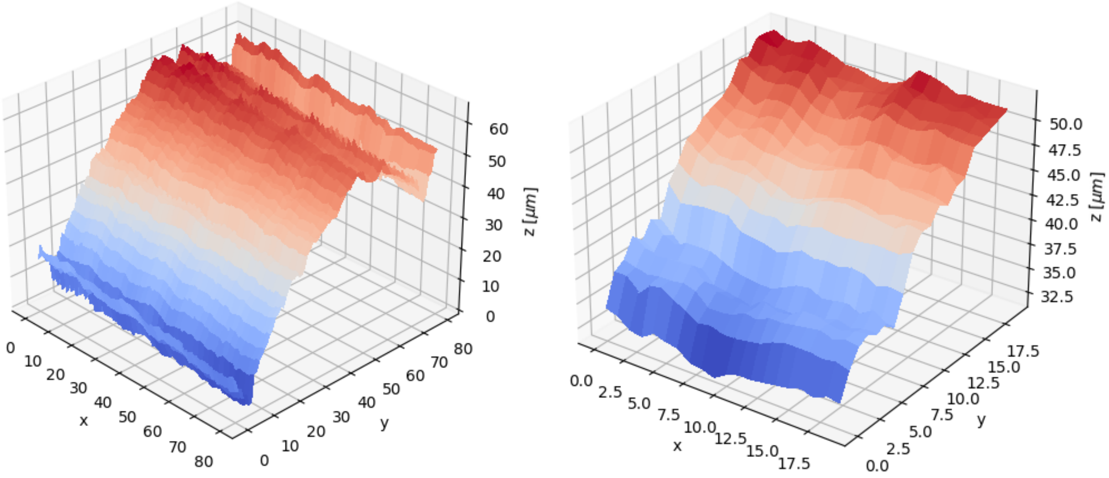
2-3-2 Tile skipping approach

The approach originally implemented on the system designed by AiDx is the called the tile skipping approach. In this method the best-in-focus position is determined through two steps. First, a z-stack is acquired and the best-in-focus position is determined by finding the position at which the FM is maximised. For the following n positions the best-in-focus position is assumed to stay constant. After the n positions have been scanned the procedure is repeated.

This approach assumes that the specimen topography alterations remain smaller than the depth of field of the camera in a range of n times the FoV dimensions. The number of frames, n, naturally depends on the DoF of the system and specimen specific characteristics. The assumption underlying this approach may hold for low magnification objective lenses with a large DoF but not for a 60x magnification objective lens with a depth of field of 1 μm.

2-3-3 Focus map surveying

According to Bian et al. [3] focus map surveying is the most adopted solution to the autofocus problem in commercially available WSI systems. In this approach a map containing the estimated best in focus positions is created prior to scanning the entire slide. The focus map is constructed through the following three steps, as depicted in Figure 2-6. First, a distribution of points in which the slide needs to be sampled is chosen. These may correspond to a regular grid with a certain spacing, a set of positions specified by the operator or a set chosen using



(a) 80x80 tiles

(b) 20x20 tiles (section from 2-4a)

Figure 2-4: Focus map of a thin smear malaria sample with a 1% infection rate made using the normalised variance as a focus metric and an average step size of 0.19 μm.

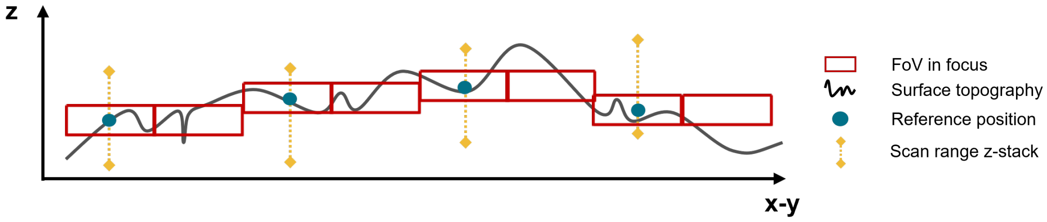
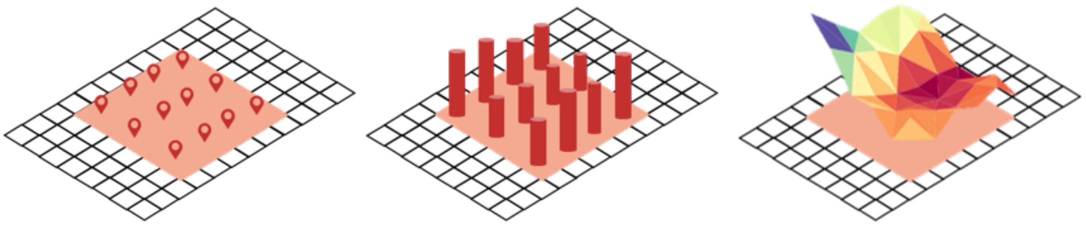


Figure 2-5: Propagation of the tile skipping algorithm using across a cross section of a sample.

another method. Secondly, z-stacks are acquired at each of the preset sampling positions. Using a FM the best-in-focus position is determined from the z-stack. Lastly, the computed best-in-focus positions are interpolated to construct a focus map. After the focus map has been constructed the entire prespecified area can be scanned. In each position the objective lens is moved to the height that is drawn from the interpolated focus map.

The focus map surveying approach has gained favour due to its simplicity, compatibility with various sample types, and minimal intellectual property concerns. By interpolating between the sampled positions, the method does not necessitate gathering a z-stack in each position and hence enables faster scanning.

Despite the advantages outlined above, there are several limitations that must be considered. Firstly, the procedure requires visiting positions all along the to-be-scanned area prior to scanning. This additional step prolongs the overall scanning time compared to methods that eliminate the need for revisiting the scanning area. Furthermore, the procedure lacks the capability to adjust the focus map based on information obtained during the scanning procedure. As a result, the focus remains static throughout the scanning process, potentially resulting in sub optimal image quality in certain regions. Another important consideration is that these methods assume that the sampled best in-focus heights are accurate. This assumption may not always hold true, as various sources of noise can affect the accuracy of the



(a) Focus point distribution

(b) Best-in-focus positions

(c) Interpolated focus map

Figure 2-6: The procedure for generating a focus map presented in chronological order. The sample slide is divided into distinct fields of view, depicted by the grid. The highlighted area on the grid indicates the specific section of the slide that is covered by the specimen.

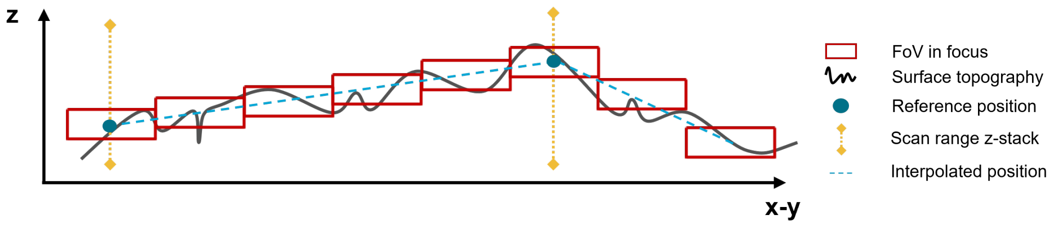


Figure 2-7: Propagation of the focus map surveying algorithm across a cross section of a sample.

obtained data. Lastly, not obtaining a z-stack in each position could also be a disadvantage, despite increasing scanning speed. The approach impedes image quality enhancement through fusion techniques.

The interpolation method used during focus map surveying depend on surface specific characteristics. In the following paragraphs different interpolation methods are described.

Bilinear interpolation This technique is used to interpolate values at points that lie inside a rectangle, given the values at the four corners of the rectangle. The method involves estimating the value at the point of interest by interpolating linearly along the x-axis and then linearly along the y-axis, using the values at the four corners of the rectangle. Bilinear interpolation assumes a linear change in the values of the function being interpolated over the rectangle. The interpolation method is implemented in python using the following method from SciPy [22]: `scipy.interpolate.RegularGridInterpolater(method = 'linear')`.

Spline interpolation Spline interpolation is a method used to interpolate data points by fitting piecewise polynomial functions to the data. The method works by dividing the domain of the function into smaller intervals, and fitting a polynomial function to each interval. The polynomial functions are chosen to be smooth, meaning that their first and second derivatives are continuous across the intervals. This results in a smooth curve that passes through all the data points. This interpolation method is implemented in python using the following method from SciPy [22]: `scipy.interpolate.RegularGridInterpolater(method = 'cubic')`.

Delaunay based interpolation Delaunay interpolation is a typical method for generating focus maps [3]. This technique involves constructing a Delaunay triangulation of the input data points, which is a triangulation where no point is inside the circumcircle of any triangle. Once the Delaunay triangulation is constructed, the value at a point of interest is interpolated by computing the barycentric coordinates of the point with respect to the triangle that contains it. These barycentric coordinates are then used to interpolate the values of the function at the vertices of the triangle. This interpolation method is implemented using `scipy.spatial.Delaunay()` to obtain the Delaunay triangulation and `scipy.interpolate.LinearNDInterpolator()` to interpolate along the triangulation [22].

2-3-4 Kalman filter based scanning

Focus map construction prior to scanning requires revisiting positions to image the entire slide. This adds a significant time burden to the scan procedure. To address this [29] proposed a Kalman filter-based method that predicts the best in-focus position of the next tile based on previous measurements and historical data. To ensure an image of the best in-focus position is made at each position, a z-stack of images in a range around the predicted in focus position can be collected. The acquired z-stacks of images can be used for multi-focus image fusion and other post processing steps to obtain a higher quality images. As was mentioned in the introduction, another important advantage of this approach is that it is less sensitive to measurement noise [29].

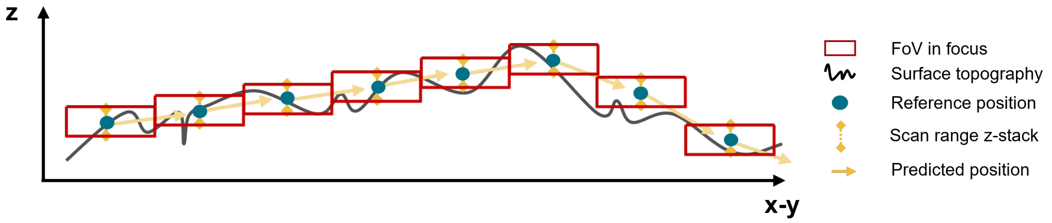


Figure 2-8: Propagation of the Kalman filter based scan algorithm across a cross section of a sample.

In this report, two modifications to the work by Zhang et al. [29] are proposed. Firstly, [29] utilised a zeroth-order Taylor expansion as a process model in their Kalman filter. This model is based on the assumption that, on average, the best prediction is made by assuming that height stays constant. Or in other words, it is assumed the to be scanned region is best modelled as a flat surface with some slight variations that are best captured by white noise. The first addition of this work is using higher order models to attempt to generate more precise estimates. This modification will be discussed more elaborately in chapter 3

Secondly, an attempt is made to improve the accuracy of the filter by incorporating neighbouring state estimates in the process model. To achieve this, the two dimensional Kalman filter models that were primarily developed for image processing tasks by Woods and Radewan [28] and Woods and Ingle [27] are extended to suit the application in this thesis. This modification is discussed in chapter 4.

2-4 Comparison of conventional scan algorithms

In this thesis the algorithms that are not the Kalman filter based algorithms will be referred to as the conventional scan algorithms. The accuracy achieved with these methods will function as a baseline to which the other methods will be compared.

The comparison of the conventional approaches was determined by applying them to the 16 measurement maps described in section 2-3-1 and comparing the resulting focus maps to the corresponding ground truths. The accuracy was computed as the mean of the difference between the ground truth and the constructed focus map in each position. The number of sampling points used was varied to check if this influenced the performance. The results are plotted in figure 2-9. The number of sampling points that give the lowest mean error are given in table 2-1 for each conventional approach.

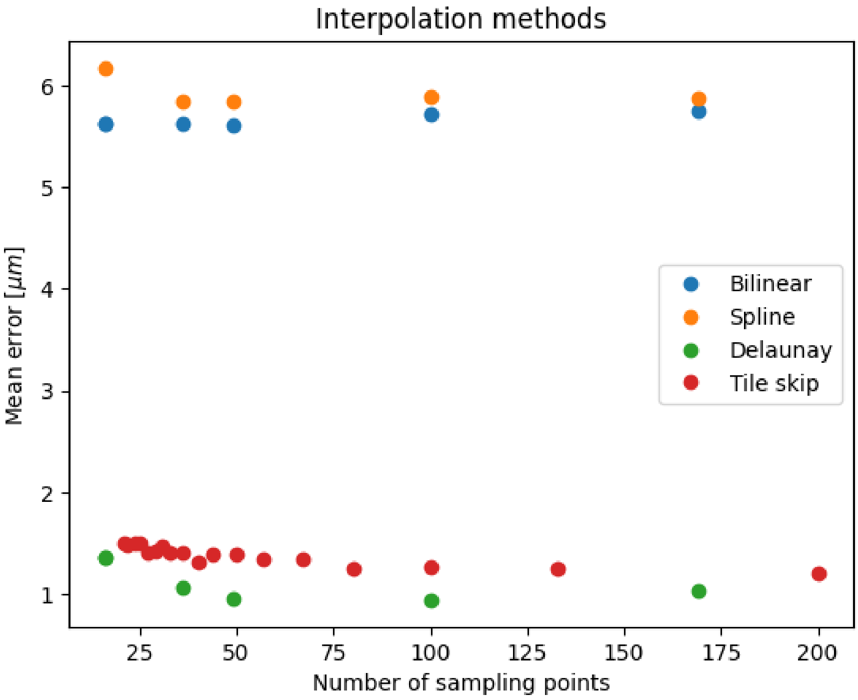


Figure 2-9: Mean positioning error achieved using different conventional focus map construction approaches.

Table 2-1: Error of the tested focus metrics

Scan algorithm	Tile skipping	Bilinear int.	Spline int.	Delaunay based int.
Mean error	1.21 μm	5.61 μm	5.84 μm	0.94 μm
Std. error	1.51 μm	6.96 μm	7.20 μm	1.17 μm
Number of points	200	49	36	100

It is found that Delaunay based interpolation outperforms the other methods. The Delaunay triangulation has some desirable properties for interpolation of data with abrupt changes. One of these properties is that it is locally optimal, meaning that it produces the triangulation that maximises the minimum angle of the triangles. In contrast, bilinear or spline interpolation rely on a regular grid of points to interpolate values at arbitrary locations. These methods do not perform well when the input data is sparse or irregularly spaced. Bilinear interpolation assumes a smooth variation of the data between the neighbouring grid points, which may not be true if the data has abrupt changes or discontinuities. The same holds for the tile skip approach. Spline interpolation can produce oscillations or artefacts if the input data has sharp features.

2-5 Conclusion

In summary, it was chosen to focus this thesis on passive autofocusing approaches because they do not require any additional components and are therefore relatively inexpensive. It was found that of the focus map surveying approaches Delaunay based interpolation outperforms the other methods with regards to both the scanning accuracy and the required number of sampling positions. Lastly, Kalman filter based scanning procedures were found to give promising results to improve both scan speed and accuracy.

Chapter 3

Kalman filtering

The Kalman filter scanning algorithm was previously mentioned in section 2-3-4. This chapter gives a closer look at the different steps of this algorithm. Figure 3-1 gives an overview of the steps in the process required to obtain a best-in-focus image. The automated microscope, referred to as 'plant' in figure 3-1, acquires a z-stack of images at position k. From this z-stack the best-in-focus position z_k is computed using a FM. The computed best in focus position is fed to the Kalman filter as the system measurement. The Kalman filter performs two tasks. Firstly, it updates the previous state prediction using the obtained measurement. The obtained state estimate is used to select an image from the z-stack at position k. Secondly, it makes a prediction of the next best-in-focus position, $\bar{x}_{k+1}$, and computes the corresponding covariance matrix $\bar{P}_{k+1}$ based on the the previous state and covariance matrix estimates. The z-stack in the next position is acquired in a range centered around the predicted best-in-focus position. The size of the range is equal to some factor times the predicted covariance matrix. In doing so, as the covariance converges and the Kalman filter becomes more certain of its prediction the scan range, and scan time will decrease.

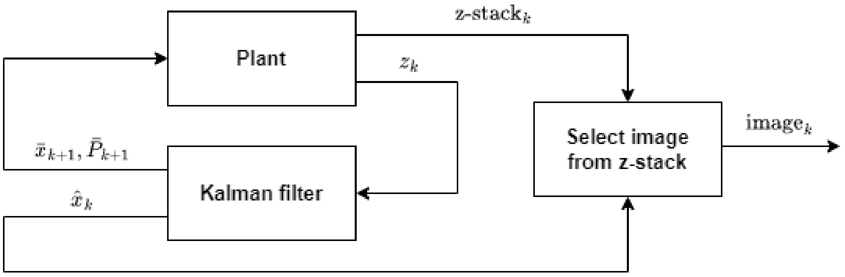


Figure 3-1: Block diagram displaying the Kalman filter based scanning concept

The goal of the Kalman filter is twofold, namely 1) minimising the scan time and 2) obtaining the best-in-focus image. Therefore, both the predicted and the updated state estimate should be as close to the best-in-focus position as possible.

3-1 Kalman filter theory

The Kalman filter finds a minimum-error variance estimate of the state $x(k)$ of a stochastic linear model from measurements of the modelled system. The discrete stochastic linear models considered in this thesis will be described by the following equations:

$$x(k+1) = Fx(k) + w(k), \qquad w(k) \sim \mathcal{N}(0, Q) \tag{3-1}$$

$$y(k) = Hx(k) + v(k), \qquad v(k) \sim \mathcal{N}(0, R) \tag{3-2}$$

where $x(k)$ is the state, F is the state transition matrix, $y(k)$ the output, H the measurement function, $w(k)$ the process noise described by the covariance matrix Q and $v(k)$ the measurement noise described by the covariance matrix R. The steps of the Kalman filter are summarised in algorithm 1. The two main steps of the algorithm will briefly be described below.

First, the predict step is performed by computing the prior state, $\bar{x}$ (line 4-6, algorithm 1). This is computed using the dynamical model to propagate the state to the next position. The state transition matrix and covariance matrix are used to compute the corresponding prior covariance matrix, $\bar{P}$. Secondly, an update step is performed (line 7-11, algorithm 1). In this step the Kalman gain is used to compute an estimate of the state, $\hat{x}$, and the corresponding covariance matrix, $\hat{P}$. Put simply, the Kalman gain is used to compute a weighted average between the prior and the measurement based on their respective covariance matrices. The covariance matrices Q and R can be used to tune the algorithm to trust the prediction over the measurements or vice versa. A more elaborate description of the Kalman filter has been published by Verhaegen and Verdult [21] and 'Labbe [8]. In the remainder of the report the (k) notation which indicates the time index will be dropped for brevity of notation. The considerations for the chosen model and filtering parameters will be described more elaborately in the following sections.

Algorithm 1 Kalman filtering

```
 1: procedure KF($zs, x_0, P_0, R, Q, dx$)
 2:     $\hat{x}, \hat{P} = x_0, P_0$                    ▷ Initialise variables
 3:     for $z$ in $zs$ do
 4:         Predict step
 5:         $\bar{x} = F\hat{x}$                          ▷ Prior state
 6:         $\bar{P} = F\hat{P}F^T + Q$                   ▷ Prior cov. matrix
 7:         Update step
 8:         $r = z - H\bar{x}$                            ▷ Residual
 9:         $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$        ▷ Kalman gain
10:         $\hat{x} = \bar{x} + Kr$                      ▷ Posterior state
11:         $\hat{P} = (I - KH)\bar{P}$                   ▷ Posterior cov. matrix
12:     end for
```

3-1-1 Assumptions

The Kalman filter assumes the discrete time model given by equation 3-1 describes a Gaussian random process. This is the case when all the probability density functions describing the process are normal in form. Gaussian random processes have several desirable properties. First of all, Gaussian random variables can be fully described by their mean and variance. Hence, the mean state and covariance matrix computed in both the predict and the update step of the Kalman filter fully describe the random variables in the state. Secondly, the output of a linearly transformed Gaussian process is also Gaussian. These two properties simplify the mathematical analysis and computational complexity significantly. Therefore, the Kalman filter requires the process model to be linear.

The superposition of independent random variables always tends towards normality regardless of the distribution of the individual random variables contributing to the sum. This is proven in the central limit theorem. Because noise is often due to a superposition of many small contributions the assumption of normality can be applied in many engineering applications.

In order for the Kalman filter to be implemented recursively it requires that the model describing the system dynamics and its uncertainties is a Markov sequence. This means that the model has the following two properties:

Property 1. ***Markov property of states*** *(Sarka [17]) The states $x_k : k = 0, 1, 2, ...$ form a Markov sequence (or Markov chain if the state is discrete). This Markov property means that x_k (and actually the hole future x_{k+1}, x_{k+2}, ...) given x_{k-1} is independent of anything that has happened before the time step $k - 1$:*

$$p(x_k|x_{1:k-1}, y_{1:k-1}) = p(x_k|x_{k-1}) \tag{3-3}$$

Also the past is independent of the future given the present:

$$p(x_{k-1}|x_{k:T}, y_{k:T}) = p(x_{k-1}|x_k) \tag{3-4}$$

Property 2. ***Conditional independence of measurements*** *(Sarka [17]) The current measurement y_k given the current state x_k is conditionally independent of the measurement and state histories:*

$$p(y_k|x_{1:k}, y_{1:k-1}) = p(y_k|x_k) \tag{3-5}$$

These properties enable circumvention of the explicit usage of the full Bayes' rule in which the number of computations per time step increases as new observations arrive. Using a Markovian model the number of computations per time step remains constant.

3-1-2 Process noise

The process noise captures uncertainties or variations in the system dynamics that are not accounted for by the model. In the case of surface modelling this could be due to modelling errors or unmodelled disturbances. More specifically, the former relates to inaccuracies or simplifications in the mathematical models used to describe the terrain dynamics. The latter relates to external factors that are not explicitly accounted for in the model. An example could be a sudden jump in the height because a particular malaria parasite happens to be in a different focus plane.

The process noise, represented by $w(k)$, is assumed to be a zero-mean white noise process. In the computation of the prior state estimate (line 5 of algorithm 1) this term is not added because the process noise is assumed to have a mean value of zero. Hence, on average the noise will have no effect on the state evolution during the prediction. The prediction of the next state, $x(k+1)$, is thus based on the system dynamics captured by the state transition matrix, F, and the current state estimate, $x(k)$.

The subsequent update step adjusts the state estimate based on the difference between the predicted and actual measurements, considering the uncertainty in the measurements and the state estimate. The uncertainty in the state estimate depends on covariance matrix of the estimate and the process noise covariance matrix Q. Therefore, the design of the process noise covariance matrix Q can have a significant influence on the filtered estimate.

As was previously stated, the process noise matrix Q represents the uncertainty or variance associated of state space description used to model the system dynamics. The elements of the matrix Q represent the (co)variances between the different state variables in the system. Note that as the process noise is assumed to have a zero mean the (co)variances are equal to the mean-squared values of the respective states. The covariance matrix for a state space model with two states is hence described by:

$$Q = \begin{bmatrix} E[x_1x_1] & E[x_1x_2] \\ E[x_2x_1] & E[x_2x_2] \end{bmatrix} \tag{3-6}$$

The system response of one of the state variables can be computed as the convolution of the impulse response function, $g(u)$, with the input function, $f(u)$, as follows:

$$x(t) = \int_0^l g(u)f(l-u)du \tag{3-7}$$

where l is the distance over which the integral is performed. Using this definition of the state response the mean-square value, $E[x_1x_1]$ can be computed as:

$$E[x_1x_1] = \int_0^l \int_0^l g(u)g(v)R_f(u-v)dudv \tag{3-8}$$

where $R_f(u-v)$ is the autocorrelation function corresponding to the input. If the input is unity white noise its autocorrelation function corresponds to the the Dirac delta function with power spectral density equal to one. Scaling the autocorrelation with amplitude A gives $R_f(x) = A\delta x$. As autocorrelation the value at $R_f(0)$ corresponds to the mean square value of the process and the mean of the process noise considered is zero mean, the mean square value will equal the variance of the process. Hence, the variance of the process will equal the amplitude of the autocorrelation function $A = \sigma^2$. Scaling the autocorrelation does not change the shape or distribution of the power across different frequencies. It does scale the amplitude of the PSD by the square of A.

Additionally, using the following property of the Dirac delta function:

$$\int_a^b f(x)\delta(x-x_0)dx = f(x_0) \qquad \text{if } x_0 \text{ belongs to the open interval} \tag{3-9}$$

the computation of the terms of the process noise covariance matrix can be simplified significantly.

3-2 The Kalman filter for Linear Space Invariant (LSI) systems

If the state space model used to model the system is space-invariant the Kalman filter converges to a stationary solution. The exact conditions under which the Kalman filter has a stationary solution are summarised in the following theorem:

Theorem 1. *(Anderson and Moore [2]) Consider the linear space-invariant system*

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{3-10}$$

$$y(k) = Cx(k) + v(k) \tag{3-11}$$

with $w(k)$ and $v(k)$ zero-mean random sequences with covariance matrix:

$$E \begin{bmatrix} w(k) \\ v(k) \end{bmatrix} \begin{bmatrix} w(k)^T & v(k)^T \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \Delta(k-j) \tag{3-12}$$

such that

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0, \qquad \textit{and} \qquad R > 0 \tag{3-13}$$

If the pair (A, C) is observable and the pair $(A, Q^{1/2})$ is reachable, then

$$P(k|k-1) = E\left[(x(k) - \hat{x}(k|k-1))(x(k) - \hat{x}(k|k-1))^T\right], \tag{3-14}$$

with $\hat{x}(k|k-1) = E[x(k)]$, satisfies

$$\lim_{k\to\infty} P(k|k-1) = P > 0 \tag{3-15}$$

for any symmetric initial condition $P(0|-1) > 0$, where P satisfies

$$P = APA^T + Q - (S + APC^T)(CPC^T + R)^{-1}(S + APC^T)^T \tag{3-16}$$

Moreover, such P is unique. If this matrix P is used to define the Kalman-gain matrix K as

$$K = (S + APC^T)(CPC^T + R)^{-1} \tag{3-17}$$

then the matrix $A - KC$ is asymptotically stable.

There exist several refinements of this theorem in which the observability and reachability refinements of this theorem are replaced by weaker noitions of detectability and stabilizability. A discussion of these refinements is outside the scope of this thesis.

3-3 State space models

3-3-1 Previous work

Zhang et al. [29] first proposed using a Kalman filter-based method scanning method for WSI systems. In the employed state space model in their system the state is a assumed to be linear plane. Because of this assumption the height was modelled as a random walk, such that the height of the next best-in-focus position is equal to the current best-in-focus position plus some noise. A more elaborate description of this process model and implementation will be given below. This method will be used as a baseline to compare the other Kalman filters to.

Wang et al. [23] proposed using a higher order state space model and combining information from two neighbouring positions. The paper is applied for digital terrain modelling. This relates very closely to the application the discussed in this thesis. The main difference lies in the scale that is considered.

The state space model proposed by Wang et al. [23] is defined as follows:

$$\begin{bmatrix} H(i,j) \\ H_x(i,j) \\ H_y(i,j) \end{bmatrix} = \begin{bmatrix} b(i,j) & b(i,j)dx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} H(i-1,j) \\ H_x(i-1,j) \\ H_y(i-1,j) \end{bmatrix} + \tag{3-18}$$

$$\begin{bmatrix} c(i,j) & 0 & c(i,j)dy \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} H(i,j-1) \\ H_x(i,j-1) \\ H_y(i,j-1) \end{bmatrix} + \begin{bmatrix} w_H(i,j) \\ w_{H_x}(i,j) \\ w_{H_y}(i,j) \end{bmatrix} \tag{3-19}$$

where (i,j) denotes the position as is shown in figure 2-3a, w_H, w_{H_x} and w_{H_y} are assumed as white sequences with known covariance structure $Q(i,j)$, dx and dy are the sampling intervals and $b(i,j)$ and $c(i,j)$ are blending factors with a summation of 1.

The observation of elevation is assumed to have a linear relationship with the state vector at the relevant position:

$$Z(i,j) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} H(i,j) \\ H_x(i,j) \\ H_y(i,j) \end{bmatrix} + v_z(i,j) \tag{3-20}$$

where $D =$ and $v_z(i,j)$ is a white sequence with known covariance structure R(i,j) and having zero cross-correlation with the $v_s(i,j)$ sequence.

There are three issues with the proposed model. First of all, and most importantly, the state space model is not observable if only the height in the to be updated position is measured (the test for observability can be found in appendix B). Due to this, the Kalman filter cannot be guaranteed to give a stationary solution. Secondly, using the proposed model does not provide a framework to include additional neighbouring positions in the estimate. Thirdly, the state space model does not propagate the neighbouring state estimate that is required to compute the prior.

3-3-2 Brownian-motion based process models

As can be seen in Figure 2-4 the height maps are not flat planes. They have a changing slope and show sudden variations related to the structure of the specimen. Therefore, the function that perfectly describes the slope along the surface is expected to be a specimen specific nonlinear function. Because the function is specimen specific it is not possible to know the exact function prior to scanning. Therefore, lower order linear models will be derived and evaluated with regards to their ability to capture the system dynamics.

In order to make a prediction of the height at the next scanning position an linear approximation of the nonlinear function will be made using a Taylor series expansion. The second order Taylor polynomial of a nonlinear function around a position, x_k, is given by:

$$H(x_{k+1}) = H(x_k) + \frac{H'_x(x)}{1!}(x_{k+1} - x_k) + \frac{H''_x(x)}{2!}(x_{k+1} - x_k)^2 + \mathcal{O}(dx^3) \tag{3-21}$$

where $(x_{k+1} - x_k)$ is the step size in the x-direction. From this point onward this will be denoted by dx. In the y-direction it will be denoted as dy. The step size is fixed in both directions. The primes are used to denote the order of the partial derivative. Hence, $H''_x(x)$ is the second order partial derivative with respect to the distance x. Lastly, $\mathcal{O}(dx^3)$ is the error term due to finite expansion. The order of the Taylor series corresponds to the number of terms used in the polynomial approximation. Zhang et al. [29] proposed using a 0^{th} order polynomial approximation. This thesis will additionally investigate 1^{st} and 2^{nd} order polynomial approximations.

In order to be able to compare the Taylor series based model to other state space models used to estimate the state dynamics, and for the ease of analysis of the model, the continuous state space model and corresponding block diagram have been derived. The block diagram of a second order Brownian noise model is given in figure 3-2. The corresponding continuous state space model is given by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \tag{3-22}$$

where x_1 is the best-in-focus height of the objective lens, x_2 the gradient, x_3 the curvature and $w(k)$ is the process noise. As is shown in 3-2 the process noise is assumed to be white. The lower order models can be derived by removing integrator blocks in from the block diagram and evaluating sub-matrices of the presented state space model. The model can be extended with higher derivatives by doing the opposite.

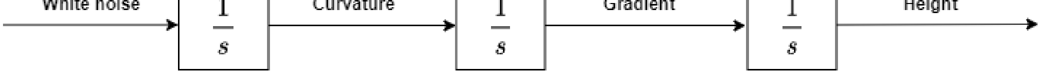


Figure 3-2: Block diagram depicting a Brownian noise second order model

In order to discretize the system the state transition matrix needs to be computed. Generally, this can be done via two methods, namely 1) by computing the inverse Laplace of $(sI - A)$ or

2) by determining the value of e^{Adx}. In this particular case the Taylor polynomials can also be used to determine the discrete state state space models. The discrete state space model is:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & dx & \frac{1}{2}dx^2 \\ 0 & 1 & dx \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(k) \tag{3-23}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} + v(k) \tag{3-24}$$

The lower order state space models can be computed in a similar and all correspond to submatrices of the state space model given above. Therefore, they will not be stated explicitly.

The different order state space models describe different dynamical processes. In this paragraph a short description of what the different order models describe is given. The zeroth order model corresponds to a Brownian-motion or Wiener process. Such a process can be characterised by an integrator driven by white noise. For such a process it can be shown that the variance after n steps is equal to n. This model is used by [29] to model the surface. The model assumes that the next position is equal to the current position with the addition of random fluctuations, characterised by white noise. Hence, this models describes flat surfaces where the deviations in the surface resemble white noise.

In the case of the first order model, where the gradient is simulated as a Wiener process, the position is obtained by integrating the random fluctuations of the gradient. The integration process accumulates these changes to determine the position at each point. In the Kalman filter the random fluctuations are not added to the process model. Hence, the model assumes that the surface is best described by a model with a constant slope. However, modelling is as a Wiener process does influence the process noise covariance matrix, Q. In doing so it allows larger changes during the correction of the gradient in the update step.

The second order state space model additionally incorporates an estimate of the curvature. A line for which the curvature is modelled as a Wiener process implies that the curvature of the line follows the statistical properties of a Wiener process. In this case the gradient is updated at each step using the state corresponding to the curvature term multiplied by the step size. This modelling approach allows fluctuations to the curvature, resulting in a surface with continuously changing curvature.

Determining the process noise covariance matrix, Q

The process noise covariance matrix Q corresponding to the second order state space model described above is derived in this section. The matrix is derived using the method described in section 3-1-2.

The relevant transfer functions are given by:

$$G(u \text{ to } x_1) = G_1 = \frac{1}{s^3}, \qquad G(u \text{ to } x_2) = G_2 = \frac{1}{s^2}, G(u \text{ to } x_3) = G_3 = \frac{1}{s} \tag{3-25}$$

The corresponding impulse response functions of the transfer functions are:

$$g_1 = \frac{1}{2}t^2 \qquad g_2 = t \qquad g_3 = 1 \tag{3-26}$$

Than using equation 3-8 and $R_f(x) = \sigma^2\delta(x)$ as the input autocorrelation function, the expected mean-square values can be computed to be:

$$E[x_2x_2] = \sigma^2 \int_0^{\Delta L} \frac{1}{4}v^4 dv = \sigma^2 \frac{1}{20}\Delta L^5 \qquad E[x_1x_1] = \sigma^2 \int_0^{\Delta L} v^2 dv = \sigma^2 \frac{1}{3}\Delta L^3 \tag{3-27}$$

$$E[x_1x_2] = \sigma^2 \int_0^{\Delta L} \frac{1}{2}v^3 dv = \sigma^2 \frac{1}{8}\Delta L^4 \qquad E[x_1x_0] = \sigma^2 \int_0^{\Delta L} v dv = \sigma^2 \frac{1}{2}\Delta L^2 \tag{3-28}$$

$$E[x_1x_0] = \sigma^2 \int_0^{\Delta L} \frac{1}{2}v^2 dv = \sigma^2 \frac{1}{6}\Delta L^3 \qquad E[x_0x_0] = \sigma^2 \int_0^{\Delta L} 1 dv = \sigma^2 \Delta L \tag{3-29}$$

This gives the following process noise covariance matrix Q:

$$Q = \begin{bmatrix} \frac{1}{20}\Delta L^5 & \frac{1}{8}\Delta L^4 & \frac{1}{6}\Delta L^3 \\ \frac{1}{8}\Delta L^4 & \frac{1}{3}\Delta L^3 & \frac{1}{2}\Delta L^2 \\ \frac{1}{6}\Delta L^3 & \frac{1}{2}\Delta L^2 & \Delta L \end{bmatrix} \sigma^2 \tag{3-30}$$

The process noise covariance matrices corresponding to the lower order models can be determined easily using the integrals above.

3-3-3 Gauss-Markov models

Obviously not all random processes are accurately modelled by a Wiener process. curvature for example is usually brief and not sustained. Therefore, a stationary Gauss-Markov process may be more appropriate than the nonstationary Brownian motion. Gauss-Markov processes have an exponential autocorrelation, $R_X(\tau) = \sigma^2 e^{-\beta|\tau|}$. The mean-square value and time constant for the process are given by the σ^2 and $1/\beta$ parameters, respectively. The exponential autocorrelation function indicates that sample values of the process gradually become less and less correlated as the separation between samples increases [4]. The block diagram corresponding to such a model is given in figure 3-3.

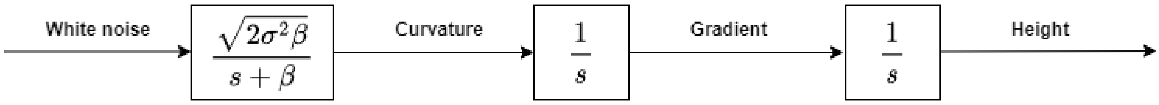


Figure 3-3: Block diagram of a second order model where the curvature is modelled as a Gauss-Markov process.

The continuous state space model corresponding to the depicted block diagram is given by the following equations:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & -\beta\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \sqrt{2\sigma^2\beta}\end{bmatrix} w \tag{3-31}$$

$$y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + v \tag{3-32}$$

The discrete state space model corresponds is given by the following equations:

$$\begin{bmatrix}x_1(k+1)\\ x_2(k+1)\\ x_3(k+1)\end{bmatrix} = \begin{bmatrix}1 & dx & \frac{1}{\beta^2}(\beta dx - 1 + e^{-\beta dx})\\ 0 & 1 & \frac{1}{\beta}(1 - e^{-\beta dx})\\ 0 & 0 & e^{-\beta dx}\end{bmatrix}\begin{bmatrix}x_1(k)\\ x_2(k)\\ x_3(k)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \sqrt{2\sigma^2\beta}\end{bmatrix} w(k) \tag{3-33}$$

$$y(k) = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1(k)\\ x_2(k)\\ x_3(k)\end{bmatrix} + v(k) \tag{3-34}$$

The lower order state space models can be derived easily by removing integral terms from the block diagram and computing the corresponding state space models.

Determining the process noise covariance matrix, Q

If the continuous system giving rise to the discrete situation has constant parameters and if the various white noise inputs have zero crosscorrelation, some simplification is possible such that the impulse response functions may be applied. A description of how the mean-square expectations are computed is given below.

The relevant transfer functions are given by:

$$G(u \text{ to } x_1) = G_1 = \frac{\sqrt{2\sigma^2\beta}}{s^2(s+\beta)}, \qquad G(u \text{ to } x_2) = G_2 = \frac{\sqrt{2\sigma^2\beta}}{s(s+\beta)}, \tag{3-35}$$

$$G(u \text{ to } x_3) = G_3 = \frac{\sqrt{2\sigma^2\beta}}{s+\beta} \tag{3-36}$$

The corresponding impulse response functions are:

$$g_1 = \sqrt{\frac{2\sigma^2}{\beta^3}}(e^{-\beta dx} + \beta dx - 1) \quad g_2 = \sqrt{\frac{2\sigma^2}{\beta}}(1 - e^{-\beta dx}) \quad g_3 = \sqrt{2\sigma^2\beta}(1 - e^{-\beta dx}) \tag{3-37}$$

Now we can use equation 3-8 to find the mean-square expectations:

$$E[x_1x_1] = \int_0^{\Delta L}\int_0^{\Delta L} g_1(u)g_1(v)\delta(u-v)dudv = \int_0^{\Delta L} g_1(v)^2 dv \tag{3-38}$$

$$= \frac{2\sigma^2}{\beta^3}(dx - \beta dx^2 + \frac{1}{3}\beta^2 dx^3 - \frac{1}{2\beta}(1-e^{-\beta dx}) - 2dxe^{-\beta dx}) \tag{3-39}$$

$$E[x_1x_2] = \int_0^{\Delta L}\int_0^{\Delta L} g_1(u)g_2(v)\delta(u-v)dudv = \int_0^{\Delta L} g_1(v)g_2(v)dv \tag{3-40}$$

$$= \frac{2\sigma^2}{\beta^2}(-dx + \frac{1}{2}\beta dx^2 + dxe^{-\beta dx} + \frac{1}{2\beta}(1+e^{-2\beta dx} - 2e^{-\beta dx})) \tag{3-41}$$

$$E[x_1x_3] = \int_0^{\Delta L}\int_0^{\Delta L} g_1(u)g_3(v)\delta(u-v)dudv = \int_0^{\Delta L} g_1(v)g_3(v)dv \tag{3-42}$$

$$= \frac{2\sigma^2}{\sqrt{\beta^5}}(-dxe^{-\beta dx} + \frac{1}{2\beta}(1-e^{-2\beta dx})) \tag{3-43}$$

$$E[x_2x_2] = \frac{2\sigma^2}{\beta}(dx - \frac{2}{\beta}(1-e^{-\beta dx}) + \frac{1}{2\beta}(1-e^{-2\beta dx})) \tag{3-44}$$

$$E[x_2x_3] = 2\sigma^2(\frac{1}{\beta}(1-e^{-\beta dx}) + \frac{1}{2\beta}(1-e^{-2\beta dx})) \tag{3-45}$$

$$E[x_3x_3] = \sigma^2(1-e^{-2\beta dx}) \tag{3-46}$$

The computed functions for the mean-square expectations can be substituted into a matrix to obtain the process noise covariance matrix Q.

$$Q = \begin{bmatrix} E[x_1x_1] & E[x_1x_2] & E[x_1x_3] \\ E[x_1x_2] & E[x_2x_2] & E[x_2x_3] \\ E[x_1x_3] & E[x_2x_3] & E[x_3x_3] \end{bmatrix} \tag{3-47}$$

The process noise covariance matrices corresponding to lower order Gauss-Markov based process models can be derived easily using the integrals given above.

3-3-4 Second-order Gauss-Markov process

Consider the case in which the position and the gradient are to be modelled as random processes and the exact derivative relationship between the two applies, and it is assumed both have finite variances. Than it can be shown that the position cannot be modelled as a first-order Markov process, because the corresponding gradient would have infinite variance. Why this is the case is described below.

Consider the position is modelled as a first-order Markov process. In that case its PSD function is given by:

$$S_{pos} = \frac{2\sigma^2\beta}{\omega^2 + \beta^2} \tag{3-48}$$

it is given that $v(t) = \frac{\partial x(t)}{\partial t}$

Property 3. *(The frequency derivative property) The multiplication of a function $x(t)$ by t in the time domain is equivalent to the differential of its Fourier transform in the frequency domain. Therefore:*

$$\mathcal{F}(\frac{\partial x(t)}{\partial t}) = j\omega G_{pos}(j\omega) \tag{3-49}$$

The Wiener-Khinchine relation states that the Power Spectral Density of a process is the Fourier transform of the autocorrelation function:

$$S_X(j\omega) = \mathcal{F}[R_X(\tau)] \tag{3-50}$$

if the transfer function from the best-in-focus position to the gradient is given by a first order Gauss-Markov model $G_{pos}(s) = \frac{\sqrt{2\sigma^2\beta}}{s+\beta}$. Than, given property 3 the transfer function from the input to position corresponds to $sG_{pos}(s)$. Thus the power spectral density of the gradient is given by $S_{vel}(s) = G_{vel}(s)G_{vel}(-s)$. In this case this corresponds to:

$$S_{vel}(s) = \frac{\sqrt{2\sigma^2\beta}s}{s+\beta}\frac{\sqrt{2\sigma^2\beta}(-s)}{-s+\beta} = \frac{(-s^2)2\sigma^2\beta}{-s^2+\beta^2} \tag{3-51}$$

or using $s = j\omega$:

$$S_{vel}(s) = \frac{2\sigma^2\beta\omega^2}{\omega^2+\beta^2} \tag{3-52}$$

The inverse Fourier transform of the spectral function gives the autocorrelation function, which in turn can be used to derive the mean square value. The inverse Fourier transform can be computed as:

$$\mathcal{F}^{-1}[S_{vel}(j\omega)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{vel}(j\omega)e^{j\omega\tau}d\omega = R_{vel}(\tau) \tag{3-53}$$

The integral of S_{vel} will go to infinity. Therefore, the corresponding gradient will have infinite variance, which clearly does not model the true system dynamics accurately. Using a second order Gauss-Markov process as a transfer function from the gradient to the position this issue can be overcome. The new transfer function will however contain smaller high frequency components compared than the first order Gauss-Markov process. As is shown by].

This derivation shows that the designer should take care when modelling random processes that are a derivative of another random variable. If the integral of the PSD is not finite the variance will also be infinite, which often is undesirable.

3-3-5 Incorporating prior knowledge into the filtering procedure

In this section two adjustments to the filtering procedure described by algorithm 1 are suggested. First, the change in scanning direction is incorporated in the model. Secondly, the gradient in the y direction is added as a state variable. Pseudo code describing the algorithm that is used to implement the higher order conventional Kalman filter is given by algorithm 2.

The change in height is equal to the step size times the gradient along the direction in which the step was made. A surface is a 2D plane steps are made in both the x and y direction. A block diagram depicting these dynamics is given in figure 3-4.

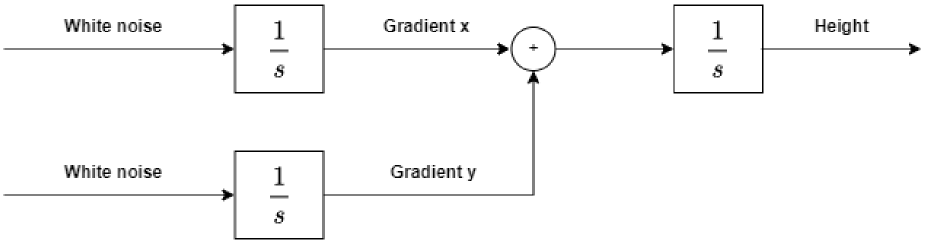


Figure 3-4: Block diagram depicting how both the x and y gradient contribute to the height in a particular position.

Changing scan direction

It was chosen to design the Kalman filter such that all the information is captured within one state space model. In the case of the first order Wiener process based model the matrices of the state space model become:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & dx & dy \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} w(k) \tag{3-54}$$

The measurement function remains the same. This approach to including the y-gradient in the state space model can be extended to second and higher order descriptions. It should be noted that the matrix pair (F, H) is no longer observable. The sub-matrices of F and H corresponding to a step in either the x or y direction are however observable. As the steps are always taken in one direction the state space model can be partitioned at each step such that the criteria of the Kalman filter stated in 1 are always satisfied. The partitioning also applies to the update step of the Kalman filter. Thus in each step the derivatives along only one direction are updated.

The design of the process noise, Q, also depends on the scanning direction. When the model is propagated along the x-direction, the model assumes the gradient in the y-direction is constant. In a way it copies the gradient in the y direction from the previous step to the next step without adding information. The process noise should account for uncertainties or variations in the system dynamics that are not accounted for by the model. When moving in the x-direction the slope in the y-direction is excluded from the model, and only propagated to the next position. Therefore, the process noise should be zero for the terms corresponding to the slope in the y-direction. The opposite holds when a step is made in the y-direction.

The designs of the process noise matrices in both cases is given by the following equation:

$$Q_x = \begin{bmatrix} \frac{1}{3}dx^3 & \frac{1}{2}dx^2 & 0 \\ \frac{1}{2}dx^2 & dx & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad Q_y = \begin{bmatrix} \frac{1}{3}dy^3 & 0 & \frac{1}{2}dy^2 \\ 0 & 0 & 0 \\ \frac{1}{2}dy^2 & 0 & dy \end{bmatrix} \tag{3-55}$$

Reversing scan direction

The previous paragraph described how the partitioned matrices can be used to change from a step in the y to a step in the x direction. Here the changes to the different matrices in order to reverse the sign of the step size are described.

Changing the sign of the step needs to be performed on every new row. In the state transition matrix F the sign of the step size dx can simply be reversed when the scanning direction along the x-direction is reversed. This corresponds to reversing the sign of dx in the state transition matrix and of all the elements on the super- and sub-diagonal elements of the covariance matrices P and Q (line 6, algorithm 2).

Corresponding algorithm

Algorithm 2 summarises the different steps accurately. At the start of each row the sign of dx is reversed in the matrices F, Q and P (line 6, alg. 2). Than an predict step is performed (line 8-9 or 12-13). In this step the fundamental matrix, F, and the process noise, Q, depend on the direction in which a step was made. The direction of propagation also determines the section or slice of the state estimate and corresponding covariance matrix that will be updated in the update step (line 10 or 14). After the update step the counter is iterated and the next iteration of the for loop is performed.

3-3-6 Checking assumptions

All the models above were tested to meet the requirements stated by theorem 1. It was found that all of the models meet the conditions stated by this theorem. Specifically for the Gauss-Markov models, for the process noise covariance matrix, Q to be positive semi-definite β has to be larger than a particular value. For the first order model this corresponds to $\beta > 5$, for the second order model this is $\beta > 20$. If this condition is not met the term $(H\bar{P}H^T + R)^{-1}$ can become singular.

Observability The Kalman filter may diverge if the system is not observable. This means that there are one or more state variables that are hidden from the view of the observer (i.e. the measurements). As a result, if the unobserved processes are unstable, the corresponding estimation errors will be similarly unstable. Another way of putting it is that the measurement do not provide enough information to estimate all the state variables of the system. The observability problem can be solved by either changing the model used to describe the system

Algorithm 2 Conventional Kalman filtering implementation higher order models

```
procedure KF($zs, x_0, P_0, R, Q, dx$)
    $\hat{x}, \hat{P} = x_0, P_0$                                ▷ Initialise variables
    $count = 1$
    for $z$ in $zs$ do
        if $(count - 1)$ modulo $NH$ is $0$:
            Change sign of $dx$ in $F$, $Q$, $P$
        if $count$ modulo $NH$ $==$ $0$:                         ▷ Predict step
            $\bar{x} = F_{b_1}\hat{x}$
            $\bar{P} = F_{b_1}\hat{P}F_{b_1}^T + Q_{b_1}$
            $slc = slc - y$
        else:
            $\bar{x} = F\hat{x}$
            $\bar{P} = F\hat{P}F^T + Q$
            $slc = slc - x$
        Update elements indexed by the slicing vector: $slc$.
        $r = z - H\bar{x}$                                       ▷ Update step
        $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$
        $\hat{x} = \bar{x} + Kr$
        $\hat{P} = (I - KH)\bar{P}$
        $count$ += 1
    end for
```

dynamics such that all its states are observable or by adding appropriate measurements to make the system completely observable.

In appendix B-1 the state space models described above are tested for observability. It is shown that the observability matrices are full rank, and the system is observable if the stepsize, dx, is chosen sufficiently large. If the step size dx becomes too small, the observability matrix may loose rank due to numerical precision issues and the accumulation of round-off errors. In the current application the step size depends on the FoV of the microscope. This can be in the order of micrometers. As described a step size of such a small order may lead to observability issues.

However because the step size is fixed it can be chosen to design the system such that the slope is not expressed as the change in height over the change in distance but rather the change in height per step. In doing so the "step size" dx changes from the width of the FoV in meters to 1 step. Substituting $dx = 1$ ensures observability is preserved.

3-3-7 Tuning filter parameters

Tuning the process noise, Q, and measurement noise, R, parameters is crucial to achieve optimal filtering performance. Careful adjustment of these parameters ensures that the Kalman filter appropriately balances the trade-off between tracking the true system state and accounting for the uncertainties in the measurements and system dynamics.

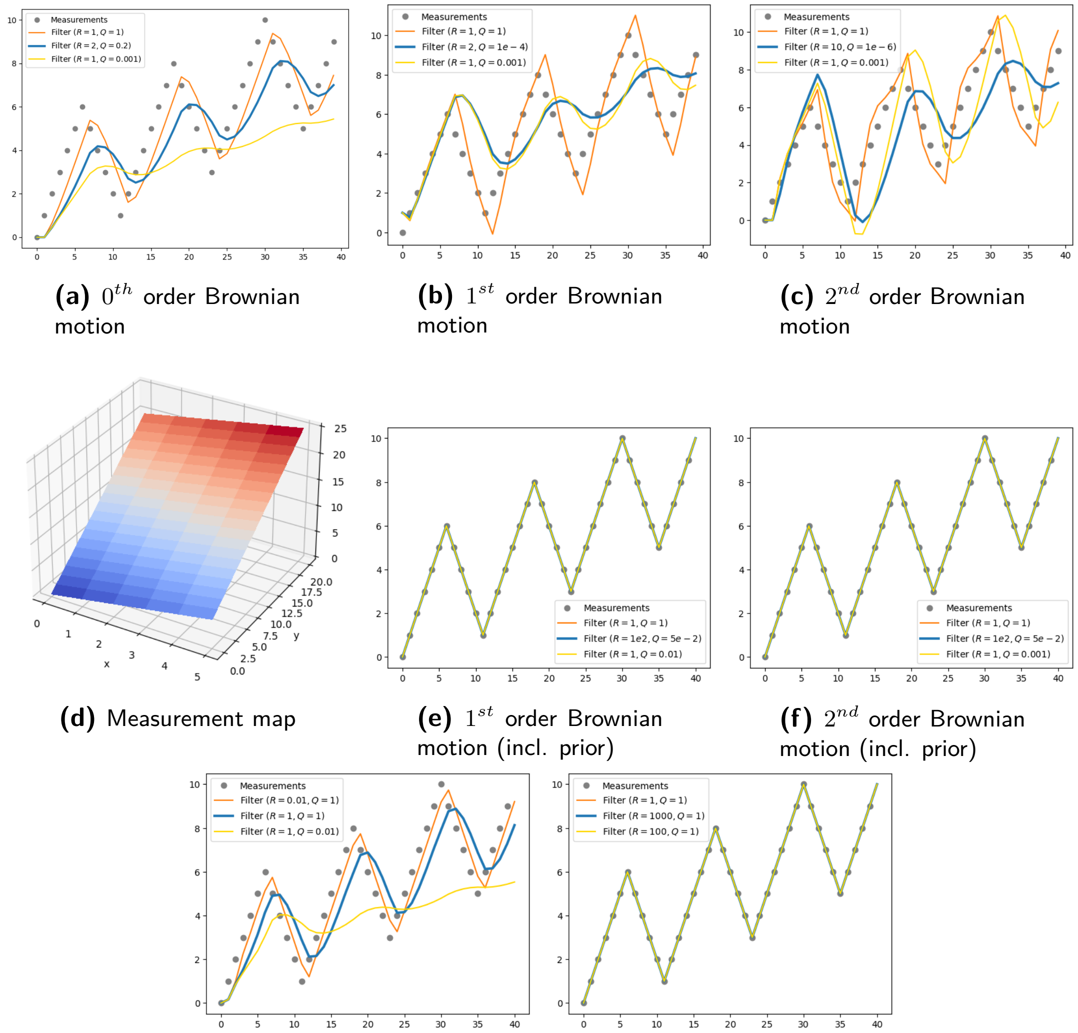
The ratio of the two parameters determines the behavior of the filter. If R is set to a very large value the covariance matrix will diverge and the filtered estimates becomes insensitive to measurements. If the wrong initial state is used or the track changes significantly the filter will require significant time to adjust. Vice versa if Q is set large relatively to R, the Kalman filter will trust the measurements over the predictions. Because of this noise will be incorporated in the output.

3-4 Results and discussion

The results in this section were obtained by altering the code of the FilterPy Kalman filtering module [9]. The code was adapted as is described in the pseudo code presented in the previous sections. All the models derived above and corresponding algorithms were implemented. The maps depicted in Figure 2-4a were flattened such that they could be used as measurements. The Gaussian smoothed maps were used as underlying true values.

The noise that is added to the ground truth to simulate the measurements, introduces additional randomness and variability, which obscures the underlying dynamics and complicates result interpretation. To mitigate this issue, all algorithms were initially tested on a simplified measurement map devoid of any noise. This particular map exhibits a gradient equal to 1 in each direction and is illustrated in Figure 3-5d. Subsequently, the algorithms were applied to filter a noisy map with a standard deviation of 0.1 μm. The outcomes of this process are presented in Tables 3-1, 3-2 and 3-3. These results serve to elucidate the performance achieved when dealing with the noisy maps during scanning.

The errors displayed in the tables correspond to the Mean Absolute Error (MAE), which are computed as the average of the absolute differences between the prior or posterior map and the 'true' Gaussian smoothed map.



(a) 0^{th} order Brownian motion

(b) 1^{st} order Brownian motion

(c) 2^{nd} order Brownian motion

(d) Measurement map

(e) 1^{st} order Brownian motion (incl. prior)

(f) 2^{nd} order Brownian motion (incl. prior)

(g) $1^{s}t$ order Gauss-Markov (incl. prior)

(h) $2^{n}d$ order Gauss-Markov (incl. prior)

Figure 3-5: For each of the implemented algorithms the first 40 steps are shown. The measurement map (shown in figure 3-5d) has a gradient equal to one in both the y and x direction. In all runs complete knowledge of the initial state of the system is assumed. In each plot the blue line corresponds to the tuning parameters that were found to achieve the best performance.

Figure 3-5 displays the predictions obtained using different order filters, using the measurements from a simple map. Beginning with the conventional filters, implemented using algorithm 1. It can be observed that the zeroth order filter consistently exhibits a lag behind the measurements. This phenomenon arises due to the absence of the gradient in the process model state. Hence, the model assumes a stationary position. As depicted in the plot, increasing the process noise effectively reduces this lag, while decreasing the process noise exacerbates the issue. Essentially, decreasing the process noise prioritises trust in the measurements over the model prediction. This approach can enhance performance when measurement noise is low. However, in the presence of substantial measurement noise, it incorporates the noise into the filter's output.

The first order filter demonstrates improved tracking of the plane. However, since the algorithm does not incorporate prior knowledge of the map, it tends to overshoot when the direction changes. Notably, the best-performing filter on the actual noisy map, depicted in blue in each plot, corresponds to the filter with the lowest process noise covariance, Q. As is described in section 3-3-7 this configuration makes the filter less responsive to changes in the measurements.

The second order filter, which incorporates curvature as one of its states (height, gradient, and curvature), demonstrates heightened responsiveness to measurement changes. While this may seem advantageous when examining the performance in figure 3-5c, this is generally not the case. As the state space order is higher than the dynamics of the system it is modelling, the higher order model is overfitting to the data, this reduces the performance when the changes in the measurements are primarily due to noise.

The figures 3-5e, 3-5f, 3-5g and 3-5h correspond to the approach where prior knowledge is incorporated in the map (described in section 3-3-5). The plots show that incorporating prior behaviour in the filter allows accurate simulation of the plane given that the initial state is correct. This observation confirms the effectiveness of the approach. This approach cannot be applied to zeroth order filters, as they do not incorporate the gradient or other higher order terms in the process model.

3-4-1 Reduced noise

The obtained results, showcasing the outcomes with the best tuning parameters, provide valuable insights into the performance of different filters. It is noteworthy that the posterior maps exhibit significantly lower MAE compared to the prior maps in all cases, as anticipated. This demonstrates that the algorithm is does effectively filter noise from the measurements in all cases. In line with the introductory statements of this chapter, the analysis proceeds by initially examining the table with less noise before delving into the table with more noise.

The findings in table 3-1 confirm that the zeroth order model outperforms the higher order models. This is expected as the effect of the changes in scanning direction are not incorporated in these higher order models. Consequently, the predictions made by these models tend to overshoot whenever a change in direction occurs, leading to less accurate prior estimates and ultimately resulting in poorer posterior maps. Moreover, the standard deviation of the prior maps for higher order models is significantly higher due to the sizeable errors introduced during direction changes.

Moving on to filters that do incorporate prior knowledge, presented in table 3-2. A notable improvement in performance is observed compared to filters of the same order without the incorporation of such knowledge. The first order filter, in particular, outperforms the zeroth order filter, indicating a better model representation of the surface. The second order filter performs slightly worse than the first order filter, suggesting a potential issue of overfitting due to the higher model order.

Shifting focus to the Gauss-Markov based filters presented in table 3-3. Here, the first order model is best compared with the zeroth order Wiener filter, while the second order model is compared to the first order filter. It is important to note that the key distinction between the Gauss-Markov filter and the Wiener model lies in the modelling of the gradient or the curvature as a Gauss-Markov process. Notably, the tuning parameters used in the Gauss-Markov based filters display significant differences, signifying the higher accuracy of models containing a integrator compared to models that do not. These findings align with the simulations conducted on the simple map depicted in Figure 3-5. However, the second order model does not surpass the performance of the first order Wiener process-based model. This suggests that the noise may be better represented as a white rather than a Gauss-Markov process.

Table 3-1: Wiener process based filters

State	Error	Zeroth order	First order	Second order
		$(R = 0.1, Q = 3, P = 5)$	$(R = 2, Q = 1e-4, P = 5)$	$(R = 10, Q = 1e-6, P = 2)$
Prior	Mean	0.286 μm	0.314 μm	0.418 μm
	St.d.	0.424 μm	0.854 μm	0.653 μm
Post	Mean	0.078 μm	0.082 μm	0.091 μm
	St.d.	0.097 μm	0.106 μm	0.120 μm

Table 3-2: Wiener process including prior knowledge

State	Error	Zeroth order	First order	Second order
			$(R = 0.1, Q = 2, P = 5)$	$(R = 0.5, Q = 0.5, P = 5)$
Prior	Mean	-	0.243 μm	0.348 μm
	St.d.	-	0.333 μm	0.487 μm
Post.	Mean	-	0.078 μm	0.081 μm
	St.d.	-	0.098 μm	0.103 μm

Table 3-3: Gaus-Markov based filters

State	Error	Zeroth order	First order	Second order
			$(R = 1e-2, P = 5, \sigma = 1, \beta = 16)$	$(R = 1e3, P = 5, \sigma = 1, \beta = 21)$
Prior	Mean	-	0.303 μm	0.252 μm
	St.d.	-	0.429 μm	0.338 μm
Post	Mean	-	0.077 μm	0.078 μm
	St.d.	-	0.097 μm	0.098 μm

3-4-2 Standard map performance

The tables 3-4, 3-5 and 3-6 display the performance obtained when applying the filters described in this chapter to the standard maps (with a lower signal-to-noise ratio). The obtained results reveal that the models perform approximately equally well when tuned such that the MAE is minimised. This outcome can be attributed to the substantial amount of noise present in the system. In the cases that the change in direction is incorporated in the algorithm, no notable improvement in prediction is observed. This lack of improvement can also be attributed to the large amount of noise present in the measurements, which obscures the benefits of incorporating such changes.

An interesting observation is that most models achieve their best performance when the process noise variance is significantly lower than the measurement noise variance. This configuration renders the filter less susceptible to the noise present in the measurements. It may however also cause the filter to 'ignore' relevant information in the measurements, causing it to overshoot. Additionally, if the initial state is wrong it takes the filter significantly longer to adjust to the measurements. This effect can be clearly observed in the error maps displayed in figure 3-6.

A comparison between the error map and the contour plot reveals that the largest errors occur when there are changes in the gradient of the map. The filters, due to low process noise variances, struggle to adequately adapt to these gradient changes, leading to heightened error levels. It is important to note that increasing the process noise as a solution to this issue is ineffective due to the relatively substantial amount of noise present in the system.

Table 3-4: Wiener process based filters

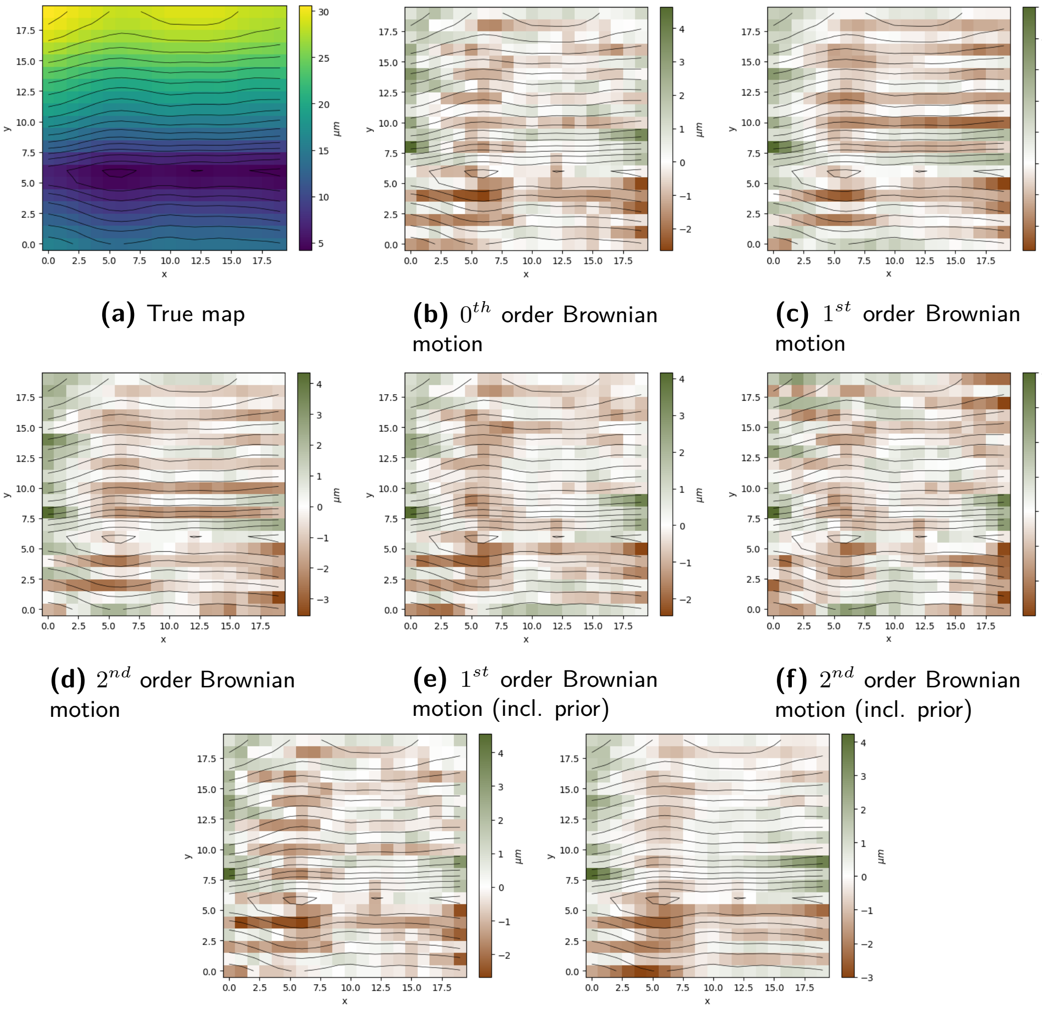
State	**Error**	**Zeroth order**	**First order**	**Second order**
		$(R = 2, Q = 0.2, P = 5)$	$(R = 2, Q = 1e - 4, P = 5)$	$(R = 10, Q = 1e - 6, P = 2)$
Prior	Mean	0.697 μm	0.665 μm	0.713 μm
	St.d.	0.875 μm	0.854 μm	0.907 μm
Post	Mean	0.598 μm	0.593 μm	0.619 μm
	St.d.	0.745 μm	0.773 μm	0.803 μm

Table 3-5: Wiener process including prior knowledge

State	**Error**	**Zeroth order**	**First order**	**Second order**
			$(R = 1e2, Q = 5e - 2, P = 5)$	$(R = 1e2, Q = 1e - 3, P = 5)$
Prior	Mean	-	0.724 μm	0.856 μm
	St.d.	-	0.900 μm	1.097 μm
Post.	Mean	-	0.621 μm	0.684 μm
	St.d.	-	0.771 μm	0.867 μm

Table 3-6: Gaus-Markov based filters

State	**Error**	**Zeroth order**	**First order**	**Second order**
			$(R = 1, P = 5, \sigma = 1, \beta = 6)$	$(R = 1e3, P = 5, \sigma = 1, \beta = 21)$
Prior	Mean	-	0.690 μm	0.732 μm
	St.d.	-	0.869 μm	0.907 μm
Post	Mean	-	0.600 μm	0.626 μm
	St.d.	-	0.749 μm	0.775 μm



(a) True map

(b) 0^{th} order Brownian motion

(c) 1^{st} order Brownian motion

(d) 2^{nd} order Brownian motion

(e) 1^{st} order Brownian motion (incl. prior)

(f) 2^{nd} order Brownian motion (incl. prior)

(g) 1^{st} order Gauss-Markov (incl. prior)

(h) 2^{nd} order Gauss-Markov (incl. prior)

Figure 3-6: Heat maps of the MAE between the prior maps and the ground truth (shown in figure 4-4d) of the Kalman filter algorithms presented in chapter 3. In each of the plots the equipotential lines of the ground truth map are added to improve the interpretability of the map.

Comparison of Brownian-motion and Gauss-Markov based process models Comparing the discrete state space models shows that the main difference between the Brownian motion based models and the Gauss-Markov based models lies within the process noise covariance matrix Q. The state transition matrices, despite containing different terms, are very similar. The term $e^{-\beta dx}$ converges to zero for low values of β. Hence, the first order Gauss-Markov based process model is approximately equal to the zeroth order Brownian motion based process model and the second order Gauss-Markov based model is approximately equal to the first order Brownian motion based process model. Comparing their respective prediction accuracy shows that the Wiener process based models perform slightly better than the Gauss-Markov based models. This suggest that the true dynamics are better characterised by a state space model driven by a Wiener process.

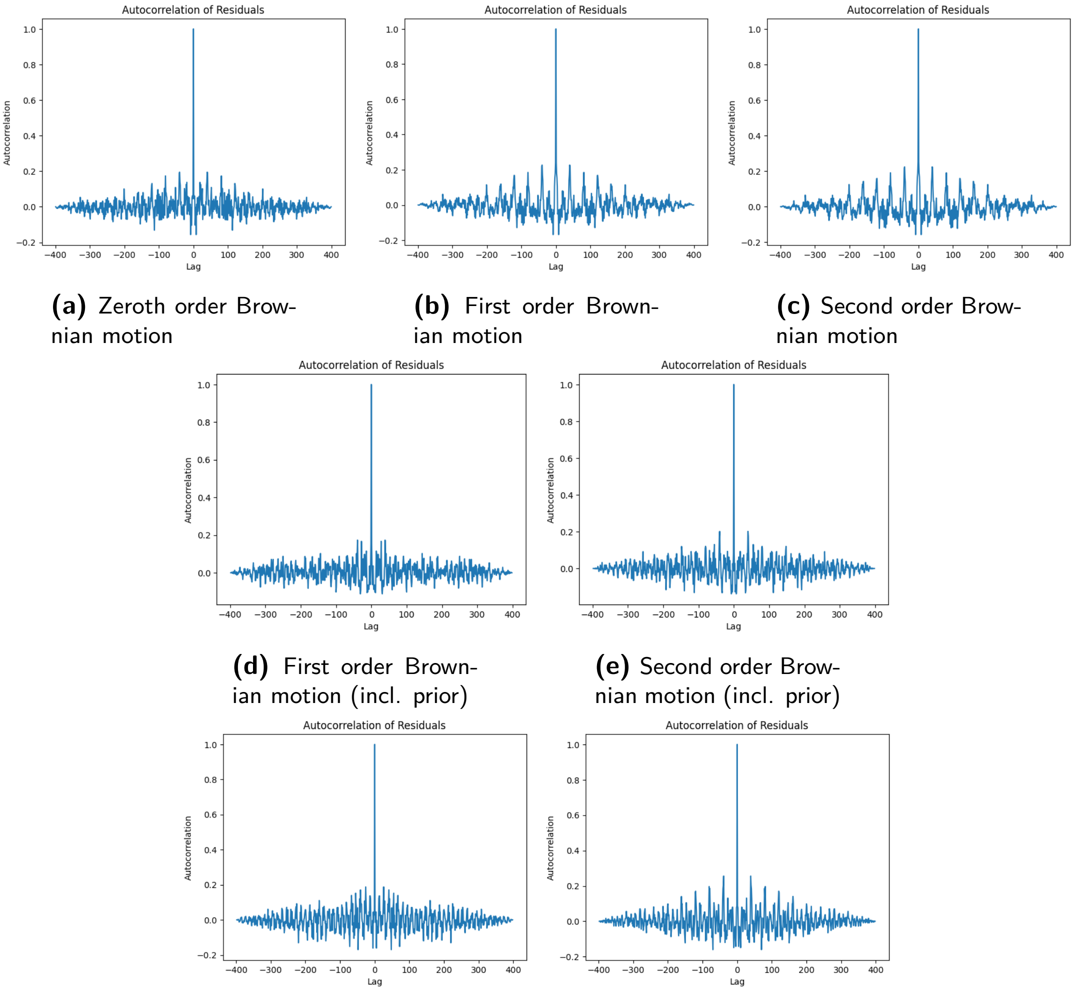
As mentioned above the main difference between the Wiener and Gauss-Markov based state space models lies in their process noise covariance matrices, Q. The (co)variance corresponding to the lower order terms of the Gauss-Markov based process model have a significantly lower variance than is the case for the Brownian motion based process models. In both cases the variance related to the highest order term is largest. The variance decreases as the order of the state decreases. In the case of a second order model, this implies that the uncertainty of the prediction is largest for the estimate of the curvature. It is thought that this may not always reflect the dynamics of the surface accurately, as it is expected that the position of a malaria parasite may change in a random fashion because the parasite is located in a different plane. Hence, for future work it is advised to alter the model such that the noise is not only considered to enter the system via the highest order system state, as is depicted in figure 3-3, but is also added after the position.

3-4-3 Testing the innovation sequences

In Kalman filter theory what has been referred to as the residual up to this point is commonly called the innovation. The innovation is the difference between the observed and predicted measurements. If the Kalman filter is implemented correctly its innovation sequence has several properties. Investigation of the innovation sequence can inform the designer of potential issues.

One property of the innovation sequence is that it is white. White noise is characterised by having a mean equal to zero. Additionally, the autocorrelation function of a white noise process is described by a Kronecker delta function. This property can be checked by plotting an estimate of the autocorrelation function of the innovation sequence. If the innovation sequence is not white, it indicates that the assumptions of the Kalman filter may not hold in the given system or measurement setup. Note that it is common to investigate the autocovariance of the Kalman filter. In the case that the mean of the process analysed is zero the autocovariance is equal to the autocorrelation.

Figure 3-7 shows the autocorrelation functions of the innovation sequences of the algorithms described above. It can be seen that the autocorrelation plots resemble Kronecker Delta functions. This confirms that the surface can be modelled as a Markovian system. It can additionally be noted that there are small peaks in the curves of the simplest Brownian motion filters. These peaks correspond to the position approximately 40 tiles before the current position. Intuitively this makes sense as these are the positions lying adjacent to the current state. The peaks are more pronounced in the models that do not incorporate a change in direction. This corresponds to what was expected because the residual in the error is larger at the edges of the surface.



(a) Zeroth order Brownian motion

(b) First order Brownian motion

(c) Second order Brownian motion

(d) First order Brownian motion (incl. prior)

(e) Second order Brownian motion (incl. prior)

(f) First order Gauss-Markov (incl. prior)

(g) Second order Gauss-Markov (incl. prior)

Figure 3-7: Autocorrelation plots of the innovation sequences of the Kalman filter algorithms implemented in chapter 3.

Table 3-7: Innovation sequence means

	Conventional filter			Brownian motion (incl. prior)		Gauss Markov (incl. prior)	
Filter	0^{th}	1^{st}	2^{nd}	1^{st}	2^{nd}	1^{st}	2^{nd}
Mean	$0.093\ \mu m$	$0.063\ \mu m$	$0.044\ \mu m$	$0.136\ \mu m$	$0.042\ \mu m$	$0.003\ \mu m$	$0.054\ \mu m$

3-5 Conclusion

This chapter started with an introduction of the Kalman filter and the requirements of the model in order for the system to converge to a stationary solution. This was followed by a short summary of the previous work. It was concluded that the state space model proposed by Wang [24] is not observable if only height measurements at the current position are used. Additionally, it was proposed to investigate whether higher order models would improve the performance of the Kalman filter proposed by Zhang et al. [29].

To test whether this is the case higher order Wiener and Gauss-Markov process based state space models were derived and tested. It was found that in the case that the signal-to-noise ratio is high and the surface is homogeneous, the first order Wiener process based state space model outperforms the zeroth order model. However, in the case that the signal to noise ratio is lowered the filters achieve similar performance. In the latter case, incorporating the changes of the scanning direction into the filter do not seem to improve the performance quality. Additionally, the results suggest that the Brownian motion based state space models outperform the models with a Gauss-Markov input.

In future work, it is suggested to alter the model of the system such that noise is not only driving the highest order term but also added separately to the height. In doing so the designer can specify the uncertainty in the curvature or gradient separately from the uncertainty in the position. This could for example be useful if there is uncertainty in the system that is specific to the position, such as a malaria parasite that is randomly located slightly higher in the blood sample.

Chapter 4

Non-Symmetric Half-Plane Kalman filtering

The Non-Symmetric Half-Plane (NSHP) Kalman filter is a recursive filtering approach which can be used to incorporate neighbouring state estimates in the prediction of the future state. The filter does so by extending the state vector so that it does not only consist of the filter estimate of the state in the previous position, but a subset of the previously visited positions is used to improve the prediction. This subset is referred to as the support. The other positions included in the state are not relevant for computing the prediction. The inclusion of these positions in the state vector is required for recursive implementation of the filter. The term Non-Symmetric Half-Plane (NSHP) refers to the specific structure used to determine which elements should be included in the state vector. This structure enables the recursive implementation of the filter. The following paragraphs will elaborate on the the specific design of the NSHP Kalman filter.

Using the NSHP was mentioned but disregarded by Wang [24] for two reasons. First, they argued that Kalman filtering is a model based filter, due to which suitability for one application (image restoration), does not guarantee its suitability for other 2D applications. Secondly, it was argued that applications other than image restoration require additional state variables, which according to [24] could not be incorporated into the approach. In this section an attempt will be made to show that the developed 2D Kalman processors can be extended to other applications by developing a filter for height map modelling.

Variogram The NSHP Kalman filter is build on the assumption that the prior state estimate can be improved by including information from not only the previous scanning position, but several neighbouring positions in the prediction. In order to test to which extend a given point on the surface is related to another point, and how this varies with the distance these points are separated, variograms of the surfaces were investigated.

Rationalised variability theory does not use autocorrelation, but instead uses a related property called the semivariance to express the degree of relationship between points on a surface.

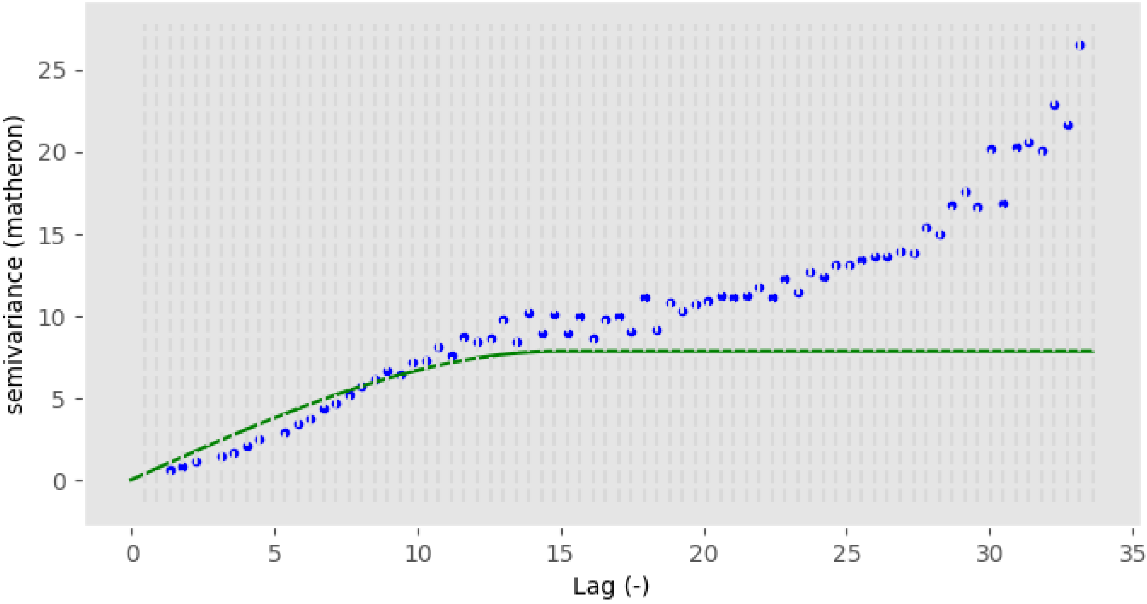


Figure 4-1: Variogram of one of the malaria maps. The lag corresponds to the separated distance in frames.

The semivariance is half the variance of the differences between all possible points spaced a constant distance apart. The range of the variogram refers to the distance at which the semivariance will become approximately equal to the variance of the surface itself. This is the greatest distance over which the value at a point on the surface are related to the value at another point. The range defines the maximum neighbourhood over which adding positions to the state is adding information and hence taking advantage of the statistical correlation among observations [11].

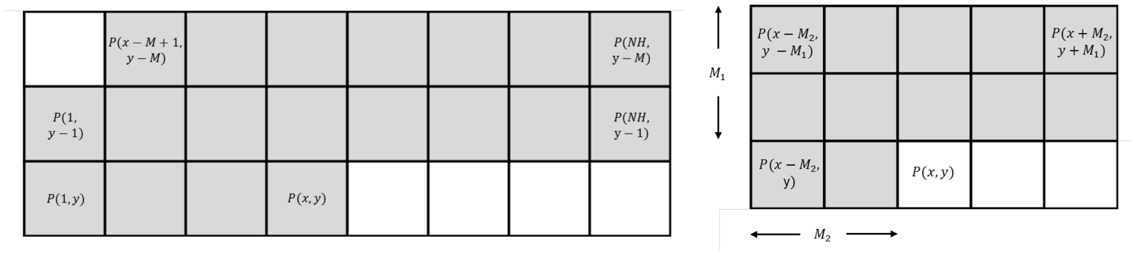
In this case the sill is manually set to the variance of the input map. It is found that the range is equal to 15 tiles. Beyond that range the value at a particular point is not related to the value at another point.

4-1 Implementation of the NSHP filter

The steps of the filter are given in algorithm 3. Each frame corresponds to a coordinate $P(i, j)$. A NSHP Kalman filter support region S can be defined for each position, the structure of the support is displayed in figure 4-2b. The recursion can be accomplished because only previously computed points are used in the computation of the present output. In this report a support order M indicates $M_1 = M_2 = M$. The state estimates corresponding to the positions in the support are used to predict the state of the next position. The state vector is a column vector that is also determined by the support order M (see figure 4-2a) and can be defined as:

$$\begin{array}{lll} [P(x,y), & \ldots, & P(1,y), \\ P(NH,y-1), & \ldots, & P(1,y-1), \\ P(NH,y-M), & \ldots, & P(x-M+1,y-M)]^T \end{array} \tag{4-1}$$

Separate lines were used to improve the readability and highlight that the positions correspond to different rows in the grid. Note that the support region is a subset of the state vector.



(a) Elements in state vector for NSHP support order M

(b) NSHP order M support

Figure 4-2: Display of the state vector and support elements in the NSHP Kalman filter

Using this definition of the state vector the state transition matrix can be defined as:

$$F = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 & C_2 & C_3 & C_4 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \tag{4-2}$$

This matrix can be split into two sections, namely the first row and the remaining rows. The latter section propagates all but the 'last' state estimates. The first row multiplies the elements of the state vector corresponding to support region with matrices C_1, C_2, C_3 and C_4 (or scalar weight in the 0^{th} order model case). The sub-matrices C_n correspond to the system matrices defined in section 3-3. The multiplications each give a prior estimate of the state. Each prior estimate is weighted using the normalised euclidean norm of the distance of the support position to the actual position. An example of the matrices corresponding to a first order implementation of the NSHP filter is given by the following matrices:

$$C_1 = w_1 \begin{bmatrix} 1 & dx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C_{2,4} = w_{2,4} \begin{bmatrix} 1 & dx & dy \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C_3 = w_3 \begin{bmatrix} 1 & 0 & dy \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-3}$$

where w_n correspond to the weights. Note that the sign of the step size dx varies depending on the position relative to the predicted state.

At the boundaries the support and state transition matrix have to be altered. The boundary conditions are invoked for computing the last and the first element of each row. Respectively, the state transition matrix used at the first boundary condition (line 8 in algorithm 3) is almost identical to the matrix given in 4-2 except the C matrix in the second to last column of the first row is removed. In the state transition matrix for the second boundary condition (line 15 of algorithm 3) the first two elements of the first row contain a C matrix and the lower diagonal identity matrix has been flipped. This ensures the elements in the state vector are ordered correctly to filter the next row. An example of the steps in each iteration are given in appendix C in order to visualise the effect of the multiplications.

4-1-1 Process noise

The process noise covariance matrix Q is modelled using the same process as is described in section 3-1-2. As was pointed out previously only the first row in the state transformation matrix is used to predict the state of the next scanning position. The remainder of the matrix is used to propagate the required state estimates of previously scanned positions for future predictions. Therefore, the process noise matrix, Q, should only increase uncertainty related to the first row and not the other elements. This leads to the following design for the process noise covariance matrix:

$$Q = \begin{bmatrix} Q_{sub} & \emptyset_1 \\ \emptyset_2 & \emptyset_3 \end{bmatrix} \tag{4-4}$$

where Q_sub is approximated to be equal to the process noise covariance matrices derived in the previous chapter.

4-1-2 Assumptions

All the models above were tested to meet the requirements stated by theorem 1. It was found that all of the models meet the conditions stated by this theorem. It should be noted that identical to the conditions mentioned in the previous chapter, for the Gauss-Markov models, for the process noise covariance matrix, Q, to be positive semi-definite β has to be larger than a particular value. For the first order model this corresponds to $\beta > 5$, for the second order model this is $\beta > 20$. If this condition is not met the term $(H\bar{P}H^T + R)^{-1}$ can become singular.

4-1-3 Pseudo code NSHP algorithm

The pseudo code describing the steps of the implemented algorithm is given below in algorithm 3. The steps in the algorithm are similar to those performed in the conventional Kalman filtering algorithm 2. The three if statements correspond to the three different cases. In this case, because the state transition matrix is observable the matrices are not sliced as in algorithm 2. Instead the update step is altered as follows.

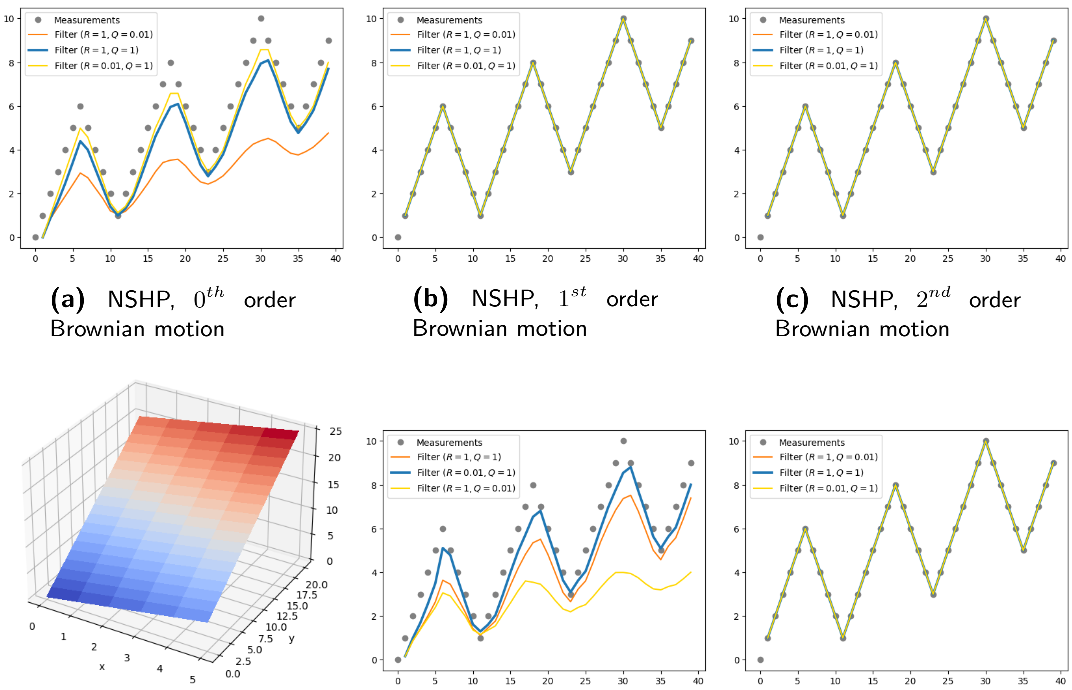
In the update step the elements in the prior state vector are updated based on the difference between the predicted and the measured value and their respective covariance matrices. However, not all elements of the state vector were used in the prediction. To ensure only the state estimates used in the prediction are updated, the elements in the Kalman gain that do not correspond to the support state estimates are set to 0 (line 13, 21 & 29 of alg. 3).

Algorithm 3 Non symmetric half plane Kalman filtering

```
procedure NSHP KF($a, b$)
A KF of the same order as the NSHP KF models is used to predict and update the the elements in the first row and first element of the second row. The corresponding state and covariance matrices are used to construct $x_0$ and $P_0$.
    $\hat{x}, \hat{P} = x_0, P_0$                                   ▷ Initialise variables
    $count$ = NH + 1
    for $z$ in $zs$ do
        if $higher\_order\_model$ and $count - 1$ modulo $NH$ is 0:
            Change sign of $dx$ in $F$, $F_{b_1}$, $F_{b_2}$ and $Q$
        if $count + 1$ mod $NH$ == 0:
            $\bar{x} = F_{b_1}\hat{x} + Bu$                          ▷ Predict step
            $\bar{P} = F_{b_1}\hat{P}F_{b_1}^T + Q$
            $r = z - H\bar{x}$                                       ▷ Update step
            $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$
            $K[slc] = 0$
            $\hat{x} = \bar{x} + Kr$
            $\hat{P} = (I - KH)\bar{P}$
        elif $count$ mod $NH$ == 0:
            $\bar{x} = F_{b_2}\hat{x} + Bu$                          ▷ Predict step
            $\bar{P} = F_{b_2}\hat{P}F_{b_2}^T + Q$
            $r = z - H\bar{x}$                                       ▷ Update step
            $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$
            $K[slc] = 0$
            $\hat{x} = \bar{x} + Kr$
            $\hat{P} = (I - KH)\bar{P}$
        else:
            $\bar{x} = F\hat{x}$                                     ▷ Predict step
            $\bar{P} = F\hat{P}F^T + Q$
            $r = z - H\bar{x}$                                       ▷ Update step
            $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$
            $K[slc] = 0$
            $\hat{x} = \bar{x} + Kr$
            $\hat{P} = (I - KH)\bar{P}$
        $count$ += 1
    end for
```

4-2 Results and discussion

The results in this section were obtained altering the code of the FilterPy Kalman filtering module [9]. The code was adapted as is described in the pseudo code above. Similar to the approach taken in the preceding chapter, we begin the analysis by examining the behaviour of the filter when applied to a simple map. Just as in the previous chapter, the gradient of this map is set to a constant value of one in all directions. The results, as depicted in the figures, clearly demonstrate that the higher order models (corresponding to figures 4-3b, 4-3c, 4-3f) accurately trace the measurement map when initialised with the correct initial condition. This alignment between the higher order models and the measurement map highlights the effectiveness of these models in capturing the underlying dynamics of the system. Note that both lower order models (corresponding to figures 4-3a, 4-3e), where the noise is not integrated, lag behind the measurement map. As is explained more elaborately in the previous chapter, this is due to underfitting.



(a) NSHP, 0^{th} order Brownian motion

(b) NSHP, 1^{st} order Brownian motion

(c) NSHP, 2^{nd} order Brownian motion

(d) Measurement map

(e) NSHP, 1^{st} order Gauss-Markov

(f) NSHP, 2^{nd} order Gauss-Markov

Figure 4-3: For each of the implemented algorithms the first 40 steps are shown. The measurement map (shown in figure 4-3d) has a gradient equal to one in both the y and x direction. In all runs complete knowledge of the initial state of the system is assumed. In each plot the blue line corresponds to the tuning parameters that were found to achieve the best performance.

4-2-1 Reduced noise

Tables 4-1 and 4-2 provide an overview of the mean absolute errors (MAE) between the prior maps and the 'true' maps obtained using algorithm 3. When compared to the preceding chapter, the models lacking an integrator manifest notably diminished performance. Again, the deficiency in fitting leads to a delay in the filters' alignment with the actual dynamics. Furthermore, this effect is exacerbated by the inclusion of neighbouring positions in the estimation process, where higher-order state derivatives such as gradient or curvature are not used to propagate the height at these positions to the present state. The higher order filters demonstrate comparable performance to the filters derived in the previous chapter. To achieve this performance level they do however require very low measurement noise. This indicates that the filter design does not accurately capture the dynamics of the modelled surface.

Table 4-1: NSHP Wiener process based filters

State	**Error**	**Zeroth order**	**First order**	**Second order**
		$(R = 1e-2, Q = 1, P = 5)$	$(R = 1e-2, Q = 1, P = 5)$	$(R = 1e-2, Q = 1, P = 5)$
Prior	Mean	0.814 μm	0.343 μm	0.332 μm
	St.d.	0.879 μm	0.586 μm	0.554 μm
Post	Mean	0.079 μm	0.079 μm	0.334 μm
	St.d.	0.099 μm	0.099 μm	0.554 μm

Table 4-2: NSHP Gauss Markov process

State	**Error**	**Zeroth order**	**First order**	**Second order**
			$(R = 1e-3, P = 5, \sigma = 1, \beta = 6)$	$(R = 1e-3, P = 5, \sigma = 1, \beta = 21)$
Prior	Mean	-	0.739 μm	0.325 μm
	St.d.	-	0.966 μm	0.556 μm
Post	Mean	-	0.079 μm	0.079 μm
	St.d.	-	0.099 μm	0.098 μm

4-2-2 Performance on standard maps

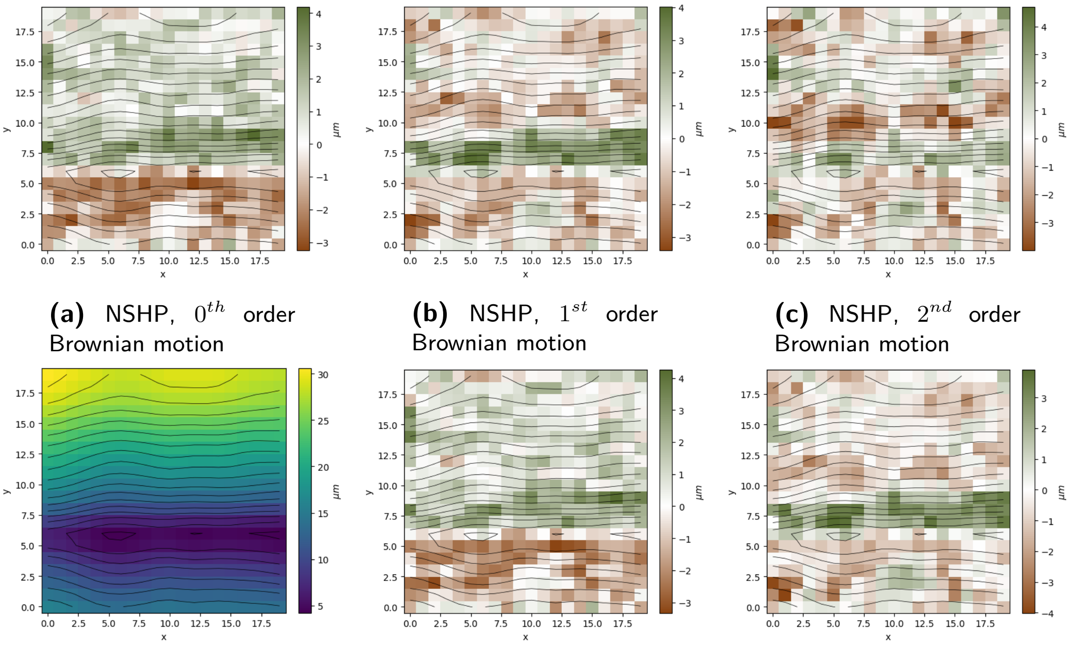
In this section the performance of the algorithms on the height maps as presented in 2-3-1 is evaluated. The MAE between the prior and 'true' underlying maps is given in tables 4-3 and 4-4. It can be observed that the performance is significantly worse than of the algorithms in the previous chapter. The MAE heat maps are displayed in figure 4-4. The heat maps show the error has been smeared out over a larger area. This indicates that the information from neighbouring positions on an adjacent row is not propagated to the to be predicted state correctly. Why this is the case will be explained more elaborately in the following section.

Table 4-3: NSHP Wiener process based filters

State	**Error**	**Zeroth order**	**First order**	**Second order**
		$(R = 1e-2, Q = 1, P = 5)$	$(R = 1, Q = 1, P = 5)$	$(R = 1, Q = 1, P = 5)$
Prior	Mean	1.015 μm	1.111 μm	1.267 μm
	St.d.	1.200 μm	1.400 μm	1.555 μm
Post	Mean	1.179 μm	0.863 μm	0.873 μm
	St.d.	1.478 μm	1.067 μm	1.079 μm

Table 4-4: NSHP Gaus Markov process including prior knowledge

State	Error	Zeroth order	First order	Second order
			$(R = 1e-2, P = 5, \sigma = 1, \beta = 6)$	$(R = 0.1, P = 5, \sigma = 1, \beta = 21)$
Prior	Mean	-	1.244 μm	1.120 μm
	St.d.	-	1.457 μm	1.420 μm
Post	Mean	-	1.174 μm	0.872 μm
	St.d.	-	1.457 μm	1.074 μm



(a) NSHP, 0^{th} order Brownian motion

(b) NSHP, 1^{st} order Brownian motion

(c) NSHP, 2^{nd} order Brownian motion

(d) Ground truth map

(e) NSHP, 1^{st} order Gauss-Markov

(f) NSHP, 2^{nd} order Gauss-Markov

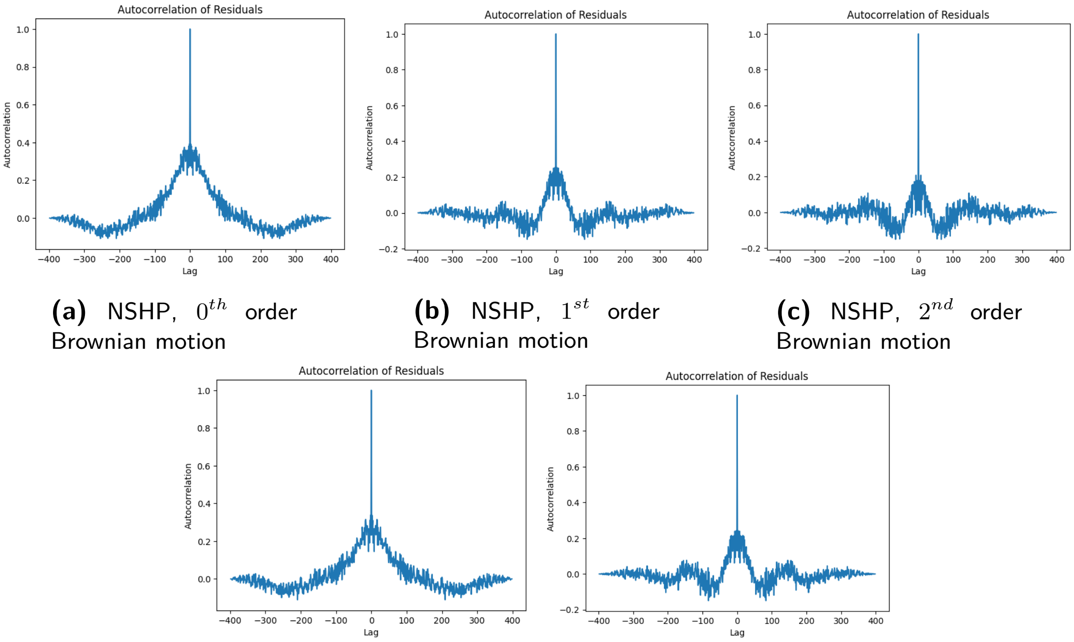
Figure 4-4: Heat maps of the MAE between the prior maps and the ground truth (shown in figure 4-4d) of the NSHP Kalman filters. In each of the plots the equipotential lines of the ground truth map are added to improve the interpretability of the map.

4-2-3 Innovation sequence analysis

Table 4-5 gives the means of the innovation sequences of the implemented NSHP filters. All means are approximately 0. The autocorrelation functions of each of the models however does not resemble a Kronecker delta function. This implies the measurements are influenced by previous or future measurements or that the system dynamics are not adequately captured by the model. In such cases, the Kalman filter's assumption of independence between time steps is violated, leading to suboptimal state estimation. Why this is the case is described in the following section.

Table 4-5: Innovation sequence means

	Brownian motion (incl. prior)			Gauss Markov (incl. prior)	
Filter	0^{th}	1^{st}	2^{nd}	1^{st}	2^{nd}
Mean	0.005 μm	0.03 μm	0.02 μm	0.009 μm	0.05 μm



(a) NSHP, 0^{th} order Brownian motion

(b) NSHP, 1^{st} order Brownian motion

(c) NSHP, 2^{nd} order Brownian motion

(d) NSHP, 1^{st} order Gauss-Markov

(e) NSHP, 2^{nd} order Gauss-Markov

Figure 4-5: Autocorrelation plots of the innovation sequences of the implemented NSHP Kalman filter algorithms.

4-2-4 Limitations of the NSHP filter

This subsection describes the issues with the NSHP filter that are hindering its performance. The deterioration of the performance of the filters is attributed to two issues, namely the observability of the states and the violation of the Markov assumption. The section starts with a discussion of the observability of the states, following a discussion on whether the current filter design satisfies property 1.

Observability Despite the (F, H) pair passing the observability test it i s thought that there are issues with the state reconstruction from the measurements. As was described in section 3-3-5 when the gradient estimate in the y direction is also incorporated in the state transition matrix, it is no longer observable. Intuitively it makes sense that using only one height measurement, made by scanning mostly along one axis, does not provide sufficient information to reconstruct the gradient in both the x and y direction accurately at each position.

In the case of the NSHP filters the (F, H) pairs are observable. This is because the state is build up of sub-state estimates at previous positions which add information to the system. Despite passing the test for observability, a poor estimate is used for the y gradient in most positions. Algorithm 2 is used to generate the state estimates of the positions along the first row and the first position on the second row. The estimates are used to initialise the NSHP filter structure. However, in algorithm 2 the state transition matrix is partitioned at each step such that only one of the gradients is updated. Due to this the gradient in the y-direction stays equal to its initial estimate along the first row. This same gradient is used by the NSHP algorithm to project the neighbouring states in the row above the current row to the current row. Therefore the y-gradient may however not be accurate at all.

This shows that the higher order models, containing higher order state estimates in multiple directions are not appropriate for the NSHP algorithm. One approach to solving this issue may be omitting states related to the y-direction all-together. When a step is made in the y-direction the state transition matrix could be partitioned as to not include the gradient and higher order derivatives in the x-direction in the prediction, and the process noise Q can be increased to reflect the uncertainty in the estimate.

The Markov property The NSHP filter was originally developed for image reconstruction [28]. The pixel values in an image often exhibit local correlations and dependencies. Each pixel's value is influenced by its neighbouring pixels, and there may be spatial coherence within the image. In this case the Markovian assumption may hold reasonably well, as the immediate past pixel values can provide useful information for predicting the current pixel value. The Kalman filter, with its ability to incorporate both measurement and state information, can be effective for smoothing the image by taking advantage of the Markovian nature of the pixel values.

Height measurements from a map are not inherently Markovian, as they typically depend on a variety of factors and considerations that extend beyond the immediate location or state. the appropriateness of applying the Kalman filter should hence carefully be considered depending on the application at hand. In cases that the surface is relatively uniform or homogeneous over small spatial scales, height measurements can exhibit Markovian behaviour. This is assumed to be the case in the previous chapter. This implies that it has to be assumed that the elevation changes are gradual and consistent within a local area.

Specifically to the current implementation of the NSHP Kalman filter. It can be derived that the filter is not strongly optimal (meaning both the prediction and update are optimal). This is due to the following reason. As the filter moves across the surface, filtered data corresponding to previous positions is used in the computation of the prior state. This violates the Markov property, due to which the filter is no longer optimal. In the correction of their previous paper [26] a modification is proposed for strong optimality. Regrettably, the existence of the correction only dawned upon the author of this thesis upon concluding the writing of it. Therefore, it is left outside the scope of the thesis.

4-3 Conclusion

As was mentioned in the previous section, Wang [24] disregarded the NSHP algorithm for two reasons. They were correct in stating that as the Kalman filter is a model based filter suitability for one application, does not guarantee its suitability for other 2D applications. More specifically, it is important that the filtered surface is homogeneous, such that the Markovian assumption holds. Secondly, it was argued that applications other than image restoration require additional state variables, which according to [24] could not be incorporated into the approach. In this section it was shown that the NSHP filter can incorporate additional state variables in the filter design.

As was pointed out in the previous section the reason that the NSHP algorithms gave poor results was due to issues with the state observability and the violation of the Markov property. In order to solve the issue with the observability of the system a different description of the system dynamics should be considered. To solve the latter issue Woods [26] proposed a correction of their previous work. In conclusion, further research into the applicability of the NSHP Kalman filter is necessary in order to use it to accurately reconstruct terrain models.

Chapter 5

Conclusion

The goal of this thesis was to improve the speed and accuracy at which high-quality, in-focus images can be acquired by an automated microscope. To do so two modifications to the work by Zhang et al. [29] were investigated. Firstly, in chapter 3 higher order process models were implemented. Secondly, in chapter 4 an attempt was made to improve the prior estimate through incorporating information from previously estimated neighbouring positions in the prediction using a two dimensional Non-Symmetric Half-Plane (NSHP) Kalman filter. The performance of the proposed scanning algorithms was tested through the simulated application of the algorithms using a height map of a thin smear malaria specimen. The accuracy of the algorithms was compared to state of the art focus map surveying procedures.

In summary, it was found that the Kalman filter based on a zeroth order model as proposed by Zhang et al. [29] outperforms all other proposed algorithms if the measurements are noisy. If the signal to noise ratio is improved it is found that a first order Wiener process model performs best. The improvement in the accuracy is significant. Using Delaunay based interpolation, which is a state of the art method, the mean error is 0.94 μm. Using the zeroth order model reduces the error to 0.70 μm for the prior estimate, and 0.60 μm for the posterior estimate. It should be noted that the performance of the algorithms is highly dependent on the characteristics of the surface modelled.

With regards to NSHP it was found that the implementation proposed in this thesis performs poorly due to issues with the state observability and the violation of the Markov property. With regard to the former, the algorithm and state space model need to be altered such that the states are observable. To solve the latter issue, Woods [26] proposed a correction of their previous work. Further research into the applicability of the NSHP Kalman filter is necessary in order to use it to accurately reconstruct terrain models.

5-1 Further research

In this thesis a first attempt was made to extend the Kalman filter based scanning approach proposed by [29] to obtain more accurate predictions of the best-in-focus position under noisy circumstances. Faster and more precise scanning remains a decisive factor in the popularisation of automated microscopy. Below direction for further research are proposed.

Kalman filtering With regards to the NSHP Kalman filter many aspects can be investigated more closely. It would be of interest to optimise the weights used, the size of the support region and boundary conditions as suggested by Woods [26]. In order to save computational power the reduced order update and the strip processor as proposed by Woods and Radewan [28] could be implemented.

Other Kalman filtering approaches that would be interesting to apply to this problem are periodic Kalman filters or Multiple Model Adaptive Estimators (MMAE). The serpentine scanning direction introduced periodic elements in the measurement data. In this thesis an attempt to include the information in the previously visited positions was made with the NSHP algorithm. A periodic Kalman filter may however be more effective at doing so. In MMAE, multiple models are ran alongside each other, at each position the state estimates of the filter are blend using their likelihoods [8]. This approach could be effective to model surfaces with changing characteristics. The surface may be flat in one segment, where higher order terms would cause overfitting to the data, and have a changing slope in another segment, requiring higher order models. MMAE provides a solution to this problem.

Delaunay triangulation Focus map surveying using Delaunay triangulation based interpolation showed promising results in this thesis. Investigating this approach further might enable improving the scan time significantly. An interesting approach would be to dynamically determine the grid in which the positions are sampled. The distance between each point could depend on the characteristics of the underlying map. For example, it would be interesting to link the sampling rate to the curvature of the surface. If the gradient is changing fast a higher sampling rate would be required to accurately reconstruct the surface and vice versa.

Appendix A

Focus metrics

This appendix describes the method applied to determine the best in focus image at each tile. It starts by describing how the acquired figures are preprocessed in order to remove noise and maximise contrast of the features related to the Giemsa stained malaria parasite. The following section describes the selection of the focus metric and related characteristics.

A-1 Preprocessing

The resolution of an imaging system is fundamentally limited by the diffraction limit of the objective lens. In accordance with Rayleigh's equations, the diffraction-limited resolution of a 60x lens with a numerical aperture of 0.85 can be calculated to be 0.39 μm. The resolution of digital imaging systems can be further restricted by the resolution of the camera, which is dependent on both the pixel size and the system magnification. The effective pixel size of the system in the object plane is equal to the pixel size divided by the system magnification, which, in the present case, amounts to 0.04 μm.

To achieve the diffraction-limited resolution, the Nyquist sampling theorem stipulates that the effective pixel size should be half the diffraction-limited resolution. Should the effective pixel size be greater than that required by the Nyquist theorem, aliasing will occur, whereas if it is significantly smaller, the image will be oversampled, which introduces noise into the image. In practice sampling with a sampling rate slightly below half the diffraction-limit gives the best result.

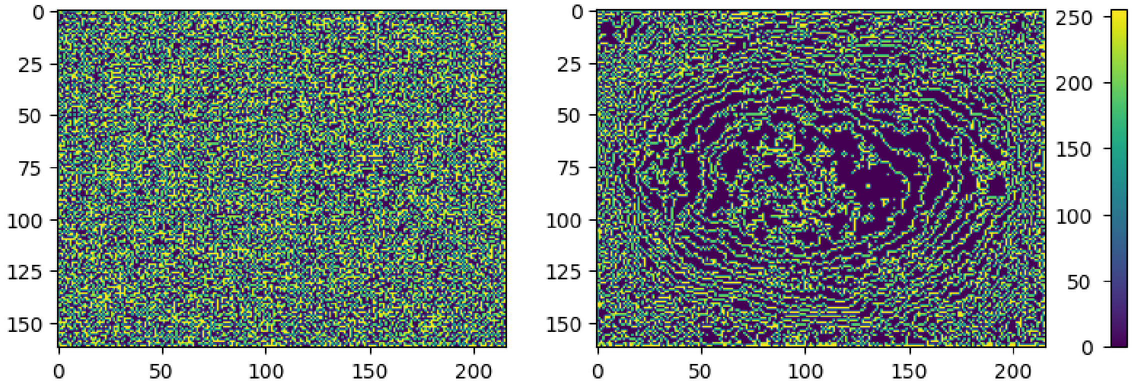
In the current application, the effective pixel size is roughly 10 times smaller than the diffraction-limited resolution. As is explained in the previous paragraph this implies the image is oversampled significantly, introducing noise into the image. Denoising can be achieved by low-pass filtering the image. In the next section a brief comparison of different low pass filtering methods is given.

A-1-1 Low pass filter design

The focus metric used by AiDx Medical was originally the variance of the Laplacian. To compute the Laplacian, the image was first resized using a bilinear method from OpenCV. This resizing acts as a type of low-pass filter. Different methods can be sued to execute the resizing. The choice of the filter size and type significantly impact the information in the resulting image as will be described in the following paragraph.

The width of the kernel used in the filter is inversely proportional to its bandwidth, meaning that a wider kernel leads to a smaller bandwidth and more aggressive filtering. When downsizing an image, the bandwidth should be reduced to a level that can be represented in the new image to prevent aliasing. This requires an increase in the kernel size of the filter. According to [14], the OpenCV resize method does not automatically adjust the kernel width based on the downsize factor, while the Pillow module does.

To evaluate the impact of these differences, an out-of-focus image was resized to $1/12^{th}$ of its original dimensions using both OpenCV and Pillow. The Laplacian of the resulting figures was computed and is shown in Figure A-1. Displaying the Laplacian was chosen because it amplifies the high-frequency content of the image.



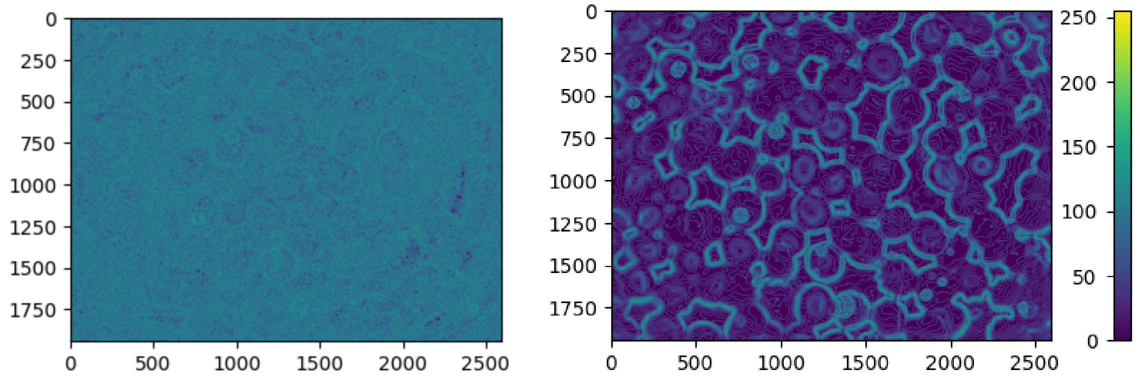
(a) OpenCV

(b) Pillow

Figure A-1: Laplacian filtered, out-of-focus images that have been resize to $1/12^{th}$ the original size using bilinear image resizing with **(a)** OpenCV and **(b)** Pillow

Additionally, the choice of filter has a significant impact on the frequency distribution of the down sampled image. An averaging kernel, which can be represented by a boxcar signal, yields a sinc() function in the frequency domain. The lobes of the sinc() function amplify high frequency content, i.e., noise, in the image. While one may consider using the Fourier transform pair to obtain an ideal low pass filter, this method results in the removal of edge details due to the sharp cut-off frequency. In practice, a Gaussian filter is found to be the best choice, as it offers a favourable trade-off between noise suppression and edge preservation. Moreover, the Fourier transform of a Gaussian filter is also a Gaussian function, which effectively mitigates ringing artefacts in the resulting image.

As a preprocessing operation all the figures in this report are filtered using a Gaussian filter. The standard deviation of the filter is set to 7 pixels. This corresponds to the kernel width required to eliminate noise in the image due to oversampling (based on the diffraction limit computed using the maximum wavelength in the visible spectrum).



(a) Original

(b) Standard deviation = 8 pixels

Figure A-2: Laplace filtered images of an in focus thin smear malaria sample. The original is displayed in **(a)** and the Gaussian filtered image **(b)**.

A-1-2 Colour selection

The malaria specimens utilised in this study were subjected to staining with Giemsa stain, resulting in a distinctive purple hue exhibited by the malaria parasites. To exploit this characteristic in the autofocus algorithm, the red-green-blue (RGB) colour channels were examined. The greatest contrast was observed in the green colour channel. This observation can be attributed to the fact that purple mainly comprises blue and red hues, resulting in minimal green channel intensity within purple areas. In contrast, the rest of the image comprises grey and white areas that show nearly uniform intensity across all channels. Consequently, the blue channel lacks contrast, while the green channel exhibits high contrast. Therefore, only the green colour channel is used to compute the focus metrics in this paper.

A-2 Focus metric selection

In order to determine which focus metric should be used to find the best in focus position a comparison of 5 focus metrics was made, namely: the focus metric originally used by AiDx medical, the mean gradient, the variance of the Laplacian, the normalised variance and the JPEG file size.

These focus metrics were selected for the following reasons. Comparative studies on the performance of focus metrics on biological specimen using brightfield microscopes by [19], [20], [5] and [10] concluded that the normalised variance provide the best overall performance. Additionally, they mentioned the derivative based metrics are fast, simple and robust to subsampling. As these were originally applied by AiDx medical they were found worth investigating further. Frequency based focus metrics were found to be to computationally expensive to be ran on our system, which has a limited computational capacity. Using the

JPEG file size however allows a computationally frequency based method that had shown promising results [7]. The results obtained are described in the following section.

A-2-1 Deep learning

Deep learning methods have been applied to obtain in focus images of pathology slides. [3] describes the different methods can be split up into two category, namely: 1) Defocus image detection and 2) Virtual refocusing. In the first category deep learning methods are applied in order to determine the distance from the measured, out of focus, image to the position where the best in focus image can be made. The second category uses deep learning methods to create a virtual in focus image from any image along the optical axis.

The paper by [3] reports promising results. The approach allows single-shot autofocusing and requires no additional optical hardware. Thus enabling fast and economical autofocusing. There are however also several disadvantage to this approach. First of all, the refocusing range of this approach is relatively short. Secondly, a change in the optical hardware may affect the autofocusing performance. Thirdly, the system may fail for new features or new types of specimen on which the employed algorithms have not been trained. Lastly, the z-stacks of the to be imaged specimens have to be available to train the algorithms. Obtaining this data can be time consuming and costly.

This approach could be implemented complementary to a different autofocusing approach which is based on first principles. As such the system will always be able to create an in focus image of a variety of specimens and simultaneously collect data on which the complementary deep learning model can be trained. If a trained deep learning algorithm is available for a particular specimen it can be employed in order to speed up the autofocusing and hence scanning procedure. Because of the disadvantages related to deep learning methods only first principle based methods were implemented in this thesis work.

A-3 Results

Focus metric accuracy The performance of the focus metrics was evaluated using six stacks obtained from different positions along a Giemsa-stained thin smear malaria pathology slide. Prior to analysis, the stacks were preprocessed as outlined in section A-1. The average step size along the z-axis used is 0.19 μm. The visually determined best in focus position of each stack served as the reference for comparison with the focus metric based results, where the best in focus position corresponds to the position of maximum focus metric value.

To determine the accuracy of the focus metrics, the mean difference between the reference and focus metric based best in focus positions was calculated for the analysed stacks. The results are presented in table A-1. Among the focus metrics examined, the frequency-based approach performed the best.

Considering that a 60x objective lens has a depth of field of 1 μm, a mean error of an order of magnitude lower is desirable in focus metric analysis. Therefore, further improvement in accuracy is desired. If computational power is available, it is recommended to explore the implementation of additional frequency-based methods.

Table A-1: Error of the tested focus metrics

Focus metric	Original	Mean grad.	Var. of Lap.	Norm. var.	JPEG file size
Error	26.17 μm	2.27 μm	2.20 μm	1.27 μm	0.97 μm

The original autofocus approach performs poorly. This is mainly attributed to the preprocessing steps. As is described in section A-1-1 improper resizing using Open CV amplifies high frequency content (noise) in the image. A Laplacian filter is sensitive to a high frequency content in the image and hence reaches its maximum variance in an out of focus position, where the image contains a lot of noise.

Comparable results are obtained using the mean of the gradient and variance of the Laplacian with the adjusted preprocessing method. It is thought that the latter slightly outperforms the former because it is more sensitive to high frequency content in the image. The smallest error is obtained using frequency based 'JPEG file size'. This is thought to be the case because frequency based method is most sensitive to small changes in high-frequency details. The generalisation of these findings to other specimens requires additional investigation. The performance of focus metrics depends on the specimen specific features and the imaging system.

It was chosen to use the normalised variance as a focus metric in the remainder of the report because of its ease of online implementation, low computational burden and relatively high accuracy.

Determining step size The step size used for acquiring a focused image must be chosen carefully to balance image quality and acquisition time. A step size that is too large will result in low-quality figures, while a step size that is too small will increase the acquisition time. The depth of field of the objective lens used provides a logical limit for the step size (1 μm for a 60x objective lens). To determine the appropriate step size, the mean accuracy for each focus metric at different step sizes is computed. The resulting plot is displayed in figure A-3.

Surprisingly, the mean error did not increase immediately for all focus metrics as the step size increased. This is because the maximum of the focus measure set with the smallest step size does not always correspond to the best position. One or more subsampled sets may contain maxima that are closer to the actual best position, leading to improved average accuracy. However, when the sample rate becomes too high, a majority of the subsampled focus measure lists maximum are worse than the original maximum, resulting in deteriorated accuracy. The JPEG file size and normalised variance have sharper peaks and are more accurate, and thus, the described effect is not as significant for those focus metrics.

The optimal step size depends on the features of the applied focus metric and the desired image quality, and should therefore be determined by expert medical staff.

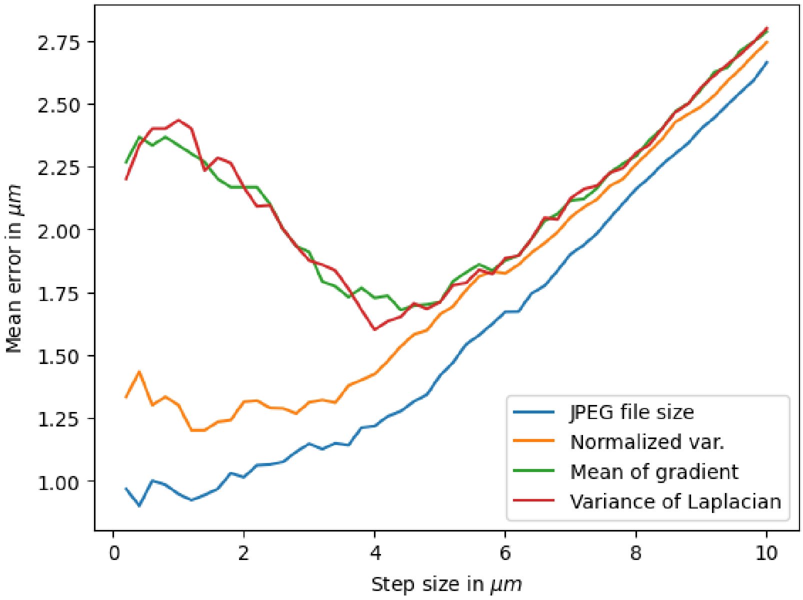


Figure A-3: Analysis of the required step size.

Appendix B

Observability

As is described in section 3-3-6 the observability of a state space model describes whether the measurements contain sufficient information in order to reconstruct all the state variables. The formal definition of the observability of a Linear Time Invariant (LTI) system is given below.

Definition 1. *The LTI system:*

$$x(k+1) = Ax(k) + Bu(k) \tag{B-1}$$

$$y(k) = Cx(k) + Du(k) \tag{B-2}$$

is observable if any initial state $x(k_a)$ is uniquely determined by the corresponding zero-input response $y(k)$ for $k_a \leq k \leq k_b$ with k_b finite.

There are two approaches that are commonly used to test whether a LTI system is observable. The two approaches are described in the lemmas below:

Lemma 1. ***Observability rank condition*** *([16]) The LTI system B-1 is observable if and only if*

$$rank(\mathcal{O}_n) = n \tag{B-3}$$

where $\mathcal{O}_n$ is the observability matrix. The observability matrix is defined as

$$\mathcal{O}_n = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{B-4}$$

Lemma 2. ***Popov-Belevitch-Hautus observability test*** *([6]) The LTI system B-1 is observable if and only if, for all $\lambda \in \mathbf{C}^n, x \neq 0$, such that $Ax = \lambda x$, it holds that $Cx \neq 0$.*

Below an example is given of how the state space models presented in the thesis will be shown to be observable (or not) using one of the lemmas presented above.

B-1 Conventional Kalman filters

The observability matrices in this section 3-3-6 correspond to the state space models presented in section 3-3. For the presented zeroth order model the the observability matrix is $\begin{bmatrix}1\end{bmatrix}$. This matrix is full rank and hence the state is observable. For the first and second order systems we obtain the following observability matrices:

$$\mathcal{O}^{1^{st}} = \begin{bmatrix} 1 & 0 \\ 1 & dx \end{bmatrix}, \qquad \mathcal{O}^{2^{nd}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & dx & \frac{1}{2}dx^2 \\ 1 & 2dx & 2dx^2 \end{bmatrix} \tag{B-5}$$

We can determine their rank through the following steps:

$$\mathcal{O}^{1^{st}} = \begin{bmatrix} 1 & 0 \\ 1 & dx \end{bmatrix} \sim \begin{bmatrix} 1 & 0 \\ 0 & dx \end{bmatrix} \sim \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{B-6}$$

$$\mathcal{O}^{2^{nd}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & dx & \frac{1}{2}dx^2 \\ 1 & 2dx & 2dx^2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & dx & \frac{1}{2}dx^2 \\ 0 & 0 & dx^2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & dx & 0 \\ 0 & 0 & dx^2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B-7}$$

It is shown that the observability matrices are full rank, and the system is observable if the stepsize, dx, is chosen sufficiently large. If the step size dx becomes to small the observability matrix may loose rank due to numerical precision issues and the accumulation of round-off errors.

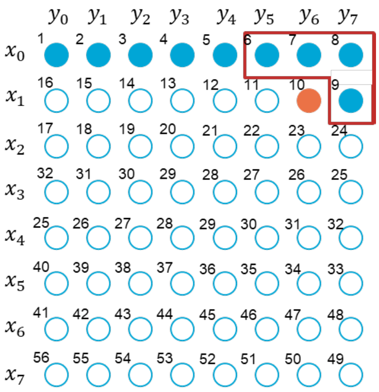
Appendix C

NSHP system matrix iterations

In this appendix the iterations using the system matrices defined in section 4 are shown in order to build intuition for the described steps. All the iterations corresponding to one row of the to be scanned area are shown.

The positions along the first row and the first position on the second row are determined using a Kalman filter as described by algorithm 1. This information is combined in the right structure to form the input matrices of the NSHP Kalman filter. How the process model propagates the states is shown in the steps below.

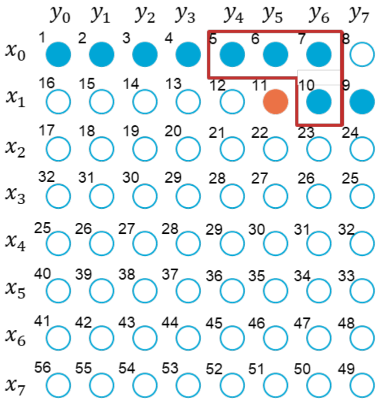
Regular case multiplication


y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7
x_0 1 2 3 4 5 6 7 8
x_1 16 15 14 13 12 11 10 9
x_2 17 18 19 20 21 22 23 24
x_3 32 31 30 29 28 27 26 25
x_4 25 26 27 28 29 30 31 32
x_5 40 39 38 37 36 35 34 33
x_6 41 42 43 44 45 46 47 48
x_7 56 55 54 53 52 51 50 49

$$\begin{bmatrix} \hat{x}_{10} \\ x_9 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 & C_2 & C_3 & C_4 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_9 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

Repetition of previous step

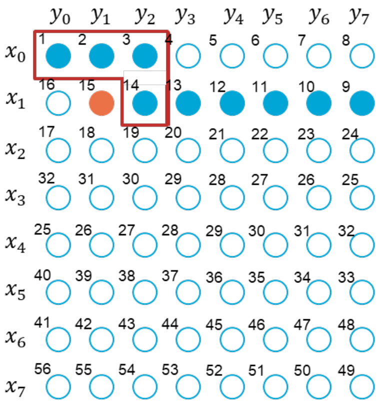
The previous multiplication is repeated to show the recursion (line 22, algorithm 3).



y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7
x_0 1 2 3 4 5 6 7 8
x_1 16 15 14 13 12 11 10 9
x_2 17 18 19 20 21 22 23 24
x_3 32 31 30 29 28 27 26 25
x_4 25 26 27 28 29 30 31 32
x_5 40 39 38 37 36 35 34 33
x_6 41 42 43 44 45 46 47 48
x_7 56 55 54 53 52 51 50 49

$$\begin{bmatrix} \hat{x}_{11} \\ x_{10} \\ x_9 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 & C_2 & C_3 & C_4 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_{10} \\ x_9 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

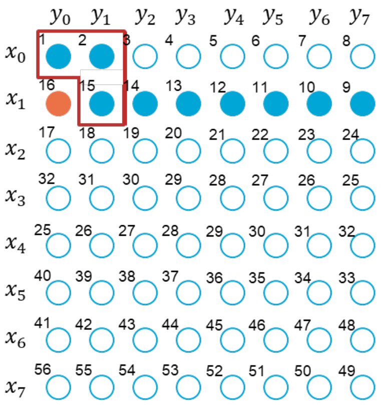
Several intermediate steps are not shown for briefity of notation. They are identical to the previous and following step.



y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7
x_0 1 2 3 4 5 6 7 8
x_1 16 15 14 13 12 11 10 9
x_2 17 18 19 20 21 22 23 24
x_3 32 31 30 29 28 27 26 25
x_4 25 26 27 28 29 30 31 32
x_5 40 39 38 37 36 35 34 33
x_6 41 42 43 44 45 46 47 48
x_7 56 55 54 53 52 51 50 49

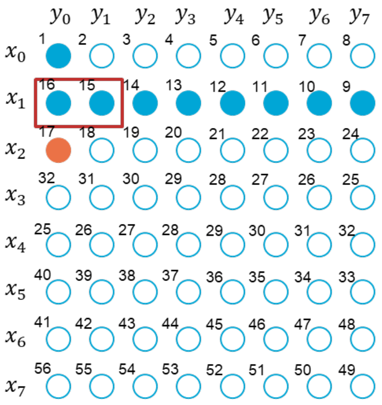
$$\begin{bmatrix} \hat{x}_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_9 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 & C_2 & C_3 & C_4 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_9 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

First boundary case (line 8 of algorithm 3).
The only difference with the previous system state transition matrix is that one less state is included in the prior computation.


y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7
x_0 1 2 3 4 5 6 7 8
x_1 16 15 14 13 12 11 10 9
x_2 17 18 19 20 21 22 23 24
x_3 32 31 30 29 28 27 26 25
x_4 25 26 27 28 29 30 31 32
x_5 40 39 38 37 36 35 34 33
x_6 41 42 43 44 45 46 47 48
x_7 56 55 54 53 52 51 50 49

$$\begin{bmatrix} \hat{x}_{16} \\ x_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_9 \\ x_1 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 & 0 & C_2 & C_3 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_9 \\ x_1 \\ x_2 \end{bmatrix}$$

Second boundary case (line 15 of algorithm 3)
In this case only two states are used to predict the next state. The identity matrices are placed along the anti diagonal to ensure that the state vector is in the right order for recursion on the following row.


y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7
x_0 1 2 3 4 5 6 7 8
x_1 16 15 14 13 12 11 10 9
x_2 17 18 19 20 21 22 23 24
x_3 32 31 30 29 28 27 26 25
x_4 25 26 27 28 29 30 31 32
x_5 40 39 38 37 36 35 34 33
x_6 41 42 43 44 45 46 47 48
x_7 56 55 54 53 52 51 50 49

$$\begin{bmatrix} \hat{x}_{17} \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{bmatrix} = \begin{bmatrix} C_1 & C_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{16} \\ x_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_9 \\ x_1 \end{bmatrix}$$

Bibliography

[1] Milon Amin, Anil V. Parwani, and Liron Pantanowitz. Digital Imaging. pages 129–145, 2014. doi: 10.1007/978-1-4614-9581-9{_}14. URL https://link.springer.com/chapter/10.1007/978-1-4614-9581-9_14.

[2] B.D. Anderson and J.B. Moore. *Optimal filtering.* Courier Corporation, 1979.

[3] Zichao Bian, Chengfei Guo, Shaowei Jiang, Jiakai Zhu, Ruihai Wang, Pengming Song, Zibang Zhang, Kazunori Hoshino, and Guoan Zheng. Autofocusing technologies for whole slide imaging and automated microscopy. *Journal of Biophotonics*, 13(12):e202000227, 12 2020. ISSN 1864-0648. doi: 10.1002/JBIO.202000227. URL https://onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/full/10.1002/jbio.202000227.

[4] Robert G. Brown and Patrick Y. C. Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises*, volume 1. J. Wiley & Sons, Hoboken, NJ, 4 edition, 2012.

[5] Frans C A Groen, Ian T Young, and Guido Ligthart. A comparison of different focus functions for use in autofocus algorithms. *Cytometry: The Journal of the International Society for Analytical Cytology*, 6(2):81–91, 1985.

[6] Thomas Kailath. *Linear systems*, volume 156. Prentice-Hall, Englewood Cliffs, NJ, 1980.

[7] Joe Knapper, Joel T Collins, Julian Stirling, Samuel McDermott, William Wadsworth, and Richard W Bowman. Fast, high-precision autofocus on a motorised microscope: Automating blood sample imaging on the OpenFlexure Microscope. *Journal of Microscopy*, 285(1):29–39, 2022. doi: https://doi.org/10.1111/jmi.13064. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/jmi.13064.

[8] Roger' 'Labbe. *Kalman Filters and Random Signals in Python.* 7.2.3 edition.

[9] Roger R. Labbe. FilterPy, 2014.

[10] X. Y. Liu, W. H. Wang, and Y. Sun. Dynamic evaluation of autofocusing for automated microscopic analysis of blood smear and pap smear. *Journal of Microscopy*, 227(1): 15–23, 2007. doi: https://doi.org/10.1111/j.1365-2818.2007.01779.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2818.2007.01779.x.

[11] Y. Zee Ma and Stephen A. Holditch. Unconventional Oil and Gas Resources Handbook - Evaluation and Development, 2016. URL https://app.knovel.com/hotlink/khtml/id:kt010QK24L/unconventional-oil-gas/glossary-u-references.

[12] Michael C. Montalto, Richard R. McKay, and Robert J. Filkins. Autofocus methods of whole slide imaging systems and the introduction of a second-generation independent dual sensor scanning method. *Journal of Pathology Informatics*, 2(1):44, 1 2011. ISSN 2153-3539. doi: 10.4103/2153-3539.86282.

[13] Mannan Muhammad and Tae Sun Choi. Sampling for shape from focus in optical microscopy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):564–573, 2012. ISSN 01628828. doi: 10.1109/TPAMI.2011.144.

[14] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On Aliased Resizing and Surprising Subtleties in GAN Evaluation. 4 2021.

[15] Marcial García Rojo, Gloria Bueno García, Carlos Peces Mateos, Jesús González García, and Manuel Carbajo Vicente. Critical comparison of 31 commercially available digital slide systems in pathology. *International Journal of Surgical Pathology*, 14(4):285–305, 10 2006. ISSN 10668969. doi: 10.1177/1066896906292274. URL https://www.researchgate.net/publication/6754764_Critical_Comparison_of_31_Commercially_Available_Digital_Slide_Systems_in_Pathology.

[16] Wilson J. Rugh. *Linear systems theory*. Prentice-Hall, 1996.

[17] Simo Sarka. *Bayesian Filtering and Smoothing*, volume 3. Institute of Mathematical Statistics Textbooks, 2013.

[18] Feimo Shen, Louis Hodgson, and Klaus Hahn. Digital Autofocus Methods for Automated Microscopy. *Methods in Enzymology*, 414:620–632, 1 2006. ISSN 0076-6879. doi: 10.1016/S0076-6879(06)14032-X.

[19] Murali Subbarao and Arman Nikzad. Focusing techniques. *Optical Engineering*, 32(11): 2824–2836, 11 1993.

[20] Yu Sun, Stefan Duthaler, and Bradley J Nelson. Autofocusing in computer microscopy: Selecting the optimal focus algorithm. *Microscopy Research and Technique*, 65(3):139–149, 2004. doi: https://doi.org/10.1002/jemt.20118. URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/jemt.20118.

[21] Michel Verhaegen and Vincent Verdult. *Filtering and system identification*. Cambridge University Press, Cambridge, 2011.

[22] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 3 2020. ISSN 1548-7091. doi: 10.1038/s41592-019-0686-2.

[23] Dai Hua Wang, Kan Wang, and Lin Sen Qiang. Depth estimation method of surface of micropart in microassembly space based on microscopic vision tomographic scanning images. *Journal of Microscopy*, 283(2):77–92, 8 2021. ISSN 13652818. doi: 10.1111/JMI.13010.

[24] Ping Wang. Applying two dimensional Kalman filtering for digital terrain modelling. *Proceedings of International Archives of Photogrammetry, Remote Sensing, and Spatial Information*, pages 649–656, 1998.

[25] Yongjie Wang, Rebecca Chandler, Kerry Israel, Wei Juan Wong, and Maria Ada Prusicki. Microscope Autofocus: Active vs. Passive Methods for Equipment Design | Olympus LS, 12 2021. URL https://www.olympus-lifescience.com/en/discovery/microscope-autofocus-active-vs-passive-methods-for-instrument-design/.

[26] John Woods. Correction to "Kalman Filtering in Two Dimensions". *IEEE Transactions on information theory*, 25:628–629, 9 1979.

[27] John W. Woods and Vinay K. Ingle. Kalman Filtering in Two Dimensions: Further results. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2):188–197, 1981.

[28] John W. Woods and C Radewan. Kalman filtering in two dimensions. *IEEE Transactions on Information Theory*, 23(4):473–482, 1977.

[29] Xuanmo Zhang, Daoxian Wang, Wenjie Zhou, Yingguo Gao, and Xiaohui Duan. A Whole Slides Imaging System for Intelligent Microscope Based on Kalman Filter. *Proceedings of 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2019*, pages 401–405, 10 2019. doi: 10.1109/IMCEC46724.2019.8984009.

[30] Xu Zhou, Rafael Molina, Yi Ma, Tianfu Wang, and Dong Ni. Parameter-Free Gaussian PSF Model for Extended Depth of Field in Brightfield Microscopy. *IEEE Transactions on Image Processing*, 29:3227–3238, 2020. ISSN 19410042. doi: 10.1109/TIP.2019.2957941.

Glossary

List of Acronyms

WSI	Whole Slide Imaging
FM	Focus Metric
FoV	Field of View
DoF	Depth of Field
NSHP	Non-Symmetric Half-Plane
LSI	Linear Space Invariant
LTI	Linear Time Invariant
PSD	Power Spectral Density
MAE	Mean Absolute Error
MMAE	Multiple Model Adaptive Estimators