The development, validation and demonstration of an automated rodent tracker and whisker detector

Brett Michael Hewitt

School of Science and the Environment

Manchester Metropolitan University

A thesis submitted in partial fulfilment of the requirements of the Manchester Metropolitan University for the degree of

Doctor of Philosophy

2018

Abstract

Quantitatively assessing behaviour to measure animal behaviour and motor control is challenging because there is a lack of unobtrusive behavioural models. Some studies have suggested that measuring whisker movements might be a good, quantitative behavioural model. However, whiskers are very thin, small and move very fast; and there is not yet an automated program that can detect whiskers in a fully-intact, freely-moving animal. Therefore, this thesis develops, validates and demonstrates a novel, fully-automated rodent tracker and a whisker annotator, that simultaneously measures locomotion and exploration behaviours as well as whisker movements.

The first step in designing an automated rodent tracker and whisker detector, is to extract a reliable ground truth from which to compare any tracked points to. Therefore, the Manual Whisker Annotator (MWA) was designed as a validator and calibrator for the subsequent trackers and detectors. The second step is to provide a reliable body and head contour. Therefore, the Automated Rodent Tracker (ART) was developed and validated, compared to a semi-automated tracker (Ethovision) and the MWA. Finally, a fully-automated whisker detector (FAWD) was designed and validated, using two existing semi-automatic whisker trackers (BWTT and Whisk) and the MWA. FAWD incorporates a variety of image-processing algorithms, including super sampling, dilation and subtraction and frangi filtering to reliably detect whiskers. Both ART and FAWD were also successfully demonstrated on videos collected from SOD1 mice, a model of Amyotrophic Lateral Sclerosis, from day 30 to 120. The development of this software enables whisker movements and locomotion to be tracked in a repeatable fashion, and the fully-automated nature of the software means that many videos can be collected and quickly processed with minimal user input.

This thesis develops and validates a suite of behavioural software that provides robust and quantitative measures of rodent behaviour for basic research or drug discovery. Future work will be to demonstrate this software on a larger range of rodent models of neurodegeneration, to further showcase the flexibility and quantitative nature of this behavioural model.

Acknowledgements

Throughout my research and preparation of this thesis, many people have guided my understanding and knowledge that allowed me to form the work presented. Firstly, I would like to thank my Director of Studies, Dr. Robyn Grant, for her kindness, patience and overall confidence in my abilities from the very beginning. I also want to express my utmost gratitude to my other supervisors, Dr. Moi Hoon Yap and Dr. Emma Hodson-Tole, who all provided their own unique perspective in different aspects of my work and contributing greatly when I was in need.

I have received help and advice, directly or indirectly, from my peers and colleagues throughout my study. These people motivated me to share ideas and build relationships to further expand my knowledge: Dr. Rachael Reynolds, Miss Regine Sonderland Saga, Mr. Thomas Higginbottom, Miss Catrin Austin, Miss Louise Tetlow, Mr. Connah Kendrick, Mr. Sean Maudsley-Barton, Dr. Adrian Davison, Mr. Peter Lawrence, Miss Mariane Delaunay, Miss Giada Porcedda, and everyone else from E121, E401 and E402.

I would like to thank the administration team during my study at MMU, Mrs Megan Paines, Ms. Francesca Robinson and Ms. Kristina Ganchenko. As well as always being available to help with a multitude of requests, they were key to helping with the success and progression of this PhD.

I would like to thank my friends from Haxby and Manchester, for always being there whenever I needed advice through the high and low points. I would also like to thank Miss Gemma Bromley, for her patience and support during the project.

My most sincere thanks goes to my family, for always believing I could succeed, even when I did not. To my mum, dad, brother and sister, I am unable to express how much I want to thank you for your support.

My final thanks goes to MMU, for providing the studentship that enabled this PhD to take place.

Contents

List of Figures

List of Publications

This thesis is based on material from the following publications:

1. Hewitt, B., Yap, M.H. and Grant, R., 2016. Manual whisker annotator (MWA): A modular open-source tool. Journal of Open Research Software, 4(1).

2. Hewitt, B., Yap, M.H. and Grant, R., 2016. ART - Automated Rodent Tracker (Conference Poster). Measuring Behaviour 2016.

3. Hewitt, B.M., Yap, M.H., Hodson-Tole, E.F., Kennerley, A.J., Sharp, P.S. and Grant, R.A., 2017. A novel automated rodent tracker (ART), demonstrated in a mouse model of amyotrophic lateral sclerosis. Journal of neuroscience methods.

4. Hewitt, B.M., Yap, M.H., Hodson-Tole, E.F. and Grant, R.A., 2018 (Under Review). Fully Automated Whisker Detector. IET Image Processing.

Chapter 1

Thesis Overview

1.1 Background

The increase in human lifespan observed in industrialized countries has been accompanied by a marked prevalence of neurodegenerative diseases [55]. The World Health Organisation (2006) have found that neurodegenerative diseases are a large cause of mortality and in 2005 constituted 11.67% towards deaths globally, and this figure is expected to increase to 12.22% by 2030 as shown in table 1.1. The WHO have also found the medical and financial burden neurodegenerative diseases places on society have been seriously underestimated [238]. In Europe alone, it is estimated that 7.5 million people suffer from either Alzheimer's or Parkinson's disease, at an estimated cost of €120 billion [91]. It is therefore imperative that more research is undertaken in this area in order to help develop new diagnostic techniques and treatments for them.

Human neurodegenerative diseases are characterized by progressive dysfunction and loss of neurons induced by particular neurological deficits[92]. Research into the pathogenesis for these disorders is exploding exponentially. To a great extent, this has been fostered by major advances in genetics, since we now know mutations that are associated with a large number of these illnesses. This includes, but not limited to, Alzheimer's Disease, Parkinson's Disease, ALS, Huntington's Disease. This discovery has led to the development of transgenic animal models of these disorders, which has increased our understanding of disease pathogenesis, as well as the effectiveness of new treatments[16].

Cause Category	**2005** (%)	**2015** (%)	**2030** (%)
Epilepsy	0.22	0.21	0.19
Alzheimer and other dementias	0.73	0.81	0.92
Parkinsons disease	0.18	0.20	0.23
Multiple sclerosis	0.03	0.03	0.02
Migraine	0.00	0.00	0.00
Cerebrovascular disease	9.90	10.19	10.63
Poliomyelitis	0.00	0.00	0.00
Tetanus	0.33	0.23	0.13
Meningitis	0.26	0.17	0.10
Japanese encephalitis	0.02	0.01	0.01
Total	**11.67**	**11.84**	**12.22**

Table 1.1: Mortality rates for various neurological disorders [238]

1.2 The effects of neurodegenerative diseases on locomotion and whisking

Unfortunately, none of the models described above completely represent the pathology of these diseases, animals that carry genetic defects similar to those found in patients do not develop identical symptoms. Nevertheless, it is undeniable that rodent models contribute significantly to the understanding of neurodegenerative diseases and the development of cures[197]. Despite these advances, generating quantifiable metrics is often intrusive, and places the animal under a great deal of stress[218, 221]. Measurable behavioural models mitigate a large part of the stress, and also allow for the animal to be studied while behaving freely. Several behavioural models have been developed, including tests on senses, memory, locomotion, motor co-ordination, strength and balance[212, 227, 247, 126, 94, 72].

The whisker system is an established model for cortical sensory pattern formation and information processing [123, 54]. The topographic nature of the system, which keeps spatial information intact has been inspiring neuroscience research for decades; in particular, the grid-like layout of the whisker follicles in the mystacial pad is conserved in the ‘barrelettes’ of the brainstem, the ‘barrloids’ of the thalamus, and the ‘barrels’ of the cortex [244, 233], so sensory information can be traced throughout the brain. However, as well as sensing the whiskers also make precise, robust and controlled movements [85, 160]. The fact that they can be precisely measured [28, 185, 37] make them a prime candidate for measuring aspects of motor control.

Indeed, another avenue of research into the whisker system is using their movements to trace neurodegenerative disease progression. The ability to trace locomotion and whisker movement deficits over time in disease models is becoming a popular avenue of exploration [87, 82, 74].

Grant et al. proposed that neurodegenerative diseases significantly affect locomotion and whisking in mice, and found significant changes in mouse models of Amyotrophic Lateral Sclerosis [86], Huntingtons Disease [74], and Anxiety [83]. This thesis will extend this proposed behavioural model, by developing, validating and demonstrating a fully automated rodent tracker and whisker detector to allow for easy mass data collection.

1.3 Problem statement

The nature of an emerging field means that research is limited and tends to be exploratory rather than focused on already established work. In chapter 2, current methods and algorithms for detecting rodents and their whiskers are discussed. Many different techniques are used with varying degrees of automation and accuracy. The main problems faced in rodent and whisker detection are automatically isolating the nosetip from an image of a rodent, which when recorded against a bright backdrop, effectively becomes a large dark object with no distinguishing features; and detecting the whiskers, which are very faint and move at high speeds. The difficulty in detecting these faint lines from video images has led to the development of novel techniques, as current well understood line trackers are unable to distinguish them from the background.

1.4 Objectives

The primary aim of this research is to develop, validate and demonstrate a fully automated rodent tracker and whisker detector. This thesis will test and evaluate established image processing techniques, and where necessary develop new techniques, on high speed video to develop a novel way of automatically detecting a rodent and its whiskers. This will help future studies analyse large video datasets quickly and accurately. To achieve this aim, the following objectives have been established:

1. Create a manual annotator software and produce a dataset with ground truth annotation for validation of the algorithms.

2. Develop and validate a robust rodent head detection and tracking algorithm using shape features.

3. Develop and validate an automated rodent whisker detector using image processing algorithms.

4. Demonstrate the final prototypes including the quantification of the mouse behaviour and statistical analysis.

1.5 Contributions

The main contributions of this thesis are as follows:

1. A manual head and whisker annotator (MWA) to provide a ground truth with which to validate the automated aspects of the final software package.

2. An automated rodent tracker (ART) which can automatically track a rodent's nose tip, orientation, and object interaction behaviours.

3. A whisker detection module that is added to ART, here as known as the Automated Rodent and Whisker Tracker (ARWT), which can automatically detect a rodent's whiskers.

4. Evaluation of the proposed methods and characterisation of locomotion and whisking behaviour in mice with ALS.

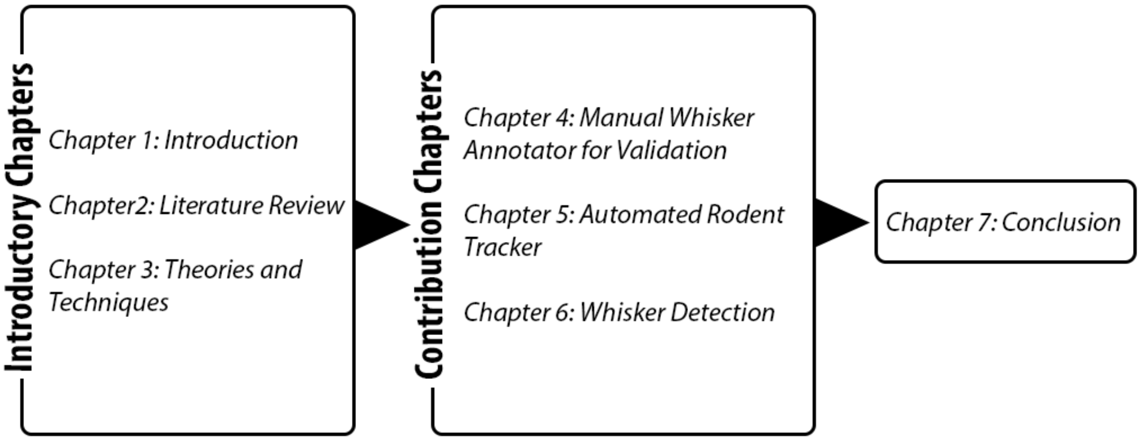

Introductory Chapters
Chapter 1: Introduction
Chapter2: Literature Review
Chapter 3: Theories and Techniques
Contribution Chapters
Chapter 4: Manual Whisker Annotator for Validation
Chapter 5: Automated Rodent Tracker
Chapter 6: Whisker Detection
Chapter 7: Conclusion

Figure 1.1: The organisational structure for this thesis. The first three chapters introduce the research to be presented as well as a thorough literature review on previous work and techniques. The next 3 chapters are the contributions made in this thesis and the final chapter concludes this work.

1.6 Thesis Organisation

As shown in figure 1.5, this thesis is split into two main sections: introductory chapters and contribution chapters.

Chapter 2 presents fundamental knowledge and a review of the literature relating to neruodegenerative diseases, mouse and behavioural models of the diseases, the whisker system.

Chapter 3 provides the technical information on the techniques explored to create a fully automated rodent tracker. This includes basic image processing techniques, filters, edge detection, object detection, image scaling, Hessian matrix, and derivative techniques of the Hessian matrix including the Frangi filter. It also looks at current whisker tracking software and how recording whisking metrics will be implemented.

The following section includes three contribution chapters. Chapter 4 introduces the development of the Manual Whisker Annotator, which was used to validate the following trackers.

Chapter 5 looks at the development of the Automated Rodent Tracker (ART) and evaluates it against current body trackers. It also includes a demonstration of the results obtained from the software on the SOD1 dataset.

Chapter 6 focuses on the development of the whisker detection module added to ART to produce the Automated Rodent and Whisker Tracker (ARWT) and evaluates it against current state of the art whisker detectors. As with the previous chapter, the results obtained from the software on the SOD1 dataset are demonstrated.

Finally, chapter 7 concludes this thesis with a summary of of contributions, the limitations of ARWT and the future research direction.

Chapter 2

Background

2.1 Chapter summary

This chapter is in ten sections. Section 2.2 explains why further research into neurodegenerative diseases is needed, and the debilitating effect it has on human lift. Section 2.3 reviews how we can model these diseases in rodents and section 2.4 looks at how we can quantify their effects through behavioural models. Section 2.5 provides a detailed discussion on the whisker system and whisking behaviour. Software designed to automate the quantification of large datasets is discussed in section 3.3, and the theory and techniques needed to create an open source tracker are considered in sections 3.4. Lastly, section 3.5 discusses the literature reviewed and its relationship to this thesis.

2.2 Neurodegenerative diseases

Neurodegenerative diseases are often debilitating, in which the symptoms can be treated, but cannot be cured [56]. The term neurodegeneration encompasses a large number of disease conditions, which affect various parts of the nervous system, individually or in combination. Under this very general definition are included diseases of the central and the peripheral nervous systems and diseases of infancy, adulthood, and old age. In certain instances, the diseases have clear genetic causes, but in most cases they arise from a complex interplay of predisposing genetic and environmental factors. Neurons are particularly vulnerable to the toxic effects of mutant or misfolded proteins [217]; Alzheimer's disease (AD), Parkinson's disease (PD), Huntington's disease (HD), Amyotrophic lateral sclerosis (ALS), prion diseases, frontotemporal dementia, and motor neuron disease all share a common feature-aggregation and deposition of abnormal proteins [217, 182]. Despite this commonality, the hallmarks and symptoms of various neurodegenerative diseases generally differ distinctly. AD, for example, is a progressive neurological disease that is characterised by amyloid plaques and neurofibrillary tangles that are formed after misfolding and aggregation of proteins, respectively [56]. Clinical signs are behavioural and psychological symptoms of dementia (BPSD) [67]. PD, also a progressive neurodegenerative disease, which is associated by the loss of dopamine-producing cells in the central nervous system and is characterised by tremor, rigidity and bradykinesia [56]. HD [111] is inherited through an autosomal dominant gene [89]. The symptoms are progressive and characterized by both behavioural and motor disturbances which usually appear in adult life, with a duration typically spanning from 11-35 years [68]. Its prevalence varies between ethnicities, affecting 0.4 people per 100,000 in Asia, compared to 5.7 people per 100,00 in North America, Europe and Australia [192]. ALS, also known as Lou Gehrig's disease, is a specific disease that is characterized by the selective loss of motor neurons leading to progressive weakness, muscle atrophy with eventual paralysis and death within 5 years of clinical onset [117]. Approximately 10% of ALS cases are familial, the remainder ALS cases being diagnosed as sporadic (90%). The discovery of mutations in the gene coding for the Cu/Zn superoxide dismutase 1 (SOD1) accounts for 20% of familial cases and has directed most ALS research, and 114 different mutations have been discovered in the SOD1 gene that contribute to these cases [198].

Currently, for most neurodegenerative diseases, diagnosis only occurs after the neurons involved are already largely dysfunctional or dead [119]. The brain also has the ability to overcome large deficits, so symptoms tend not to appear until a large amount of damage has been done [162]. Scans such as MRI, CAT and PET can detect some of the physical symptoms, however, definitive results can only be obtained by a post mortem (or in exceptional circumstances a brain biopsy) [154]. This makes developing preventative treatment, and tracking the effects they have extremely difficult. Since treatments will be most effective at the onset of the disease, early diagnosis is essential to curing these diseases [213]. Early diagnosis would also allow the effects of drugs and/or treatments to be better monitored for their effectiveness.

2.3 Mouse models

The identification of genes mutated or deleted in the inherited forms of neurodegenerative diseases has allowed investigators to create *in-vivo* and *in-vitro* model systems relevant to a wide variety of human neurological disorders. Genetically engineered mice that recapitulate some of the features of human diseases have provided substantial progress in understanding the molecular mechanisms of neurodegeneration. They are also instrumental in identifying targets for mechanism-based therapeutics [242]. However, a considerable gap remains between the significant advances made in the understanding of the neurodegenerative diseases and the identification of new and effective therapies [115].

While genetically engineered mice have been, by far, the most popular and widely used animal model for the study of neurodegenerative diseases [115], the tests carried out on the mice are intrusive and often involve making in-situ recordings from the target organ [218, 221]. Generating measurable behavioural models would reduce animal numbers and increase longitudinal studies of disease progression [228].

2.4 Behavioural models

Behavioural models are used in animals with the belief that they can model human behaviours. Behavioural models fall into three classes: to model brain functions, psychological processes, or drug actions [239]. Several behavioural models used to model the effects of neurodegeneration on brain function are described in further detail below, with a focus on how behaviours change with respect to AD, PD, HD and ALS.

2.4.1 Sensory models

Eyesight

Eyesight has been shown to be affected by the onset of AD, PD, HD and ALS, either as a result of retinal degeneration, or the development of cataracts within the lens [97, 156, 140]. Various tests were conducted to test different aspects of a mice's vision. These included physical observation of the eye, such as the thickness of the retinal wall [97], staining optic nerves for review under a microscope [140], or observing the iris with a microscope [140, 156]. Other tests not related to physical properties of eye are also conducted, such as suspending a mouse by the tail to a height of approximately 15 cm and lowered to a metal grid within seconds. Scoring is based on the distance of the animal's nose from the grid before extending the forelimbs toward it [188], or by employing what's known as the visual cliff test, which measures the ability of a mouse to the drop-off at the edge of a horizontal surface [42]. All mice suffered from some form of vision impairment across all diseases, with the exception of PD models, where no differences in retinal function between transgenic and wild type mice were observed [219]. However, in other models, the retina in ALS and HD mouse models had morphological and functional alterations that increased with age [97, 100]. Mouse models of AD were shown to develop far more cataracts than their wild type counterparts [156].

Smell

Olfactory dysfunction is an early clinical sign of AD, PD and HD [148, 58, 137], with only slight olfactory deficits associated with ALS [199]. Presently, the reasons for the olfactory deficit in such disorders is unknown [148], although deposits of various proteins associated with these diseases have been found in the olfactory bulb [50, 176]. Mouse models that overexpress these proteins provide a potential means for determining these observed olfactory deficits. To quantify olfactory dysfunction, physical and behavioural tests were conducted. Physical tests performed included staining of the olfactory bulb for viewing with a microscope [148]. Various behavioural tests exist, with most of them consisting of some form of introduction to a new odour, and the latency or investigation time of the odour recorded [148, 247]. Microscopic analysis of the olfactory bulb revealed higher concentrations of several proteins associated with these diseases [128, 148]. Behavioural tests showed a decrease in time spent investigating new odours, and an increase in latency before the first sniff of a new odour [148, 137].

Hearing

Hearing impairment has been shown to be another early clinical sign of AD, PD, HD and ALS [141, 120, 230, 226]. Several mechanisms have been proposed for the association between auditory impairment and ND diseases, either through degeneration of ear structures such as the hair cells [120], or where auditory stimuli is not adequately processed by the central nervous system [226]. The acoustic startle response [46] is widely used to quantify hearing impairment. Mice are placed in a cylinder of a chamber and a white noise (70-75 dB) is delivered as background level. Brief tones from 75 to 120 dB are then delivered randomly. Amplitude of the whole body flinch in response to tones is recorded [145]. Hair cell degeneration was found to be more prevalent in models of ALS [120], and and a reduction in startle response was recorded in AD, PD and HD [225, 232, 31, 2].

2.4.2 Exploratory Behaviour

Mice spend long periods exploring new environments, driven by an attraction to novel stimuli [44]. When a rodent explores a novel environment, it will learn the areas where to find food or water, and will return to these places more quickly than an animal with no such previous experience [21]. Exploration can therefore be defined as "*exploration is evoked by novel stimuli and consists of behavioural acts and postures that permit the collection of information about new objects and unfamiliar parts of the environment*" [43]. Spatial information collected during exploration allows the mouse to construct an internal representation of the environment within their hippocampus [177], which has been shown to be affected by AD, PD, HD and ALS [168, 161, 27, 105]. Mazes have played an important role in the development of psychology, as they often introduce novel stimuli and therefore exploratory behaviour, and are particularly suited for analysis of several types of cognitive function such as spatial memory, mental imagery and reasoning [178]. Many different types of mazes have been developed [163, 13, 194, 51, 178], Some mazes define a path for the mouse to follow, either by the use of walls tall enough to prevent the mouse climbing over, or by the use of elevated paths placed far enough part to discourage the mouse from jumping across [178]. Other mazes don't define any paths, but consist of an open arena where the rodent is tasked to find a target area, either directly from memory or from the use of visual cues [163, 13]. Below two types of maze are looked at, the T-Maze (and also a slight variation known as the Y-Maze), and the Morris Water Maze.

T and Y-Maze

The T-Maze [194] and Y-Maze [51] are two behavioural tests measuring exploratory behaviour through the willingness of mice to explore new environments. Both mazes consists of three arms, with the arms arranged in either a "T" or "Y" shape as shown in figure 2.1a and 2.1b respectively.

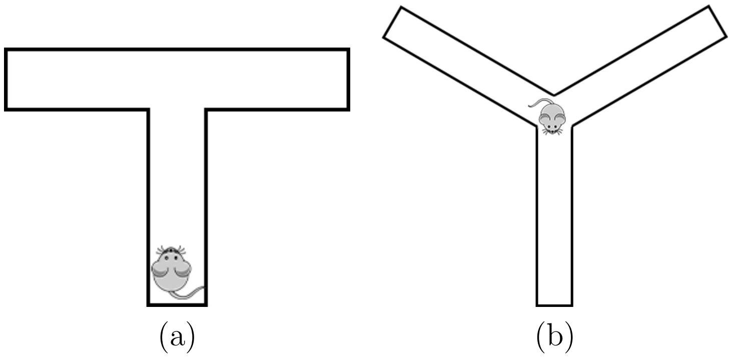

(a) (b)

Figure 2.1: An example of (a) The T-Maze; (b) The Y-Maze

Both mazes come in two variations: spontaneous alternation and forced alternation [49, 134, 240, 39, 205, 35, 110]. For spontaneous alternation, the mouse is placed within the maze, with the starting location for each type of maze show in figure 2.1. For "T" mazes, the mouse can freely choose one of two arms. Once a choice has been made, the mouse is prevented from leaving the arm for a pre-determined amount of time, and then removed and placed back in the starting location. The mouse is then again allowed to freely choose one of the arms. This procedure is repeated many time, and for each time, if the mouse chooses the other arm, it is recorded as a spontaneous alternative behaviour [234, 110]. The procedure for the "Y" maze differs slightly, with the mice being placed in the center of the maze, they have free choice over 3 different arms. Instead of being put back to the starting location after an arm is chosen, mice are allowed to explore the maze for a set amount of time, with each arm the mouse explores being recorded. The spontaneous alternation behaviour is recorded as the number of three successive choices that include on instance of each arm, divided by the total number of opportunities for alternation [110]. Forced alternation versions of these mazes work by blocking off one of the arms, allowing access only to a specific arm. For the "Y" maze, the starting location changes to the base of one of the arms [240]. After the mouse was allowed to explore the arms, it was returned to its home cage for a pre-determined amount of time, and then returned back to the maze with the block remove, allowing access to all arms. Forced alternation behaviour was defined as the percent of mice entering the unblocked (unexplored) arm first [205, 240, 35]. The T and Y Maze have proven to be popular ways to study spatial memory in mouse models of neurodegenerative diseases, and have shown that for mouse models of AD, PD, HD and ALS, the alternation percentage decreased [240, 66, 22, 184, 181, 107]. Some studies however have found no differences in alternation percentage for specific genotypes of these diseases[240, 107].

Morris water maze

Another type of maze designed to test spatial memory is the Morris Water Maze [163]. The maze is essentially a circular pool of water containing a hidden platform. The platform is hidden by ensuring it is just below the water line, and by using an opaque liquid (water coloured with non-fat milk or non-toxic tempera paint [173]), thus rendering the platform invisible and ensuring no local cues can be used to help locate the platform. With no prior knowledge of the maze, rats and mice swim randomly looking for the platform, however, normal rats and mice learn very quickly to swim directly towards the platform from any starting position at the edge of the pool, as shown in figure 2.2 [163]. The two most common metrics recorded are time to reach the platform (escape latency), and the length of the path swam (path length). To a much lesser degree, swimming velocity is also used, and another variation of the maze whereby after repeated attempts at the maze, the platform is removed, and the time spent swimming in the quadrant where the platform was is recorded [53]. AD, PD, HD and ALS have all been shown to increase escape latency and path length [53, 143, 187, 248].

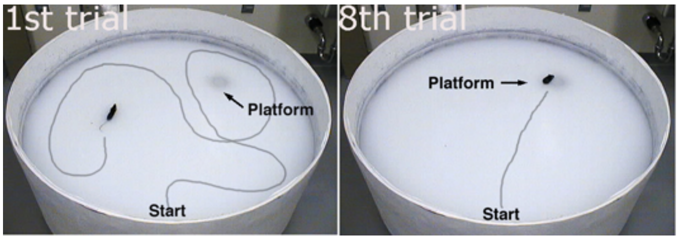


Figure 2.2: The Morris Water Maze

2.4.3 Motor Control

The pathways in the brain affected by neurodegeneration do not only lead to cognitive declines as described above, but also result in motor impairment. AD, PD, HD and ALS display movement abnormalities such as chorea, dyskinesia, dystonia, rigidity, tremor, bradykinesia, muscle twitches, spasticity, impaired posture and balance, and a loss of automatic movements [31, 92, 154, 17]. Several motor related behavioural models are described in this section.

Gait analysis

Gait analysis reflects the characteristics of the animal's walking pattern [38]. In its most sophisticated form, analysis of gait includes the collection of ground reaction forces and kinematic and electromyographic data [243]. The difficulty of scaling and applying these techniques to mice has limited their use despite a steady increase in studies of the rodent motor system [127]. A simple but related method that has been successfully used in mice is footprint analysis. Obtaining the footprints usually requires application of ink to the animal's paws, with the resulting footprints being used to measure static gait parameters such as stride length [193]. Wooley et al. (2005) has shown that transgenic mice had significantly longer stride and stance times than controls by 8 weeks. Consistent with disease progression, hindpaw measures of transgenic mice showed larger changes than front paws.

Balance

Two methods for evaluating motor skill methods are the balance beam and rotarod. The balance beam is a narrow length of material that a rodent can traverse across [64]. It is divided into ten segments of 10 cm each with a black marker. Each mouse is placed in the center segment and is allowed to walk on the beam for 60 seconds. The number of segments crossed and the latency to fall are recorded. Slips from the beam may also be counted. The rotarod also measures balancing ability, but instead of using a beam for the rodent to traverse, it must keep itself balanced on a rotating beam [114]. Studies have shown balance is significantly affected by neurodegenerative diseases, with control rodents performing significantly better in both tests[65].

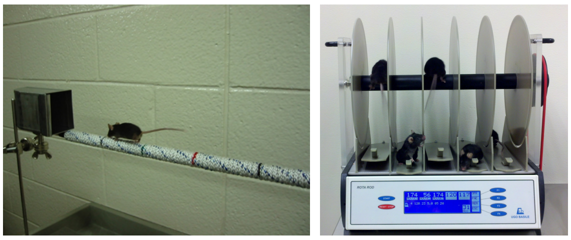


Figure 2.3: The balance beam and rotarod tests used to measure balance [133, 12]

Muscular strength

Muscular strength can be used to determine disease progression. There are two popular techniques used to measure muscle strength: Hanging wire [130] and direct force measurements [139]. For the hanging wire test , rodents are placed in the middle of a metal grid. The grid is gently shaken so the animal grips firmly to it, then the grid is placed upside down at a height of 30-40cm with padding underneath to prevent any potential injury by falling. The amount of the time the rodent can suspend itself before falling is measured to determine strength [130]. Direct force measurements involve placing the animal onto a metal bar that can measure force, and allowing it to grip firmly to it. The rodent is then gently pulled away from the bar. The force recorded just prior to the rodent releasing its grip is used to determine strength. For tests on experimental mice, muscular strength has been shown to deteriorate as the diseases progresses [207].

2.4.4 SOD1 model

Transgenic mice overexpressing SOD1 mutants linked to familial ALS develop progressive motor neuron disease with many pathological features observed in both familial and sporadic ALS cases [36]. Specifically, the gene was mutated by substituting a glycine93 → alanine (G93A). Wild-type mutations of the SOD1 gene show no signs of overt motor neuron disease but do have mild pathologic changes in the innervation of muscle that are suggestive of premature aging [36]. This mouse model has an onset of paralysis at 90 days, accompanied by degenerative changes to motor neurons that compare well with human ALS pathology [216], and death by 135 days, depending on genetic background. The SOD1G93A mouse has been widely used for research ranging from basic molecular cell biology through to extensive drug trials [36]. This model was the first mutant SOD1 transgenic produced, and is freely available, which adds to its popularity.

2.4.5 Locomotion and whisking

Locomotion describes a variety of movements that the rodent uses to move from one place to another, which include walking, running, burrowing, climbing and swimming. Locomotion is relatively easy to analyse compared to other techniques. It can be done manually by an observer, although this can be tedious it is by no means impossible. Another more popular alternative is to automatically analyse locomotion with the use of a camera and tracking software. Whisking is a prominent behaviour observed in most rodents with the exception of a few species [245]. As whiskers are controlled by muscles, it stands to reason that the whisking behaviour would be affected by the onset of a neuro-degenerative disease. However, current methods to accurately record whisker movements are either very expensive, require the mouse to be modified in some way (whiskers cut to a single row for example, or restraining of the head) or the measurements have to be done by hand.

2.5 The whisker system

2.5.1 Anatomy of the whisker system

The whiskers or *vibrissae* are prominent sinus hairs found on almost all mammals [159]. Vibrissae differ from ordinary hairs by being longer and thicker, having large follicles containing blood-filled sinus tissues, and by having an identifiable representation in the somatosensory cortex. Vibrissae are found on various parts of the body, but those most frequently studied are the facial or mystacial vibrissae, also called whiskers. Long facial whiskers, or macrovibrissae, are found in many mammalian species, projecting outwards and forwards from the snout of the animal to form a tactile sensory array that surrounds the head [191].

Vibrissal follicles are surrounded by striated muscles that have been grouped in two categories: extrinsic and intrinsic muscles. Extrinsic muscles belong to the group of facial muscles moving the upper lip and the wing of the nose. Some of these muscles are so closely linked to the mystacial region that the latter may be considered as their principal insertion zone. They are called extrinsic because of their origins on different regions of the skull outside the mystacial region. Intrinsic (follicular) muscles are associated solely with the vibrissal follicles and have no bony attachment. They are found around follicles α, β, γ, δ, around all follicles of rows A and B, and around the first six follicles of rows C, D and E as shown in Figure 2.4. The form of the follicular muscle is a sling connecting two adjacent follicles of the same row. The arc of the sling surrounds the inferior part of the rostral follicle and the two extremities insert to the conical body of the caudal follicle and to the neighbouring corium. They are the protractors of the vibrissae. The inferior parts of the vibrissal follicles of a given row are fixed in a fibrous band which inserts in the anterior part of the muzzle. It is proposed that these bands become stretched during the protraction of vibrissae and contract, by their elasticity, immediately upon the end of the follicular muscles' contraction, executing the fast return of vibrissae to their resting, retracted position [57].

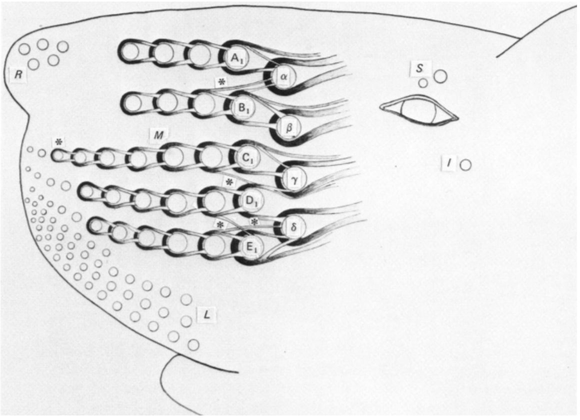


Figure 2.4: Schematic drawing of the intrinsic musculature of the left mystacial region[57].

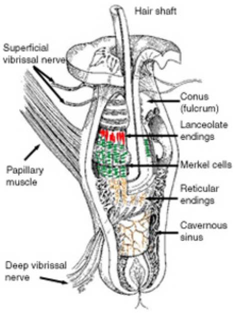


Figure 2.5: The anatomy of a whisker folicle[48]

2.5.2 Whisking behaviour

The characteristic pattern of whisker movements is for all the whiskers to move forward (protraction), followed by the whiskers moving backwards (retraction). This sweeping motion, also known as whisking, happens many times a seconds, usually over a period of a few seconds. Cinematographic measurements of whisker movements generated by behaving rats were compared with electromyographic (EMG) activity recorded simultaneously from mystacial pad musculature [160]. Muscle activity consists of repetitive bursts, each of which initiated a "whisking" cycle consisting of a protraction followed by a retraction. Protraction amplitude and velocity are directly proportional to the amount of EMG activity during forward whisker movement. Over time, the intensity of muscle discharge determines the set point about which the vibrissae moved; higher levels of muscle activity results in a greater degree of overall whisker protraction. These findings are consistent with the known anatomy of the facial musculature and underscore the importance of whisker protraction in the acquisition of tactile information by the vibrissae [32].

In rats and mice the facial whiskers are repetitively and rapidly swept back and forth, during many behaviours including locomotion and exploration. These whisking movements generally occur in bouts of variable duration, and at rates between $3 - 25Hz$ (with rates generally being somewhat faster in mice than in rats). Movements of the whiskers are also closely coordinated with those of the head and body allowing the animal to locate interesting stimuli through whisker contact, then investigate them further using both the macrovibrissae and an array of shorter, non-actuated microvibrissae on the chin and lips [191].

Since the rapid movement of the vibrassae consumes energy, and has required the evolution of specialised muscles, it can be assumed that whisking conveys an advantage for the animal, and subsequent studies have confirmed that whisking is a form of active sensing [159]. Whisking is a very complex process, with the small, thin whiskers moving very fast, and delivering a wide array of information back to the brain about an object's shape, size, texture and direction if the object in question is moving.

As well as whisking, other aspects of whisker control can also be seen in rodents. Extensive studies laboratory rats and mice have revealed that whisker movements can be actively controlled during locomotion and object exploration, in particular that the spacing, speed and position of the whiskers can be changed [85, 160]. Indeed, during object exploration, rats reduce the spacing, or spread, of their whisker, so that they bunch up on a surface and enable more whisker contacts [85, 86]. They also reduce the angular position and retraction speed following a contact [85, 86]. This enables gentle touches against a surface, and increases the duration of surface contact. Asymmetry, or more specifically contact-induced asymmetry, also often occurs following a unilateral contact. It is characterised by the whiskers contralateral to the contact increasing in amplitude and the whiskers ipsilateral to the contact decreasing in amplitude, enabling asymmetry between the two whisker fields [160]. Therefore, it is important to measure elements of whisker angle, amplitude, speed and spread during recordings of whisker behaviour.

Vibrissae play such an important part in behaviour, that experiments in adult rats and shrews involving whisker clipping, cauterization of the whisker follicle, section of the peripheral nerves, or lesion of critical structures in the vibrissal pathway, have found significant deficits in exploration, thigmotaxis, locomotion, maintenance of equilibrium, maze learning, swimming, locating food pellets, and fighting. Posture, social, and developmental behaviours are also affected following whisker removal or damage [90, 5, 191]).

Changes in behaviours such as maze learning, food finding, gap crossing, and fighting, are presumably the consequence of the loss of fundamental tactile sensory abilities that use vibrissal signals. Specifically, the ability to localise, orient to, and track objects or surfaces in space, and to discriminate between objects or surfaces based on their tactile properties.

2.6 Conclusion

Using behavioural models in mice to increase our understanding of ND diseases will provide an important contribution to diagnosing and monitoring neurodegenerative diseases, and will also allow easier analysis of new treatments and their efficiency. Furthering the locomotion and whisking models in particular will reduce animal testing numbers, as well as reducing any suffering that might inflict the mouse. Recording locomotion and whisking behaviour is currently a long tedious task, and the ability to automate this process would bring a significant increase to our knowledge in this area. This project aims to do exactly that, by developing software which can automatically quantify locomotion and whisking characteristics; and using the software to evaluate these characteristics over a range of ND diseases. While locomotion characteristics refer to the forms of movements that animals naturally use, whether it be running, flying, swimming, or hopping, we will specifically be looking at the speed, distance travelled, and frequency of movement. Several whisking behaviours were found to be affected as the disease progressed. The parameters that were affected most were the whisker starting position (offset value), move more (larger amplitude), move slower (lower frequency) and retract faster (retraction speed). Protraction speed was not found to be significantly altered by the onset of the disease.

Chapter 3

Background Theory and Techniques

3.1 Chapter summary

The software covered in section 3.3 all relied on fundamental, and more advanced image processing techniques. This chapter focuses on the techniques used, and introduce other techniques that are used throughout the thesis, covering basic image processing, filtering algorithms, object detection, image scaling and feature extraction. It will summarise existing trackers and state recommendations for this thesis.

3.2 Image Processing Techniques

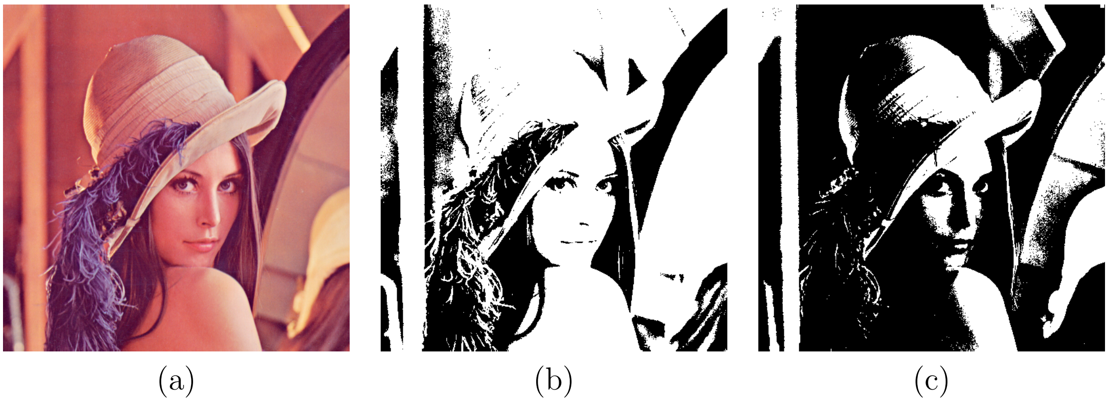
3.2.1 Binary Images

A binary image is a digital image that has only two possible values for each pixel, either 0 or 1 (black or white respectively), and thus has a bit depth of 1 bit.

Thresholding

Converting an image into a binary image is possible through the use of thresholding. If the pixel P value is larger than a Threshold T, the new pixel P_n is assigned a value of 1, otherwise it is assigned a value of 0 (equation 3.1)

$$P_n = \begin{cases} 1, & \text{if } P > T \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$



(a) (b) (c)

Figure 3.1: (a) Original image; (b) Threshold image ($T = 100$); (c) Threshold image ($T = 150$)

AND, OR, NOT

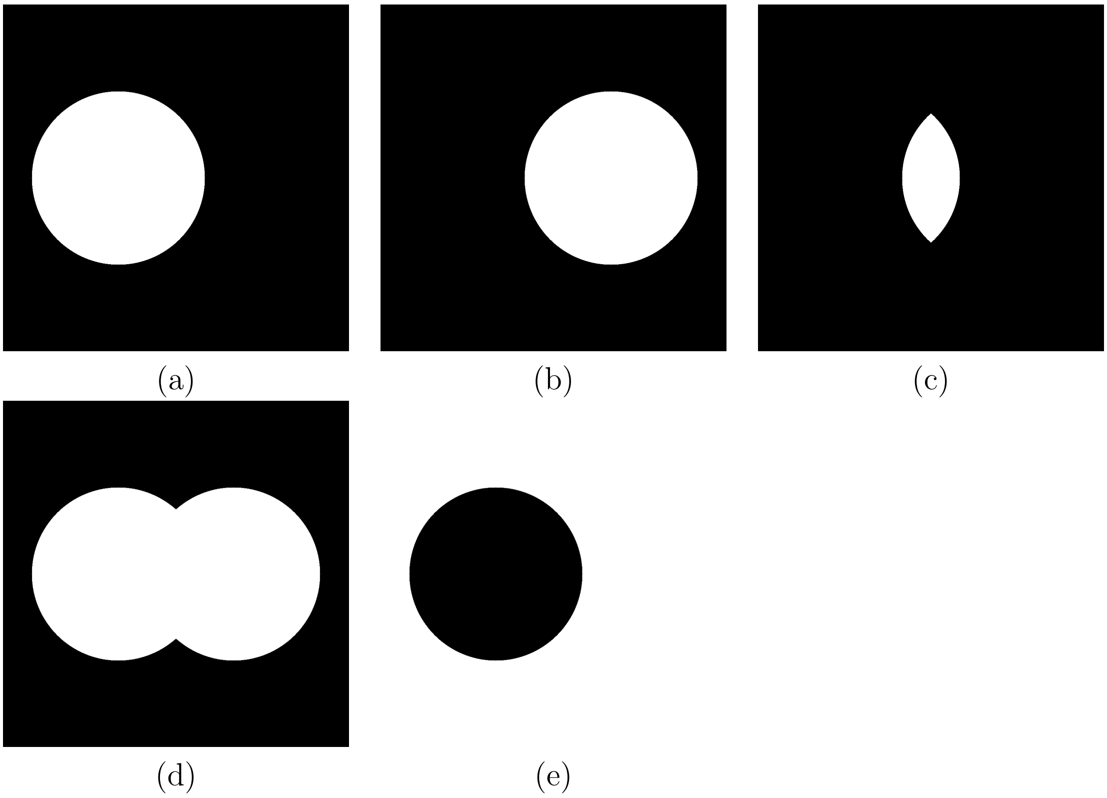
And, Or and Not are logical operators that can be applied, but not limited to, binary images.

P_1	P_2	Result
0	0	0
0	1	0
1	0	0
1	1	1

P_1	P_2	Result
0	0	0
0	1	1
1	0	1
1	1	1

P_1	Result
1	0
0	1

Table 3.1: AND truth table (left), OR truth table (center) and NOT truth table (right)

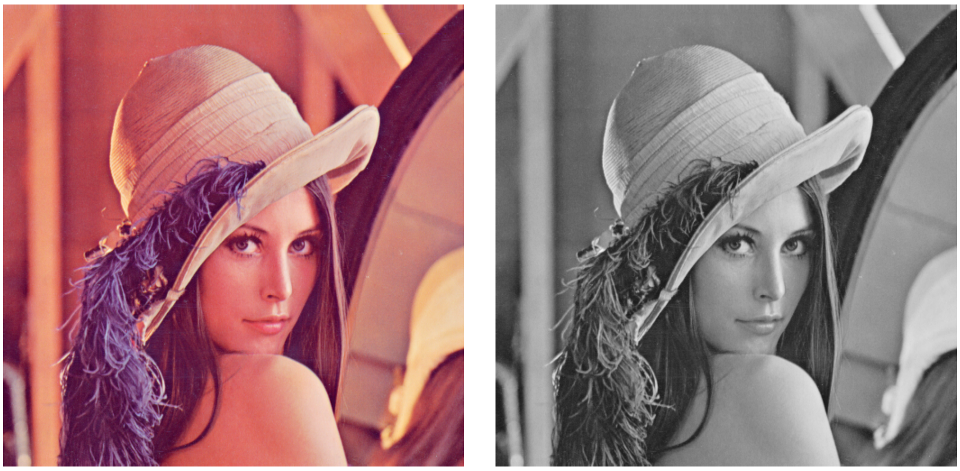


(a) (b) (c) (d) (e)

Figure 3.2: (a) Image 1; (b) Image 2; (c) Image 1 AND Image 2; (d) Image 1 OR Image 2; (e) NOT Image 1

3.2.2 Greyscale Images

Greyscale images contain more than just black and white pixels. They include shades of grey. In a greyscale image, each pixel has more information encoded in it than a binary value, allowing more shades to be recorded. The number of grey levels that may be represented is 2^n, where n is the bit depth of the image (256 for an 8 bit image).



(a) (b)

Figure 3.3: (a) Original Image; (b) Greyscaled image

3.2.3 Filtering Algorithms

Filtering is a technique for modifying or enhancing an image by emphasizing or removing certain features, and is represented as a matrix of numbers (known as a kernel). This matrix is applied to each pixel of an image and its surrounding neighbours, to produce another image, this process is known as convolution [147]. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement. This section looks at smoothing filters designed to eliminate noise and also filters designed to enhanced certain features.

Gaussian Filter

The Gaussian Filter suppresses signal noise using a lowpass filter that approximates a gaussian profile as shown in figure 3.4 [47]. For a 1 dimensional signal, the gaussian profile takes the form shown in equation 3.2, where σ is the standard deviation of the distribution.

$$g(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \tag{3.2}$$

For a 2 dimensional signal, such as an image, the equation takes the form shown in equation 3.3, with the profile shown in figure 3.5.

$$g(x,y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.3}$$

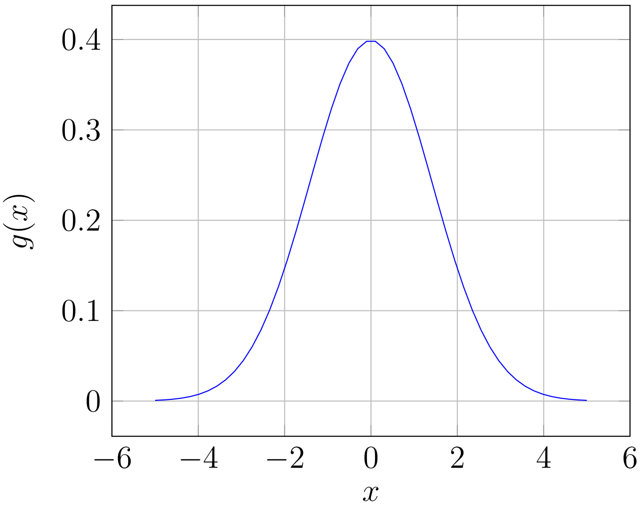


Figure 3.4: Gaussian Profile where $\sigma = 1$

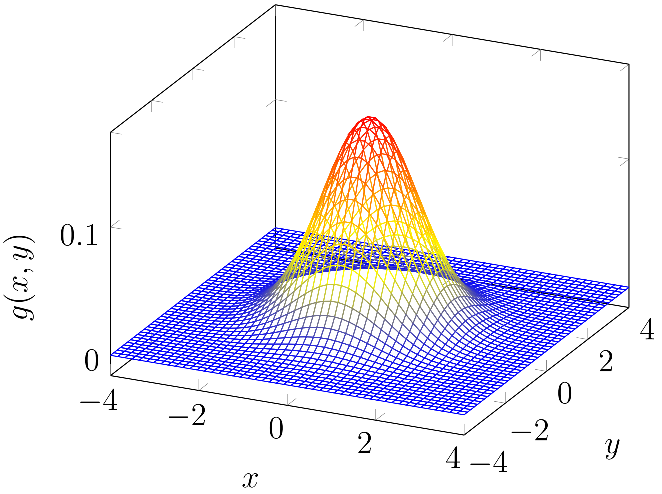


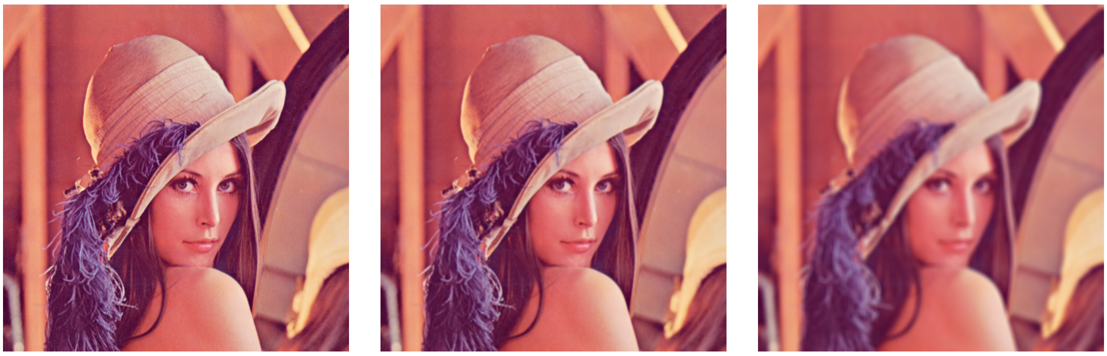
Figure 3.5: 2 dimensional Gaussian Profile where $\sigma = 1.0$

The idea of Gaussian smoothing is to use this 2-D distribution as a point-spread function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. Figure 3.6 shows a suitable integer-valued convolution kernel that approximates a Gaussian with $\sigma = 1.0$. It is not obvious how to pick the values of the mask to approximate a Gaussian. One could use the value of the Gaussian at the centre of a pixel in the mask, but this is not accurate because the value of the Gaussian varies non-linearly across the pixel. The integrated value of the Gaussian profile over the whole pixel (by summing the Gaussian at 0.001 increments) is chosen, and then these values are rescaled so that the corners have a value of 1. Finally, the 273 is the sum of all the values in the mask, this ensures that there is no intensity change.

1/273

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 3.6: Discrete approximation to Gaussian function with $\sigma = 1.0$

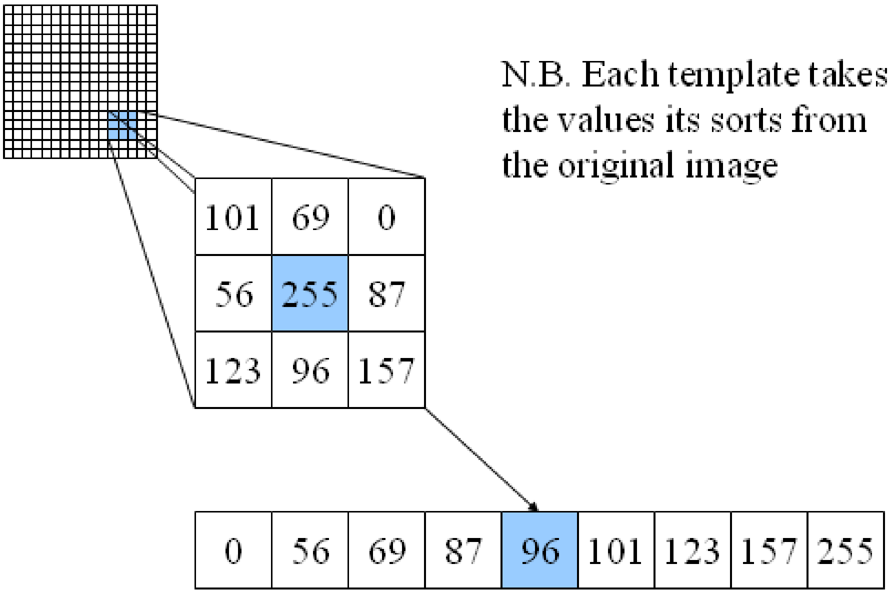


(a) (b) (c)

Figure 3.7: The effects of Gaussian Smoothing (a) Original image; (b) Output image with $\sigma = 1, kernel = 3$; (c) Output image with $\sigma = 5, kernel = 9$.

Median Filter

The median filter [224, 109] is another widely used filter. This filter has an added advantage over the Gaussian filter as under certain conditions it can preserve edges while removing noise. As with the Gaussian filter, this filter works by evaluating each pixel based on its neighbours. But instead of using a function to obtain the final value, the median value of all the surrounding pixels is chosen. An example is shown in figure 3.8 using a 3x3 kernal.

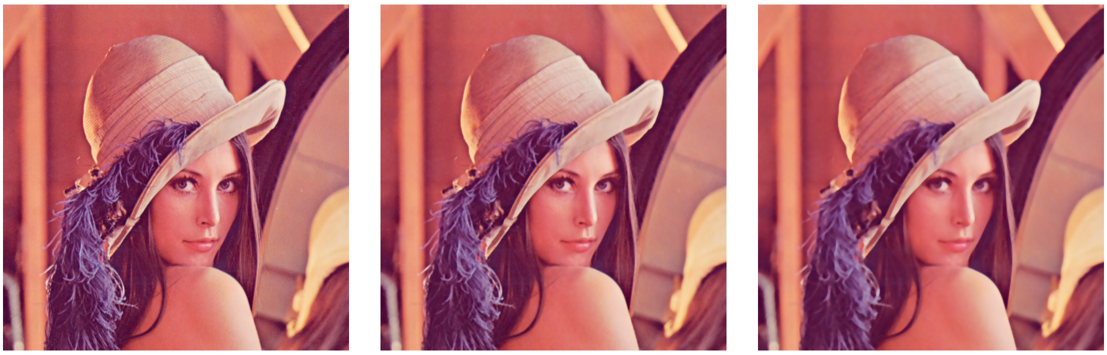


N.B. Each template takes the values its sorts from the original image

101	69	0
56	255	87
123	96	157

0	56	69	87	96	101	123	157	255

Figure 3.8: Visual representation of how the pixel is chosen in a median filter



(a) (b) (c)

Figure 3.9: (a) Original image; (b) Output image of median filter with 3x3 kernel; (c) Output image of median filter with 5x5 kernel

3.2.4 Edge Detection

Edge detection is a technique in image processing for finding object boundaries by detecting variances in brightness.

Sobel

The Sobel-Feldman Operator (also known as the Sobel Filter) is used to emphasise edges [208]. It computes the intensity gradient at each point in the image. The result of the Sobel Filter is either the corresponding gradient vector or the normal of this vector. The Sobel Filter is based on convolving the image with a small, separable, and integer valued filter in the horizontal and vertical directions and is therefore relatively computationally inexpensive compared to other filters. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image. Two 3x3 kernels are used which are then convolved onto the original image, one for horizontal derivatives (G_x) and another for vertical derivatives (G_y) and are shown in Figure 3.10.

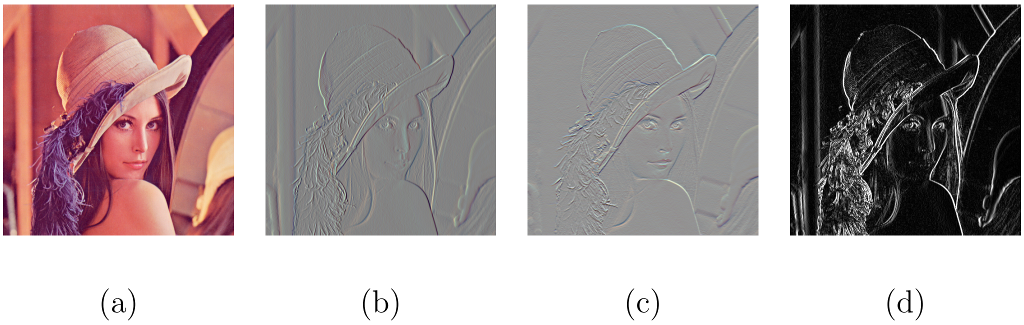
$$G_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad G_y = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Figure 3.10: Horizontal (G_x) and Vertical (G_y) kernels for the Sobel Filter

Both these kernels can be defined as the increase in intensity, with G_x representing the increase in the "right" direction, and G_y representing the increase in the "down" direction. By using these directional intensity changes, we can calculate the gradient magnitude G using equation 3.4, and the gradients direction θ using equation 3.5.

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{3.4}$$

$$\theta = atan2(G_x, G_y) \tag{3.5}$$



(a) (b) (c) (d)

Figure 3.11: (a) Original Image; (b) G_x; (c) G_y; (d) Normalised Gradient Magnitude

Canny

Canny edge detection [30] is a multi step process that has 5 distinct steps.

1. Apply Gaussian Filter to remove noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges

After applying a Gaussian filter as described in section 3.2.3, the intensity gradients of the image are found using the Sobel operator as described above. After applying these two steps, an image with thick edges is generated. Edges should only be marked once, and to reduce the thickness non-maximum supression is applied.

Non-Maximum Suppression is an edge thinning technique, where the highest magnitude pixel in the perpendicular axis to the edge is chosen to represent the edge [122]. The first implementation of Non-Maximum Suppression would round the edge angle to one of 4 values, 0, 45, 90, 135, degrees. If the gradient angle was 0 degrees (the edge is running North to South), the pixels to the left and right would be used to determine the local maximum(Figure 3.12a). If the gradient angle was 45 degrees (edge running North-East to South-West), the pixels to the upper left and lower right would be used (Figure 3.12b). Later interpretations of this method do not round the gradient angle, and interpolate between the chosen pixels intensity.

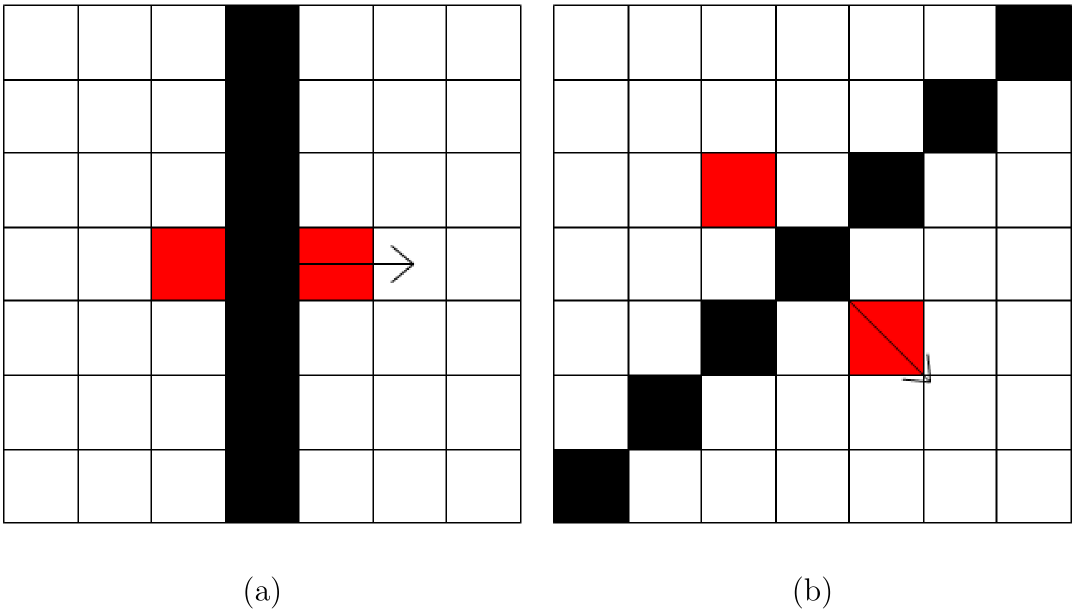


Figure 3.12: Local pixels to determine non-maxima suppression for (a) 0 degrees; (b) 45 degrees

Once an image has had potential edges marked, some of these potential edges will be from noise and should be removed. To do this, the first step is to apply Double Thresholding. Two threshold values T_{High} and T_{Low} are set. If the gradient intensity is less than T_{Low}, the pixel is suppressed. If the gradient intensity is larger than T_{Low} but smaller than T_{High}, it is classed as a weak edge, and finally if the gradient intensity is larger than T_{High} it is classed as a strong edge. Strong edges are taken as definite edge points, and weak edges will still be considered as potential points.

The final step is to distinguish the weak edges between these real and false, and to do that hysteresis is used. Hysteresis checks the 8 neighbouring pixels of a weak edge point, and if at least one of these points contains a strong edge pixel, the pixel will become an edge. If none of the neighbouring pixels are a strong edge, then the pixel is suppressed. Figure 3.13 shows 3 weak edge pixels. As pixel 1 is touching a strong edge, it will be accepted. Pixel 3 would be suppressed as it is not touching a strong edge. In the initial implementation of hysteresis, pixel 2 would also be suppressed as it is not touching a strong edge. However, an improvement has been developed which would traverse all weak edges, from a strong edge starting positioning [175]. This implementation would accept pixel 2 as an edge.

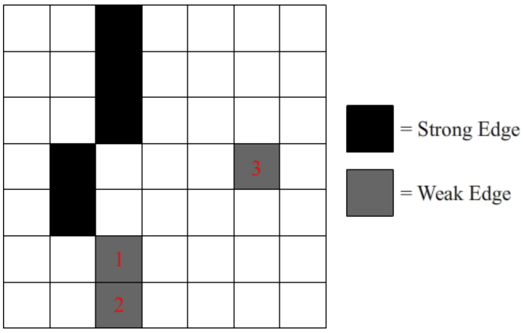
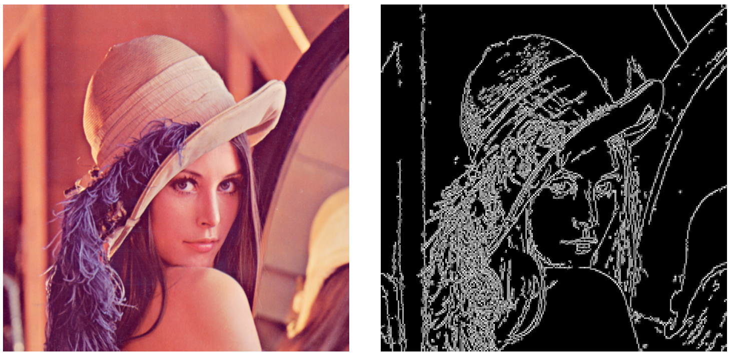


Figure 3.13: Hysteresis edges



(a)

(b)

Figure 3.14: The effects of Canny edge detection (a) Original Image; (b) Detected Edges.

3.2.5 Object Detection

Object detection is the process of localising objects in images or videos, typically by the use of looking for specific features. There are a wide array of methods and techniques available to detect objects in images, and they have their strengths and weaknesses, and in certain cases, novel techniques may need to be developed to overcome the accuracy and performance limitations.

Template Matching

Template matching [60] attempts to find an object from a target image or template. It uses what's known as the sliding window approach, in which a window of smaller size than the image, scans from left to right and top to bottom in an attempt to locate the target image. This approach means that the method is not scale-invariant, nor rotationally-invariant. It is faster than other methods described below, with detection times as low as $\approx 40ms$ [76]. However, due to the method using a template, it is sensitive to illumination and occlusion.

SIFT & SURF

Scale-Invariant Feature Transform (SIFT) [210] and Speeded Up Robust Features (SURF) [15] are two methods of feature based detection. Both methods describe local features of a target object, then use these features to try and find the object within an image. They are both scale-invariant, meaning it does not matter how large the target object appears. SURF is partly inspired by SIFT, but is claimed to be several times faster and more robust, managing to find objects in $\approx 255ms$ and $\approx 1036ms$ respectively [15]. However, depending on other factors such as illumination, blur, scale and rotation, SIFT can still match SURF and in some cases still outperform it [116]. These methods require the target object to have enough defined features that it can accurately describe it. If the object does not have enough features then these methods will fail.

Contour Matching

Contour Matching [144] can detect partially distorted 2D shapes without regard to their orientation, location, and size. The algorithm first calculates the curvature function from the digitized image of an object. The points of local maxima and minima extracted from the smooth curvature are used as control points to segment the boundary and to guide the boundary-matching procedure. The boundary-matching procedure considers two shapes at a time, one shape from the template databank, and the other from the object being classified. The procedure tries to match the control points in the unknown shape to those of a shape from the template databank, and estimates the translation, rotation, and scaling factors to be used to normalize the boundary of the unknown shape. The chamfer 3/4 distance transformation and a partial distance measurement scheme constitute the final step in measuring the similarity between the two shapes. The unknown shape is assigned to the class corresponding to the minimum distance. The algorithm has been successfully tested on partial shapes using two sets of data, one with sharp corners and the other with curve segments. This algorithm not only is computationally simple, but also works reasonably well in the presence of a moderate amount of noise.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification [132]. They work similar to other Neural Networks, by taking an input, and applying learned weights to these, to produce an output or to pass on to another layer [98]. However, a convolutional layer in a network will create (or learn) a kernel, rather than a series of weights, to apply to the input image [200] before passing on to the next layer.

Background Subtraction

Background Subtraction [223] (also known as Foreground Detection) is widely used for detecting moving objects from static video cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called the "background image" or "background model". As a basic, the background image must be a representation of the scene with no moving objects and must be kept regularly updated so as to adapt to the varying luminance conditions and geometry settings. More complex models have extended the concept of "background subtraction" beyond its literal meaning.

In the simplest implementation, known as Adjacent Frame Differencing [223], background subtraction takes the pixel intensity (P) of the current image ($I(t)$), and subtracts the reference (background) image (B) from it.

$$P = P[I(t)] - P[B] \tag{3.6}$$

This difference image would only show some intensity for the pixel locations which have changed in the two frames. Though we have seemingly removed the background, this approach will only work for cases where all foreground pixels are moving and all background pixels are static. A threshold Th is applied on the difference image to improve the subtraction, and to try and remove as many false positives as possible that may be due to luminance changes.

$$|P[I(t)] - P[I(t-1)]| > Th \longrightarrow \mathit{Foreground} \tag{3.7}$$

$$|P[I(t)] - P[I(t-1)]| \leq Th \longrightarrow \mathit{Background} \tag{3.8}$$

Adjacent Frame Differencing however provides a background image that is susceptible to sudden changes in appearance, whether that be luminosity or an object such as tree moving [223]. One way of reducing these errors is to generate a background image made from a series of preceding images that are averaged using a mean filter. For calculating the background image at the instant t,

$$B(x, y, t) = \frac{1}{N} \sum_{i=1}^{N} V(x, y, t - i) \tag{3.9}$$

where N is the number of preceding images taken for averaging. This averaging refers to averaging corresponding pixels in the given images. N would depend on the video speed (number of images per second in the video) and the amount of movement in the video. After calculating the background $B(x, y, t)$ we can then subtract it from the image $V(x, y, t)$ at time $t = t$ and threshold it. Thus the foreground is

$$|V(x, y, t) - B(x, y, t)| > Th \longrightarrow \textit{Foreground} \tag{3.10}$$

$$|V(x, y, t) - B(x, y, t)| \leq Th \longrightarrow \textit{Background} \tag{3.11}$$

where Th is threshold. Similarly we use median instead of mean in the above calculation of $B(x, y, t)$.

Usage of global and time-independent Thresholds (same Th value for all pixels in the image) may limit the accuracy of the above two approaches.

Another method for generating the background image called Running Gaussian Average [246], calculates the background independently at each pixel location. The model is based on ideally fitting a Gaussian probability density function (PDF) on the last n pixel's values. For optimisation purposes, the pdf is not created from scratch on each new frame, instead a running average (or on-line cumulative average) is used. As the background may change over time (e.g. due to illumination changes or non static background objects), every pixels mean and variance is calculated as follows:

$$\mu_t = \rho I_t + (1 - \rho)\mu_{t-1} \tag{3.12}$$

$$\sigma_t^2 = d^2 \rho + (1 - \rho)\sigma_{t-1}^2 \tag{3.13}$$

$$d = |(I_t - \mu_t)| \tag{3.14}$$

where I_t is the pixel's current value at time t and μ_t is the previous average; ρ is an empirical weight often chosen as a trade off between stability and speed (usually

$\rho = 0.01$). d is the Euclidean distance between the mean and the current pixel. The pixel can then be classified using the following:

$$\frac{|(I_t - \mu_t)|}{\sigma_t} > Th \longrightarrow \mathit{Foreground} \tag{3.15}$$

$$\frac{|(I_t - \mu_t)|}{\sigma_t} \leq Th \longrightarrow \mathit{Background} \tag{3.16}$$

where Th is a threshold determined by the user (usually $Th = 2.5$).

Koller et al. [129] noted that equation 3.12 is unduely updated also at the occurrence of foreground values. For this reason, they propose to use the following equation:

$$\mu_t = M\mu_{t-1} + (1 - M)(\rho T_t + (1 - \rho)\mu_{t-1}) \tag{3.17}$$

where the binary value M is 1 in correspondence of a foreground value, and 0 otherwise. This method is also known as selective background update.

3.2.6 Image Scaling

Bilinear

Bilinear interpolation is an extension of linear interpolation in a 2D space. Bilinear interpolation works by performing linear interpolation in one direction (along x-axis), then again in the other direction (y-axis). As linear equations only require two data points, extending this into 2D space means bilinear interpolation requires a 2x2 area to work. Despite the name, bilinear interpolation is not linear, but rather a product of two linear functions [190].

Bicubic

Similar to bilinear, bicubic interpolation is an extension of cubic interpolation, and is constructed using the product of two cubic equations in two directions (x and y). Cubic equations require 4 data points, and extending this into 2D space means bicubic interpolation requires a 4x4 area to work [190].

Nearest-Neighbour

The nearest neighbor method differs from the previous two interpolation methods by simply selecting the value of the nearest pixel and ignoring the values of other neighbouring points. This is the most computationally inexpensive method and is commonly used [190]. However, this type of scaling is susceptible to aliasing effects for both enlarging, and reduction of images [190].

Lanczos

Lanczos sampling applies a kernel of window size a to determine the new value. The kernel is generated from a sinc function, multiplied by the central lobe of a second, longer sinc function [135] as shown in equation 3.18.

$$L(x) = \begin{cases} 1 & \text{if } x = 0, \\ \dfrac{a\sin(\pi x)\sin(\pi x/a)}{\pi^2 x^2} & \text{if } -a \leq x < a \text{ and } x \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

Area

Area scaling is another fast method, albeit slower than nearest neighbour, that takes an average of all the surrounding pixels within a window size a.

3.2.7 Hessian

A Hessian matrix (H) is a square matrix of second-order partial derivatives and is used to describe the local curvature of a function of many variables and is generally defined by the matrix shown in equation 3.19 [157].

$$H = \begin{bmatrix} \frac{\delta^2 f}{\delta x_1^2} & \frac{\delta^2 f}{\delta x_1 \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_1 \delta x_n} \\ \frac{\delta^2 f}{\delta x_2 \delta x_1} & \frac{\delta^2 f}{\delta x_2^2} & \cdots & \frac{\delta^2 f}{\delta x_2 \delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2 f}{\delta x_n \delta x_1} & \frac{\delta^2 f}{\delta x_n \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_n^2} \end{bmatrix} \tag{3.19}$$

For an image given by $f(x, y)$, a 2 × 2 matrix is used, as shown in equation 3.20.

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} \tag{3.20}$$

Eigenvalues are a special set of scalars associated with a linear system of equations (i.e., a matrix equation) that are sometimes also known as characteristic roots, characteristic values, proper values, or latent roots [151, 106]. In image processing the eigenvalues represent information about the change in intensity around a pixel, and are computed by solving the equation $\det(H - \lambda I) = 0$ where H is the Hessian matrix and I is the identity matrix. Expanded out, the equation looks like

$$H - \lambda I = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.21}$$

$$= \begin{bmatrix} f_{xx} - \lambda & f_{xy} \\ f_{yx} & f_{yy} - \lambda \end{bmatrix} \tag{3.22}$$

$$\det(H - \lambda I) = 0 \tag{3.23}$$

$$(f_{xx} - \lambda)(f_{yy} - \lambda) - (f_{xy})(f_{yx}) = 0 \tag{3.24}$$

$$\lambda^2 - f_{xx}\lambda - f_{yy}\lambda + f_{xx}f_{yy} - f_{xy}^2 = 0 \tag{3.25}$$

which becomes a quadratic equation for discrete values of f_{xx}, f_{yy} and f_{xy} and can be solved using the quadratic formula to obtain the two eigenvalues.

When calculating the Hessian matrix, it is done over a patch of pixels rather than for individual pixels using a weighting function $w(u, v)$ as in equation 3.26. This weighting function can be uniform, but is more typically an isotropic, circular Gaussian, that acts to average in a local region while weighting those values near the center more heavily. If a Gaussian weighting is used, the derivatives must be scaled appropriately by a factor related to the Gaussian kernel: σ_I^2.

$$w(x, y) = g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \tag{3.26}$$

At each scale, interest points are those points that simultaneously are local extrema of both the determinant (DET) and trace (TR) of the Hessian matrix. The trace of Hessian matrix is identical to the Laplacian of Gaussians [158].

$$DET = \sigma_I^2(f_{xx}f_{yy} - f_{xy}^2) \tag{3.27}$$

$$TR = \sigma_I(f_{xx} + f_{yy}) \tag{3.28}$$

The Hessian matrix forms the basis for many other feature detection techniques. Two of these techniques,

Harris Corner Detection

The Harris corner detector relies on the principle that an image intensity will experience large changes in multiple directions at a corner. These changes in intensity can be found by looking at the intensity changes when a window around the point shifts in an arbitrary direction [95]. Harris denotes the matrix A (equation 3.29 as the autocorrelation matrix, which closely relates to the derivatives of image intensity.

$$A(x) = \sum_u \sum_v w(u,v) \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} \tag{3.29}$$

where f_x and f_y are the intensity derivatives in the x and y direction respectively at point (u, v).

By analysing the eigenvalues of matrix A, the following assumptions can be made: The corresponding eigenvalues of matrix A provide a quantitative measure

1. If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$ then the pixel at (x, y) has no features of interest
2. If $\lambda_1 \approx 0$ and λ_2 has a large positive value, then an edge is present
3. If λ_1 and λ_2 both have large positive values, then a corner is present.

Harris and Stephens note that exact computation of the eigenvalues is computationally expensive [95], since it requires the computation of a square root, and instead suggest the following function M_c:

$$M_c = \lambda_1\lambda_2 - \kappa\,(\lambda_1 + \lambda_2)^2 = \det(A) - \kappa\ \mathrm{trace}^2(A) \tag{3.30}$$

Where κ is an empirically determined constant in the range $0.04 - 0.06$

M_c will be large for corners, M_c is negative with a large magnitude for an edge and $|M_c|$ is small for a flat region. This improves computational efficiency by eliminating the need to compute the eigenvalues and solely using the determinant and trace of matrix A.

Frangi Filter

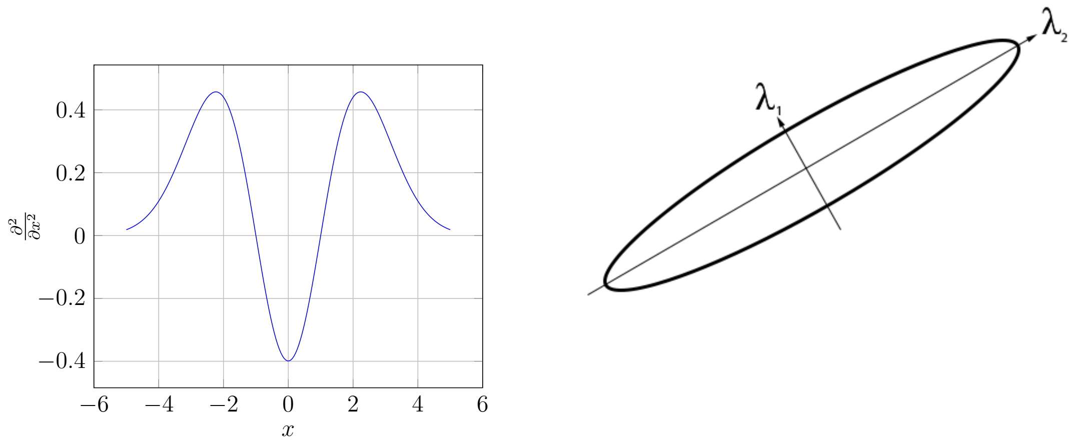
The Frangi filter was designed for vessel enhancement in medical images, and was designed for use in 2D and 3D images [70]. As our images are always 2D, we will only evaluate the 2D version. It achieved this by searching for geometrical structures which can be regarded as tubular. It does this by analysing the the eigenvalues of the Hessian matrix to look for tube like structures (Table 3.2.7). It then applies a second order derivative of a Gaussian kernel at scale σ (equation 3.31) to measure the contrast difference between the regions inside and outside the range $(-\sigma, \sigma)$ in the direction of the derivative [70]. For the remainder of this section, λ_1 is treated as the eigenvalue with the lowest magnitude ($|\lambda_1| < |\lambda_2|$).

λ_1	λ_2	Orientation Pattern
L	L	Flat region
L	H-	Tubular structure (bright)
L	H+	Tubular structure (dark)
H-	H-	Blob like structure (bright)
H+	H+	Blob like structure (dark)

Table 3.2: Possible patterns depending on the eigenvalues (L = low, H = high, +/- indicates the sign of the eigenvalue) [70].

Eigenvalue analysis of the Hessian allows the principal directions in which the local second order structure of the image can be found. Since this directly gives the direction of the tube like structure, the second order gaussian can be applied once, rather than at several different orientations, reducing the computational complexity [70].

$$\frac{\partial^2}{\partial x^2} = \frac{1}{\sqrt{2\pi\sigma^5}}(-\sigma^2 + x^2)e^{-x^2/(2\sigma^2)} \tag{3.31}$$



(a)

(b)

Figure 3.15: (a) Second order gaussian profile where $\sigma = 1$; (b) Tubular like ellipse with small λ_1 and large λ_2

As tube like structures will have a small λ_1 and large λ_2, a dissimilarity measure, R_B, is introduced (equation 3.32), which will approach 0 as the eccentricity approaches infinity.

$$R_B = \frac{\lambda_1}{\lambda_2} \tag{3.32}$$

The above ratio is based solely on geometry, and is thus grey-level invariant. It is possible for background pixels to produce an unpredictable filter response due to random noise fluctuations. However, knowing that vessel structures are brighter than the background and occupy a relatively small volume, background noise will produce small magnitude eigenvalues, which are quantified and accounted for using equation 3.33.

$$S = \sqrt{\lambda_1^2 + \lambda_2^2} \tag{3.33}$$

This measure will be low in the background where no structure is present and the eigenvalues are small for the lack of contrast. In regions with high contrast compared to the background, the norm will become larger since at least one of the eigenvalues will be large. The following combination of the components is therefore used to define a vesselness function with size σ:

$$V_o(\sigma) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp(-\frac{R_B^2}{2\beta^2})(1 - \exp(-\frac{S^2}{2c^2})) & \end{cases} \tag{3.34}$$

where β and c are thresholds which control the sensitivity of the line filter to the measures R_B and S respectively.

3.3 Existing tracking software

The increase in computing power and knowledge of computer vision has brought many automated tracking applications [210], of which some are open source and/or freely available [186, 37, 23, 118, 125]. Some of these trackers have been created to demonstrate the effectiveness of newly developed techniques [23], while others have been created using existing methods, but are used to address a specific problem such as tracking a particular object across a large dataset [186]. Tracking programs are being used to successfully, and quickly quantify locomotion and whisker metrics for a range of research topics. Several body and whisker tracking programs have been developed [186, 37, 125] for this purpose, but there is a need to evolve these programs based on new requirements and improving technology.

Existing trackers work by automatically quantifying metrics (such as angle, velocity, curvature) over a dataset, or in some cases a range of datasets [186]. A typical dataset would consist of a mouse, being recorded from a top-down view while performing a certain task or behaving naturally as shown in Figure 3.16. Some tasks require the mouse to be head fixed [209], or record how the mouse reacts when a whisker touches an object [37]. These programs increase productivity in researchers as the process generally requires little intervention from the user, and can analyse the videos much faster than if a human were doing it manually [174]. However, these programs do have weaknesses, such as struggling to analyse low quality videos, their processing speed, or usability. This section will focus on these new requirements and technologies, and how they can be used to create a new and improved whisker tracker.

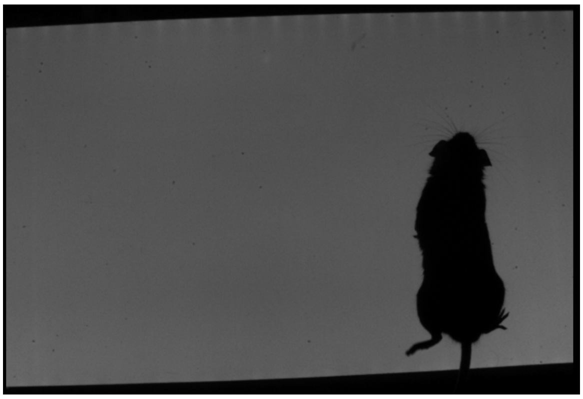


Figure 3.16: Example frame from the dataset [86].

3.3.1 Body trackers

ZooTracer

Developed by Microsoft Research [23], Zootracer claims to be capable of accurately tracking multiple, unmarked, interacting individuals in arbitrary video footage. It is provided free of change, and is semi-automated, requiring the user to select a point of interest at the start of the video and the same point of interest at the end of the video. The tracking is done by a modified version of template matching [23]. ZooTracer is provided free of charge, but requires some tedious setting up. It depends on the Open CV 2.4.8 libraries to run, and this particular library is not easily available. The user would have to download and compile the source code themselves in order to generate the necessary DLL's, and although the program also boasts of high accuracy rates, it did not perform well on our datasets, with the detected nosetip often rapidly moving from the left to the right side of the head. ZooTracer does provide very good visual feedback for displaying the trace line, but accessing this data in raw format proved troublesome. There is also very limited documentation available for it.

Ethovision XT

Originally developed in 1990 by Lucas Noldus, Ethovision XT [172] is the most versatile and popular mouse tracker. It is capable of tracking multiple rodents, behaviour recognition such as rearing or interaction with another rodent, and tracking multiple body points. It has also been developed to be used with specific behavioural models such as the Morris Water Maze and the T-Maze. The user can also define zones within a video, and record the time spent in these zones by the rodent. Ethovision is a commercial solution, and as such is very well documented and comes with 24 hour technical support. Being a commercial solution however, it costs $5850 for the most basic package. Its major limitations are that is not capable of detecting the nose point on a rodent, and it would fail in certain circumstances, such as a reflection being present, or if the rodent moved a static object, it would detect this object as part of the rodent. It should be noted however, that these problems were found on Ethovision version 5. The most recent version (12) may have solved some of these issues, but due to cost we were unable to obtain a more recent copy for evaluation.

EthoWatcher

Ethowatcher [118] was developed by a team of post-graduate students working at the Laboratory of Bioengineering of the Institute of Biomedical Engineering (IEB-UFSC, Dept. of Electrical Engineering, EEL) and at the Laboratory of Comparative Neurophysiology (Dept. of Physiological Sciences) of the Federal University of Santa Catarina (UFSC), to support detailed ethography, video-tracking, and the extraction of kinematic variables from digital video files in laboratory animals. It detects objects of interest using background subtraction and thresholding [118]. Ethowatcher requires certain conditions to work however, and perhaps the most challenging of these conditions is that it needs part of the video to not contain the tracking object so it can generate a background image. This is a major problem, as all the videos always contain at least one object we wish to track. Installation of the software was straight forward, and while the UI provides some guidance on how to use the software, it does have some major disadvantages. When previewing video files, and selecting a frame, the preview is not scaled, and if you're trying to use the software on a small screen, you would only be able to see a small segment of the preview. It will also only work with Audio Video Interleaved (AVI) video formats.

3.3.2 Whisker Tracking Approaches

Whisker tracking was first performed in a qualitative manner by use of motion picture film capturing successive photographs of whisking behaviour [236]. The analysis would measure the temporal rate of certain actions, such as protraction and retraction of the mystacial vibrissae, protraction and retraction of the nose or tip of the snout, head approach and withdrawal (or extension and retraction), and rapid expiration and inspiration. This was achieved by calibrating the camera to 30, 32, or 64 frames per second, and noting how many frames had elapsed since the start of the action [236], however, there is no substitute for quantitative measurements. Quantifying whisking kinematics was achieved by tracing 6 large, middle-row whiskers, specifically, the 3 most caudal ones on each side of the face, on transparencies overlaid on a video monitor [33]. While this was a manual process and would have taken some time to complete for an entire video, it did provide direct measurements of whisker angles, which could be used to analyse the velocity, amplitude, and temporal pattern of whisking [33].

Later studies began to develop semi-automated techniques to measure the whisker angle. One approach developed was to use a charged coupled device (CCD), lit uniformly by a light source such as fiber-optic devices [18] or by an infrared laser curtain [19]. This uniform light would generate a constant baseline voltage for each element in the CCD. As a whisker travelled over the CCD, and thus break the light source over a subset of elements, a voltage shift would occur in the respective elements and allow for the position of the whisker to be calculated [18, 19]. This technique was subsequently used in other studies [241, 112, 121], with some studies adapting the approach to measure movement in 2 dimensions (horizontal and vertical) [8].

Advancements in camera technology, allowed high speed footage with a large spatial resolution to be obtained. These cameras have very low shutter speeds ($< 1ms$), with frame rates more consistent than in previous studies [236], which in turn allowed for more accurate analysis. Some studies used these cameras could to capture whisker movements in greater detail, and then manually track the whiskers on an LCD screen using custom built software [160, 85]. It was noted here that methods for automated tracking of whisker movement had been created [125, 124, 231]; but automated tracking of multiple whiskers in a complete, intact whisker field had yet to be developed [85]. Various methodologies were attempted to overcome this issue. Dyes and markers were used by several studies to track specific points along on the whisker, or the entire whisker itself [124, 164, 195, 229], using a mixture of freely available software and plugins [195], commercial tracking software [164], or custom software [124, 229]. The tracking software used in these studies was designed to track specific locations based on the dyes or makers, rather than the whiskers themselves. Advances in freely available computer vision libraries allowed for development of trackers which overcame this issue [196, 231, 185, 28]. While previous methods would allow for reconstructing the whisker and hence quantifying it, fully automated solutions allowed metrics to be garnered over larger datasets at a much quicker pace.

3.3.3 Whisker trackers

Manually tracking one point across a video is tedious, but if you were wanting to track whiskers manually which consist of 3 or more points across a video, then it is almost impossible [174]. As a result, and with the increase in research on whisker movements, several programs have been created to try and automatically track whiskers.

BIOTACT Whisker Tracking Tool (BWTT)

The BIOTACT Whisker Tracking Tool [186] is a MATLAB Software Package used for whisker tracking and implements the Vibrassae and Snout Analyser (ViSA). ViSA consists of two modules, one module tracks head contour, position and orientation, the other module detects and tracks whiskers shafts located at a user defined distance from the head contour. The software is freely available and easy to install, providing you have Matlab installed. The head detection works by first converting the frame into a binary image (3.2.1) using a threshold found to work reliably over all tested datasets. The head is then detected by using a head template (3.2.5). The head template was generated by approximating the shape of a rats head with a cubic spline in a frame where the rats head was symmetric and not rotated. Whiskers are detected by looking for whisker like segments around the head contour by searching for intensity peaks. It is not fully automated, and requires the user to give the location of the nosetip in the first frame it becomes visible. Results were very accurate, but only after manually fine tuning many of the settings.

Whisk

Whisk [37] is a whisker tracking application developed by N. G. Clack [37]. This program requires the rodents to be head fixed, and their whiskers cut into a single row with 4 whiskers either side. As the rodents are head fixed, there is no head detection, and the software is purely for whisker tracking. To do this, Whisk detects whiskers using a two step process, Tracing and Linking. Tracing identifies initiation sites, and highlights potential whiskers. Linking removes false positives and also handles whiskers that overlap or are in contact with another object as these would not have been fully traced. To do this, they assumed whiskers were of a particular length, and that they would maintain the same ordering along the face. For the majority of images, a simple length threshold was used to remove the false positives and leave one traced curve per whisker.

WhiskerTracker

Developed by Knutsen [125], WhiskerTracker is another MATLAB Software Package, allowing for easy installation. WhiskerTracker detects the head by following the eye reflections, which requires the dataset to have been filmed with two overhead infra-red LED spotlights. In the first frame where both eyes appear, the user has to manually locate them. Then in the following frames, the program expects the eyes to have not moved much (2mm), and the peak luminance within a 2x2 mm area from the last known position indicates the new position of the eyes. Once the position of the head is known, the program would then attempt to track the whiskers. First, stationary objects were removed from the image (3.2.5), then a region of interest parallel to the face is filtered (3.2.4) according to the local angle of the whiskers in the previous frame. Splines are then fitted to whisker like features in the image.

WhiskerMan

WhiskerMan [11, 29] differs from existing trackers in that it is designed to extract kinematic parameters of an individual whisker near the whisker base, from mice where multiple rows of whiskers are intact. It achieved this by accurately approximating the bottom portion of the whisker as a quadratic bezier curve. The bezier curve is then fitted in the image by a local, gradient based searched. Tracking errors due to whisker crossovers were almost completely eliminated by using knowledge of the whisker shape in the previous frames.

3.3.4 Summary of existing software

Each piece of software attempts to locate the body or nosetip using different a method, with varying degrees of accuracy and speed. The ease of using ZooTracer is perhaps its greatest quality over other trackers, but its low accuracy and slow speed let it down. Ethovision required significant user input to set up, and was susceptible to tracking reflections. It also required a background image with no rodent present, which was sometimes unobtainable with our dataset. Ethovision will be discussed in further detail in chapter 5. EthoWatcher is widely used, it could not be fully evaluated due to our dataset always containing a rodent in every frame, and is the only piece of evaluated software that had this requirement. BWTT boasts high accuracy rates, and once set up is fully automated, but it can only process 2 frames per second, and if the initial parameters are not entered correctly, the accuracy rate can drop considerably. Whisk is very accurate, with the authors claiming an accuracy of $0.23 \pm 0.2 pixels$.

It also performs very well, being able to process $8MPx/s/cpu$ (35 fps for 640 x 352 pixel video). Whisk supports parallel processing using inexpensive cluster nodes [37]. However, Whisk requires datasets where the mice have to be head fixed with their whiskers cut to a single row. WhiskerTracker is capable of tracking individual whiskers when the quality is good enough for them to be resolved. However, it requires occasional intervention by the operator (when whiskers overlap for example), relies on high quality datasets, and assumes whisker displacement is small from frame to frame [125]. WhiskerMan is unfortunately not available for download and as such can not be evaluated.

BWTT is the most accurate tracker, although Whisk and WhiskerTracker are not port of the final review as they require a specific type of dataset. BWTT however is also the slowest, averaging only $2fps$ for head detection alone. A complete software package should therefore be accurate, fast, allow free movement of the mouse with any number of whiskers, have an intuitive and easy to use user interface, provide error feedback with potential ways of resolving them, and work across the entire dataset, including low quality videos.

3.4 Creating a complete package

In order to create a software package for this project, several aspects need to be reviewed. Good usability, rapid prototyping, ease of install and intuitive user interfaces are heavily dependant on the programming language used. Sustainable and upgradeable software can be made by adopting an appropriate design pattern. Highly optimised image processing techniques have been created in a wide array of freely available libraries, and some of these can be sped up even further by taking advantage of computing advances.

3.4.1 Developmental languages

There are hundreds of different programming languages [179], and all of them have their own specific strengths and weaknesses, and many were created with a specific objective in mind. Despite the different options available, some programming languages have come to the fore when being used to develop desktop applications.

MATLAB (Matrix Laboratory) is a fourth generation language created by Mathworks [152]. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. MATLAB users come from various backgrounds of engineering, science, and economics [152]. Although it is very powerful, and comes with its own image processing library which is described in section 3.4.2 in greater detail, its user interface options are quite poor and it also requires the user to have Matlab installed.

C++ was created by Bjarne Stroustrup while working on his PhD thesis, with work first beginning in 1979 [215]. At this time the C language was already in use, but lacked the object-orientated programming paradigm. Bjarne found this paradigm very useful, and so began to incorporate it into C, hence why the new language was called C++. It was designed with a bias toward system programming (e.g. for use in embedded systems or operating system kernels), with performance, efficiency and flexibility of use as its design requirements [1]. It has grown in popularity, and is used in some of the most demanding desktop applications (Adobe products for example [214]). Due to its popularity, all major Operating Systems support C++ programs, with little to no changes needed when porting.

C$^\sharp$ was designed to be a simple, modern, general-purpose, object-oriented programming language, in order to decrease development time and increase productivity [99]. It achieves this in many ways, but probably most notably is that it's designed for the Common Language Infrastructure. This removes memory management, and allows the language to easily communicate with many other languages such as C++. C$^\sharp$ was not designed to compete with other languages on performance [202].

Windows Presentation Foundation (WPF) is a graphical subsystem for rendering user interfaces in Windows-based applications [71]. This dramatically speeds development time, and allows you to create visually stunning user experiences [165]. However it is dependant on Windows libraries, and would therefore require a significant effort to port to Mac and Linux. Although this may seem like a major disadvantage, it provides a major reduction in development time as you can use windows themes and styles and not have to make your own from scratch.

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible [79]. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. It shares several similarities to the C based languages, but is organised rather differently, omitting some parts of C and C++ and including some aspects of other languages [79]. Recent optimisations in Java have also allowed its performance to become comparable to C++ for low level benchmarks [138], but is still slower for more complex operations [77].

We believe there are 2 major points I need to consider when choosing the preferred development language, performance and development time. For performance, C++ is the obvious choice, where as for development time, C$^\sharp$ is the best choice. C$^\sharp$ is also the language we're most familiar with, which would reduce development time even further. As C$^\sharp$ can communicate with C++, it would be possible to develop the application in C$^\sharp$, with the performance critical aspects written in C++. Although Matlab is extremely powerful, it is still quite slow when compared to C++ [7]. Coupled with the need that Matlab must also be installed as well, I have decided to reject this language. Java is too general purpose for my application, it lags behind C++ in terms of performance and C$^\sharp$ in terms of development time.

3.4.2 Image processing libraries

The domain of image processing has increased vastly in recent years [34], and has become increasing complex and inconsistent due to the various programming languages and variation of available libraries [204]. Apart from being a development language, Matlab also comes with its own image processing library [78]. Matlab is incredibly useful for rapid prototyping and testing, and how easy it is to learn [136]. However, a program written in Matlab would also require the user to have Matlab installed, and as it isn't open source, the user may have to pay a large amount for it. There is a popular alternative library to Matlab called OpenCV which provides an extensive set of algorithms [45]. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. It is also very well documented, with a large user base.

OpenCV has by far the most extensive set of image processing functions, from Image Manipulation to GPU acceleration. It is also the fastest image processing library as C++ code is much faster than Matlab code [153]. Not only that, but OpenCV is also open source so further optimisations could be made. There is also a commercial variant of OpenCV that is compiled using the Intel C++ Compiler, TBB & IPP for even more performance, but this will not be free. Matlab is certainly powerful enough for this project, but its speed and the fact it requires Matlab to be installed are a major disadvantage.

3.4.3 Computing advances

There are several advances in computing architecture made recently that can be exploited to dramatically increase the efficiency of the package [101]. Multi core CPU's are now standard in even the most basic computers, and by utilising all the cores in an efficient manner, it will provide a significant increase in performance [62]. This is well suited for complex operations which can be run in parallel. For example, each frame of video is analysed independently of the other frames, so it would be possible to analyse 2 frames at the same time on a dual-core processor, decreasing execution time by a potential 50%. Graphical Processing Units (GPUs) are also being utilised to help with certain problems, especially in image processing [201]. As GPUs tend to have a lot more cores than CPUs, but aren't as powerful, they are well placed for dealing with simple parallel problems [220]. Modifying images, for example, where each pixel undergoes a simple transformation, is perfectly suited for GPU processing.

3.4.4 Design patterns

In large and complex software applications, re-usability and interoperability of software components are very important. This need, led to the development of design patterns, which are in essence a proven solution to commonly occurring problems in software design [40]. They provide a template to design your code around, rather than give you specific directions on how to write code. They are a necessity when working on large projects, as they also allow someone who has never seen your code to easily navigate around the source files and understand how the underlying mechanisms work [73].

While many different types of patterns will eventually be used throughout the application, there will only be one architectural pattern as this describes the organisational structure of the application [25]. Depending on the development language(s) chosen, there are three widely adopted architectural patterns, the Model View Controller (MVC) [131], Model View Presenter (MVP) [189] and Model View View-Model (MVVM) [206]. Although the patterns are different, they address the problem of separating the logical part of the application (Model) from from the user interface (View) [69]. This provides code re-usability by allowing the Model to be used throughout the program without the need to change it, even if the developer wishes to change how the information is presented to the user [183].

The differences between the design patterns are how the model reflects data back to the view. For MVC, a controller is used to process view events (eg. a button a click), and uses this to manipulate the model. The view is subscribed to the change events within the model, and will update itself when a change happens [131]. This was the first design pattern to be created out of the three, and as a result, many web content management systems adopted this pattern, and subsequently, the majority of web applications were developed using MVC [131].

For MVP, a presenter is used instead of a controller, which will process input events and send them to the model, but the model will return the information back to the presenter rather than update the view directly. The presenter will then update the view via an interface [189]. This pattern has become widely adopted for applications developed using WinForms, as the view in WinForms is updated via an interface [189].

MVVM was developed to simplify event driven behaviour, by exploiting features in Windows Presentation Foundation [166]. MVVM uses a View-Model instead of a presenter, which, like MVP, manipulates the model with input events, but it does not update the view via an interface. Instead, WPF allows data bindings between the View-Model and View, providing another layer of separation. This allows for View-Models to be used against multiple Views, and multiple Views to be used with different View-Models [75].

3.5 Tracker recommendations

All the currently available tracking programs have their strengths and weaknesses. Performance (defined as the execution speed of a piece of software) and usability weaknesses can definitely be removed. Accuracy rates are already quite high amongst all the programs, but with the latest image processing libraries and computer advances, accuracy will be expected to increase, even for low quality datasets. The application will also assume the datasets contain unrestrained, freely behaving animals. This way, even if restrained animals were used, it wouldn't pose a problem. One of the main limitation to all the programs though is that they are only semi-automated, with very little error feedback if something fails. The new program will aim to be fully automated, with errors logged and stored for the user. All the programs are also limited in their output. They all expose the raw data (each x and y co-ordinate of the various points), and some will generate graphs, but many of the metrics described in section 4.5 would require the user to manually calculate them from the raw data. The program being developed will therefore include a comprehensive statistical analysis toolbox. Other features which will help users such as batch and parallel processing will also be included.

Using a combination of C++ and C$^\sharp$ / WPF, it will be possible to create a piece of software to the desired specification. OpenCV will provide all the image processing algorithms I need, as well as provide support for GPU processing for the computers that have compatible hardware. I will also adopt the MVVM design pattern, this will make it easier should the need to port to other platforms arise. This is due to being able to separate the View from the Model, so if I were to port it to Mac OSX for example, I would only need to create a new user interface, rather than re-write any logic. Using Emgu CV in combination with OpenCV will also decrease development time, and since Emgu CV can be compiled in Mono, it won't present a problem to portability.

3.6 Metric Requirements

The metrics extracted from behavioural trackers can be quantified in various ways. This section looks at the different methods used in order to extract meaningful information from them.

3.6.1 Whisker angle

The angle of the whisker is possibly one of the most important metrics the program must be able to quantify. Whisker angle is defined as the angle between the center line of the rodent, and the tangent to a given point on this whisker shaft. Many studies that involve rodents use this metric [209, 37, 4, 180, 186]. However, how the angle is measured can vary considerably from study to study. The state-of-the-art methods to measure whisker angle include:

1. Measuring the angle that the whisker makes with the centerline of the rodent [209], where an angle of 0 degrees corresponds to a whisker that is perpendicular.

2. Measuring the angle at different points along the whisker, and this is measured against the horizontal (the rodent is head fixed so the head orientation never changes) [37].

3. Measuring the azimuthal angle, that is the angle between the whisker in its current position, against the whisker in its resting position [4, 180].

4. Measuring the angle between two vectors, $\vec{M}$ and $\vec{S}$, where $\vec{M}$ is parallel to the mid-line of the snout and pointing in the opposite direction to that in which the animal is facing (i.e. backwards). $\vec{S}$ is the vector parallel to the base of the whisker shaft and facing outwards from the animal [186].

3.6.2 Whisker angular velocity

Angular velocity is measured as degrees per second [125, 33] or in degrees per millisecond [85]. Once a method for detecting the angle is chosen, angular velocity (ω) will be easy to calculate as the only extra variable needed is the change in time which can be found from the frame rate, as shown in equation 3.35. However, since the whisker deforms, the angular velocity is not constant along the entire length of the whisker. Fortunately, as mentioned in section 3.6.3, it is possible to locate any point along the whisker, and as such, determine the angular velocity of that point. There is quite a high probability though that the dataset footage may have been slowed down as well, so the frame rate of the video would not give an accurate measure of the change in time. Providing an option to manually specify the frame rate or how much the video has been slowed down (e.g. 2x, 4x) could overcome this problem.

$$\omega = \frac{\Delta\theta}{\Delta T} \tag{3.35}$$

3.6.3 Whisker curvature

To quantify curvature, a polynomial to degree n is used to model the whisker [125, 124, 180, 222, 37]. Studies vary between expressing this polynomial as a parametric equation or a bezier curve [63]. If it were expressed in parametric form, a separate co-ordinate system would need to be created for each whisker to ensure a parametric equation could be generated. However, this would be difficult and time consuming to implement. Bezier curves are already widely used for creating curved lines in computer graphics, and all work off the same co-ordinate system, regardless of position, orientation, or shape, and have been used to measure whisker curvature [11]. The curvature (K) can also be calculated for any plane curve given parametrically in Cartesian coordinates as $B(t) = (x(t), y(t))$ [180], as shown in equation 3.36.

$$K = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{\frac{3}{2}}} \tag{3.36}$$

3.6.4 Whisking frequency and amplitude

Whisking frequency is the amount of full whisk cycles (peak to peak) per second. However, the signal generated by a moving whisker is quite noisy and it is not obvious which peaks should be used. To solve this, Grant et al. [88] proposed using a low pass filter to smooth the signal. The peaks were then detected, before finally finding the closest peak on the unfiltered signal. To calculate the amplitude, the mean value was removed from the whisking angle time series and the root mean square value was computed to give the root mean square (RMS) whisking amplitude. These time series were approximately sinusoidal, so the ”peak-to-peak whisking amplitude” was estimated by multiplying the RMS whisking amplitude by $2 \times \sqrt{2}$ [87]. Other methods for determining amplitude include filtering the whisker angle between 6 and 60 Hz (Butterworth Filter [26]) and then performed a Hilbert Transformation [93] on the filtered signal [180].

3.6.5 Whisker spread

Whisker spread can be calculated in two ways. The first way is to calculate the angle between two vectors $\vec{F}$ and $\vec{R}$. Where $\vec{F}$ is the vector of the foremost whisker, and $\vec{R}$ is the vector of the rearmost whisker [191]. The second way is to measure the angles between adjacent whiskers and take the standard deviation of this [85].

3.6.6 Whisker stress and bending moments

Some datasets include whisker interaction with an object (pole for example). Here, it is possible to estimate contact forces and bending moments. For some studies, the equations to describe these though rely upon knowing the radius of the whisker at any point and the bending stiffness of the whisker [96, 180]. Previous studies performed deflection tests to estimate bending stiffness, and modelled the whisker as a cone to estimate the radius [180]. Other studies however, found the change in curvature (Δk) at the base of the whisker to be directly proportional to the bending moment and contact force, and can be used as a proxy to acquire the bending moment using equation 3.37 [37, 28].

$$M_p(t) = \Delta k_p(t) E_p I_p \tag{3.37}$$

where $M_p(t)$ is the bending moment around a point p at time t, $\Delta k_p(t)$ is the change in curvature at point p at time t and $E_p I_p$ is a constant of proportionality.

3.6.7 Mean angular velocity and offset

Whisker offset was calculated as the mean angular position for each clip. Mean angular retraction and protraction velocities were calculated from the angular position, as the average velocity of all the backward (negative) whisker movements, and forward (positive) whisker movements, respectively. These measures were all averaged to give a mean value per clip, and each side (left and right) was also averaged together [87].

3.6.8 Locomotion velocity

Locomotion velocity is calculated as the speed of the movement forward by the animal, as a mean over the clip [87]. This is done by tracking the animals nose tip, however, it is possible that the animal is standing still but moving its head.

3.6.9 Summary of metrics

As the program can detect the head and whiskers, the data required to quantify the locomotion velocity and whisker angle will be available. Measuring the whisker angle is the main objective of this work. Since there are several different methods used, and no literature about which one may give better results, all the different methods will be incorporated and evaluated to determine if one gives a larger signal than the others. Once the angle is known, the angular velocity can also be easily calculated for any point along the whisker. The program will detect the whisker as a long sequence of points (up to 20), then generate a curve based on a certain number of these points. This means any polynomial (up to a possible degree of 20) could be used to create the equation. The higher the degree, the better the equation will fit the points. However, too high a degree can lead to over-fitting and produce inaccurate results. For this reason, WhiskerMan uses a quadratic equation [28], and the largest degree observed in the literature was 5 [37]. As each whisker will have its angle calculated, the whisker spread also becomes easy to calculate. Calculating the frequency and amplitude should also become quite straight forward, as the point where the angular velocity crosses 0 should be clearly visible. If the signal is not clear, the method described in 3.6.4 will enable us to calculate this. The mean offset and mean angular velocity also become straight forward.

The locomotion velocity will need careful consideration. The nosetip velocity will be easy to calculate, but body velocity will be more difficult. If the dataset provides continuous footage of the entire mouse, then it will certainly be possible, but many of the datasets available tend to show mice with much of their bodies out of the cameras field of view.

The stresses, bending moments and forces will be too difficult to calculate automatically by approximating the whisker as a cone. This is due to being unable to accurately estimate unknown parameters. It could be possible that these are left for the user to manually input, as then bending moments and forces can be calculated by the equations provided by Pammer et al. (2013). Although it would be possible to acquire the change in curvature, as curvature can be measured at any point along the whisker at any time, and therefore use this as a proxy to bending moments as described by equation 3.37, it is not necessary for this work and will not be included. This will be a limitation of the current software compared to existing trackers, and will be something to consider for future work.

3.7 Summary

All of the theories and techniques described in this chapter provides the foundation on which the following contribution chapters will be based. Chapter 5 will describe in detail how the head and orientation of the rodent is found, utilising basic filters, background subtraction and object detection techniques. Chapter 6 will introduce a new whisker detection technique which takes advantage of the feature extraction, and image scaling methods.

Chapter 4

Development of a Manual Whisker Annotator (MWA) for Validation

Based on publication: Manual Whisker Annotator (MWA): A Modular Open-Source Tool

4.1 Chapter Summary

This chapter focuses on the development of the manual whisker annotator which was used to validate the planned algorithms and trackers. The whisker metrics are determined and defined here.

4.2 Introduction

There is no freely available software that has been designed solely to manually annotate whiskers and gather the desired data in an easy and intuitive manner. Although some automated trackers come with a manual annotation component, they have not been designed for manually annotating an entire video and extracting a variety of metrics from it. WhiskerTracker [125] provides manual annotation to correct automated tracking mistakes, and Whisk [37] provides manual annotation in order to validate the automated tracking. Many studies have developed their own tracking software to address this, but these programs have not been published and are not available for download [13, 12, 14]. Freely available programs have provided some good analytical tools for modelling an object's trajectory or acceleration, but they were not designed for tracking multiple objects or handling whiskers [15]. The aim of this software package is to improve upon existing manual annotation software, by increasing annotation speed. We will do this by i) creating an easy to use and intuitive interface; ii) allowing easy extraction of all raw data in an intuitive format; iii) creating an analytical suite to analyse the raw data; and iv) creating the software package in a format that is easy to install with as little effort required from the user as possible (i.e. no additional assemblies required).

4.3 Implementation and Architecture

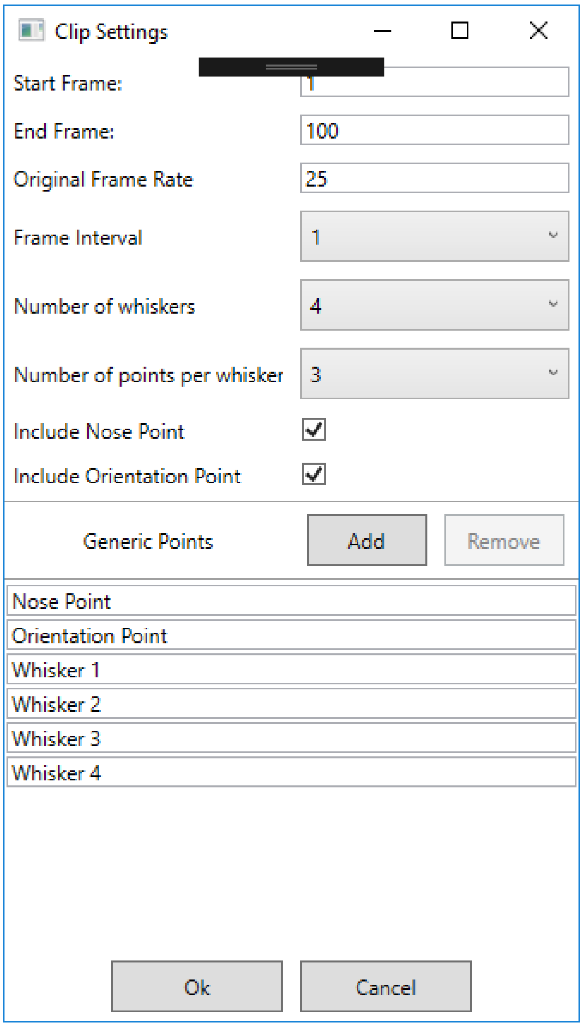
MWA decreases annotation speed of manual tracking of whiskers by minimising the total number of clicks required by the user to annotate an entire video. This is achieved by automatic progression from one point on the whisker to another, from whisker to whisker, and once all whiskers have been annotated, from frame to frame. The tracked positions from the previous frame and overlaid onto the current frame, with a slight transparency, making it easier to locate the specific whisker being tracked. A full guide into how the user annotates the software is provided in section 4.4. Once all frames have been tracked, the tracked points can be instantly exported into Excel, or if the user wishes, they can analyse the results in the analytical suite. This instantly provides access to metrics such as those described in section 4.5. To further increase ease of use, the software comes in a single executable package, allowing for easy installation on any Windows computer, without the need to install any dependencies.

Using C$^\sharp$, WPF and the .NET framework allows high performance, robust and scalable solutions for Windows [12]. It was developed using the Model View View-Model (MVVM) design pattern, and as such, the code can be split into 3 main sections, the View, Model and Repository. The View folder contains all the windows. Converters, Behaviours, Commands, Controls and other classes created specifically for the windows can be found in their respective folders. The Model layer contains all the objects which represent the real-world content and can be found in the Model folder. The interfaces for these objects can be found in the ModelInterface folder. The model and view are linked with a View-Model, and these can be found in the ViewModel folder. Finally, the repository handles loading and saving of data, which is in XML format for this project. The files for these can be found in the Repository folder.

4.4 User Operation

The user operates the software by first loading in a video, and then presented with video settings window (figure 4.1). Here you provide the settings for how you wish to annotate the video.

1. Start Frame - The frame from which you wish to start the annotation
2. End Frame - The frame from which you wish to end the annotation
3. Original Frame Rate - The original frame rate of the video. Many high speed video clips are slowed down, and as such the current frame rate can not be used to calculate time dependant metrics such as velocity. Setting this option adjusts the metrics accordingly
4. Frame Interval - You can decide to annotate the video every 1, 2, 3, 4, or 5 frames
5. Number of Whiskers - How many whiskers you wish to annotate
6. Number of points per whisker - The number of points that will be used to define the whisker, 2 for linear, 3 for quadratic or 4 for cubic
7. Include Nose Point - Whether you would like to annotate the nose point
8. Orientation Point - Whether you would like to annotate another point, that when a vector is drawn between this point and the nose tip, the orientation can be deduced.
9. Generic Points - Whether you wish to annotate any other points that aren't part whiskers.



Clip Settings

Start Frame: 1

End Frame: 100

Original Frame Rate 25

Frame Interval 1

Number of whiskers 4

Number of points per whisker 3

Include Nose Point ☑

Include Orientation Point ☑

Generic Points

Nose Point

Orientation Point

Whisker 1

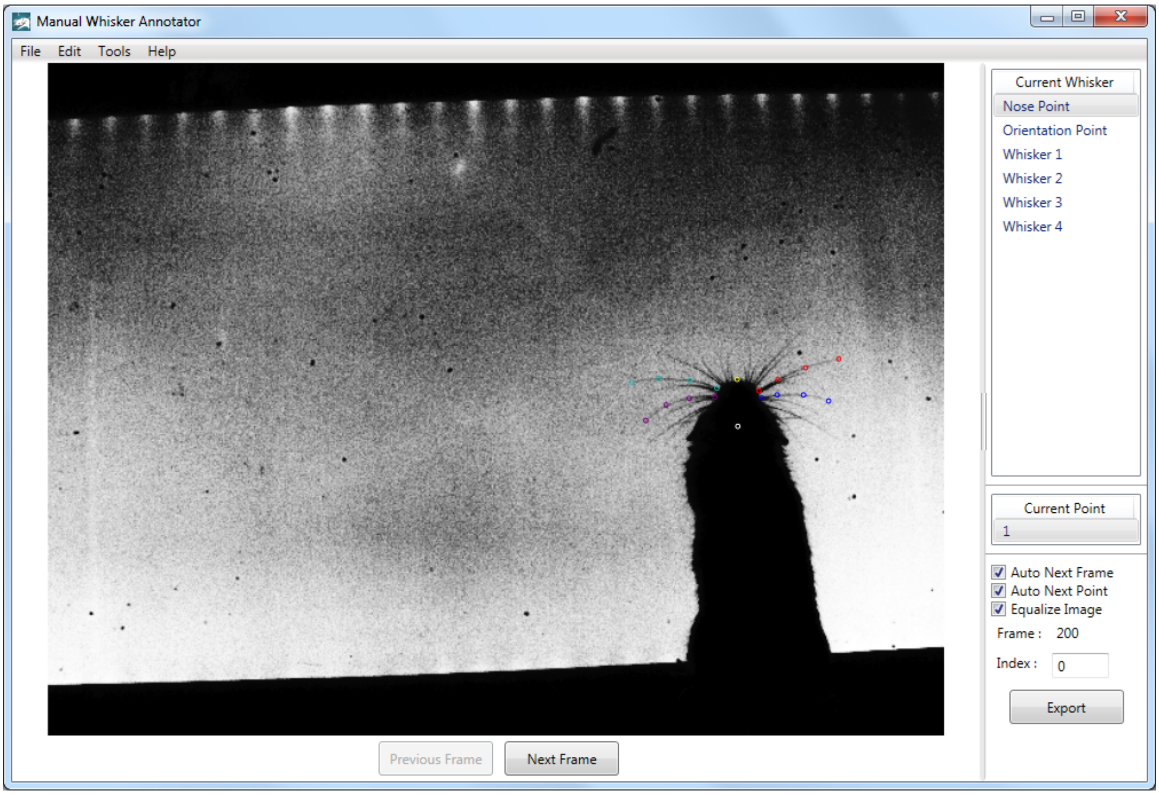
Whisker 2

Whisker 3

Whisker 4

Figure 4.1: Video clip settings.

Once the video settings have been input, the user will be presented with the screen shown in figure 4.2. This is the main user interface and where annotation begins. The current part you are tracking is shown on the right hand side, and the user begins tracking by clicking on the screen where that particular part is. Once clicked, the next tracking part will be automatically selected, allowing the user to click in quick succession. Once all tracking parts have been acquired for that frame, the software will automatically progress onto the next frame.



Current Point: 1

☑ Auto Next Frame
☑ Auto Next Point
☑ Equalize Image

Frame : 200

Index : 0

Figure 4.2: Overview of the Manual Whisker Annotator.

Three settings are provided on this page:

1. Auto Next Frame - Whether the software should automatically progress on to the next frame
2. Auto Next Point - Whether the software should automatically progress on to the next point
3. Equalize Image - An option to provide image enhancement by use of histogram equalisation

When the user has completed the annotation, they can export the raw data points using the export button, or use the analytical suite to export a range of metrics described in section 4.5. The analytical suite can be accessed by going to Tools -¿ Analysis.

4.5 Metrics

Whisker Curvature

Whisker curvature is used to estimate bending moments and forces in the whisker follicle in order to make associations with neural measurements [16, 17, 18, 19]. To model curvature, Bezier curves were used [20]. Given that the user had manually annotated the points along a whisker, using either 2, 3 or 4 points per whisker, a Bezier equation (linear, quadratic or cubic respectively) was easy to implement. Any point on the line can also be found easily by specifying how far along the line you want to be (0 = end, 0.5 = midway, 1 = beginning). This value is known as the T Value. It was quickly discovered though that Bezier curves will not go through the control points, and as such, the program would display a curve that did not perfectly overlap the whisker. It was therefore encouraged that the user places the points with equal distance between them. This allows the program to estimate the T Value of each point along the Bezier curve, and calculate the control points needed to create the correct curve. Curvature is measured in pixels^{-1}.

Given a quadratic Bezier curve with a start point P_0, control point P_1, and end point P_2, as shown in equation 4.1, the control point P_1 can be found using equation 4.2 which passes through the point $\mathbf{B}(t)$ with a T Value of $t = 0.5$

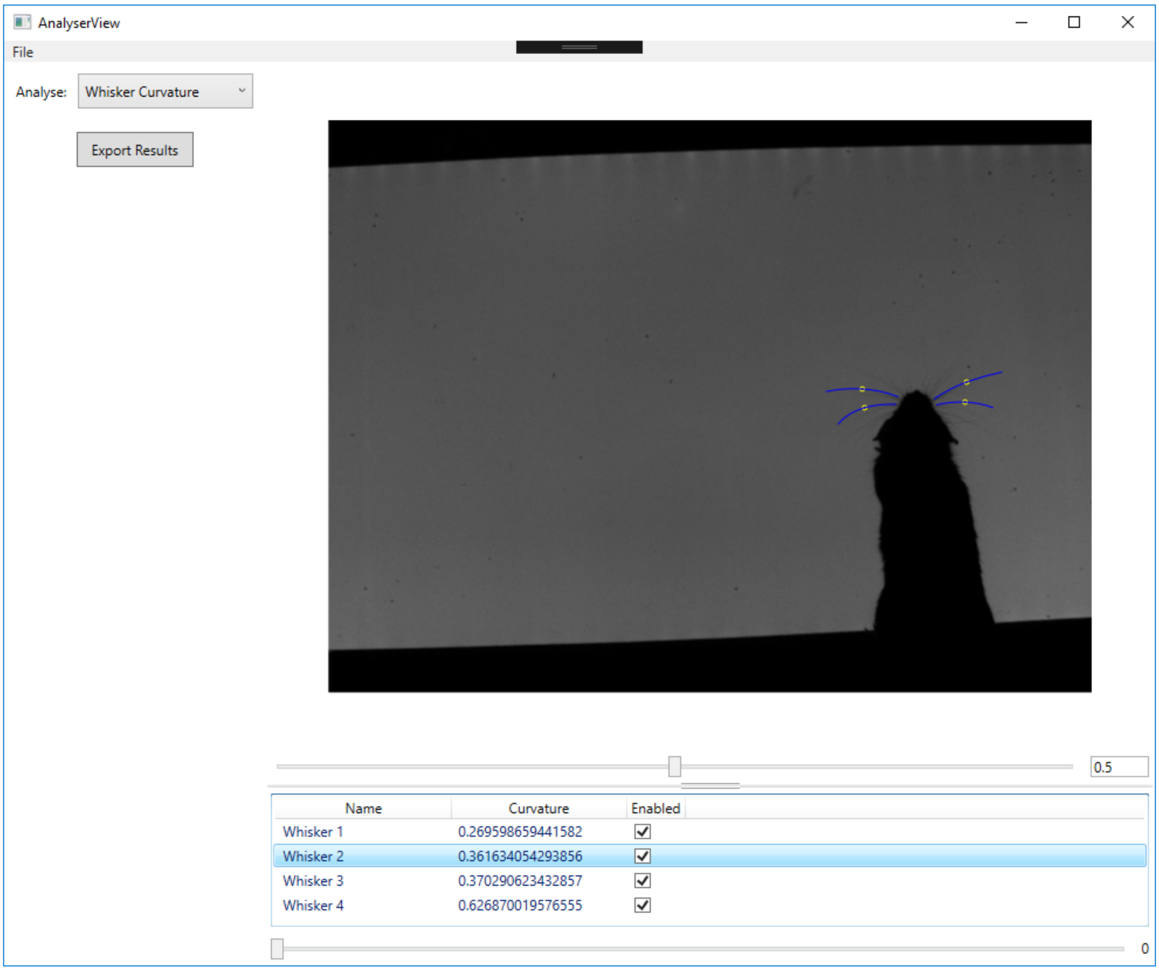
$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2 \ , \ 0 \leq t \leq 1. \tag{4.1}$$

$$\mathbf{P}_1 = \frac{\mathbf{B}(t) - (1-t^2)\mathbf{P}_0 - t^2\mathbf{P}_2}{2(1-t)t} \tag{4.2}$$

Given a cubic Bezier curve with a start point P_0, control points P_1 and P_2, and end point P_3, as shown in equation 4.3, the control points P_1 and P_2 can be found with T values $t_1 = 1/3$ and $t_2 = 2/3$ using the equations shown in 4.4

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , \ 0 \leq t \leq 1. \tag{4.3}$$

$$
\begin{aligned}
\text{Let } a &= 3t_1(1-t_1)^2 \\
\text{Let } b &= 3{t_1}^2(1-t_1) \\
\text{Let } c &= \mathbf{B}(t_1) - (P_0 \times (1-t_1)^3) - (P_3 \times {t_1}^3) \\
\text{Let } d &= 3t_2(1-t_2)^2 \\
\text{Let } e &= 3{t_2}^2(1-t_2) \\
\text{Let } f &= \mathbf{B}(t_2) - (P_0 \times (1-t_2)^3) - (P_3 \times {t_2}^3) \\
P_1 &= \frac{c-(af/d)}{(b-ae/d)} \\
P_2 &= \frac{c-(b \times P_1)}{a}
\end{aligned}
\tag{4.4}
$$



Analyse: Whisker Curvature

0.5

Name	Curvature	Enabled
Whisker 1	0.269598659441582	☑
Whisker 2	0.361634054293856	☑
Whisker 3	0.370290623432857	☑
Whisker 4	0.626870019576555	☑

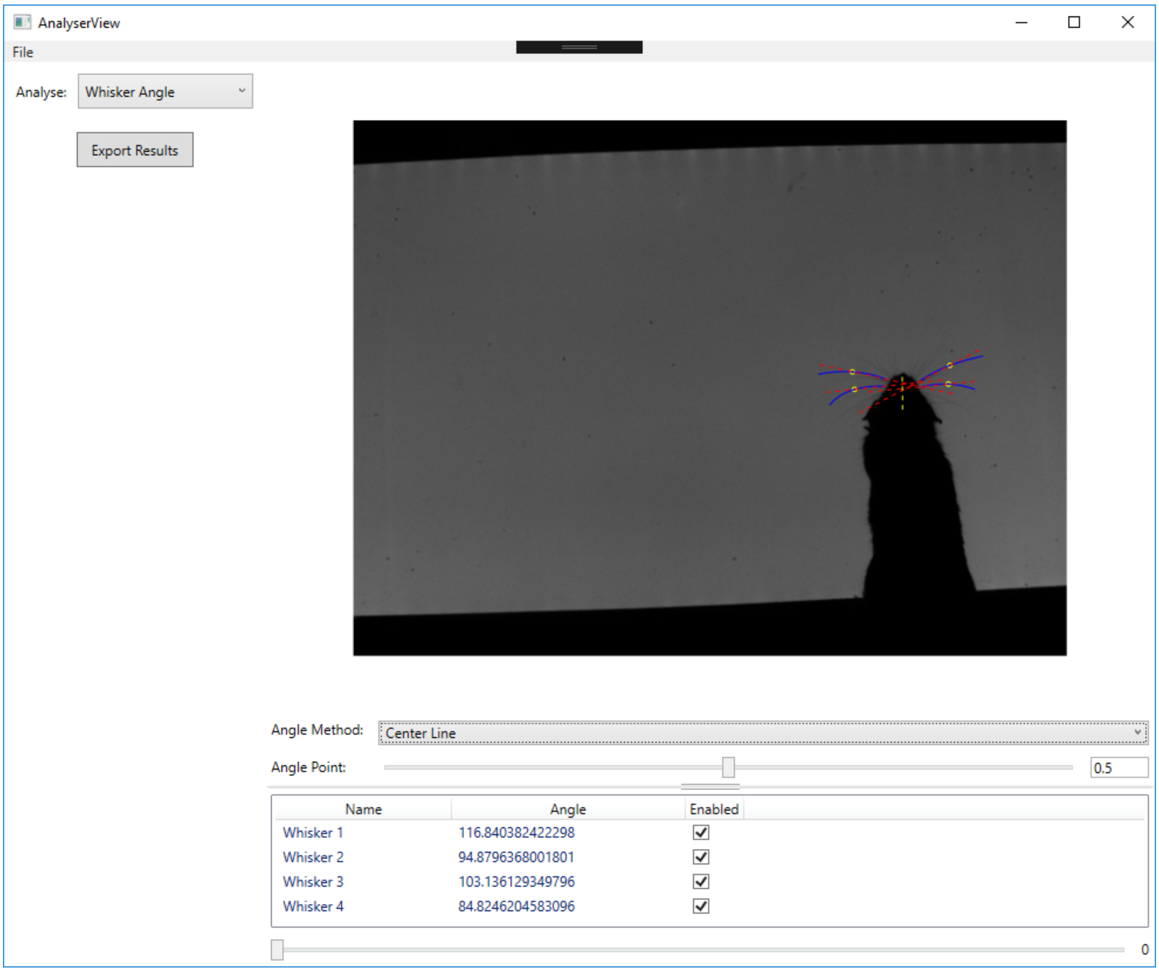
0

Figure 4.3: Curvature results

Whisker Angle/Mean Offset

Possibly the most important metric to detect is the angle of the whisker, and its average position (mean offset). This is due to the amount of other metrics which are calculated from the whisker angle and used in the subsequent studies, including, but not limited to, angular velocity, frequency and amplitude [7]. Mechanoreceptors situated around the follicle-sinus complex provide input to 150 - 400 neurons, and these neurons can encode the whiskers position with high precision by firing at specific deflection angles [21]. Mean offset is used by some studies to determine how far a whisker has moved [21], or to compare the resting position of whiskers of experimental models against healthy groups after contact with an object [7 , 8, 10].

Several methods were identified for calculating the whisker angle [16, 22], and a method was developed that would cater for all of them. The main differences being where the angle was measured against, and where along the whisker shaft the angle should be taken. To solve this, 3 options are provided from where to take the angle from the horizontal, vertical, and body center line. Another option is then provided to choose how far along the whisker shaft to take the angle (0 being the tip, 0.5 the middle, 1 the base) as shown in Fig 1. Angles are measured in degrees.



Analyse: Whisker Angle

Angle Method: Center Line

Angle Point: 0.5

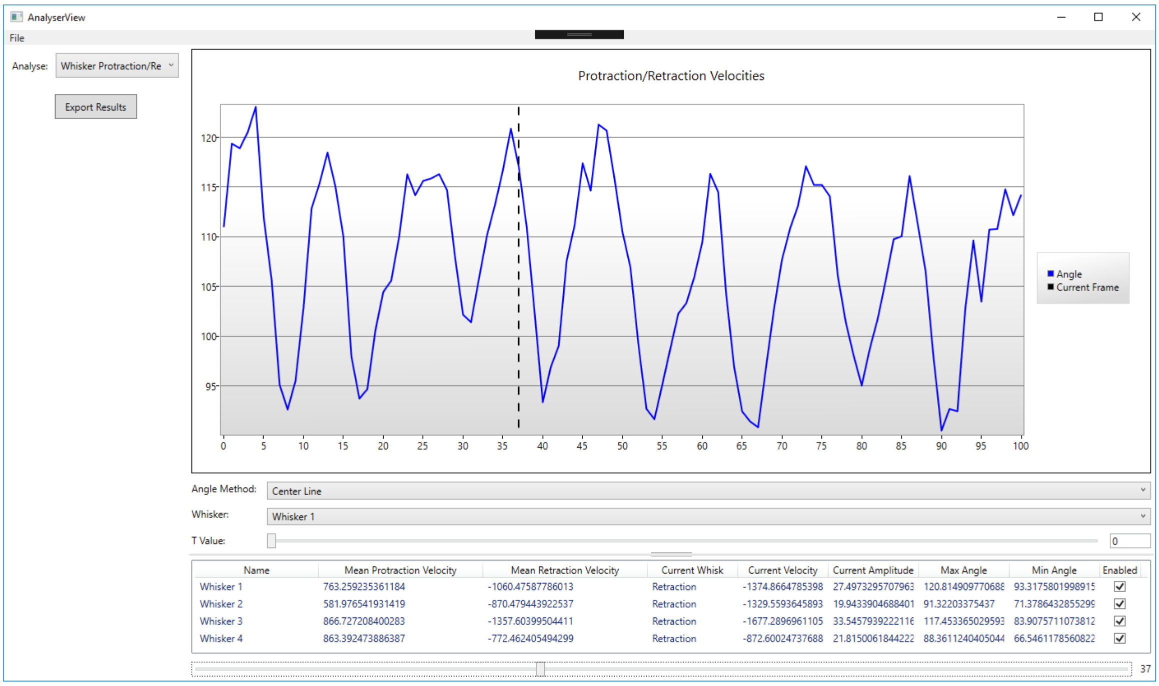
Name	Angle	Enabled
Whisker 1	116.840382422298	☑
Whisker 2	94.8796368001801	☑
Whisker 3	103.136129349796	☑
Whisker 4	84.8246204583096	☑

Figure 4.4: Whisker Angle results

Whisker Protraction/Retraction Velocities

Every whisk cycle will begin with a protraction, followed by a retraction. The velocity of the protractions and retractions are closely linked to behaviour, and can achieve speeds of up to 1000 degrees/second [3]. During contact with the object, retraction velocity is reduced leading to a longer contact duration [8]. Protraction velocities are slower than retraction velocities, and it was believed that retraction was a passive process without muscular activation. It has now been shown this is true for foveal whisking, but retractions during exploratory whisking are under active control [23].

Once the angle has been found, individual protraction/retraction cycles can be tracked. By smoothing the angle signal, locating the peaks and then finding the nearest peaks on the unsmoothed signal (like the Autocorrelogram [24]), the cycles could be identified. The protraction and retraction velocities are measured in degrees/second.



Angle Method: Center Line

Whisker: Whisker 1

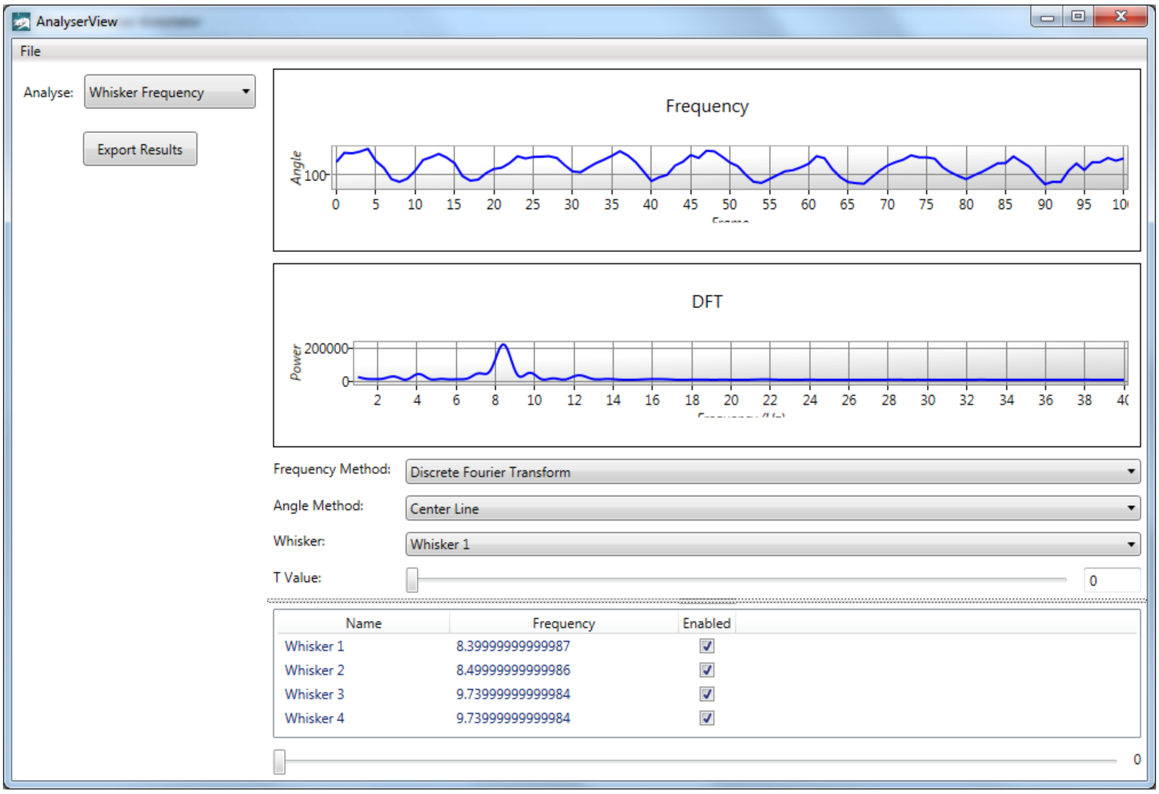
T Value: 0

Name	Mean Protraction Velocity	Mean Retraction Velocity	Current Whisk	Current Velocity	Current Amplitude	Max Angle	Min Angle	Enabled
Whisker 1	763.259235361184	-1060.47587786013	Retraction	-1374.8664785398	27.4973295707963	120.814909770688	93.3175801998915	☑
Whisker 2	581.976541931419	-870.479443922537	Retraction	-1329.5593645893	19.9433904688401	91.32203375437	71.3786432855299	☑
Whisker 3	866.727208400283	-1357.60399504411	Retraction	-1677.2896961105	33.5457939222116	117.453365029593	83.9075711073812	☑
Whisker 4	863.392473886387	-772.462405494299	Retraction	-872.60024737688	21.8150061844222	88.3611240405044	66.5461178560822	☑

Figure 4.5: Protraction and Retraction results

Whisking Frequency

The frequency at which mice whisk is closely related to certain behaviours such as locomotion and object exploration, with lower frequencies observed during locomotion and higher frequencies observed during foveal whisking against objects [15]. Two methods are provided for calculating frequency: Autocorrelogram [24] and Discrete Fourier Transforms [25] (Fig 2). The signal used to obtain the frequency is the angle of the whisker over the duration of the video. As there are multiple ways of defining the angle, the option to choose which method and the T Value is provided. The frequency is measured in Hertz (Hz), and is shown in figure 4.6.



Analyse: Whisker Frequency

Frequency Method: Discrete Fourier Transform

Angle Method: Center Line

Whisker: Whisker 1

T Value: 0

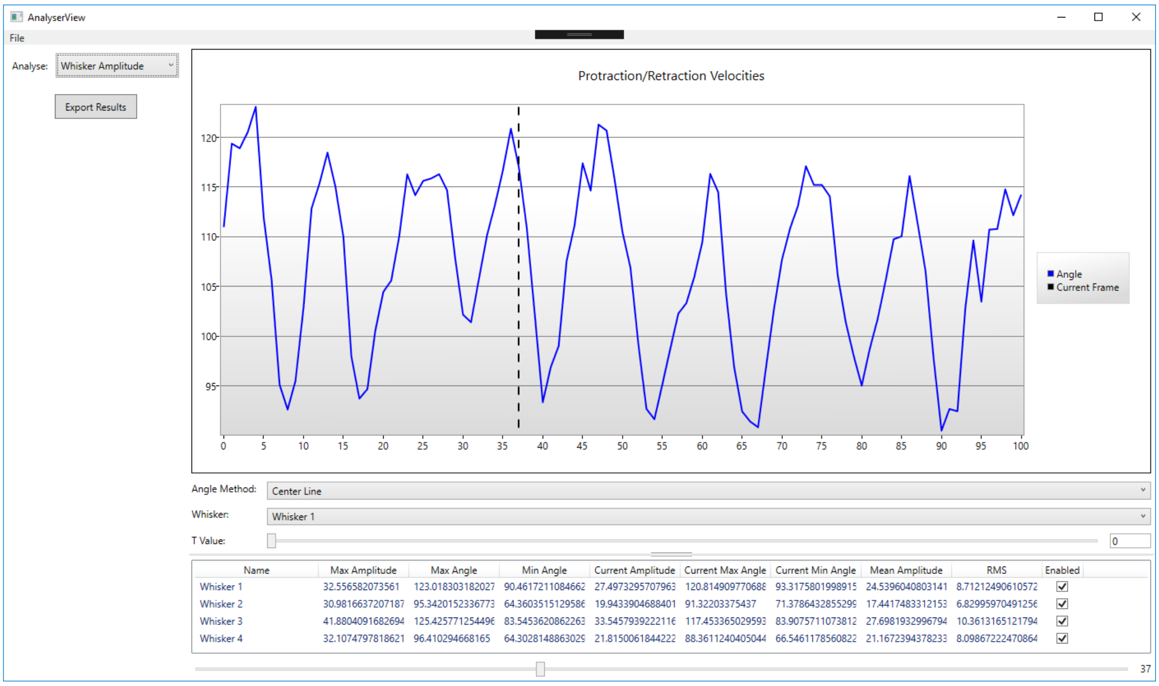
Name	Frequency	Enabled
Whisker 1	8.39999999999987	☑
Whisker 2	8.49999999999986	☑
Whisker 3	9.73999999999984	☑
Whisker 4	9.73999999999984	☑

0

Figure 4.6: The frequency section in the analytical suite. The top graph shows the angle of the whisker over time, with the bottom graph showing the power spectrum of the Discrete Fourier Transform.

Whisker Amplitude

During whisking, amplitudes vary from between 10 100 degrees [3], with much larger amplitudes observed during exploration, and smaller amplitudes during foveal whisking [15]. Upon contact with an unexpected object, ipsilateral whiskers will cease protraction and reduce amplitude to investigate the object further [10]. This enables the whiskers to make light touches upon the surface, which is thought to improve the quality of the information gathered by the whisker follicle [23]. As the angles are already known, and the protraction/retraction cycles identified, the amplitude is calculated as the difference in angle between the beginning and end of the cycle. The average amplitude, and minimum and maximum angles are given for each whisk cycle, and the amplitudes are measured in degrees.



Analyse: Whisker Amplitude

Angle Method: Center Line

Whisker: Whisker 1

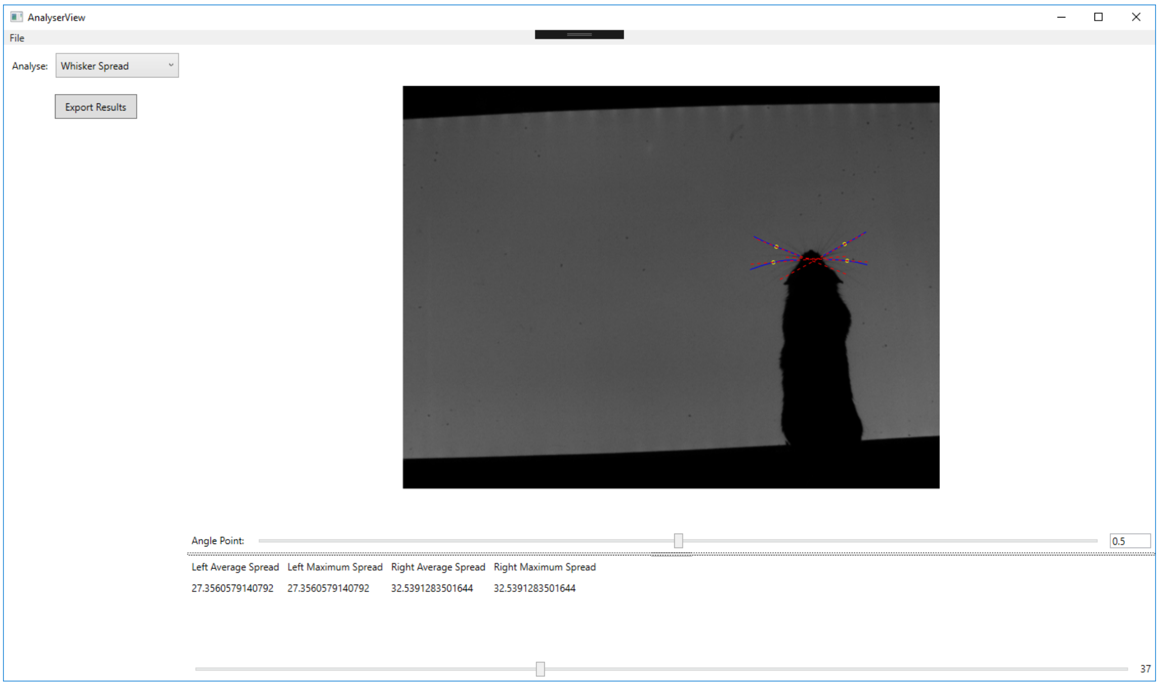
T Value: 0

Name	Max Amplitude	Max Angle	Min Angle	Current Amplitude	Current Max Angle	Current Min Angle	Mean Amplitude	RMS	Enabled
Whisker 1	32.556582073561	123.018303182027	90.4617211084662	27.4973295707963	120.814909770688	93.3175801998915	24.5396040803141	8.71212490610572	☑
Whisker 2	30.9816637207187	95.3420152336773	64.3603515129586	19.9433904688401	91.32203375437	71.3786432855299	17.4417483312153	6.82995970491256	☑
Whisker 3	41.8804091682694	125.425771254496	83.5453620862263	33.5457939222116	117.453365029593	83.9075711073812	27.6981932996794	10.3613165121794	☑
Whisker 4	32.1074797818621	96.410294668165	64.3028148863029	21.8150061844222	88.3611240405044	66.5461178560822	21.1672394378233	8.09867222470864	☑

Figure 4.7: Whisker Amplitude results

Whisker Spread

Another important component in the whisking cycle is the spread of the whiskers. This spread varies over the course of a whisking cycle, with whiskers spreading out during protraction, and reducing in spread during retraction [8]. During tactile discrimination, whisker spread is substantially reduced in order to increase the number of whisker-surface contact points [8]. As each individual whisker being annotated is identifiable and the angle of the whisker is already known, calculating the spread is as simple as the difference in angle between the whiskers. The only difficulty is identifying which side of the head the whisker is on. This can only be achieved if the center line is known, and as such, this metric can only be calculated if the user has chosen to include the nose point and the orientation point. Two spread measurements are given, the maximum spread (angle between foremost and rearmost whisker), and the average spread. Both are measured in degrees.



Analyse: Whisker Spread

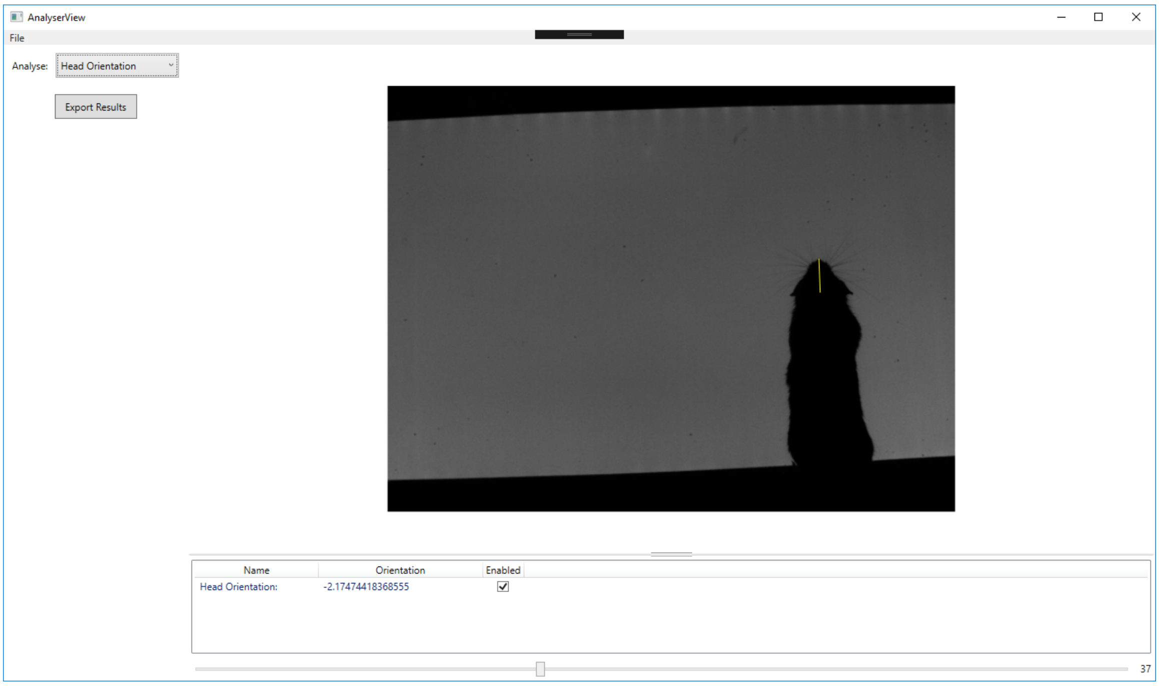
Angle Point: 0.5

Left Average Spread	Left Maximum Spread	Right Average Spread	Right Maximum Spread
27.3560579140792	27.3560579140792	32.5391283501644	32.5391283501644

Figure 4.8: Whisker Spread results

Head Orientation

As a rat orientates its head, whiskers will move asymmetrically in order to search the area that the head is moving towards [26]. It has also been found that mice will re-orientate their head after a whisker has come in contact with an object to align itself with the object for further investigation [26]. Head orientation can only be calculated by the software if the user has selected to include a nose point and a orientation point in the annotation settings. If both of these parameters have been included, then head orientation is calculated as the angle between the vertical, and a line generated by the head point and orientation point. Head orientation is measured in degrees.

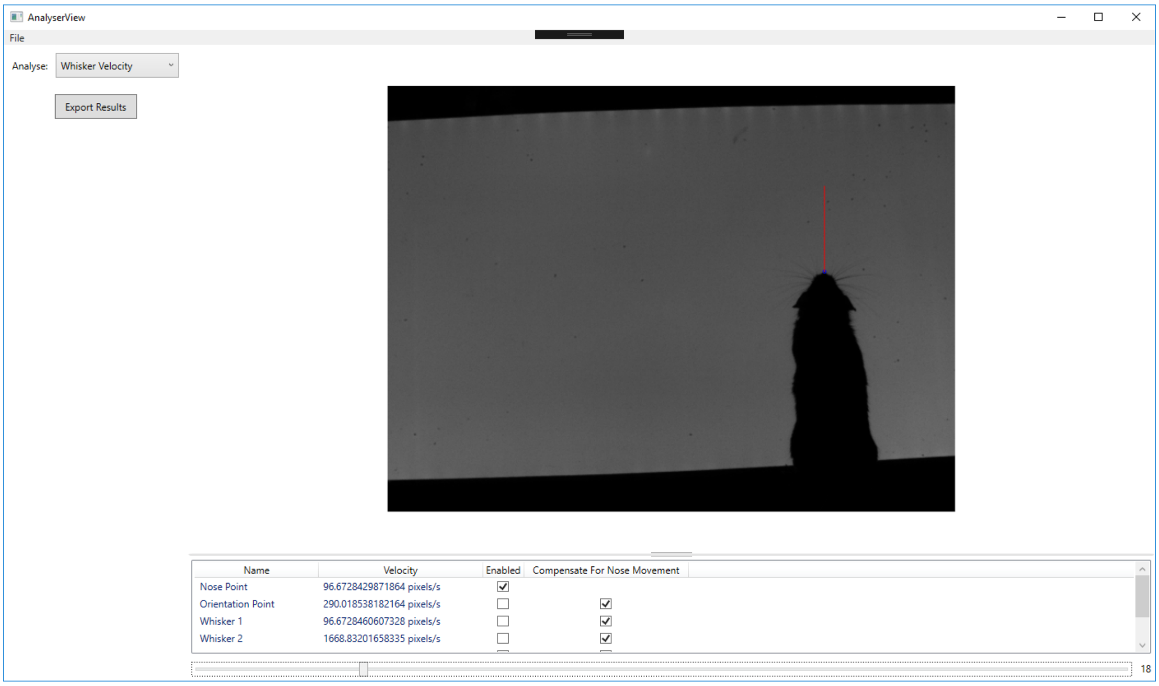


Name	Orientation	Enabled
Head Orientation:	-2.17474418368555	☑

Figure 4.9: Head Orientation results

Locomotion Velocity/Distance Travelled

Whisking behaviour is also tightly coupled with locomotion. Rats use their whiskers as collision detectors when running, by ceasing whisking and positioning their whiskers far out in front of them [27, 28]. Due to the association with whisking and locomotion, locomotion always needs to be controlled for, to make sure observed differences in whisking are not simply caused by changes in locomotor activity [7]. Nose Displacement is the sum of distances travelled between frames by the nose point. Point Velocity is calculated as the change in position over time, and since we know how far the point has moved between frames, and the frame rate of the video, this was quite easy to calculate. If the animal in the video is moving though, and a point attached to the animal was not moving, the attached point would still appear to have a velocity. Because of this, nose point compensation was created, allowing the user to subtract the nose points velocity from the other points. The distance and velocity and measured in pixels and pixels/second respectively.



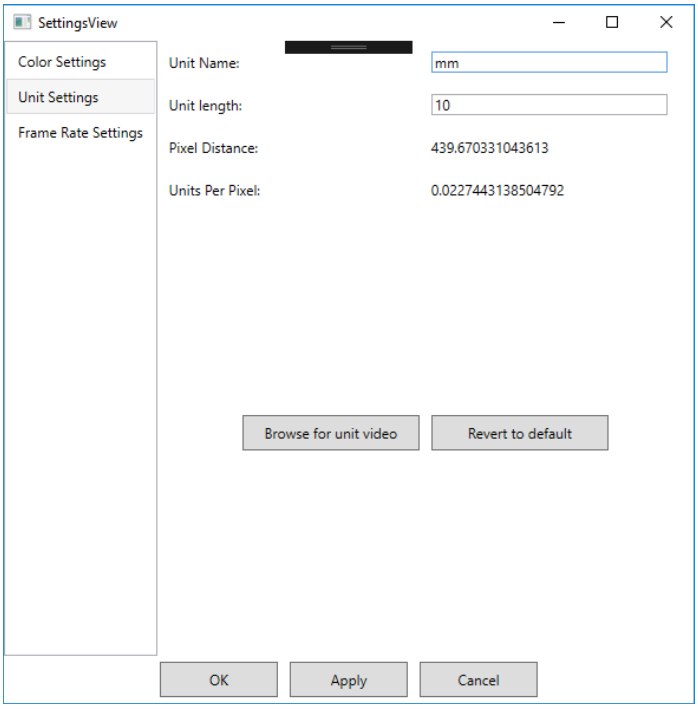
Analyse: Whisker Velocity

Name	Velocity	Enabled	Compensate For Nose Movement
Nose Point	96.6728429871864 pixels/s	☑	
Orientation Point	290.018538182164 pixels/s	☐	☑
Whisker 1	96.6728460607328 pixels/s	☐	☑
Whisker 2	1668.83201658335 pixels/s	☐	☑

Figure 4.10: Locomotion Velocity results

Calibration

Metrics involving velocity were originally calculated in pixels per second, but as development proceeded a potential error was discovered. Many of the videos were taken using a high speed camera, and the footage had been significantly slowed down for the user. Unfortunately, it is not possible to determine the original frame rate from the meta-data. So if a user wishes to alter the frame rate used to calculate the velocity, they must manually enter the original frame rate. Many videos also come with an additional calibration video, allowing you to calculate the real world size. The program therefore also provides the option of loading in one of these videos, specifying the calibration points and enter the real distance between these points. Doing this will allow the user to take measurements in m/s, mm/s or any other unit of choice, instead of pixels/s.



Unit Name: mm

Unit length: 10

Pixel Distance: 439.670331043613

Units Per Pixel: 0.0227443138504792

Figure 4.11: Calibration window

Summary of Metrics

Whisker and locomotion movements can be quantified in many different ways, table 4.5 shows the metrics produced by the manual whisker annotator.

Variable	**Unit**	**Description**
Locomotion Variables		
Distance Travelled	mm	The total distance travelled by the rodent throughout the annotation
Velocity	mm/s	The velocity the rodent is travelling at
Head Orientation	$degrees$	The orientation of the head of the rodent
Single whisker variables		
Protraction	$degrees$	Forward(rostral) movement
Retraction	$degrees$	Backwards (caudal) movement
Angular Position	$degrees$	The angle the whisker makes with the center line of the snout
Angular Velocity	$degrees/s$	The speed at which the whisker is rotating
Linear Velocity	mm/s	The linear speed of a point on the whisker shaft
Curvature	mm^{-1}	How much the whisker bends
Mean Offset	$degrees$	The average angular position of a whisker
Frequency	Hz	The rate at which rodents whisk
Amplitude	$degrees$	The size of the whisk cycle (peak to peak)
Multiple Whisker Variables		
Spread	$degrees$	The size of the spread of the ipsilateral whiskers
Whisk Cycle		The amount of times a full whisk cycle is completed

4.6 Application of MWA for validation

The Manual Whisker Annotator (MWA) was designed as a validator for the subsequent trackers and detectors, by providing ground-truth data of coordinates and whisker metrics. In the subsequent chapters, (Chapter 5 and Chapter 6), the MWA is used to validate the outputs of both the Automated Rodent Tracker (ART) and the Fully Automated Whisker Detector (FAWD), by comparing their coordinate positions to MWA. It is also incorporated in to the software as a calibrator, to provide known measurements to coordinates within an image or video.

4.7 Discussion

We present here the Manual Whisker Annotator (MWA), a piece of software designed specifically for tracking a rodents nose tip and whiskers. MWA is a freely available, open source alternative to other trackers [160, 85]. It has been designed to be intuitive, easy to use, and require as little action as possible from the user. Annotation speed is increased up by allowing the user to track each frame, whiskers, and points on each whisker sequentially. Once annotated, whiskers are automatically constructed from the annotated points, removing the need for the user to do this manually when using other trackers. An analytical suite is also provided, again decreasing overall time needed by supplying the required metrics immediately after annotation. In addition, it uses the MVVM software design pattern, allowing easy integration of any metric analysis that is not provided by the software.

Chapter 5

Development, demonstration and validation of an Automated Rodent Tracker

Based on publication: A novel automated rodent tracker (ART), demonstrated in a mouse model of amyotrophic lateral sclerosis.

5.1 Chapter summary

This chapter discusses the development of the head and nose tip tracking software using a novel method. The proposed method is fully automated, capable of locating the nose tip in under 4ms, and is validated against ground truth annotations. The dataset the software was validated against was a series or high speed recordings collected by Dr. Robyn Grant for use in publications [84, 86, 85, 81].

5.2 Introduction

There are a number of commercial rodent trackers, that are able to capture aspects of rodent movement in an open arena (i.e Ethovision [211]; Any Maze (Stoelting CO, USA); VideoTrack (View Point, France), and Smart Video Tracking (PanLab Harvard Apparatus, USA). These are usually able to measure distance travelled, speed of movement, position, and sometimes even identify the occurrence of certain behaviours, such as wall-following, or rearing. However, these programs can be expensive, require quite a lot of initial user input and are unable to be modified by users (i.e are not open source) [167]. A variety of open source alternatives exist; however, these tend to be very specific to certain experimental lab set-ups (i.e. written to interact with experimental arenas, such as OpenControl) [3] or still require a lot of manual intervention during set-up (Ethowatcher). Many of these programs have not been reliably validated against other methods, or in other labs, and require coding skills to adapt the code to other set-ups or extract alternative metrics [167]. Therefore, despite advances in technology and tracking software, there does not yet exist a free, open source rodent tracker that requires little user interaction in order to process large quantities of behavioural data from video footage. Indeed, as technology continues to develop, current abilities to collect large video-based datasets are not matched by a concomitant ability to reliably and efficiently analyse such data volumes. Such mismatch in capabilities limits potential knowledge advances in associated fields and as such requires urgent attention to amend the current situation. This chapter, therefore, introduces a novel Automatic Rodent Tracker which can batch process large numbers of high quality video files, and provide measurements of locomotion (such as speed and position) and identify behaviours (such as exploration, moving forward, turning), with minimal user interaction. Only one setting is required from the user prior to execution, and error feedback is provided post-analysis, with the option to re-analyse specific videos with a different setting. The tracker itself is modular and open source, enabling other modules to be developed and added to its framework; for instance, future automatic tracking modules, such as for whisker tracking or gait analysis, providing a means for it to be developed by and for the wider research community.

5.3 Datasets

5.3.1 Animals

Mice were originally obtained from the Jackson Laboratory, B6SJL-Tg (SOD1-G93A) 1Gur/J (stock number 002726), and subsequently backcrossed onto the C57Bl/6 background (Harlan UK, C57Bl/6 J OlaHsd) for >20 generations (Mead et al. 2011). Our model, on a defined inbred genetic background, shows no effect of sex or litter of origin on survival (Mead et al. 2011). The transgenics have an average survival of 140 days (Mead et al. 2011). For this study five SOD1G93A mice and five control mice were imaged in a 7 T magnet, at post-natal day (P) 30, P60, P90 and P120 (3 days), to measure loss of pelvic muscle volume. They were also filmed in an open arena three days post-imaging. All the animals were female kept on a 12:12 light schedule at 22 C, with water and food ad libitum. All procedures were approved by the local Ethics Committee and UK Home Office, under the terms of the UK Animals (Scientific Procedures) Act, 1986.

5.3.2 Behavioural Data Collection

High-speed digital video recordings were made using a Photron Fastcam PCI camera, recording at 500 frames per second, shutterspeed of 0.5 ms, and resolution of 1024 1024. The camera was suspended from the ceiling, above a custom-built rectangular (40 cm 40 cm) viewing arena with a glass floor, ceiling, and endwall. A Perspex block (55 cm x 55cm) was placed in the arena to encourage exploration of the space (Figure 1a). The mice were recorded at P30, P60, P90 and P120 on two consecutive days. Video data was collected in near darkness and back-lit using a normal spectrum lightbox for illumination. Multiple 1.6 second video clips were collected opportunistically (by manual trigger) when the animal moved beneath the field of view of the camera. Approximately 12 videos were collected from each animal per filming day, which only took around 5-10 minutes.

5.4 Methodology

5.4.1 Mouse and Background Detection

Firstly, videos were saved from the camera as Audio Video Interleaved (.avi) format and all videos were loaded to the Automatic Rodent Tracker (ART) for analysis via batch processing. The tracker firstly identified the presence of a mouse by using background subtraction to find moving parts within the image sequences (Figure 5.1b) in each video frame. A minimum convex hull [61, 6] was used to enclose the mouse and was adapted by using hull bending to better fit the shape of the mouse in each frame. The enclosed mouse was subtracted from the frame to predict the background. Canny edge detection [30] was then used to detect the background contours, including the edges of the Perspex arena, the Perspex block and other contours, like sawdust or faecal material within the arena. A binary background image was then created (Figure 5.1c), which allowed all background objects to be subtracted from each frame, leaving only the mouse visible (yellow contour in Figure 5.1d). The proposed method enabled the detection of the mouse and is robust to occlusion or when the mouse is interacting with other objects (Figure 5.1d).

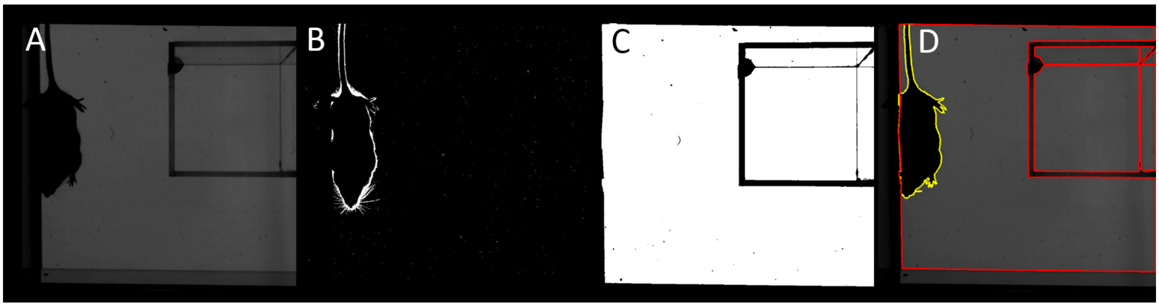


Figure 5.1: Processing of the videos on a per-frame basis. a) original image; b) background subtraction finds the moving mouse c) binary image of the predicted background, following preprocessing; d) contour image of the mouse in yellow, Perspex block and arena edges in red.

5.4.2 Mouse Head Detection

For this work, it was necessary to detect the positioning and orientation of a rodent's head. The datasets available would show the mouse from a top down view, against a bright backdrop. The problems mainly encountered were that the mice could be pointed in any direction, it could be only partially visible, obscured by an artefact or not present at all. Different species of mice can also have very differently shaped heads. Rule based sliding keypoints addresses all of these issues.

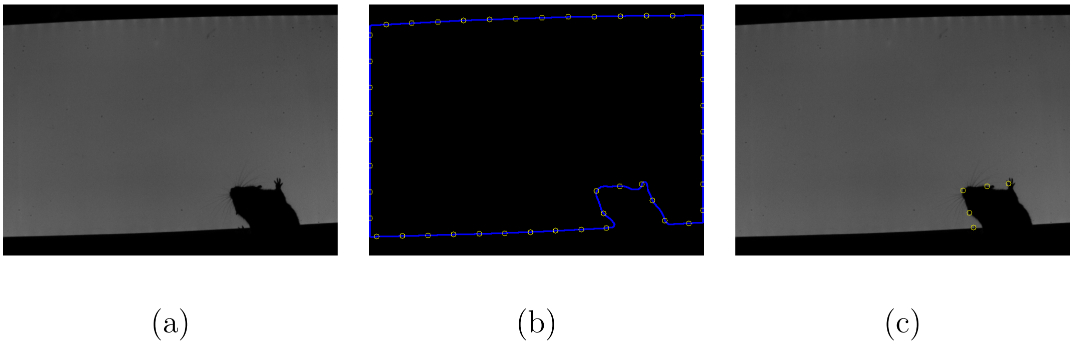


Figure 5.2: The RBSK overview: (a) Original Image; (b) Keypoints placed along the contour; (c) The keypoints that best match the rules.

Pre-Processing

For the detection algorithm to run at its most optimal, the pre-processing steps should remove as many contours as possible that are definitely not part of the mouse. The image is first converted to a grayscale image, and a median filter [108] is applied. This removes noise from the image while preserving edges. The image was then converted to a binary image [203]. Each pixel intensity $I_{(x,y)}$ is converted to either white (255) or black (0) using the following formula:

$$I_{(x,y)} = \begin{cases} 255 & \text{if } I_{(x,y)} \geq 30 \\ 0 & \text{if } I_{(x,y)} < 30 \end{cases}$$

The threshold value of 30 was obtained from multiple experiments. It can remain constant as the way the videos are taken means the mouse will always nearly have an intensity of 0.

Finally the contours are detected by using the Canny Edge Detection algorithm [30].

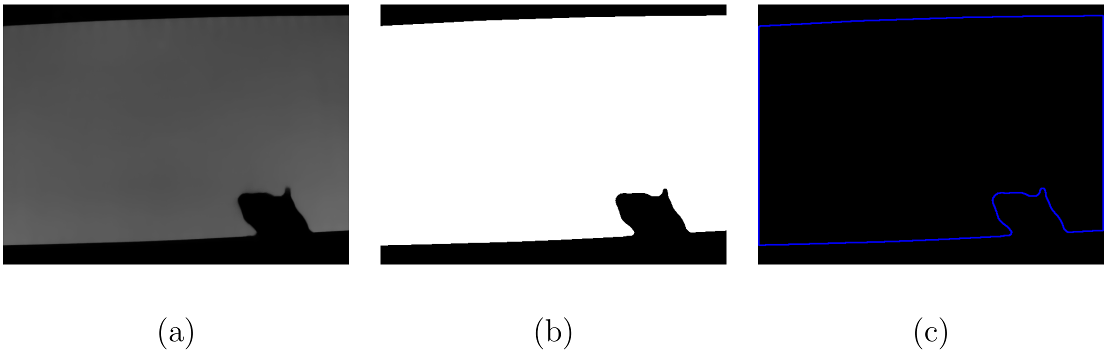


Figure 5.3: The results from Pre-Processing steps: (a) Filtered Image; (b) Binary Image; (c) Contour Image.

In the next section we will introduce our proposed method for detecting the head and nosetip of the mouse.

Rule Based Sliding Keypoints

Rule based sliding keypoints is a novel method for detecting objects using a set of rules. Keypoints are generated at set distances around the contours, and by defining rules which can be applied to these keypoints (like certain keypoints must be within a certain distance of each other, or keypoints may have to be "behind" other keypoints), we can test groups of points to see if any fit the rules defined. However, it is likely that the keypoints won't be generated in exactly the right place, we therefore slide the keypoints several times so they fall as close as possible to the right place on one of the slides.

There are two key settings for this method. The distance between the points d, and the number of times to slide them n. The distance between the points is set dynamically to cater for videos with different resolutions (and as such a different sized mouse in terms of number of pixels). The number of slides is a trade off between accuracy and speed, with each slide comes an increase in accuracy, but a decrease in speed. For a mouse, 20 slides appeared to consistently give accurate results while still executing relatively quick. On every slide, the points are shifted by an offset ϕ which is equal to d divided by n.

Since the keypoints are checked in groups, there are some keypoints that wouldn't get checked. For example, if there were 20 points, and we were taking groups of 5 points, the first group to be checked would consist of points 1, 2, 3, 4 and 5 (with point 3 being the potential head point). The last group would consist of points 16, 17, 18, 19 and 20 (with point 18 as the potential head point). This means points 1, 2, 19 and 20 never get considered to be potential head points. To solve this, several keypoints get generated after the starting keypoint, as shown in figure 5.4.

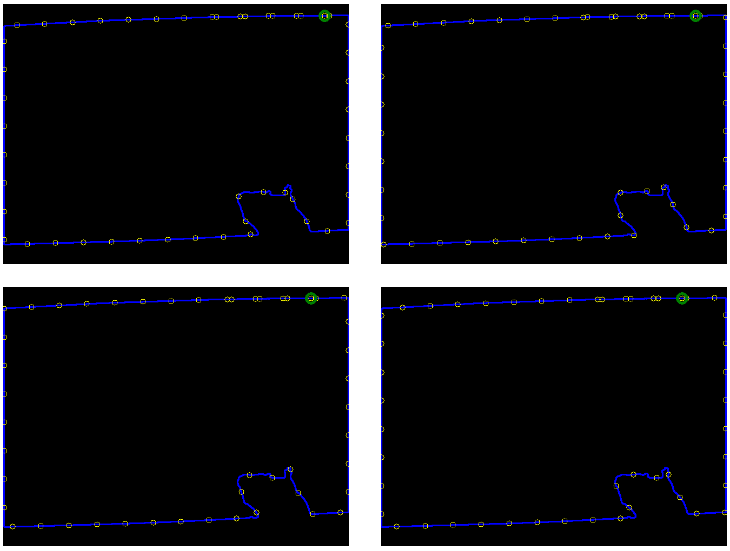
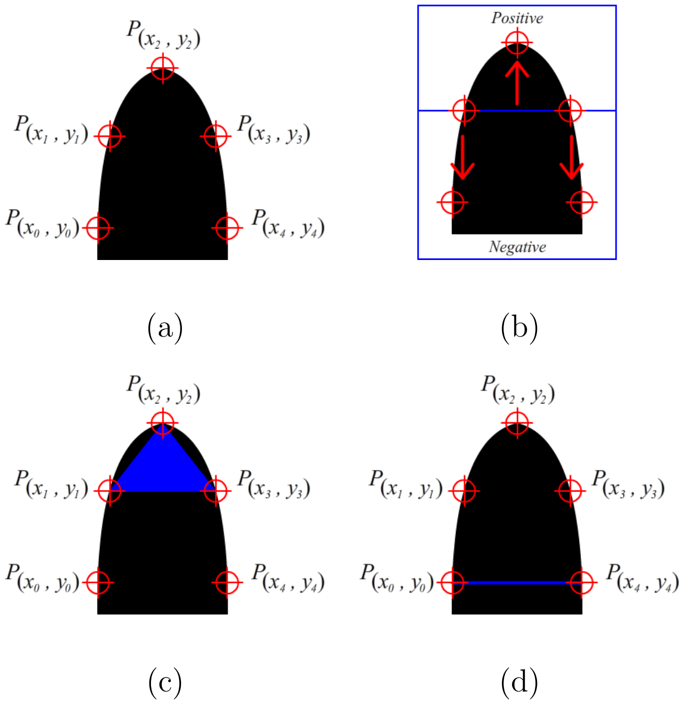


Figure 5.4: The keypoints generated for slides 5, 10, 15 and 20. The green circle indicates the starting point

The Rules

The rules created for finding a mouse head are as follows:

1. Using groups of 5 keypoints with ordering as follows: $P_{(x_0,y_0)}$, $P_{(x_1,y_1)}$, $P_{(x_2,y_2)}$, $P_{(x_3,y_3)}$, $P_{(x_4,y_4)}$ as shown in figure 5.5a.



(a) (b) (c) (d)

Figure 5.5: The various images explaining the rules

2. When a line is drawn between points $P_{(x_1,y_1)}$, and $P_{(x_3,y_3)}$, points $P_{(x_0,y_0)}$, and $P_{(x_4,y_4)}$ must lie on the opposite side of the line from point $P_{(x_2,y_2)}$ as shown in figure 5.5b.

3. The area created by the head points $P_{(x_1,y_1)}$, $P_{(x_2,y_2)}$ and $P_{(x_3,y_3)}$ (figure 5.5c), must exceed a threshold T_A. The threshold is defined as the maximum area a triangle can make, divided by two. This prevents the algorithm from accepting shapes with thin bases (such as a leg) or wide bases (an object poking out from one of the edges).

4. The line drawn between points $P_{(x_1,y_1)}$, and $P_{(x_3,y_3)}$ (bottom of the triangle in figure 5.5c) must cover black pixels. Ten points along the line are checked, and the sum of the black pixels must be greater than the sum of the white pixels.

5. The distance between points $P_{(x_1,y_1)}$, and $P_{(x_3,y_3)}$ (bottom of the triangle in figure 5.5c) must be below an upper threshold T_U. After several tests, the best results were given for $T_U = 1.4 \times d$.

6. The distance between points $P_{(x_0,y_0)}$, and $P_{(x_4,y_4)}$ (figure 5.5d), must lie between between a lower threshold T_L and an upper threshold T_U. After testing, the following values gave the best results.

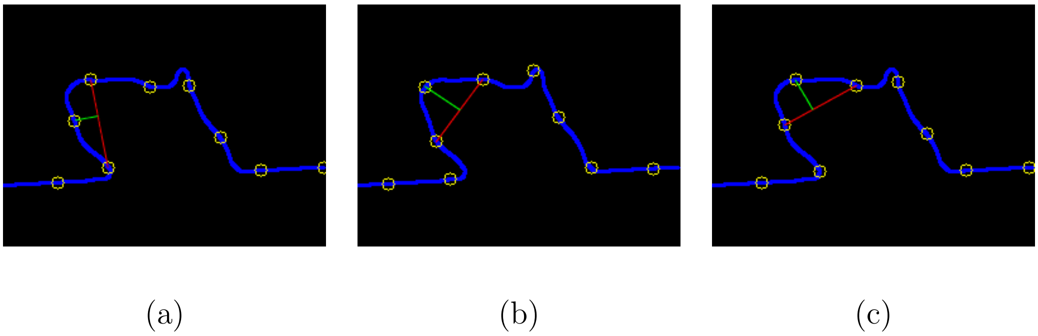
$$T_L = 1.2 \times d \qquad T_U = 2.5 \times d$$

Advanced Rules

Advanced rules are similar to the rules above, as in they remove false positives. However, they take significantly more time to execute than the rules described above, and if executed for every single slide, would cause a significant decrease in performance. If used correctly though, these rules can be executed without a noticeable increase in execution time, and they are explained in greater detail in section 5.4.2. The average intensity of the area defined by points $P_{(x_1,y_1)}$, $P_{(x_2,y_2)}$ and $P_{(x_3,y_3)}$ must fall below an average threshold intensity T_I. It is possible that the edge of a nose is curved and some white pixels may appear in the area generated so this value must be slightly greater than zero. After testing, the following threshold value was found to give consistent results. $T_I = 10$.

Choosing the best slide

If an object is detected, it is more than likely that several slides will have evaluated correctly. To choose the slide that will give us the most accurate result, a probability function is used. For this case, the shortest distance between the point $P_{(x_2,y_2)}$ and the line drawn between $P_{(x_1,y_1)}$, and $P_{(x_3,y_3)}$ will be maximised when a keypoint falls on the nosetip. The diagram below shows the nosetip distance (green line) from the drawn line (red) for 3 different slides.


(a)

(b)

(c)

Figure 5.6: The keypoints from 3 different slides: (a) The first slide; (b) The slide where the Nose tip distance is maximised; (c) The last slide.

Optimisation

Parallel Processing: In the example above the program only has to check one contour. For more complex images with more than one contour, it is possible to run each contour analysis in parallel, since every contour is independent of each other. It is also possible to run each slide in parallel, as again it is independent of the other slides. This can result in super quick processing even for complex images. However, if the image is simple like the one above, then it is best to run the algorithm with no parallel processing, as the overhead generated negates any performance improvement.

Rule Ordering: The ordering of the rules can have a large impact on performance. As soon as a rule fails, the rest of the rules will not be checked. You can improve execution time by putting the simplest rules first, and then following on with more complex rules. If the rule is particularly complex, you may wish to make it an advanced rule.

Rules: Some rules will be computationally expensive, and this expense is multiplied when used over a large dataset. In order to reduce execution time further, we can run these expensive rules after the initial rules have found a list of potential locations and their corresponding probability. The advantage of this is that if a lower probability potential match is found, it will not run the advanced rule.

Sorting: Execution time can be reduced even further by sorting the results found based on probability. This enables the first potential match, which validates against the advanced rules to be used, rather than searching through all the lists for the one with the highest probability.

Using the GPU: For computers running an nVidia graphics card, there are some operations which can be offloaded to the GPU in order to improve performance. The main improvement will be on pre-processing, with filtering times dramatically reduced by using the GPU.

Performance

The main factors that will affect performance are the image resolution, number of contours in the image, the number and complexity of the rules, the gap distance and the number of slides. From the available dataset, the object we're finding doesn't have many features, nor are there many other objects to distinguish from, so the rules are quite simple and take very little time to execute. The number of slides is a trade off between accuracy and performance. The more slides, the longer it will take, but the more accurate it will become, as shown in Figure 5.7. This was performed on the original image above (640 x 480), with a gap distance of 50 pixels. All of the tests are run on an Intel(R) Core(TM) i5 2500 CPU.

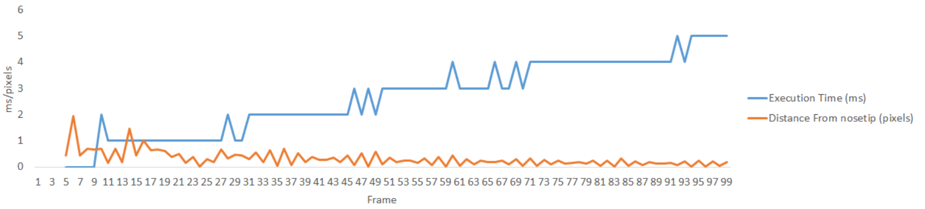
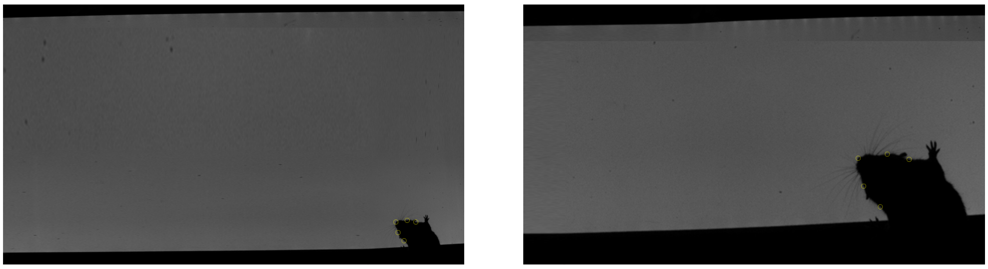


Figure 5.7: Number of iterations vs Execution time

The gap distance is determined by the size of the object that you want to locate. The smaller it is, the more keypoints will be placed and analysed. Below are two versions of the image above, both have been scaled to HD resolutions (1920 x 1080).



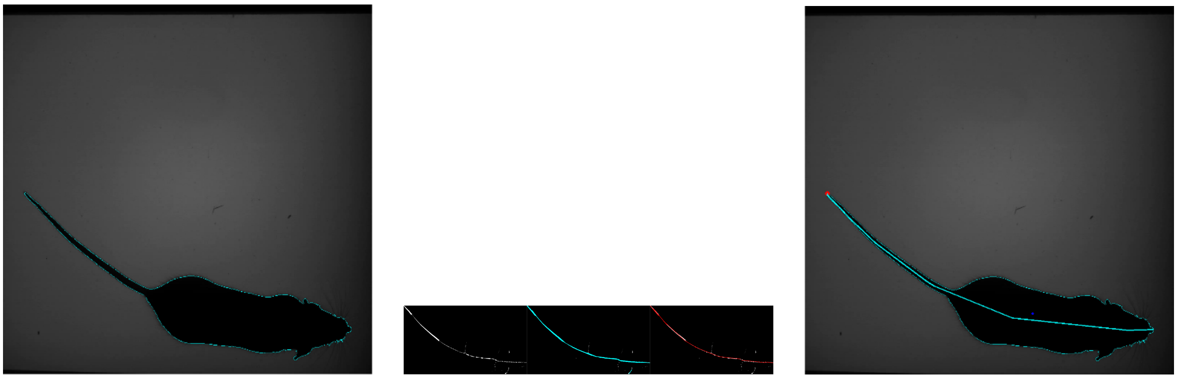
(a) $d = 50$, Execution Time = $5ms$ (b) $d = 150$, Execution Time = $4ms$

Figure 5.8: Execution time for two different gap distances

5.4.3 Body Detection

Spine Fitting

Once the head of the mouse had been found, it was combined with the earlier located mouse contour to define the whole shape of the mouse, along with the position and orientation of the head. When the whole body of the mouse is visible (figure 5.9a), a modified version of the RBSK was used to find both the head and tail tips. Once located, the body contour was isolated from the image and skeletonisation [150] was performed. Hough lines [59] was then used to obtain all line like segments within the skeletonised image (figure 5.9b). The spine was then defined as the shortest route between the nose and tail tip by only traversing the found lines (figure 5.9c).



(a) (b) (c)

Figure 5.9: Mouse contour

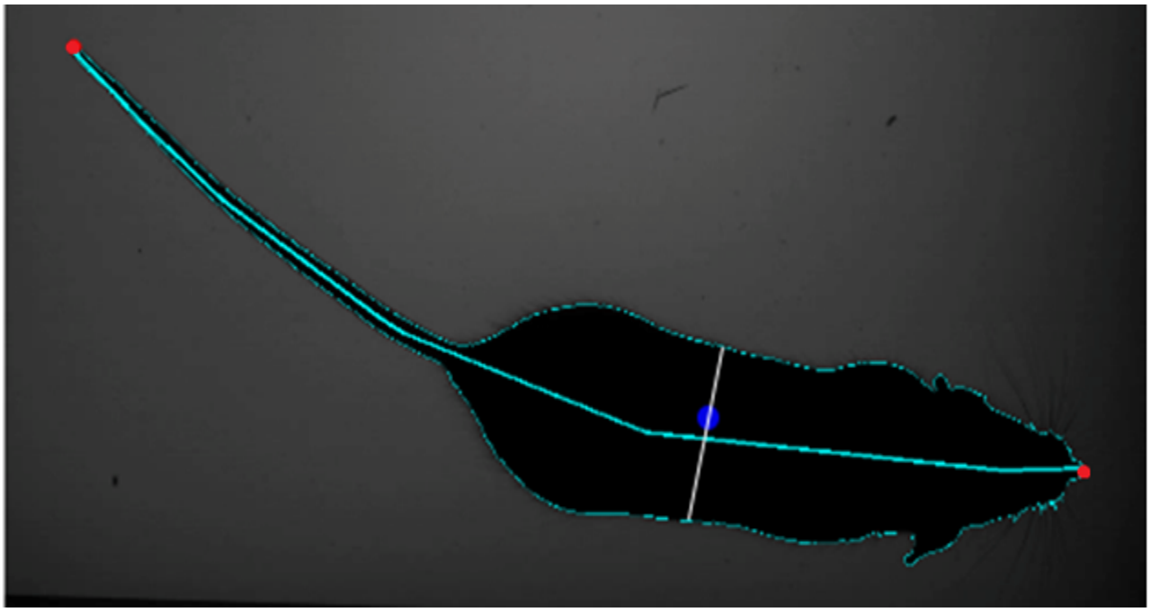


Figure 5.10: Whole mouse contour and spine. Spine is indicated by the light blue line, the white line indicated the centroid width. The nose and tail tips are shown by the red dots, and the center of mass of the body by the blue dot

Centroid Width and Body Weight Approximation

The center of mass of the body segment was located, and a line drawn through this point, perpendicular to the spine and terminated at the outer contour. This line was termed the centroid width (Figure 5.10). This width measurement could be considered as a slice along the width of the animal, and was later correlated to the body weight of the rodent.

5.4.4 Software Architecture

MVVM

The goal of this project was to create a sustainable, open source piece of software that could be continually developed. In order to achieve this, the MVVM design pattern was adopted. This allowed certain aspects of the software to be developed, such as the rodent tracker, before adding more modules to handle whisker tracking.

UI Automation

As the software was designed to be as intuitive as possible, a large amount of effort was taken in order to make the application as user friendly as possible. Providing wizards where possible to guide the user through the necessary procedures in order to start tracking, providing detailed error feedback in the event of a tracking failure, and not allowing the user to change settings that could disrupt tracking once it has begun by disabling certain aspects of the application.

Batch Processing

As the software would be mainly used on large datasets, batch processing was integrated into the application. Again, with the end user in mind, this was developed to be as intuitive and as easy as possible to use. Settings could be applied to the entire dataset, or to specific videos. Once ran, if there were any issues with particular videos, these single videos could be ran again with different settings, or excluded from the results entirely.

5.5 Validation

ART was validated against manual tracking and Ethovision XT 5.1.220. ART contains a validation option, so a user is able to validate the accuracy of ART for their set-up against a manual tracking example. In this instance, the accuracy of ART to find the nose point pixel coordinate (in red in Figure 5.10), was compared to manual tracking of the nose point of 43 videos, using the Manual Whisker Tracker (MWA) [103], by a trained and experienced experimenter. The error difference (in mm) between the centre of the nose point in ART and MWA was calculated as an average in each of the 43 videos, over a period of 302-800 frames, and presented as a histogram. Five videos were tracked an additional time by the same experimenter to approximate manual error. In addition, a Bland-Altman plot [20] was constructed to compare manual tracking and ART in all the frames that contained nose tracking, in both the x and y planes.

As Ethovision XT 5.1.220 did not find a nose point, but rather used the center of mass of the mouse body, this pixel coordinate was compared between ART and Ethovision XT 5, over the same 45 videos. The error difference (in mm) between the centre of mass point in ART and Ethovision was calculated as an average in each of the 43 videos, over the same period of 302-800 frames. In addition, a Bland-Altman plot was constructed to compare Ethovision tracking and ART in all the frames that contained centre of mass tracking, in both the x and y planes. During the tracking set-up in ART and Ethovision, the minimum number of clicks out of the total number of 43 videos was also recorded, to estimate user intervention. Ethovision and ART have rather different tracking strategies, for example ART found both the nose point and the center of mass, while Ethovision only tracked the center of mass. Therefore, due to the additional processing steps in ART, running time was not compared between these programs.

5.6 Statistical Considerations

All metrics were normally distributed, therefore parametric statistical tests were used throughout. A multivariate ANOVA was conducted on all the measures collated from the video footage. Average speed, average angular speed, centroid width and the percentages of time moving forward, turning, staying still and interacting were all inputted to the ANOVA as dependent variables. Genotype (SOD1 or control) and age (P30, P60, P90 and P120) were added as independent variables and Mouse ID as a covariate, to control for individual differences. Significant main and interaction effects were reported in the results to a significance level of <0.05, along with effect sizes (Partial Eta Squared: η^2p). Individual ANOVAs were then conducted as post hoc tests at each age, and indicated on the figures with an asterisk (*). Body weight and pelvic muscle volume were also examined using a multivariate ANOVA and were entered as dependent variables, with genotype and age as independent variables. As only one measure was collected per mouse for these variables, mouse ID did not need to be included in the analyses. Centroid width was also averaged, per mouse, and correlated to body weight and pelvic muscle volume using a Pearson's correlation.

5.7 Results

5.7.1 Validation against manual tracking

The average difference (error, mean ± sd) in the nose point coordinate between ART and a manual tracker was 0.60 ± 0.24 mm. This can be seen in Figure 5.11e, which shows the average range of errors per clip in a histogram, with no error being larger than 1.4 mm. The experimenter error, defined as the average distance between multiple annotations of the same video, was calculated as 0.62 ± 0.37 mm. Figure 5.11a and 5.11c show a Bland-Altman plot of the agreement between manual tracking and ART, in the x and y plane, respectively. The vast majority of points fell around zero, and were well within 3 standard deviations of zero (dashed line in figures), especially in the x plane, which had a particularly low difference in tracking between the two methods (Figure 5.11a).

5.7.2 Validation against Ethovision

The average difference (error, mean ± sd) in the centre of mass coordinate between ART and Ethovision was 0.78 ± 0.68 mm (n=43 videos). This can be seen in Figure 5.11f, which shows the average range of errors, per clip, in a histogram, with no error being larger than 4.6 mm. As well as the error, the minimum number of clicks it took to set up each tracking video was also recorded. Ethovision took a minimum of 30 clicks, whereas ART took a minimum of 6 clicks. Figure 5.11b and 4d show a Bland-Altman plot of the agreement between Ethovision and ART, in the x and y plane, respectively. The vast majority of points fell around zero, and were well within 3 standard deviations of zero (dashed line in figures). The difference in tracking between ART and Ethovision, was much lower than that of ART and Manual Tracking, for instance the 3 standard error lines (dashed lines) in Figure 5.11b and 4d were much lower than those in Figure 5.11a and 5.11c.

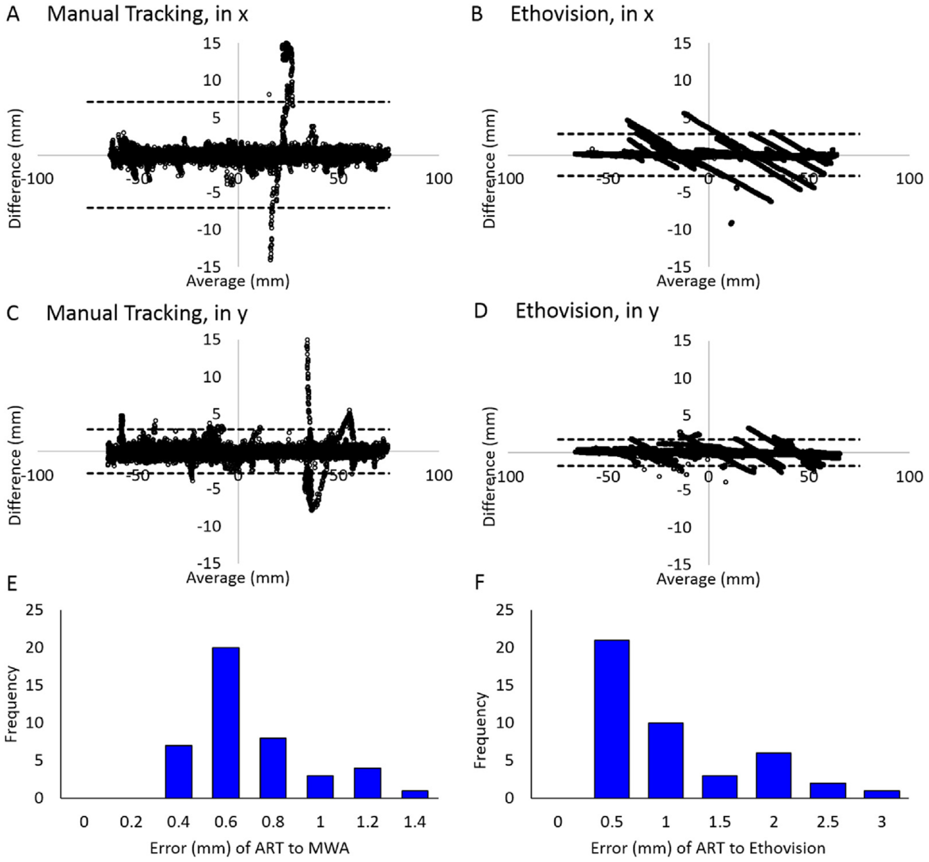


Figure 5.11: a) Bland-Altman plot comparing ART and manual tracking for the x coordinates of the nose point; b) Bland-Altman plot comparing ART and Ethovision for the x coordinates of the centre of mass; c) Bland-Altman plot comparing ART and manual tracking for the y coordinates of the nose point; d) Bland-Altman plot comparing ART and Ethovision for the y coordinates of the centre of mass. Dashed lines correspond to 3 standard deviations. e) a histogram of error (in mm) of the coordinate point of the nose, between ART and manual tracking; f) a histogram of error (in mm) of the coordinate point of the centre of mass, between ART and Ethovision.

5.7.3 Demonstration of ART on an experimental dataset of SOD1 and control mice

Locomotion was impacted in SOD1 mice

The average locomotion velocity was significantly reduced in SOD1 mice (Multivariate ANOVA Main Effect: F(1,327) = 16.414, p<0.001, η^2p = 0.049), at both P30 and P120 (Figure 5.12a). The SOD1 mice had a tendency to rotate more, indicated by elevated angular velocities at P60 and P90 (Figure 5.12b), however, this was not a significant finding (Multivariate ANOVA Interaction Effect: F(3,327) = 2.4432,

$p=0.065$, $\eta^2p = 0.022$).

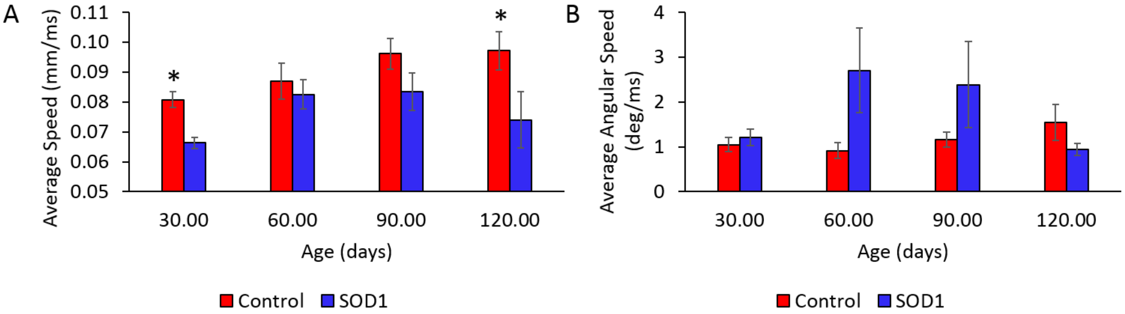


Figure 5.12: Locomotion comparisons in SOD1 and control mice. a) Average locomotion speed was significantly slower in SOD1 mice at P30 and P120. b) Average angular (turning) speed was not significantly affected, despite some elevations at P60 and P90. Graphs show mean values, with standard error bars, significant interactions are indicated by an asterisk (*).

General behaviours were altered in SOD1 mice

Overall, SOD1 mice spent less time interacting with the Perspex block within the arena (Multivariate ANOVA Main Effect: $F(1,327) = 4.762$, $p=0.030$, $\eta^2p = 0.015$) and more time stood still (Multivariate ANOVA Main Effect: $F(1,327) = 6.198$, $p=0.013$, $\eta^2p = 0.019$) (Figure 5.13). In particular, at P120 SOD1 mice spent less time moving forwards (Multivariate ANOVA Interaction Effect: $F(3,327) = 6.508$, $p<0.001$, $\eta^2p = 0.058$) and more time stood still (Multivariate ANOVA Interaction Effect: $F(3,327) = 7.209$, $p=<0.001$, $\eta^2p = 0.058$) (Figure 5.13b). There were no significant effects of genotype or age on the percentage of time spent turning.

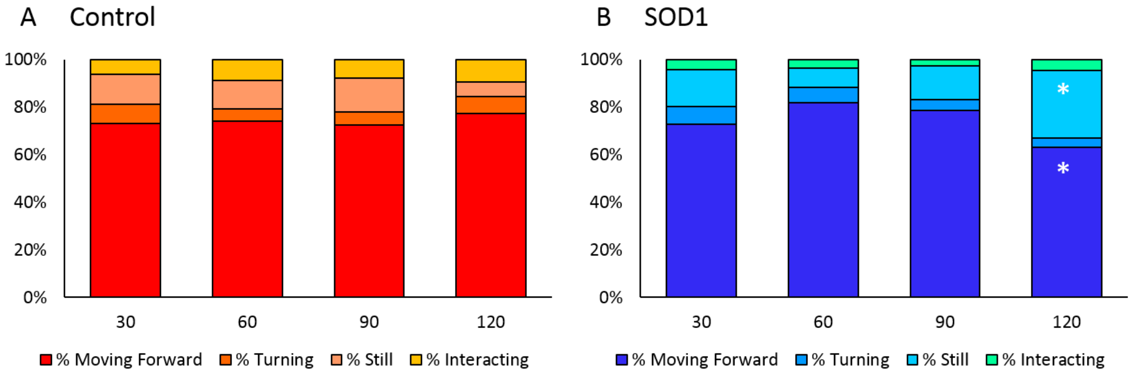


Figure 5.13: Percentage of time spent moving forward, turning, still and interacting in SOD1 (a) and control mice (b). Overall, SOD1 mice spent less time interacting with objects. At P120 SOD1 mice spent less time moving forwards and more time still. Graphs show the percentage time allocated to each of the defined behaviours, significant interactions are indicated by an asterisk (*).

Body size and weight were affected in SOD1 mice

SOD1 mice weighed significantly less than control mice at P90 and P120 (Multivariate ANOVA Main Effect: F(1,327) = 84.691, p<0.001, η^2p = 0.738) (Figure 5.14a), and also had a reduced pelvic muscle volume at P60, P90 and P120 (Multivariate ANOVA Main Effect: F(1,327) = 251.874, p<0.001, η^2p = 0.894) (Figure 5.14c). This can be seen clearly in the example MRI Pelvic Muscle images in Figure 5.14f. Centroid width was significantly reduced in SOD1 mice, in all age groups (Multivariate ANOVA Main Effect: F(1,327) = 201.733, p<0.001, η^2p = 0.388) (Figure 5.14e). Centroid width was also significantly correlated to both body weight (Pearsons Correlation: r=0.766, p<0.001)(Figure 5.14b) and pelvic muscle volume (Pearsons Correlation: r=0.420, p=0.009) (Figure 5.14d).

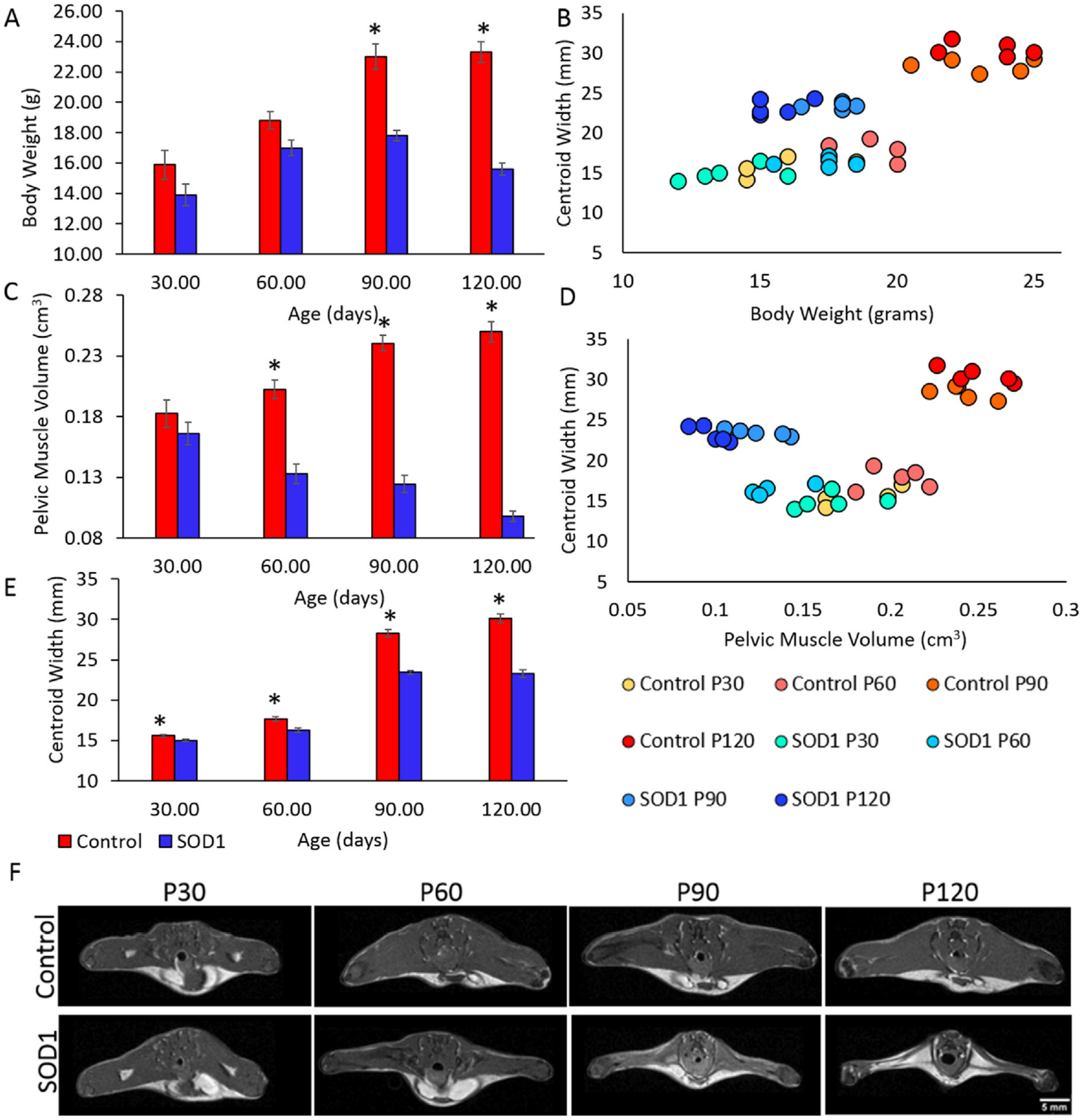


Figure 5.14: Body size and weight measurements in SOD1 and control mice. a) Body weight was significantly lower in SOD1 mice at P90 and P120. c) Pelvic muscle volume, taken from MRI measurements (an example shown in panel f) were reduced in SOD1 mice at P60, P90 and P120. e) Centroid width, taken from the video, was lower in SOD1 mice throughout. Centroid width was significantly correlated to both body weight and pelvic muscle volume. a, c and e panels show mean values, with standard error bars, significant interactions are indicated by an asterisk (*). b and d panels show a result per mouse, per age.

5.8 Discussion

We present here the Automatic Rodent Tracker (ART), and show that it i) performs similarly to manual tracking; ii) performs well compared to Ethovision; and iii) can extract important locomotion measurements, and behavioural classifications, from video data of freely behaving healthy and mutant mice.

5.8.1 Validating and comparing ART

ART performed well, in terms of accuracy, when it was compared to both manual tracking (0.60 ± 0.24 mm) and Ethovision (0.78 ± 0.68 mm) (Figure 5.11). These error values were particularly low, considering that the total viewing area was 40 x 40 cm. A critique of some past open-source trackers have been their lack of validation [167]; ART overcomes this by containing a validation option that allows users to calculate error from manual tracking. In addition, in ART, the tracking was overlaid on to the video, and users could validate it frame-by-frame. In other trackers (such as Ethovision), the tracking is previewed as a line, and cannot be explored on a frame-by-frame basis in tandem with the position of the mouse. Manual tracking was used here as the ground truth of the nose point coordinate; however, when the experimenter manually tracked videos again, the nose point coordinate varied by around 0.620.37 mm. This experimenter error was very low; however, automatic tracking further reduces errors of this type, and provides a more repeatable alternative [211].

As well as reducing manual error, automatic tracking should also reduce manual intervention. ART has much lower manual intervention scores per video analysed than Ethovision (6 clicks as opposed to 30), indicating an increased ease of setting up videos for analysis. In addition, ART can also batch-process videos, so large amounts of data can be processed at once further reducing demands on operator time. With the increases in the amount of video data collected, caused by recent technology developments [113, 171] and a growing emphasis on behavioural data collection for animal welfare [221], it has become even more important for video analysis to be conducted quickly and in large batches with minimal user intervention. Furthermore, behavioural data is relatively variable, especially in SOD1 mice [87, 24], and it is recommended to have large animal numbers for all preclinical research, according to updated guidelines [146]. This will continue to increase the amount of video data collected for behavioural studies, and the need for efficient analysis programs.

Automatic trackers vary in the way they process background and foreground images. ART uses a rule-based system that is quick to process. However, there are some instances where it does struggle to identify and track the mouse. One such instant is when the mouse rears, as the head changes shape significantly, and the nose point can become obscured (Figure 5.15a). Another example is if the mouse crosses a defined object (Figure 5.15b), for example if its head crosses over the Perspex block by climbing on top of it. ART primarily focusses on tracking the position and movement of the rodent head (such as its orientation and turning speed) and these measurements are not accurate if the mouse positions its head with severe pitch, roll and yaw; therefore, it is preferential for episodes of rearing and climbing to be omitted from any data processing of the head. If a user does need to make measurements when the rodent is rearing, an additional module could be added to future versions of ART to incorporate this.

Ethovision also had some episodes where it did not track. Ethovision calculates the background image by processing one video frame that does not contain the mouse. This background has to be constant for all the subsequent frames, for instance, the camera view cannot move, the Perspex block cannot be moved, and the mouse cannot defecate nor brush sawdust in to the shot. If any of these things happen, then the mouse tracking becomes inaccurate (Figure 5.15c). ART overcomes this by processing the background in each individual frame, rather than just using one set frame. Another example where Ethovision had trouble tracking the footage was when the mouse got close enough to the Perspex arena wall that there was a reflection of the mouse appearing on the wall, and the refection was sometimes tracked instead of the actual mouse (Figure 5.15d). This was not a problem for ART, as a region of interest could be specified for processing within the arena walls, which omits areas of reflection. Indeed, we demonstrate here, that ART can robustly track a variety of videos with a very similar accuracy to costly proprietary software, such as Ethovision, with minimal user intervention.

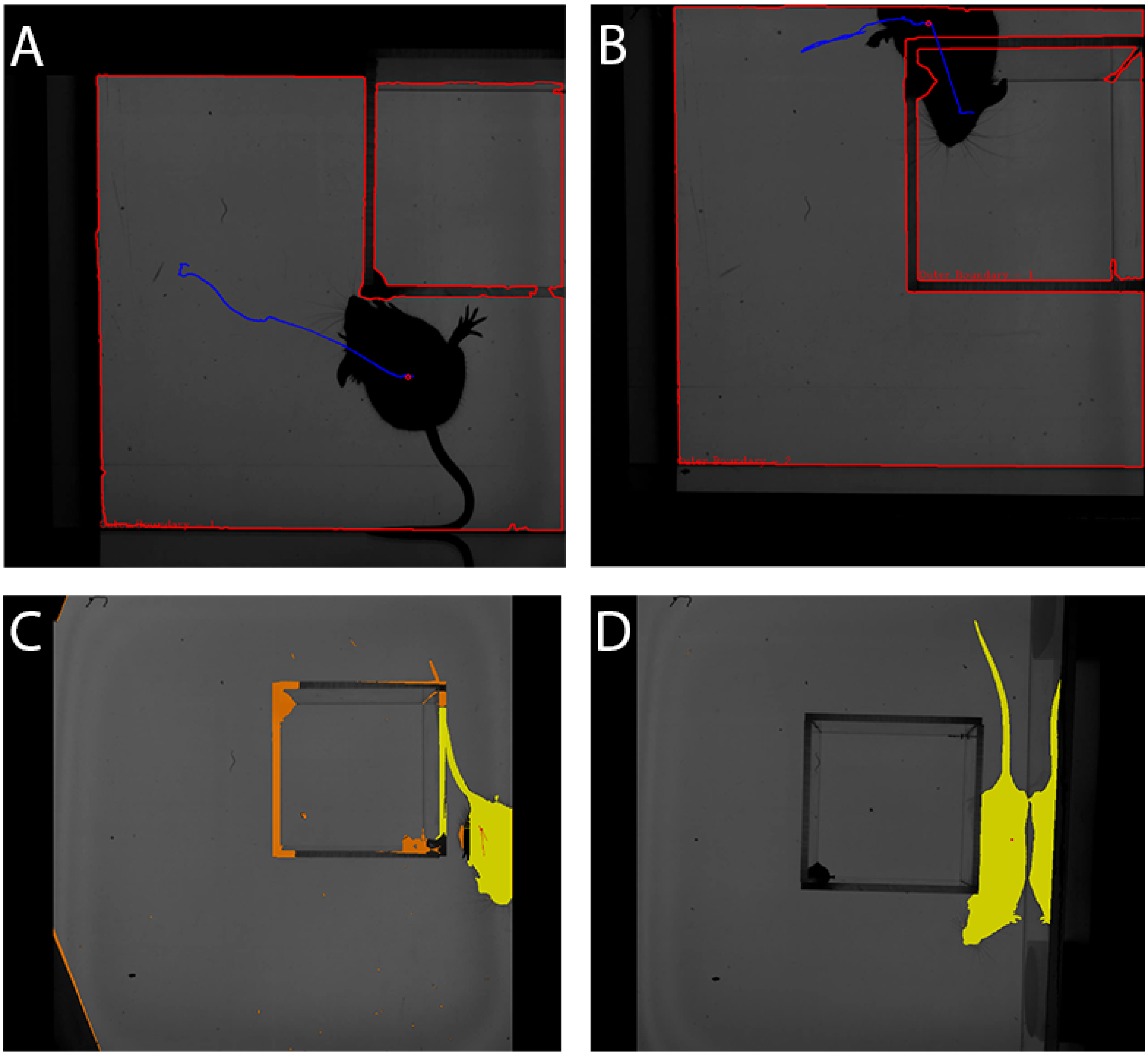


Figure 5.15: Failure modes for ART (a and b) and Ethovision (c and d). a) ART: Mouse rearing and deforming the head silhouette, preventing the nose tip from being detected. b) ART: Mouse climbing over a static object, background subtraction would then cause the mouse to appear as two separate entities. c) Ethovision: A video where the Perspex box was in a unique position and the program could not therefore find an accurate background image, therefore identified both mouse and object as mouse (in yellow). d) Ethovision: A mouse standing adjacent to its reflection, so both the mouse and its reflection was identified as the mouse (in yellow).

5.8.2 Demonstration of ART on experimental data

We show here that locomotion speeds are significantly reduced in SOD1 mice at P30 and P120 (Figure 5.12). Other studies have also found that locomotion speeds are reduced in SOD1 mice. Grant et al. 2014 [87] found that locomotion speeds were reduced in SOD1 mice from P30-P120, revealed by head-tracking of high-speed video footage, using an open source tracker, BWTT [185]. Only 4-8 videos were tracked per mouse in total in the previous study (compared to 42-69 here) as the BWTT required manual intervention to locate the nose and head points in one key frame [185]. Other behaviours, such as travelling and interacting could not be extracted from the BWTT tracker. Many studies have also examined locomotion from a gait analysis perspective by having mice walk on treadmills or catwalks [243, 149, 14]. However, data from these studies are highly dependent on the speed setting of the treadmill [14], and does not truly represent freely-moving rodent experiments. In addition, in order to measure other behaviours, such as rotating, interacting or staying still, requires an additional behavioural test, as these behaviours are unable to be carried out on a treadmill. There are not many studies that have looked at general behaviour changes in SOD1 mice [14], and the majority of studies have focussed on motor assessments, such as treadmill running, rotorod and grip strength [14]. Therefore, the finding here, that SOD1 mice reduce their interaction (object exploration behaviours) overall, and especially at is a new result (figure 5.13). We also show that SOD1 mice are less active at P120, by staying still more and moving forward less. This agrees with other studies that have found activity levels and locomotion to be significantly impacted in these mice [87, 243, 149].

We also found a decrease in SOD1 body weight from P90 and a decrease in pelvic muscle volume from P60 (Figure 5.14a and c). While these are not new findings, and can also be found in other studies [87, 155, 237], this is the first study to show that video analysis measurements of the centroid body width also correlated well with body weight (Figure 5.14b), and was significantly reduced from P30 (Figure 5.14d). This approximation of size, using the centroid width, is a novel way to non-invasively monitor rodent health, without directly having to handle the animal for weighing. Developing video analysis software, such as ART, allows many videos to be collected and processed from rodents behaving freely within a semi-natural environment. Not only this, but it characterises a number of quantifiable metrics of locomotion behaviour, general behaviours and rodent size measurements. This approach allows the collection of a variety of metrics, within the same behavioural set-up, thus decreasing the time of experimental data collection and the amount of animal handling.

5.8.3 The ART Approach

ART is a freely available, open source alternative to currently available rodent trackers, which performs at a very similar level of accuracy. It has been designed to be intuitive, easy to use, and require as little action as possible from the user, improving upon many of the trackers described in section 5.2. This is achieved by automating each step as much as possible. In addition, it uses the software design patterns and plug-in architecture [204], so that other behavioural trackers can be incorporated, such as gait analysis or whisker tracking. A batch processing component is also included, allowing for automated analysis over a large dataset which none of the stated trackers are capable of. Future work will be to expand ART to include more behavioural metrics and to test it on a wide range of mouse models with motor deficits.

Chapter 6

Development, demonstration and validation of an Automated Whisker Detection method

Based on publication (under review): Fully Automated Whisker Detector

6.1 Chapter summary

In this chapter, the whisker detection method is described. Here, we propose a new method that improves upon existing methods by being fully automated, while retaining the same accuracy levels of current state of the art methods that require manual intervention from the user. The proposed method is able to distinguish faint lines from a variety of brightness and contrast variations along the contour of the rodent. The method is validated against manually annotated images and other state of the art techniques, using a dataset obtained from Dr. Robyn Grants pre-existing video database for publications [84, 86, 85, 81].

6.2 Introduction

Behavioural assessment of animal movements using high speed videography produces large amounts of video data, which can often vary in terms of scale, resolution and image intensity due to a variety of experimental set-ups and equipment across different laboratories. In addition, when rodents move freely, their heads engage in episodes of pitch, yaw and roll and will also sometimes be obscured. While high speed videography allows whisker movements to be seen, detecting whiskers is challenging. Whiskers are very thin, have low contrast, can move with high angular velocity (>10,000 degrees per second) [37], and there are often several whiskers overlapping each other, and passing under one another. Videos can also be of low quality, introducing background noise. Detecting whiskers, therefore, provides an ideal candidate for developing image processing techniques, in particular validating line detectors. There are some freely available packages [186, 37], but these come with limitations, such as requiring a significant amount of settings that must be input manually, and using these same settings over the course of an entire video where brightness and contrast could change. We aim to overcome some of the limitations of currently available software and, therefore, present preliminary steps in a new automated approach for whisker detection, evaluating its accuracy against currently available software and manual annotations, and show that whiskers provide an ideal candidate for testing line detection techniques.

6.3 Methodology

6.3.1 Introduction

Prior to developing a whisker detector, the automated rodent head tracker (ART) (section 5) was developed and made freely available [104], providing information on the location of the nosetip, the orientation of the head and the contour of the body. For this work, it was necessary to determine the location of an animals whisker's. The problems encountered were: i) variations in illumination between videos and from one side of the arena to the other; ii) distinguishing the whisker from background noise that could influence the orientation angle; and iii) distinguishing individual whiskers when they were close together. The area immediately surrounding the contour also contained a large amount of noise, making it difficult to determine the exact location of the whisker root. Figure 6.1 illustrates the 3-step process for our proposed FAWD method: i) enhancing the whiskers and normalising for brightness variations within the Region Of Interest (ROI); ii) finding potential locations of the whiskers, and iii) detecting the whiskers. The output generated is the location and angle of all the whiskers around the head contour.

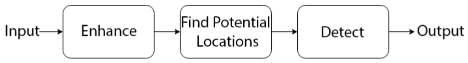


Figure 6.1: Flow diagram of detection process.

6.3.2 Validation Dataset

Brown Rats (*Rattus norvegicus*) (n=9 clips), Harvest Mice (*Micromys minutus*) (n=1 clips), House Mice (*Mus musculus*) (n=36 clips), and Gray Short-tailed Opossum (*Monodelphis domestica*) (n=4 clips) were recorded from a top-down view against a bright backlight, using a high-speed digital video camera recording at 500 Frames Per Second (FPS), with a shutter speed of 0.5ms. Camera types, settings, lighting and resolution varied over the 50 clips. The back-light and high speed camera were required as whiskers are thin, low contrast, and move very fast. To validate the performance of FAWD, 50 images were extracted from random videos, including a variety of all species, experimental set ups, and brightness variations. The images also showed the animal performing different behaviours such as locomotion and exploration of static objects placed within the set-up. Using the ART[102], the nosetip, orientation angle, and body contour of the animal was identified, as well as all static objects in the scene, as illustrated in figure 6.2b.

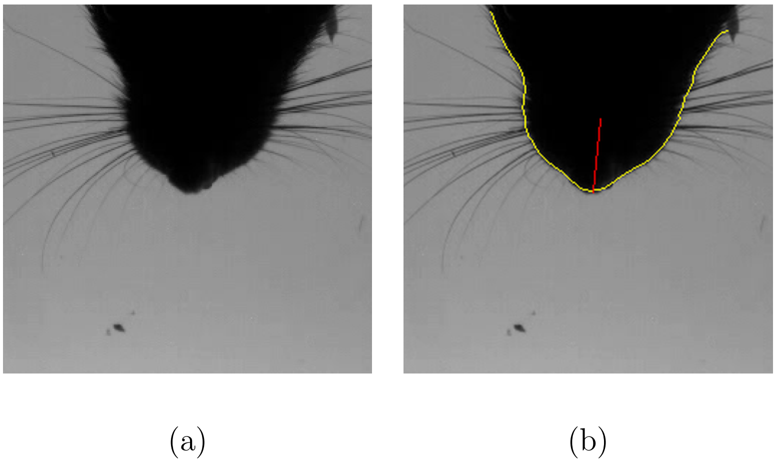


Figure 6.2: a) Example image from typical dataset b) Detected contours and orientation of the animal head using ART [102]

6.3.3 Enhancing The Whiskers And Normalising Brightness Levels

The first step in enhancing the whiskers was to increase the resolution by a factor of 2 using nearest neighbour interpolation[41] in order to achieve larger gaps between adjacent whiskers. As the whiskers had very low contrast, and could begin in a region with low brightness, extending into a brighter region, it was necessary to enhance them and normalise the brightness levels. This was achieved by dilating the image using a flat 3x3 structuring element (equation 6.1)[235] over 3 iterations, to effectively remove the whiskers while preserving the rest of the image.

$$(f \oplus b)(x) = \sup_{z \in B}[f(x-z)] \tag{6.1}$$

where $\oplus$ denotes dilation of an image $f(x)$ by a structuring element $b(x)$, sup is the supremum and z is the kernel.

Elementwise subtraction was then performed between inverted versions of the original image and the dilated image and normalised using a linear equation (equation 6.2); as shown in figure 6.3b.

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin \tag{6.2}$$

where I_N is the normalised image, I is the original image, Min and Max are the respective minimum and maximum intensity values of the original image, and $newMin$ and $newMax$ are the new minimum and maximum values (0 and 255 respectively for an 8 bit image).

The equalisation was kept linear as any other form of equalisation introduced too much noise into the image and created additional line like segments. A Frangi filter [70] was then applied to enhance the whiskers even further. Even though the Frangi filter was designed to detect vessels in medical images and facial wrinkles [169], it also works very well on the resulting whisker images (figure 6.3c). The enhancement process is shown in algorithm 1.

6.3.4 Identifying Potential Whisker Locations

Although the whiskers were enhanced, they were still shown as several pixels wide, with some noise still present. In order to highlight the whiskers further, potential whisker locations were identified. This was achieved by the use of a constructed mask around the rodent contour, that was extended by a set distance, of only 1 pixel in width for all values between the inner and outer extension distances (Figure 6.4). For each mask, a Gaussian filter was applied to remove noise and to create a “smoother” gradient profile of the pixels. Non-maxima suppression [52] was then applied to the

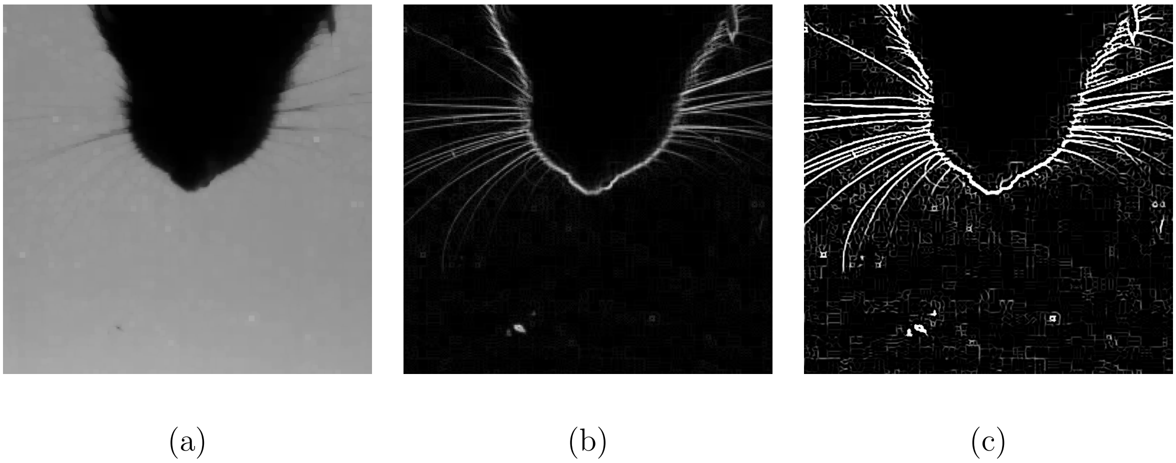


Figure 6.3: a) The dilated image b) Subtracted image c) Frangi image.

```
Data: Input image
begin
    Increase resolution by factor of 2 (HighResImg);
    Invert HighResImg (ImgNot);
    Dilate HighResImg (ImgDilate, Figure 6.3a);
    Invert ImgDilate (ImgDilateNot);
    Subtract ImgDilateNot from ImgNot (DilateEnhancedImg);
    Normalise DilateEnhancedImg (Figure 6.3b);
    Apply Frangi filter to DilateEnhancedImg (FrangiImage, Figure 6.3c);
end
```

Algorithm 1: Enhancement

entire kernel, rather than in a specified direction, to leave only the largest intensity pixel. The size of the kernel was a trade off between sensitivity and specificity. A larger kernel would result in fewer false positives, but will increase false negatives. This was especially true for whiskers that were close together, and would thus both appear in the same kernel (with the lower intensity one getting suppressed). After empirical testing kernel size was determined as 2% of the resolution. The resulting pixels could be considered probable locations for whisker positions. The process was repeated several times with the contour being extended by a further pixel each time, leaving an area several pixels wide with potential whisker locations as shown in Figure 6.5a. Blob detection [142] was used to locate groups of pixels of a certain area, allowing pixels generated from noise in the image to be filtered out. This vastly improved the performance of the algorithm, as it needed to check fewer pixels for potential whiskers. This combination of techniques were shown to significantly improve the number of whiskers detected, as shown in Appendix A. There were many different settings that could be applied to each of these techniques. In order to select the best settings, a systematic testing of each possible combination was conducted, and the combination which had the largest number of whiskers detected was selected, as shown in Appendix B.

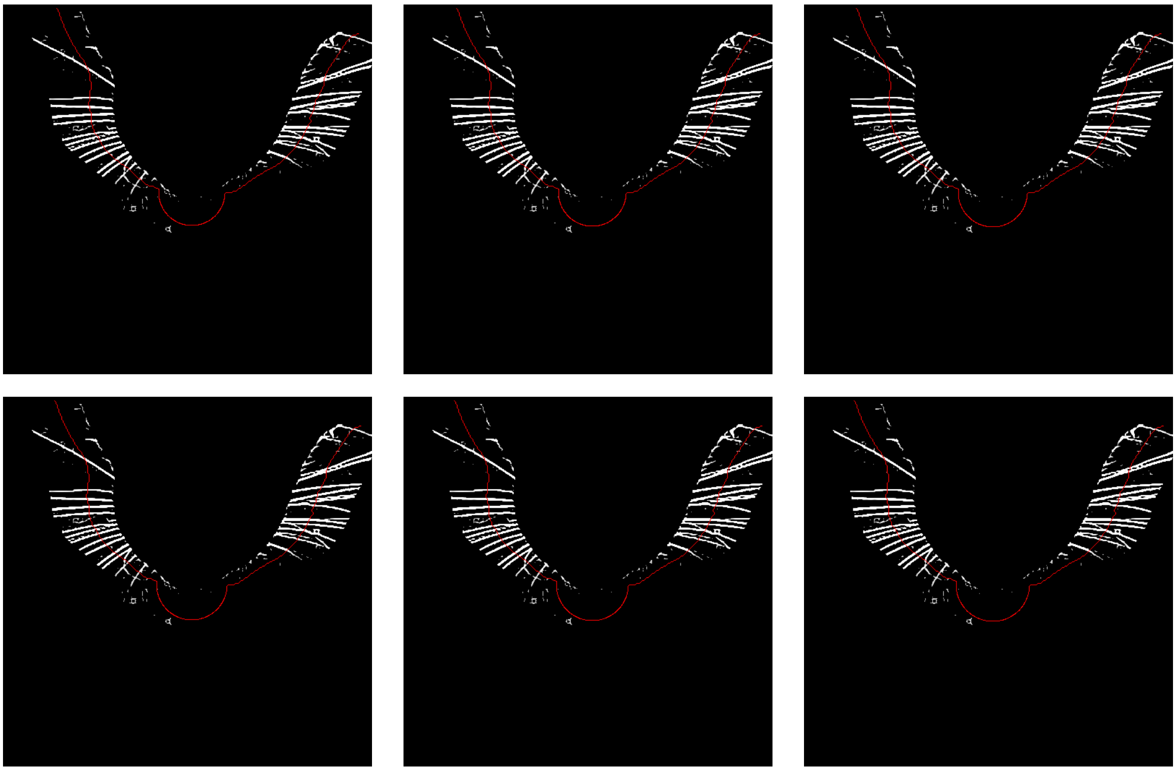
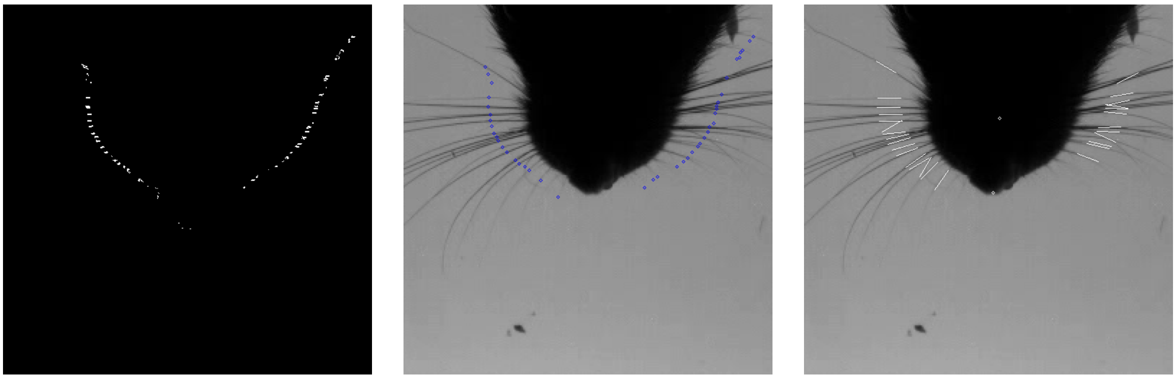


Figure 6.4: The 1 pixel mask lines

6.3.5 Whisker Detection

The centroid from each group of pixels in the non-maximum suppressed image was chosen, and a line mask was created for every angle (0 - 180°), the length of which was kept short so curved whiskers could still be approximated by a straight line. This mask was then used to calculate the average intensity of the Frangi image for that particular whisker. If the average intensity increased above a certain threshold (40% of the bit depth, 102 for 8-bit), then the orientation with the largest response was saved for that pixel. All potential whiskers are shown in Figure 6.5c. While the above process achieved a very high true positive rate, it also generated a significant amount of false positives. In order to reduce the false positives, rule based logic was applied



(a) (b) (c)

Figure 6.5: a) Probable whisker locations b) Centroid locations after blob detection c) The detected whiskers

to each detected whisker, such that if the whisker was found to be parallel to the orientation of the rodent's face, it was removed.

```
Data: FrangiImage, Extended Mask Lines
begin
    for Each extended mask line do
        Obtain mask image from FrangiImage (MaskImg);
        Apply Non-Max Suppression on a 5x5 kernel to MaskImg
          (SuppressedImg);
        Add SuppressedImg to temporary output image (TempImg,
          Figure 6.5a);
    end
    Use blob detection on TempImg;
    Filter blobs by a minimum area;
    for Each blob do
        if blob centroid intensity less than a minimum value then
            discard;
        else
            Add blob centroid to potential whisker list (PotentialWhiskers,
              Figure 6.5b);
        end
    end
end
```

Algorithm 2: Identifying Potential Whisker Locations

```
Data: PotentialWhiskers, FrangiImg
begin
    for Each blob centroid in PotentialWhiskers do
        for Each orientation do
            Obtain average intensity of frangi image for whisker at centroid
              point with orientation;
            if average intensity greater than a minimum value then
                if average intensity is greatest so far then
                    Store orientation at centroid location (Figure 6.5c);
                else
                    Discard;
                end
            else
                Discard;
            end
        end
    end
end
```

Algorithm 3: Whisker Detection

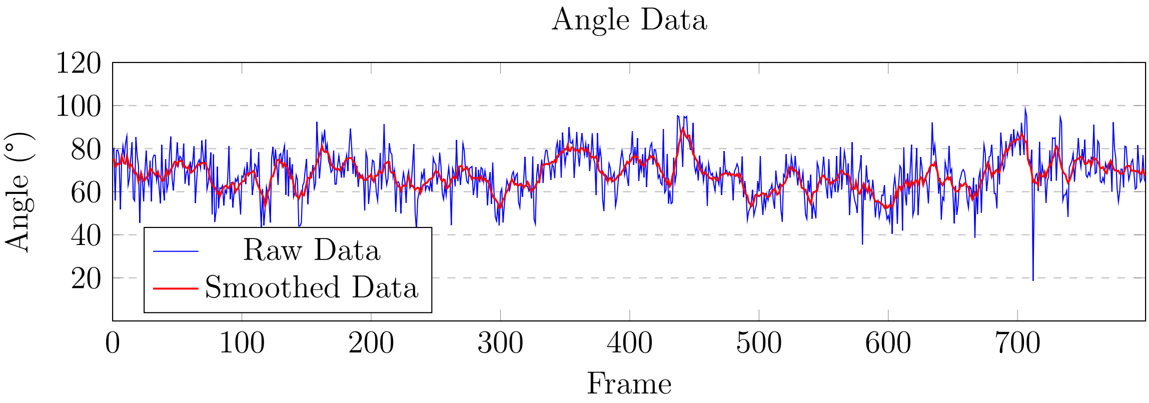


Figure 6.6: Angle data against time

6.4 Extracting Whisker Data

Given the position and angle of all detected whiskers per frame, the average angle was taken from all of these for both the left and right side of the head. This average angle is used to determine amplitude, velocity, mean offset, and frequency. Due to variances in the amount of the detected whiskers, the raw signal contained a lot of a noise, and a smoothing operator (moving average in this case) was applied to the signal to minimise this.

6.4.1 Validation against manual tracking

The ability of FAWD to detect whiskers was measured by counting how many whiskers were correctly detected ($TruePositive$ - TP) in 50 video-still images. True negative values were unable to be generated due to the nature of the images, as this would cover almost the entire ROI and would skew the accuracy results [185].

To determine how well FAWD correctly estimated a line/whisker segment, the Jaccard Similarity Index (JSI) was used to measure the reliability of the detection. The JSI was calculated by the intersection of A (detected segment) and B (annotated segment) divided by the union of A and B (equation 6.3) [170]. A subset of the 50-video still images (n=10) were manually annotated using the Manual Whisker Annotator [103] to obtain B. Experimental error was also calculated by tracking all the ten clips twice and calculating JSI.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}. \tag{6.3}$$

where J is the JSI, A and B is image A and image B respectively.

6.4.2 Comparison against existing trackers

To compare FAWD to the existing trackers, Whisk and BWTT, the TP rate from 50 images was identified, by counting the number of true whiskers detected for each tracker. The same subset (n=10) of images (as above) was then used to measure how accurately each tracker found the whiskers by calculating the JSI compared to the manually annotated images.

6.5 Results

6.5.1 Validation against manual tracking

The average TP whisker detection rate for FAWD against 50 images was 15.84 ± 5.9 (n=50), with a JSI of 0.06 ± 0.01 (n=10). The experimenter error across all images had a JSI of 0.7 ± 0.08 (n=10). The comparatively low JSI of FAWD's detection, compared to manual tracking, was mainly due to the manually annotated images being a trace of the entire whisker shaft, whereas FAWD only detected a portion of the whisker near the base.

6.5.2 Comparison against existing trackers

The average TP whisker detection rate for Whisk was 6.8 ± 6.97 (n=50), and had a JSI of 0.07 ± 0.06 (n=10). FAWD found significantly more whiskers than Whisk (Wilcoxon Signed Rank: Z=-6.0927, p<0.01).

The average TP whisker detection rate for BWTT was 14.76 ± 4.7 (n=50), and had a JSI of 0.17 ± 0.02 (n=10). Although the TP detection rate for FAWD was slightly higher, there was no significant difference between the trackers (Wilcoxon Signed Rank: Z=-1.0554, p=0.2891).

6.5.3 Comparison of whisker detection

Although FAWD did not significantly improve upon BWTT over the full dataset, it performed significantly better on videos with high spatial resolution of the whiskers, (Wilcoxon Signed Rank: Z=-2.6672, p<0.001, n=13), from here known as the $Z1$ dataset, where as there was no significant difference on videos with lower spatial resolution (Wilcoxon Signed Rank: Z=-0.9381, p=0.3472, n=37), from here known as the $Z2$ dataset (Table 6.1).

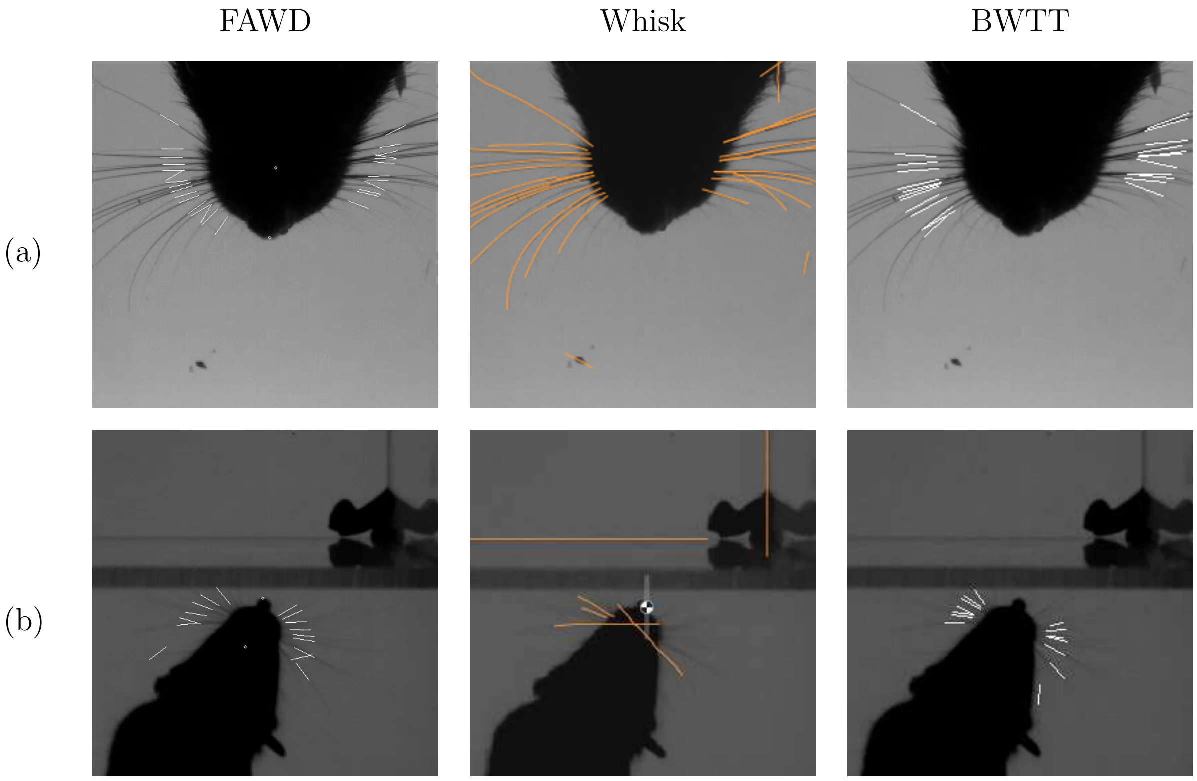


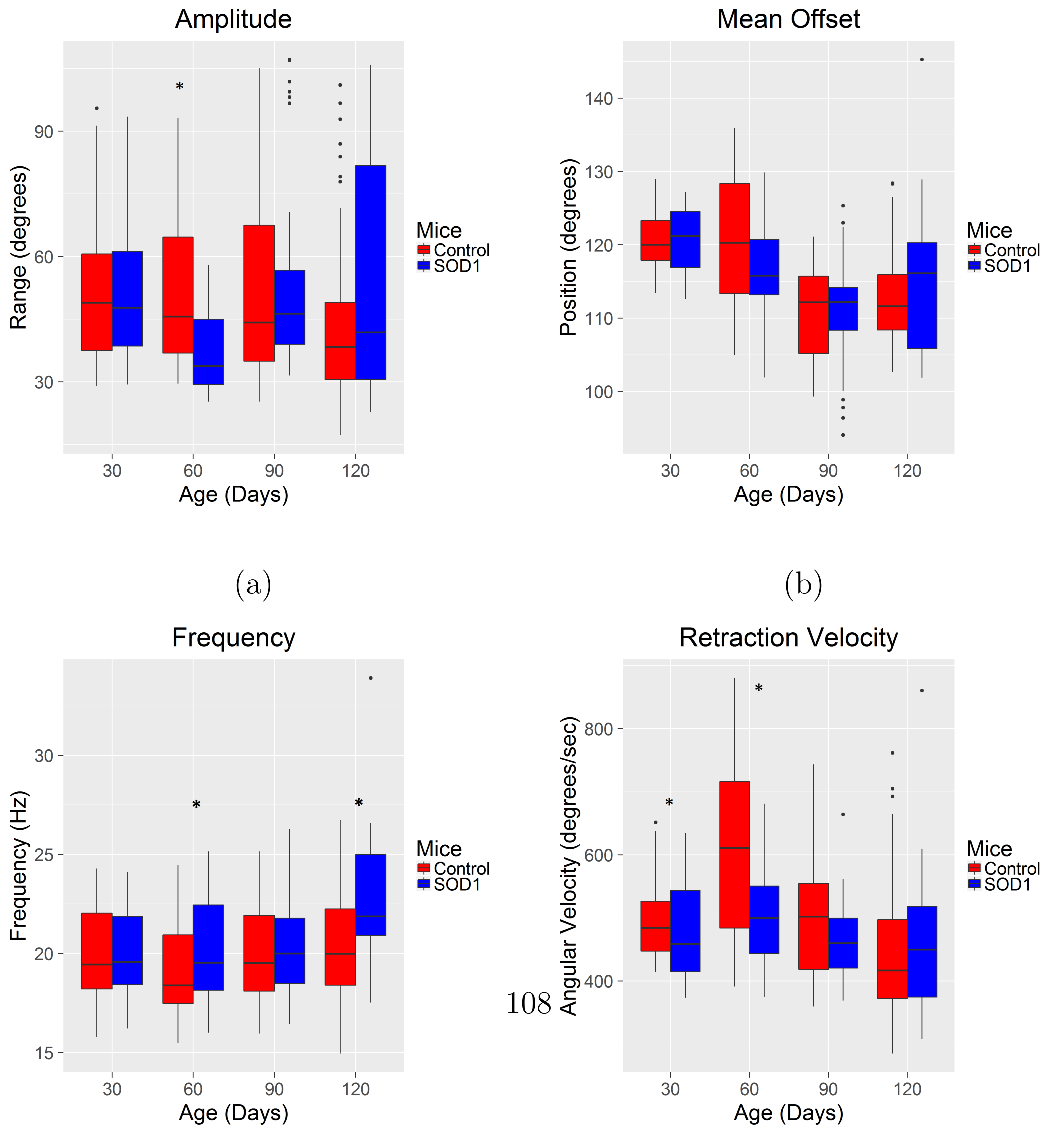
Figure 6.7: Detection rates at two different zoom levels; a) $Z1$ and b) $Z2$.

6.5.4 Demonstration of FAWD on an experimental dataset of control and SOD1 mice

To demonstate FAWD the same video dataset was used as ART (chapter 5). In particular, control mice (non-transgenic) were tested against SOD1^{G93A} mice. Results were only taken from when mice were moving forward at a speed greater than or equal to 25mm/sec. Amplitude was found to only be significantly reduced in transgenic mice at 60 days of age (U=362, p<0.01, Z=3.98) (Figure 6.8a). There were no significant differences for the mean offset of the whiskers at any age (Figure 6.8b). Frequency was found to be significantly increased in transgenic mice at 60 (U=556.5, p=0.0394, Z=-2.06) and 120 (U=336, p<0.01, Z=-3.26) days (Figure 6.8c). Retraction velocity was found to be significantly reduced in transgenic mice at 30 (U=3480, p=0.0394, Z=2.06) and 60 (U=461, p<0.01, Z=3.01) days of age (Figure 6.8d).

Method	Accuracy (JSI)	Mean No. of detections			Automation	Dynamic Settings
		All	$Z1$	$Z2$		
	(n=10)	(n=50)	(n=13)	(n=37)		
FAWD	0.06	15.84	22.62	13.5	Fully automated	All settings calculated per frame
Whisk	0.07	6.8	16.69	3.32	Fully automated	All settings calculated per frame
BWTT	0.17	14.76	17.23	13.89	Some settings are manually input at the beginning	Settings calculated on keyframe and propagated throughout the clip

Table 6.1: Detection and accuracy results for FAWD, BWTT and Whisk



(a)

(b)

	Offset	Retraction Velocity	Frequency	Amplitude
Ntg mean	112.31 ± 6.04	439.97 ± 105.42	20.34 ± 2.84	42.5 ± 18.63
Sod1 mean	116.25 ± 11.47	467.34 ± 136.12	23.35 ± 3.69	52.03 ± 27.33

Table 6.2: Results for mice at p120

6.6 Discussion and Conclusions

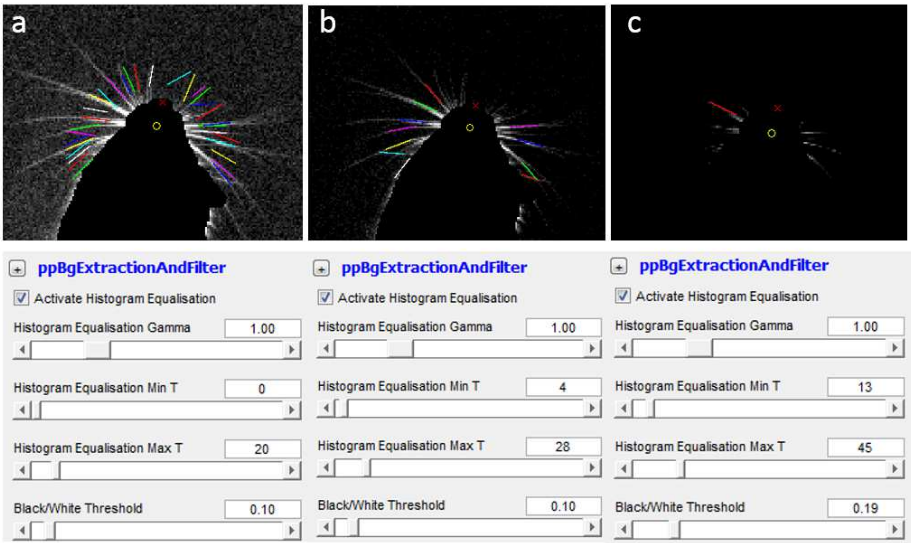
We suggest here that whiskers are a useful tool for testing line detection techniques; and present the Fully Automated Whisker Detector (FAWD). FAWD improves on existing trackers by essentially combining the best elements. FAWD has similar accuracy levels compared with state of the art trackers such as BWTT, while requiring less user input as it is fully automated, such as Whisk. A further improvement FAWD has on existing trackers, is its ability to work across a wide range of brightness and contrast variations. FAWD was still able to detect whiskers where elements of the background were darker than the actual whiskers.

The challenging and varying nature of whiskers are a good test for line detection algorithms. Whereas many algorithms require well defined edges to work [30], these would fail to detect a faint line amongst background noise and other artefacts. In order to accurately detect whiskers, a more complex line detector should be used, such as in FAWD.

6.6.1 Program Comparison

Even though BWTT performed comparably against FAWD on the detection rate for all whiskers, the TP rate was significantly higher for the high resolution ($Z1$) dataset, when compared to BWTT. This highlights FAWD's potential to automatically detect whiskers which have high spatial resolution. Also, all the settings are dynamically computed for each individual frame in FAWD. Multiple settings that must be manually adjusted into BWTT can have a major impact on detection rates (Figure 6.9). These settings are then used across an entire video, meaning if the rodent were to travel from an area of high brightness into an area of low brightness which was common for these datasets, the program may begin to fail to detect all the whiskers, whereas FAWD would continue to operate at the same detection rate regardless of brightness variation.

FAWD has been developed as an alternative to existing whisker detectors. It has been designed to reduce as much manual intervention as possible, giving the user the same ease of use of programs such as Whisk, while maintaining accuracy levels similar to that of other methods, such as BWTT. This was achieved by dynamically calculating each required setting per frame, and can thus be used on datasets consisting of various brightness and contrast changes. As FAWD processes standard video files, no specialist equipment is needed other than a high speed camera, and no tracking dye need to be applied to the whiskers, providing an advantage over other detection techniques such as those described by Bermejo et al. (1998) and Knutsen et al. (2008). However, limitations of the software are its inability to measure curvature, whisker cross-overs, and that it can only detect multiple whiskers, rather than track. Other trackers such as WhiskerMan [28] address all of these limitations, but requires rodents to be head fixed, whereas FAWD does not. This is only a preliminary step in developing FAWD as a comprehensive whisker tracker program. Whisker tracking algorithms will also have to be developed to identify the same whisker over a number of video frames. FAWD represents the first step in developing a robust, fully-automatic whisker tracker to be used on freely behaving small mammals over a range of experimental conditions.



a

ppBgExtractionAndFilter

☑ Activate Histogram Equalisation

Histogram Equalisation Gamma 1.00

Histogram Equalisation Min T 0

Histogram Equalisation Max T 20

Black/White Threshold 0.10

b

ppBgExtractionAndFilter

☑ Activate Histogram Equalisation

Histogram Equalisation Gamma 1.00

Histogram Equalisation Min T 4

Histogram Equalisation Max T 28

Black/White Threshold 0.10

c

ppBgExtractionAndFilter

☑ Activate Histogram Equalisation

Histogram Equalisation Gamma 1.00

Histogram Equalisation Min T 13

Histogram Equalisation Max T 45

Black/White Threshold 0.19

Figure 6.9: Variation in detection rates for different settings in BWTT

6.6.2 Whisking Results

Amplitude and retraction velocity differences in whisking between control and transgenic mice became apparent from an early age, with differences in retraction velocity visible from just 30 days old. However, with no significant differences at later ages suggests as the mice age there whisking characteristics become more aligned. The exception to this is frequency, with transgenic mice whisking more frequently than their control counterparts at 120 days of age (Figure 6.8. Our results were taken from clips where mice were displaying forward movement as it has been shown that whisking kinematics are altered when moving at speed. Specifically that rats adopt a look ahead strategy when running at fast locomotion speeds [9]. As such, the amount of results generated for control mice was much higher than transgenic mice, as locomotion velocity is significantly affected in transgenic mice. While there was still enough data for the results to still be normally distributed, there was vastly more data for control mice, especially at 120 days of age.

These results are in partial contrast to previous work carried out on SOD1^{G93A} mice [86] which found all whisker variables were significantly affected by 120 days of age. Specifically that transgenic mice held their whiskers further back (lower mean offset) and moved at a lower frequency. While amplitude and retraction velocity results were not significantly different at 120 days of age, larger means for transgenic mice do correlate with previous work. This may be explained by the lower sample size (n=34) compared to our dataset (n=97) and that all clips were analysed regardless of forward velocity, where as our results were taken from mice when only moving forward. This also helps explain the increase in offset values compared to Grant et al (2014) of approximately 20 degrees, 89.9-99.35 compared to 112.3-116.3, as mice hold their whiskers further forward when moving at speed [9], the increase in frequency, 12.18-17.97 compared to 20.3-23.4, and the increase in retraction velocity, 260-320 compared to 440-467.3 as they whisker faster [9]. However, amplitude results were also higher (42-52 compared to 35-42) which was is in contrast to expected results.

6.7 Conclusion

This work has allowed for a large dataset (n=877), more than any other study, to be automatically analysed with no human intervention. We have also shown our method for whisker detection is comparable to current state of the art, semi automated trackers. Out of the 877 clips, 470 were found to display forward movement by the mouse and were included in the final results. We found significant differences in whisking kinematics between control and SOD1^{G93A} mice at varying ages, with the most notable difference being an increase in frequency at 120 days of age for transgenic mice. As our study only analysed mice moving forward, the discrepancies between the results of previous studies can be accounted for by the differences in whisking kinematics during locomotion movement.

6.8 Chapter Summary

This chapter has outlined a new method for fully automated whisker detection using a 3 step process. Most notably, using dilation to control for videos with varying brightness and the use of a Frangi filter for whisker enhancement. We have also shown the results obtained from the method, and how they compare with previous studies as well as the variations between them.

Chapter 7

Discussion

7.1 Chapter summary

In this final chapter, a summary of the contributions of this thesis on rodent and whisker detection are discussed. A critical analysis of the work completed is done with a focus on the strengths and limitations found during the research. It also highlights potential future improvements to this field and the direction in which it is heading for researchers in a continuously growing area.

7.2 Summary of findings

A summary of the research objectives in shown in table 7.2 along with the corresponding outcomes. These findings will detail the reason for each objective and how the outcome was achieved.

The first objective was to create a manual annotator software package. This was achieved by creating the Manual Whisker Annotator (MWA, section 4.2), which was subsequently used to create a ground truth dataset for validation. MWA turned out to be incredibly successful, with 90 downloads at the time of writing, and being used in other research for annotation purposes other than tracking whiskers on rodents [80, 10].

The second objective was to develop and validate a rodent head detection algorithm. A robust and novel method was successfully created, by using basic rules to define the shape of the head, and a separate algorithm for precise location of the nosetip, it is able to quickly and accurately find the rodent, its nosetip, and head orientation. It also works for a variety of species with different shaped heads. This method was validated against a manually annotated dataset, and against a widely used automated tracker, Ethovision [172]. As the detection algorithm required knowledge of static objects in the scene, an automated background detection was developed. This detection method did not require the animal to be absent in the videos, providing an advantage over other trackers which did so [172]. This knowledge of static objects also allowed interaction behaviours to be obtained, by recording how long the nosetip would be touching an object. This method was incorporated into a software package known as the Automated Rodent Tracker (ART) and at the time of writing, has been downloaded 310 times.

The third objective was to develop and validate a whisker detection method. This was achieved by the creation of a Fully Automated Whisker Detector (FAWD). With prior knowledge of the nosetip location and head orientation, whiskers were able to be detected in the surrounding area to a high accuracy and in a fully automated manner. This method was validated against the ground truth dataset created with MWA. This method was developed into the Automated Rodent and Whisker Tracker (ARWT). An extension of ART, this package included the whisker detection module. It allowed for batch processing, with parameters assigned to each video such as age, transgenic or non transgenic, and mouse ID. Once evaluated, the dataset can be filtered by locomotion velocity or rotational velocity.

Finally, the fourth objective was to demonstrate the software package outlined in objectives 2 and 3 to characterise locomotion and whisking behaviour in the SOD1 dataset. This was achieved by utilising the software, mainly the batch processing functionality, on every video in the dataset. It was conducted in a fully automated fashion, with no user input required after the initial parameters of the mice had been input (age, tg or ntg, ID). Using the obtained locomotion, orientation, and whisker positions, all metrics mentioned in section 4.5 were obtained and analysed.

No.	Objective	Outcome
1	Create a manual annotator software package and produce a dataset with ground truth annotation.	An intuitive, and easily deployable manual annotator was created, capable of annotating nosetip location, whiskers, and any other points wishing to be tracked. This software was subsequently used to produce a ground truth dataset.
2	Develop and validate a robust rodent head detection and tracking algorithms using shape features.	A novel, robust head detection method was created, capable of automatically tracking a rodents orientation and nosetip location, as well as identifying background objects.
3	Develop and validate an automated rodent whisker detector using image processing algorithms.	Using a combination of existing techniques, a fully automated, accurate whisker detector was created, capable of detecting whiskers across a range of videos with variable brightness and contrast.
4	Demonstrate the final prototypes including the quantification of the mouse behaviour and statistical analysis.	The Automated Rodent and Whisker Tracker (ARWT) was demonstrated on the SOD1 dataset.

Table 7.1: The research objectives (defined in section 1.4) against the actual outcomes.

7.3 Limitations of work

7.3.1 Head and Body Tracking

During development of the proposed detection algorithms, certain limitations became apparent. As mentioned in section 5.8.1, there are some failure modes for which the head detection algorithm can fail, such as when the rodent rears its head. Another instance where detection can fail was if the rodent moves over a static object, as these objects are subtracted from the scene, any portion of the rodent over the object would also be removed, leading to an irregular shaped rodent.

7.3.2 Whisker Detection

While the accuracy of FAWD was validated, there were certain conditions which would give undesirable results. When the whiskers were compressed against an object and had little to no spacing between them, they would be treated as one whisker. The static object removal algorithm could also remove portions of the whisker that were in direct contact with the object. This could lead to a small amount, or none at all, of the whiskers been detected, and as an average of all whiskers was taken, this would introduce unintended noise into the obtained whisking signal.

7.3.3 Datasets

Although the total dataset was quite large (consisting of 877 videos), there was a notable reduction in data for transgenic mice at P120 moving at a speed greater than 25mm/sec, which was unsurprising given the reduction in locomotion at this age for transgenic mice. While there was still enough data for statistical analysis to be performed, a larger dataset would have been more desirable.

7.4 Applications for understanding Amyotrophic Lateral Sclerosis

Using ARWT, and outlined in section 5.7, locomotion was impacted in SOD1 mice, with a significantly reduced average speed at both P30 and P120. SOD1 mice also had a tendency to rotate more, but was not a significant result. General behaviours were found to be affected, with SOD1 mice spending less time interacting with the static objects within the arena, less time moving forwards, and more time stood still. Body size and weight were also significantly reduced in SOD1 mice. While ARWT was not used to determine these, it was used to measure the centroid width which correlated with body weight.

Unlike the above results which were obtained from across the entire dataset, whisking results were only taken from clips where the mouse was moving at a speed greater than to or equal 25mm/sec due to the effect locomotion has on whisking. For this sub sample of the dataset, amplitude was found to be significantly lower in SOD1 mice at P60. Retraction velocity was also found to be significantly lower in SOD1 mice at P30 and P60. A significant increase in frequency was found for SOD1 mice at P60 and P120, and there were no significant differences in mean offset for all ages.

For this study, SOD1 mice, representing a model of Amyotrophic Lateral Sclerosis (ALS), were used. However, many mouse models for many other neurodegenerative diseases exist, including, but not limited to, Parkinsons, Alzheimers and Huntingtons. Whisker movements have been found to be strongly impacted in mouse models of Huntingtons Disease [74], and whisker motor deficits were revealed earlier than any other behavioural test, including gait assessments. While the whisker behavioural model is still in an early phase of development, we think that further exploring the relationship between disease progression and behaviour is a useful and important step in quantifying clinical aspects of neurodegenerative disorders, for better characterisation and treatment. The next step for this project is to test a number of other models, including Parkinsons Disease, Alzheimers Disease, Cerebellar Ataxia and Cortical Developmental Disorders, and will be presented at the Measuring Behavior Conference in June 2018.

7.5 Recommendations for the future

7.5.1 Individual Whisker Tracking

Although FAWD was capable of detecting multiple whiskers, tracking individual whiskers across a video is extremely challenging. As the whiskers move very fast, and are extremely close together, being able to distinguish individual whiskers between frames is a very complex task. Not FAWD or any other whisker detector had a perfect accuracy rate, and when whiskers weren't found, this becomes a major problem for tracking multiple objects within a close proximity, and would likely require the development of new techniques to accomplish.

We present here that whisker analyses could be used as a new behavioural model of ALS. While whisker movements are fairly robust and resistant to disease progression, in comparison to weight changes and locomotion deficits (Figure 5.14), the changes in amplitude and retraction speeds were still clear and significant (Figure 6.8). These changes are thought to be associated with neurodegeneration of the Facial Nucleus and wastage of white muscle fibers in the mystacial retraction muscles [86]. Further exploring the links between between neurodegeneration and resultant whisker movements would progress our understanding of facial muscle wastage and control in Amyotrophic Lateral Sclerosis.

7.5.2 Utilisation of ARWT to measure whisker movements: a new behavioural model

Quantitatively assessing behaviour to measure animal health is challenging because there is a lack of unobtrusive behavioural models to refine animal testing procedures. Our novel software solution simultaneously measures locomotion behaviours and whisker movements, providing robust and quantitative measures of rodent behaviour for basic research or drug discovery. Several other behavioural models exist, such as the Morris Water Maze, Rotorod and grip strength task, but these require extensive animal training (which can be very stressful for the animals), and usually only measure the duration and frequency of certain behaviours, being only semi-quantitative. Most laboratories will expose their animals to a host of these behavioural tests in order to capture a range of behaviours, requiring training for each individual test and repeated testing of each individual animal. Locomotion testing can be more measureable, but still requires extensive animal training (i.e. Noldus catwalk). Filming whisker movements do not require any animal training, and takes only around 10 minutes per session, reducing stress and time away from the home cage. Filming whiskers can also be conducted in the dark, under infrared lighting, thus reducing stress and encouraging more active and natural behaviours.

The software developed in this thesis utilises videos from high-speed video cameras and standard arenas, which are present in many behavioural testing facilities. It allows wholly automatic extraction of whisker movements from the collected video footage, using advanced image processing techniques. Movement measures include estimates of angle, frequency, spread, amplitude, speed, curvature and asymmetry, which are highly quantitative and sensitive to change under a range of health conditions.

Uptake of the ARWT by academic researchers and industry will significantly improve animal welfare by alleviating the stress and suffering that arises from long experiments away from the home cage under bright lights. So far, we suggest that adopting the ARWT approach and software will significantly improve upon traditional behavioural methods by:

1. Making sure all animals complete the task (none lost due to not being able to learn).

2. Reducing stress in animals by having short experimental times and enabling experiments to be conducted in the dark.

3. Generating wholly quantitative data using specialist image-processing software that can measure very precise aspects of motor control from short movement bouts.
4. Reducing severity of disease symptoms by diagnosing motor deficits earlier in disease progression.
5. Reducing the number of animals in each condition, as significant volumes of quantitative data can be collected from each animal. Indeed, significant results can be seen in this thesis between groups containing only five animals, rather than the large numbers that are usually recommended for behavioural testing.

7.6 Conclusions

The work presented in this thesis forms contributions in the tracking of a rodent, its nosetip, and orientation, as well as whisker detection, where a focus is placed on automation and the processing of large datasets. By outperforming current methods, either in terms of accuracy or automation, the contributions show promising results for the future of this field, where it can only get better with new advances in image processing and computer vision. Detecting whiskers and tracking movements was incredibly challenging, especially as even humans can struggle to see them under certain conditions. However, as seen in the literature and work presented, analysing locomotion and whisker movements is a promising way to monitor disease progression within rodents.

Appendix A

Univariate Analysis of Variance for the selection of whisker detection techniques

The techniques that significantly improved the number of whiskers detected included Super Sampling (S.Sample), Dilation, followed by Elementwise Subtraction (D.S.), Frangi Filter (Frangi) and the distance away from the head contour (Distance). Binary Thresholding was not found to significantly improve the number of whiskers detected.

Source	**Type III Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Corrected Model	98058.38	609.00	161.02	60.66	0.00
Intercept	31684.63	1.00	31684.63	11937.17	0.00
S.Sample	142.76	4.00	35.69	13.45	0.00
D.S	26.06	3.00	8.69	3.27	0.02
Frangi	79635.56	4.00	19908.89	7500.67	0.00
Distance	395.95	1.00	395.95	149.18	0.00
Threshold	0.93	1.00	0.93	0.35	0.55

Appendix B

Systematic testing of each possible combination of settings

The mean number of whiskers detected was counted for all possible combinations of super sampling methods, dilation level ('d' - dilation level -'s'), the Frangi settings (Sigma Start - Sigma End - Sigma Step - Beta One - Beta Two), the distance away from the body contour in pixels, and the thresholding value, although this was not incorporated as it was shown to not significantly improve the number of whiskers detected, as shown in Appendix A.

Super Sample Method	**Dilate level**	**Frangi Settings**	**Distance settings**	**Thresh. settings**	**Mean**
BiCubic	d3s	0.25-0.75-0.1-0.91-51	16-19	60	3.11
BiCubic	d3s	0.25-0.75-0.1-0.91-51	16-19	80	3.11
BiCubic	d3s	0.25-0.75-0.1-0.91-51	7-10	60	3.33
BiCubic	d3s	0.25-0.75-0.1-0.91-51	7-10	80	3.33
BiCubic	d3s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.44
BiCubic	d3s	1.37-1.371-0.5-2.85-3.67	7-10	60	10.78
BiCubic	d3s	1.37-1.371-0.5-2.85-3.67	7-10	80	10.89
BiCubic	d3s	1-1.01-0.5-0.91-51	16-19	60	0.00
BiCubic	d3s	1-1.01-0.5-0.91-51	7-10	60	0.00
BiCubic	d3s	1-1.01-0.5-0.91-51	7-10	80	0.00
BiCubic	d3s	1-2.75-0.1-0.6-79	16-19	60	0.00

BiCubic	d3s	1-2.75-0.1-0.6-79	7-10	60	0.00
BiCubic	d3s	1-2.75-0.1-0.6-79	7-10	80	0.00
BiCubic	d3s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
BiCubic	d3s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
BiCubic	d5s	0.25-0.75-0.1-0.91-51	16-19	60	3.89
BiCubic	d5s	0.25-0.75-0.1-0.91-51	16-19	80	4.22
BiCubic	d5s	0.25-0.75-0.1-0.91-51	7-10	60	5.56
BiCubic	d5s	0.25-0.75-0.1-0.91-51	7-10	80	6.00
BiCubic	d5s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.44
BiCubic	d5s	1.37-1.371-0.5-2.85-3.67	7-10	60	11.56
BiCubic	d5s	1.37-1.371-0.5-2.85-3.67	7-10	80	11.00
BiCubic	d5s	1-1.01-0.5-0.91-51	16-19	60	0.00
BiCubic	d5s	1-1.01-0.5-0.91-51	7-10	60	0.00
BiCubic	d5s	1-1.01-0.5-0.91-51	7-10	80	0.00
BiCubic	d5s	1-2.75-0.1-0.6-79	16-19	60	0.00
BiCubic	d5s	1-2.75-0.1-0.6-79	7-10	60	0.00
BiCubic	d5s	1-2.75-0.1-0.6-79	7-10	80	0.00
BiCubic	d5s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
BiCubic	d5s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Cubic	d3s	0.25-0.75-0.1-0.91-51	16-19	60	5.33
Cubic	d3s	0.25-0.75-0.1-0.91-51	16-19	80	5.22
Cubic	d3s	0.25-0.75-0.1-0.91-51	7-10	60	5.56
Cubic	d3s	0.25-0.75-0.1-0.91-51	7-10	80	5.89
Cubic	d3s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.67
Cubic	d3s	1.37-1.371-0.5-2.85-3.67	7-10	60	11.00
Cubic	d3s	1.37-1.371-0.5-2.85-3.67	7-10	80	10.89
Cubic	d3s	1-1.01-0.5-0.91-51	16-19	60	0.00

Cubic	d3s	1-1.01-0.5-0.91-51	7-10	60	0.00
Cubic	d3s	1-1.01-0.5-0.91-51	7-10	80	0.00
Cubic	d3s	1-2.75-0.1-0.6-79	16-19	60	0.00
Cubic	d3s	1-2.75-0.1-0.6-79	7-10	60	0.00
Cubic	d3s	1-2.75-0.1-0.6-79	7-10	80	0.00
Cubic	d3s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Cubic	d3s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Cubic	d5s	0.25-0.75-0.1-0.91-51	16-19	60	5.00
Cubic	d5s	0.25-0.75-0.1-0.91-51	16-19	80	5.67
Cubic	d5s	0.25-0.75-0.1-0.91-51	7-10	60	6.22
Cubic	d5s	0.25-0.75-0.1-0.91-51	7-10	80	6.89
Cubic	d5s	1.37-1.371-0.5-2.85-3.67	16-19	60	9.56
Cubic	d5s	1.37-1.371-0.5-2.85-3.67	7-10	60	11.33
Cubic	d5s	1.37-1.371-0.5-2.85-3.67	7-10	80	11.56
Cubic	d5s	1-1.01-0.5-0.91-51	16-19	60	0.00
Cubic	d5s	1-1.01-0.5-0.91-51	7-10	60	0.00
Cubic	d5s	1-1.01-0.5-0.91-51	7-10	80	0.00
Cubic	d5s	1-2.75-0.1-0.6-79	16-19	60	0.00
Cubic	d5s	1-2.75-0.1-0.6-79	7-10	60	0.00
Cubic	d5s	1-2.75-0.1-0.6-79	7-10	80	0.00
Cubic	d5s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Cubic	d5s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Lanczos4	d3s	0.25-0.75-0.1-0.91-51	16-19	60	5.00
Lanczos4	d3s	0.25-0.75-0.1-0.91-51	16-19	80	5.00
Lanczos4	d3s	0.25-0.75-0.1-0.91-51	7-10	60	5.89
Lanczos4	d3s	0.25-0.75-0.1-0.91-51	7-10	80	6.44
Lanczos4	d3s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.33

Lanczos4	d3s	1.37-1.371-0.5-2.85-3.67	7-10	60	11.56
Lanczos4	d3s	1.37-1.371-0.5-2.85-3.67	7-10	80	11.56
Lanczos4	d3s	1-1.01-0.5-0.91-51	16-19	60	0.00
Lanczos4	d3s	1-1.01-0.5-0.91-51	7-10	60	0.00
Lanczos4	d3s	1-1.01-0.5-0.91-51	7-10	80	0.00
Lanczos4	d3s	1-2.75-0.1-0.6-79	16-19	60	0.00
Lanczos4	d3s	1-2.75-0.1-0.6-79	7-10	60	0.00
Lanczos4	d3s	1-2.75-0.1-0.6-79	7-10	80	0.00
Lanczos4	d3s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Lanczos4	d3s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Lanczos4	d5s	0.25-0.75-0.1-0.91-51	16-19	60	5.11
Lanczos4	d5s	0.25-0.75-0.1-0.91-51	16-19	80	6.11
Lanczos4	d5s	0.25-0.75-0.1-0.91-51	7-10	60	6.78
Lanczos4	d5s	0.25-0.75-0.1-0.91-51	7-10	80	6.78
Lanczos4	d5s	1.37-1.371-0.5-2.85-3.67	16-19	60	9.44
Lanczos4	d5s	1.37-1.371-0.5-2.85-3.67	7-10	60	11.56
Lanczos4	d5s	1.37-1.371-0.5-2.85-3.67	7-10	80	11.56
Lanczos4	d5s	1-1.01-0.5-0.91-51	16-19	60	0.00
Lanczos4	d5s	1-1.01-0.5-0.91-51	7-10	60	0.00
Lanczos4	d5s	1-1.01-0.5-0.91-51	7-10	80	0.00
Lanczos4	d5s	1-2.75-0.1-0.6-79	16-19	60	0.00
Lanczos4	d5s	1-2.75-0.1-0.6-79	7-10	60	0.00
Lanczos4	d5s	1-2.75-0.1-0.6-79	7-10	80	0.00
Lanczos4	d5s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Lanczos4	d5s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Linear	d3s	0.25-0.75-0.1-0.91-51	16-19	60	3.78
Linear	d3s	0.25-0.75-0.1-0.91-51	16-19	80	4.00

Linear	d3s	0.25-0.75-0.1-0.91-51	7-10	60	4.78
Linear	d3s	0.25-0.75-0.1-0.91-51	7-10	80	4.89
Linear	d3s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.00
Linear	d3s	1.37-1.371-0.5-2.85-3.67	7-10	60	10.44
Linear	d3s	1.37-1.371-0.5-2.85-3.67	7-10	80	9.89
Linear	d3s	1-1.01-0.5-0.91-51	16-19	60	0.00
Linear	d3s	1-1.01-0.5-0.91-51	7-10	60	0.00
Linear	d3s	1-1.01-0.5-0.91-51	7-10	80	0.00
Linear	d3s	1-2.75-0.1-0.6-79	16-19	60	0.00
Linear	d3s	1-2.75-0.1-0.6-79	7-10	60	0.00
Linear	d3s	1-2.75-0.1-0.6-79	7-10	80	0.00
Linear	d3s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Linear	d3s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Linear	d5s	0.25-0.75-0.1-0.91-51	16-19	60	3.89
Linear	d5s	0.25-0.75-0.1-0.91-51	16-19	80	4.00
Linear	d5s	0.25-0.75-0.1-0.91-51	7-10	60	4.33
Linear	d5s	0.25-0.75-0.1-0.91-51	7-10	80	4.67
Linear	d5s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.11
Linear	d5s	1.37-1.371-0.5-2.85-3.67	7-10	60	10.11
Linear	d5s	1.37-1.371-0.5-2.85-3.67	7-10	80	10.22
Linear	d5s	1-1.01-0.5-0.91-51	16-19	60	0.00
Linear	d5s	1-1.01-0.5-0.91-51	7-10	60	0.00
Linear	d5s	1-1.01-0.5-0.91-51	7-10	80	0.00
Linear	d5s	1-2.75-0.1-0.6-79	16-19	60	0.00
Linear	d5s	1-2.75-0.1-0.6-79	7-10	60	0.00
Linear	d5s	1-2.75-0.1-0.6-79	7-10	80	0.00
Linear	d5s	3.25-3.26-0.5-0.91-51	7-10	60	0.00

Linear	d5s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Nearest Neighbour	d3s	0.25-0.75-0.1-0.91-51	16-19	60	3.11
Nearest Neighbour	d3s	0.25-0.75-0.1-0.91-51	16-19	80	3.33
Nearest Neighbour	d3s	0.25-0.75-0.1-0.91-51	7-10	60	4.56
Nearest Neighbour	d3s	0.25-0.75-0.1-0.91-51	7-10	80	4.89
Nearest Neighbour	d3s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.44
Nearest Neighbour	d3s	1.37-1.371-0.5-2.85-3.67	7-10	60	10.78
Nearest Neighbour	d3s	1.37-1.371-0.5-2.85-3.67	7-10	80	10.89
Nearest Neighbour	d3s	1-1.01-0.5-0.91-51	16-19	60	0.00
Nearest Neighbour	d3s	1-1.01-0.5-0.91-51	7-10	60	0.00
Nearest Neighbour	d3s	1-1.01-0.5-0.91-51	7-10	80	0.00
Nearest Neighbour	d3s	1-2.75-0.1-0.6-79	16-19	60	0.00
Nearest Neighbour	d3s	1-2.75-0.1-0.6-79	7-10	60	0.00
Nearest Neighbour	d3s	1-2.75-0.1-0.6-79	7-10	80	0.00
Nearest Neighbour	d3s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Nearest Neighbour	d3s	3.25-3.26-0.5-0.91-51	7-10	80	0.00
Nearest Neighbour	d5s	0.25-0.75-0.1-0.91-51	16-19	60	3.89
Nearest Neighbour	d5s	0.25-0.75-0.1-0.91-51	16-19	80	4.22

Nearest Neighbour	d5s	0.25-0.75-0.1-0.91-51	7-10	60	5.56
Nearest Neighbour	d5s	0.25-0.75-0.1-0.91-51	7-10	80	6.00
Nearest Neighbour	d5s	1.37-1.371-0.5-2.85-3.67	16-19	60	8.44
Nearest Neighbour	d5s	1.37-1.371-0.5-2.85-3.67	7-10	60	11.56
Nearest Neighbour	d5s	1.37-1.371-0.5-2.85-3.67	7-10	80	11.00
Nearest Neighbour	d5s	1-1.01-0.5-0.91-51	16-19	60	0.00
Nearest Neighbour	d5s	1-1.01-0.5-0.91-51	7-10	60	0.00
Nearest Neighbour	d5s	1-1.01-0.5-0.91-51	7-10	80	0.00
Nearest Neighbour	d5s	1-2.75-0.1-0.6-79	16-19	60	0.00
Nearest Neighbour	d5s	1-2.75-0.1-0.6-79	7-10	60	0.00
Nearest Neighbour	d5s	1-2.75-0.1-0.6-79	7-10	80	0.00
Nearest Neighbour	d5s	3.25-3.26-0.5-0.91-51	7-10	60	0.00
Nearest Neighbour	d5s	3.25-3.26-0.5-0.91-51	7-10	80	0.00

References

[1] The features of c++ as a language, 2000.

[2] Abraham Acevedo-Arozena, Bernadett Kalmar, Shafa Essa, Thomas Ricketts, Peter Joyce, Rosie Kent, Claire Rowe, Andy Parker, Anna Gray, Majid Hafezparast, et al. A comprehensive assessment of the sod1g93a low-copy transgenic mouse, which models human amyotrophic lateral sclerosis. *Disease models & mechanisms*, 4(5):686–700, 2011.

[3] Paulo Aguiar, Luís Mendonça, and Vasco Galhardo. Opencontrol: a free opensource software for video tracking and automated control of behavioral mazes. *Journal of neuroscience methods*, 166(1):66–72, 2007.

[4] E. Ahissar and P.M. Knutsen. Object localization with whiskers. *Biological Cybernetics*, 98(6):449–458, 2008.

[5] Alwynelle S Ahl. Evidence of use of vibrissae in swimming in sigmo don fulviventer. *Animal Behaviour*, 30(4):1203–1206, 1982.

[6] Ezak Ahmad, Moi Hoon Yap, Hans Degens, and Jamie S McPhee. Atlas-registration based image segmentation of mri human thigh muscles in 3d space. In *SPIE Medical Imaging*, volume 9037, pages 90371L–90371L. International Society for Optics and Photonics, 2014.

[7] Tyler Andrews. Computation time comparison between matlab and c++ using launch windows. 2012.

[8] Ehsan Arabzadeh, Erik Zorzin, and Mathew E Diamond. Neuronal encoding of texture in the whisker sensory pathway. *PLoS biology*, 3(1):e17, 2005.

[9] Kendra Arkley, Robyn A Grant, Ben Mitchinson, and Tony J Prescott. Strategy change in vibrissal active sensing during rat locomotion. *Current Biology*, 24(13):1507–1512, 2014.

[10] Kendra Arkley, Guuske P Tiktak, Vicki Breakell, Tony J Prescott, and Robyn A Grant. Whisker touch guides canopy exploration in a nocturnal, arboreal rodent, the hazel dormouse (muscardinus avellanarius). *Journal of Comparative Physiology A*, 203(2):133–142, 2017.

[11] Michael R Bale, Dario Campagner, Andrew Erskine, and Rasmus S Petersen. Microsecond-scale timing precision in rodent trigeminal primary afferents. *Journal of Neuroscience*, 35(15):5935–5940, 2015.

[12] Carrolee Barlow, Shinji Hirotsune, Richard Paylor, Marek Liyanage, Michael Eckhaus, Francis Collins, Yosef Shiloh, Jacqueline N Crawley, Thomas Ried, Danilo Tagle, and Anthony Wynshaw-Boris. Atm-deficient mice: a paradigm of ataxia telangiectasia. *Cell*, 86:159–171, 1996.

[13] Carol A Barnes. Memory deficits associated with senescence: a neurophysiological and behavioral study in the rat. *Journal of comparative and physiological psychology*, 93(1):74, 1979.

[14] Richard J Batka, Todd J Brown, Kathryn P Mcmillan, Rena M Meadows, Kathryn J Jones, and Melissa M Haulcomb. The need for speed in rodent locomotion analyses. *The Anatomical Record*, 297(10):1839–1864, 2014.

[15] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[16] M Flint Beal, Anthony E Lang, and Albert C Ludolph. *Neurodegenerative Diseases: Neurobiology, Pathogenesis and Therapeutics*. Cambridge University Press, 2005.

[17] Ellen J. Bennett, Richard J. Mead, Mimoun Azzouz, Pamela J. Shaw, and Andrew J. Grierson. Early detection of motor dysfunction in the sod1g93a mouse model of amyotrophic lateral sclerosis (als) using home cage running wheels. *PLOS One*, 9(9), 2014.

[18] R Bermejo, D Houben, and H Philip Zeigler. Optoelectronic monitoring of individual whisker movements in rats. *Journal of neuroscience methods*, 83(2):89–96, 1998.

[19] Roberto Bermejo, Akshat Vyas, and H Philip Zeigler. Topography of rodent whisking–i. two-dimensional monitoring of whisker movements. *Somatosensory & motor research*, 19(4):341–346, 2002.

[20] J Martin Bland and DouglasG Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The lancet*, 327(8476):307–310, 1986.

[21] Hugh Carlton Blodgett. The effect of the introduction of reward upon the maze performance of rats. *University of California publications in psychology*, 1929.

[22] R Braga, I Kouzmine, Newton S Canteras, and C Da Cunha. Lesion of the substantia nigra, pars compacta impairs delayed alternation in a y-maze in rats. *Experimental neurology*, 192(1):134–141, 2005.

[23] Aeron Buchanan and Andrew Fitzgibbon. Interactive feature tracking using kd trees and dynamic programming. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 626–633. IEEE, 2006.

[24] Selina Bucher, Kerstin E Braunstein, Heiko G Niessen, Thomas Kaulisch, Michael Neumaier, Tobias M Boeckers, Detlef Stiller, and Albert C Ludolph. Vacuolization correlates with spin–spin relaxation time in motor brainstem nuclei and behavioural tests in the transgenic g93a-sod1 mouse model of als. *European Journal of Neuroscience*, 26(7):1895–1901, 2007.

[25] Frank Buschmann, Kelvin Henney, and Douglas Schimdt. *Pattern-oriented Software Architecture: on patterns and pattern language*, volume 5. John wiley & sons, 2007.

[26] Stephen Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[27] Richard Camicioli, M Milar Moore, Anthony Kinney, Elizabeth Corbridge, Kathryn Glassberg, and Jeffrey A Kaye. Parkinson's disease is associated with hippocampal atrophy. *Movement Disorders*, 18(7):784–790, 2003.

[28] Dario Campagner, Mathew H Evans, Michaela SE Loft, and Rasmus S Petersen. What the whiskers tell the brain. *Neuroscience*, 2017.

[29] Dario Campagner, Mathew Hywel Evans, Michael Ross Bale, Andrew Erskine, and Rasmus Strange Petersen. Prediction of primary somatosensory neuron activity during active tactile exploration. *Elife*, 5, 2016.

[30] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[31] Rebecca J Carter, Lisa A Lione, Trevor Humby, Laura Mangiarini, Amarbirpal Mahal, Gillian P Bates, Stephen B Dunnett, and A Jennifer Morton. Characterization of progressive motor deficits in mice transgenic for the human huntingtons disease mutation. *Journal of Neuroscience*, 19(8):3248–3257, 1999.

[32] George E Carvell, Daniel J Simons, Seth H Lichtenstein, and Patrick Bryant. Electromyographic activity of mystacial pad musculature during whisking behavior in the rat. *Somatosensory & motor research*, 8(2):159–164, 1991.

[33] George E Carvell and DJ Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *The Journal of Neuroscience*, 10(8):2638–2648, 1990.

[34] Ping-Lin Chang and Wei-Guang Teng. Exploiting the self-organizing map for medical image segmentation. In *Computer-Based Medical Systems, 2007. CBMS'07. Twentieth IEEE International Symposium on*, pages 281–288. IEEE, 2007.

[35] Paul F Chapman, Gail L White, Matthew W Jones, Deirdre Cooper-Blacketer, Vanessa J Marshall, Michael Irizarry, Linda Younkin, Mark A Good, TVP Bliss, Bradley T Hyman, et al. Impaired synaptic plasticity and learning in aged amyloid precursor protein transgenic mice. *Nature neuroscience*, 2(3), 1999.

[36] Wenje Chen, Ping Zhai, Robert L Sufit, and Teepu Siddique. Motor neuron degeneration in mice that express a human cu, zn superoxide dismutase mutation. *Science*, 264:17, 1994.

[37] N.G. Clack, D.H. O'Connor, D. Huber, L. Petreanu, A. Hires, S. Peron, K. Svoboda, and E.W. Myers. Automated tracking of whiskers in videos of head fixed rodents. *PLOS Computational Biology*, 8(7), 2012.

[38] KA Clarke and J Still. Gait analysis in the mouse. *Physiology & behavior*, 66(5):723–729, 1999.

[39] Cheryl D Conrad, Liisa AM Galea, Yasukazu Kuroda, and Bruce S McEwen. Chronic stress impairs rat spatial memory on the y maze, and this effect is blocked by tianeptine treatment. *Behavioral neuroscience*, 110(6):1321, 1996.

[40] Ruth Cortez and Alexander Vazhenin. Developing re-usable components based on the virtual-mvc design pattern. In *Databases in Networked Information Systems*, pages 132–149. Springer, 2013.

[41] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[42] Jacqueline N Crawley. Behavioral phenotyping of transgenic and knockout mice: experimental design and evaluation of general health, sensory functions, motor abilities, and specific behavioral tests. *Brain research*, 835(1):18–26, 1999.

[43] WE Crusio and JH Van Abeelen. The genetic architecture of behavioural responses to novelty in mice. *Heredity*, 56(Pt 1):55–63, 1986.

[44] Wim E Crusio. Genetic dissection of mouse exploratory behaviour. *Behavioural brain research*, 125(1):127–132, 2001.

[45] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, and Mario Cifrek. A brief introduction to opencv. In *MIPRO, 2012 Proceedings of the 35th International Convention*, pages 1725–1730. IEEE, 2012.

[46] Peter Curzon, Min Zhang, Richard J Radek, and Gerard B Fox. The behavioral assessment of sensorimotor processes in the mouse: acoustic startle, sensory gating, locomotor activity, rotarod, and beam walking. 2009.

[47] E Roy Davies. *Machine vision: theory, algorithms, practicalities.* Elsevier, 2004.

[48] Centre de recherche de l'Institut universitaire en sant mentale de Qubec. Structure and innervation of a whisker follicle, 1993.

[49] Robert M. J. Deacon and J. Nicholas P. Rawlins. T-maze alternation in the rodent. *Nature Protocols*, 1:7–12, 2006.

[50] Kelly Del Tredici, Udo Rüb, Rob Ai De Vos, Jürgen RE Bohl, and Heiko Braak. Where does parkinson disease pathology begin in the brain? *Journal of Neuropathology & Experimental Neurology*, 61(5):413–426, 2002.

[51] F Dellu, W Mayo, J Cherkaoui, M Le Moal, and H Simon. A two-trial memory task with automated recording: study in young and aged rats. *Brain research*, 588(1):132–139, 1992.

[52] Frédéric Devernay. *A non-maxima suppression method for edge detection with sub-pixel accuracy*. PhD thesis, INRIA, 1995.

[53] Rudi DHooge and Peter P De Deyn. Applications of the morris water maze in the study of learning and memory. *Brain research reviews*, 36(1):60–90, 2001.

[54] Mathew E Diamond, Moritz Von Heimendahl, Per Magne Knutsen, David Kleinfeld, and Ehud Ahissar. 'where'and'what'in the whisker sensorimotor system. *Nature Reviews Neuroscience*, 9(8):601, 2008.

[55] Xiao-xia Dong, Yan Wang, and Zheng-hong Qin. Molecular mechanisms of excitotoxicity and their relevance to pathogenesis of neurodegenerative diseases. *Acta pharmacologica Sinica*, 30(4):379–387, 2009.

[56] Janine Doorduin, Erik FJ de Vries, Rudi A Dierckx, and Hans C Klein. Pet imaging of the peripheral benzodiazepine receptor: monitoring disease progression and therapy response in neurodegenerative disorders. *Current pharmaceutical design*, 14(31):3297–3315, 2008.

[57] Josef Dorfl. The musculature of the mystacial vibrissae of the white mouse. *Journal of anatomy*, 135(Pt 1):147, 1982.

[58] Brian P Dranka, Alison Gifford, Donna McAllister, Jacek Zielonka, Joy Joseph, Crystal L OHara, Cheryl L Stucky, Anumantha G Kanthasamy, and Balaraman Kalyanaraman. A novel mitochondrially-targeted apocynin derivative prevents hyposmia and loss of motor function in the leucine-rich repeat kinase 2 (lrrk2 r1441g) transgenic mouse model of parkinson's disease. *Neuroscience letters*, 583:159–164, 2014.

[59] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[60] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[61] Bradley Efron. The convex hull of a random set of points. *Biometrika*, 52(3-4):331–343, 1965.

[62] Accelerating Video Feature Extractions. Multi-core software. 2007.

[63] Gerald Farin. Algorithms for rational bézier curves. *Computer-aided design*, 15(2):73–77, 1983.

[64] Dennis M Feeney, Abel Gonzalez, and Wendy A Law. Amphetamine, haloperidol, and experience interact to affect rate of recovery after motor cortex injury. *Science*, 217(4562):855–857, 1982.

[65] H-L Feng, Y Leng, C-H Ma, J Zhang, M Ren, and D-M Chuang. Combined lithium and valproate treatment delays disease onset, reduces neurological deficits and prolongs survival in an amyotrophic lateral sclerosis mouse model. *Neuroscience*, 155(3):567–572, 2008.

[66] Mohammed Filali, Robert Lalonde, and Serge Rivest. Sensorimotor and cognitive functions in a sod1 g37r transgenic mouse model of amyotrophic lateral sclerosis. *Behavioural brain research*, 225(1):215–221, 2011.

[67] Sanford I Finkel, Jorge Costa e Silva, Gene Cohen, Sheldon Miller, and Norman Sartorius. Behavioral and psychological signs and symptoms of dementia: a consensus statement on current knowledge and implications for research and treatment. *International Psychogeriatrics*, 8(S3):497–500, 1997.

[68] Tatiana Foroud, Jacqueline Gray, Julia Ivashina, and P Michael Conneally. Differences in duration of huntingtons disease based on age at onset. *Journal of Neurology, Neurosurgery & Psychiatry*, 66(1):52–56, 1999.

[69] Martin Fowler. *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[70] Alejandro Frangi, Wiro Niessen, Koen Vincken, and Max Viergever. Multiscale vessel enhancement filtering. *Medical Image Computing and Computer-Assisted InterventationMICCAI98*, pages 130–137, 1998.

[71] Adam Freeman. Windows presentation foundation. In *Introducing Visual C# 2010*, pages 1069–1098. Springer, 2010.

[72] Gilbert Gallardo, Oliver M. Schlter, and Thomas C. Sdhof. A molecular pathway of neurodegeneration linking alpha-synuclein to apoe and abeta peptides. *Nature Neuroscience*, 11:301–308, 2008.

[73] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software.* Pearson Education, 1994.

[74] Huw Garland, Nigel I Wood, Elizabeth A Skillings, Peter J Detloff, A Jennifer Morton, and Robyn A Grant. Characterisation of progressive motor deficits in whisker movements in r6/2, q175 and hdh knock-in mouse models of huntingtons disease. *Journal of Neuroscience Methods*, 2017.

[75] Raffaele Garofalo. *Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern.* Microsoft Press, 2011.

[76] Dariu M Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 439–444. IEEE, 1998.

[77] Luca Gherardi, Davide Brugali, and Daniele Comotti. A java vs. c++ performance evaluation: a 3d modeling benchmark. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 161–172. Springer, 2012.

[78] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB.* Pearson Education India, 2004.

[79] James Gosling. *The Java language specification.* Addison-Wesley Professional, 2000.

[80] Manu Goyal, Neil D Reeves, Satyan Rajbhandari, Jennifer Spragg, and Moi Hoon Yap. Fully convolutional networks for diabetic foot ulcer segmentation. *arXiv preprint arXiv:1708.01928*, 2017.

[81] Robyn A Grant, Vicki Breakell, and Tony J Prescott. Whisker touch sensing guides locomotion in small, quadrupedal mammals. *Proc. R. Soc. B*, 285(1880):20180592, 2018.

[82] Robyn A Grant, Nele Cielen, Karen Maes, Nele Heulens, Gina LJ Galli, Wim Janssens, Ghislaine Gayan-Ramirez, and Hans Degens. The effects of smoking on whisker movements: A quantitative measure of exploratory behaviour in rodents. *Behavioural processes*, 128:17–23, 2016.

[83] Robyn A Grant, Nele Cielen, Karen Maes, Nele Heulens, Gina LJ Galli, Wim Janssens, Ghislaine Gayan-Ramirez, and Hans Degens. The effects of smoking

on whisker movements: A quantitative measure of exploratory behaviour in rodents. *Behavioural processes*, 128:17–23, 2016.

[84] Robyn A Grant, Sebastian Haidarliu, Natalie J Kennerley, and Tony J Prescott. The evolution of active vibrissal sensing in mammals: evidence from vibrissal musculature and function in the marsupial opossum monodelphis domestica. *Journal of Experimental Biology*, 216(18):3483–3494, 2013.

[85] Robyn A Grant, Ben Mitchinson, Charles W Fox, and Tony J Prescott. Active touch sensing in the rat: anticipatory and regulatory control of whisker movements during surface exploration. *Journal of neurophysiology*, 101(2):862–874, 2009.

[86] Robyn A. Grant, Paul S. Sharp, Aneurin J. Kennerley, Jason Berwick, Andrew Grierson, Tennore Ramesh, and Tony J. Prescott. Abnormalities in whisking behaviour are associated with lesions in brain stem nuclei in a mouse model of amyotrophic lateral sclerosis. *Behavioural Brain Research*, 259:274–283, 2013.

[87] Robyn A Grant, Paul S Sharp, Aneurin J Kennerley, Jason Berwick, Andrew Grierson, Tennore Ramesh, and Tony J Prescott. Abnormalities in whisking behaviour are associated with lesions in brain stem nuclei in a mouse model of amyotrophic lateral sclerosis. *Behavioural brain research*, 259:274–283, 2014.

[88] Robyn A Grant, Anna L Sperber, and Tony J Prescott. The role of orienting in vibrissal touch sensing. *Frontiers in behavioral neuroscience*, 6, 2012.

[89] James F Gusella, Nancy S Wexler, P Michael Conneally, Susan L Naylor, Mary Anne Anderson, Rudolph E Tanzi, Paul C Watkins, Kathleen Ottina, Margaret R Wallace, Alan Y Sakaguchi, et al. A polymorphic dna marker genetically linked to huntington's disease. *Nature*, 306(5940):234–238, 1983.

[90] John W Gustafson and Sherry L Felbain-Keramidas. Behavioral and neural approaches to the function of the mystacial vibrissae. *Psychological bulletin*, 84(3):477, 1977.

[91] A Gustavsson, M Svensson, F Jacobi, C Allgulander, J Alonso, and E Beghi. Cost of disorders of the brain in europe. *Eur Neuropsychopharmacol*, 21(10):718–719, 2011.

[92] SM Hague, S Klaffke, and O Bandmann. Neurodegenerative disorders: Parkinsons disease and huntingtons disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(8):1058–1063, 2005.

[93] Stefan L Hahn. *Hilbert transforms in signal processing*, volume 2. Artech House Boston, 1996.

[94] C. Hall and E. L. Ballachey. A study of the rat's behavior in a field. a contribution to method in comparative psychology. *University of California Publications in Psychology*, 6:1–12, 1932.

[95] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[96] Mitra Hartmann. Vibrissa mechanical properties. In *Scholarpedia of Touch*, pages 591–614. Springer, 2016.

[97] Kouhei Hashizume, Manabu Hirasawa, Yutaka Imamura, Setsuko Noda, Takahiko Shimizu, Kei Shinoda, Toshihide Kurihara, Kousuke Noda, Yoko Ozawa, Susumu Ishida, et al. Retinal dysfunction and progressive retinal cell death in sod1-deficient mice. *The American journal of pathology*, 172(5):1325–1331, 2008.

[98] Simon Haykin and Neural Network. A comprehensive foundation. *Neural networks*, 2(2004):41, 2004.

[99] Anders Hejlsberg, Scott Wiltamuth, and Peter Golde. *C# language specification*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[100] Dominique Helmlinger, Gaël Yvert, Serge Picaud, Karine Merienne, José Sahel, Jean-Louis Mandel, and Didier Devys. Progressive retinal degeneration and dysfunction in r6 huntington's disease mice. *Human molecular genetics*, 11(26):3351–3359, 2002.

[101] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[102] Brett Hewitt, Moi Hoon Yap, and Robyn Grant. *ART Automated Rodent Tracker*. Measuring Behavior 2016 (Dublin, Ireland, May 2016). 2016.

[103] Brett Hewitt, Moi Hoon Yap, and Robyn Grant. Manual whisker annotator (mwa): A modular open-source tool. *Journal of Open Research Software*, 4(1), 2016.

[104] Brett M Hewitt, Moi Hoon Yap, Emma F Hodson-Tole, Aneurin J Kennerley, Paul S Sharp, and Robyn A Grant. A novel automated rodent tracker (art), demonstrated in a mouse model of amyotrophic lateral sclerosis. *Journal of Neuroscience Methods*, 2017.

[105] J Graeme Hodgson, Nadia Agopyan, Claire-Anne Gutekunst, Blair R Leavitt, Fred LePiane, Roshni Singaraja, Desmond J Smith, Nagat Bissada, Krista McCutcheon, Jamal Nasir, et al. A yac mouse model for huntingtons disease with full-length mutant huntingtin, cytoplasmic toxicity, and selective striatal neurodegeneration. *Neuron*, 23(1):181–192, 1999.

[106] Kennether Hoffman and Ray Kunze. Linear algebra, 2nd. *New J ersey: Prentice*, 1971.

[107] Leigh Holcomb, Marcia N Gordon, Eileen McGowan, Xin Yu, Stan Benkovic, Paul Jantzen, Kristal Wright, Irene Saad, Ryan Mueller, Dave Morgan, et al. Accelerated alzheimer-type phenotype in transgenic mice carrying both mutant amyloid precursor protein and presenilin 1 transgenes. *Nature medicine*, 4(1):97–100, 1998.

[108] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Trans. Acoust., Speech, Signal Processing*, 27:13–18, 1979.

[109] T Huang, GJTGY Yang, and G Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979.

[110] Robert N Hughes. The value of spontaneous alternation behavior (sab) as a test of retention in pharmacological investigations of memory. *Neuroscience & Biobehavioral Reviews*, 28(5):497–505, 2004.

[111] George Huntington. *On chorea.* Medical and Surgical Reporter, 1872.

[112] Shantanu P Jadhav, Jason Wolfe, and Daniel E Feldman. Sparse temporal coding of elementary tactile features during active whisker sensation. *Nature neuroscience*, 12(6):792, 2009.

[113] Hueihan Jhuang, Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D Steele, and Thomas Serre. Automated home-cage behavioural phenotyping of mice. *Nature communications*, 1(6):68, 2010.

[114] BJ Jones and DJ Roberts. The quantitative measurement of motor incoordination in naive mice using an accelerating rotarod. *Journal of Pharmacy and Pharmacology*, 20(4):302–304, 1968.

[115] JPND. Experimental models for neurodegenerative diseases. *JPND*, 2014.

[116] Luo Juan and Oubong Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[117] Jean-Pierre Julien and Jasna Kriz. Transgenic mouse models of amyotrophic lateral sclerosis. *Biochimica Et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1762(11):1013–1024, 2006.

[118] Carlos Fernando Crispim Junior, Cesar Nonato Pederiva, Ricardo Chessini Bose, Vitor Augusto Garcia, Cilene Lino-de Oliveira, and José Marino-Neto. Ethowatcher: validation of a tool for behavioral and video-tracking analysis in laboratory animals. *Computers in biology and medicine*, 42(2):257–264, 2012.

[119] Hibiki Kawamata and Giovanni Manfredi. *Introduction to Neurodegenerative Diseases and Related Techniques*, volume 793. Humana Press, 2011.

[120] Elizabeth M Keithley, Cecilia Canto, Qing Yin Zheng, Xiaobo Wang, Nathan Fischel-Ghodsian, and Kenneth R Johnson. Cu/zn superoxide dismutase and age-related hearing loss. *Hearing research*, 209(1):76–85, 2005.

[121] Vivek Khatri, Roberto Bermejo, Joshua C Brumberg, Asaf Keller, and H Philip Zeigler. Whisking in air: encoding of kinematics by trigeminal ganglion neurons in awake rats. *Journal of neurophysiology*, 101(4):1836–1846, 2009.

[122] Les Kitchen and Azriel Rosenfeld. Gray-level corner detection. Technical report, DTIC Document, 1980.

[123] David Kleinfeld, Ehud Ahissar, and Mathew E Diamond. Active sensation: insights from the rodent vibrissa sensorimotor system. *Current opinion in neurobiology*, 16(4):435–444, 2006.

[124] Per Magne Knutsen, Armin Biess, and Ehud Ahissar. Vibrissal kinematics in 3d: tight coupling of azimuth, elevation, and torsion across different whisking modes. *Neuron*, 59(1):35–42, 2008.

[125] Per Magne Knutsen, Dori Derdikman, and Ehud Ahissar. Tracking whisker and head movements in unrestrained behaving rodents. *Journal of neurophysiology*, 93(4):2294–2301, 2005.

[126] D. T. Kobayashi and K. S. Chen. Behavioral phenotypes of amyloid-based genetically modified mouse models of alzheimers disease. *Genes, Brains and Behaviour*, 4:179–196, 2005.

[127] Vikrant Kobla, Yiqing Liang, Feng-Feng Wang, and Yi Zhang. System and method for animal gait characterization from bottom view using video analysis, August 20 2013. US Patent 8,514,236.

[128] Zacharias Kohl, Martin Regensburger, Robert Aigner, Mahesh Kandasamy, Beate Winner, Ludwig Aigner, and Jürgen Winkler. Impaired adult olfactory bulb neurogenesis in the r6/2 mouse model of huntington's disease. *BMC neuroscience*, 11(1):114, 2010.

[129] Dieter Koller, J Weber, T Huang, J Malik, G Ogasawara, B Rao, and S Russell. Towards robust automatic traffic scene analysis in real-time. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision &amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 126–131. IEEE, 1994.

[130] Jiming Kong and Zuoshang Xu. Massive mitochondrial degeneration in motor neurons triggers the onset of amyotrophic lateral sclerosis in mice expressing a mutant sod1. *Journal of Neuroscience*, 18(9):3241–3250, 1998.

[131] Glenn E Krasner, Stephen T Pope, et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3):26–49, 1988.

[132] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[133] R. Lalonde, K. Hayzouna, F. Selimic, J. Marianic, and C. Strazielled. Motor coordination in mice with hotfoot, lurcher, and double mutations of the grid2 gene encoding the delta-2 excitatory amino acid receptor. *Physiology and Behaviour*, 80:333–339, 2003.

[134] Robert Lalonde, MI Botez, and Diane Boivin. Spontaneous alternation and habituation in a t-maze in nervous mutant mice. *Behav Neurosci*, 100(3):350–352, 1986.

[135] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[136] D Brian Larkins and William Harvey. Introductory computational science using matlab and image processing. *Procedia Computer Science*, 1(1):913–919, 2010.

[137] A M Lehmkuh, E R Dirr, and S M Fleming. Olfactory assays for mouse models of neurodegenerative disease. *J. Vis. Exp*, 90, 2014.

[138] JP Lewis and Ulrich Neumann. Performance of java versus c++. *Computer Graphics and Immersive Technology Lab, University of Southern California (January 2003)*, 2004.

[139] X Li, S Mohan, W Gu, J Wergedal, and DJ Baylink. Quantitative assessment of forearm muscle size, forelimb grip strength, forearm bone mineral density, and forearm bone size in determining humerus breaking strength in 10 inbred strains of mice. *Calcified tissue international*, 68(6):365–370, 2001.

[140] Richard T Libby, Michael G Anderson, Iok-hou Pang, Zachary H Robinson, Olga V Savinova, I MIHAI Cosma, AMY Snow, Lawriston A Wilson, Richard S Smith, Abbot F Clark, et al. Inherited glaucoma in dba/2j mice: pertinent disease features for studying the neurodegeneration. *Visual neuroscience*, 22(5):637–648, 2005.

[141] Yow-Sien Lin, Chiung-Mei Chen, Bing-wen Soong, Yih-Ru Wu, Hui-Mei Chen, Wen-Ying Yeh, Dai-Rong Wu, Yi-Jun Lin, Paul Wai-Fung Poon, Mei-Ling Cheng, et al. Dysregulated brain creatine kinase is associated with hearing impairment in mouse models of huntington disease. *The Journal of clinical investigation*, 121(4):1519, 2011.

[142] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[143] Lisa A Lione, Rebecca J Carter, Mark J Hunt, Gillian P Bates, A Jennifer Morton, and Stephen B Dunnett. Selective discrimination learning impairments in mice expressing the human huntington's disease mutation. *Journal of Neuroscience*, 19(23):10428–10437, 1999.

[144] Hong-Chih Liu and Mandyam D Srinath. Partial shape classification using contour matching in distance transformation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(11):1072–1079, 1990.

[145] S. F. Logue, E. H. Owen, and D. L. Rasmussen. Assessment of locomotor activity, acoustic and tactile startle, and prepulse inhibition of startle in inbred mouse strains and f1 hybrids: implications of genetic background for single gene and quantitative trait loci analyses. *Neuroscience*, 1997.

[146] Albert C Ludolph, Caterina Bendotti, Eran Blaugrund, Adriano Chio, Linda Greensmith, Jean-Philippe Loeffler, Richard Mead, Heiko G Niessen, Susanne Petri, Pierre-Francois Pradat, et al. Guidelines for preclinical animal research in als/mnd: A consensus meeting. *Amyotrophic Lateral Sclerosis*, 11(1-2):38–45, 2010.

[147] Jamie Ludwig. Image convolution. *Portland State University*, 2013.

[148] Jonathan B Macknin, Makoto Higuchi, Virginia M-Y Lee, John Q Trojanowski, and Richard L Doty. Olfactory dysfunction occurs in transgenic mice overexpressing human τ protein. *Brain research*, 1000(1):174–178, 2004.

[149] Renzo Mancuso, Sara Oliván, Rosario Osta, and Xavier Navarro. Evolution of gait abnormalities in sod1 g93a transgenic mice. *Brain research*, 1406:65–73, 2011.

[150] Antoine Manzanera, Thierry M Bernard, Françoise Prêteux, and Bernard Longuet. Ultra-fast skeleton based on an isotropic fully parallel algorithm. In *International Conference on Discrete Geometry for Computer Imagery*, pages 313–324. Springer, 1999.

[151] Marvin Marcus and Henryk Minc. *Introduction to linear algebra.* Courier Corporation, 1965.

[152] Mathworks. Matlab, 2015.

[153] Slavomir Matuska, Robert Hudec, and Miroslav Benco. The comparison of cpu time consumption for image processing algorithm in matlab and opencv. In *ELEKTRO, 2012*, pages 75–78. IEEE, 2012.

[154] Yvonne McCulloch. Diagnosis and assessment, 2011.

[155] Richard J Mead, Ellen J Bennett, Aneurin J Kennerley, Paul Sharp, Claire Sunyach, Paul Kasher, Jason Berwick, Brigitte Pettmann, Guiseppe Battaglia, Mimoun Azzouz, et al. Optimised and rapid pre-clinical screening in the sod1g93a transgenic mouse model of amyotrophic lateral sclerosis (als). *PloS one*, 6(8):e23244, 2011.

[156] Simon Melov, Norman Wolf, Dorothea Strozyk, Susan R Doctrow, and Ashley I Bush. Mice transgenic for alzheimer disease β-amyloid develop lens cataracts that are rescued by antioxidant treatment. *Free Radical Biology and Medicine*, 38(2):258–261, 2005.

[157] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Computer VisionECCV 2002*, pages 128–142. Springer, 2002.

[158] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[159] Ben Mitchinson, Robyn A. Grant, Kendra Arkley, Vladan Rankov, Igor Perkon, and Tony J. Prescott. Active vibrissal sensing in rodents and marsupials. *Phil. Trans. R. Soc. B*, 366:3037–3048, 2011.

[160] Ben Mitchinson, Chris J Martin, Robyn A Grant, and Tony J Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proceedings of the Royal Society of London B: Biological Sciences*, 274(1613):1035–1041, 2007.

[161] KK Moodley and D Chan. The hippocampus in neurodegenerative disease. In *The Hippocampus in Clinical Neuroscience*, volume 34, pages 95–108. Karger Publishers, 2014.

[162] Stephen Morairty. Detecting neurodegenerative diseases before damage is done, 2013.

[163] Richard Morris. Developments of a watermaze procedure for studying spatial learning in the rat. *Journal of Neuroscience Methods*, 11:47–60, 1984.

[164] Mostafa A Nashaat, Hatem Oraby, Laura Blanco Peña, Sina Dominiak, Matthew E Larkum, and Robert NS Sachdev. Pixying behavior: A versatile real-time and post hoc automated optical tracking method for freely moving and head fixed animals. *eNeuro*, 4(1):ENEURO–0245, 2017.

[165] Adam Nathan. *Windows presentation foundation unleashed.* Pearson Education, 2006.

[166] Adam Nathan. *Windows presentation foundation unleashed.* Pearson Education, 2006.

[167] Shubham Nema, Whidul Hasan, Anamika Bhargava, and Yogesh Bhargava. A novel method for automated tracking and quantification of adult zebrafish behaviour during anxiety. *Journal of neuroscience methods*, 271:65–75, 2016.

[168] Manuela Neumann, Deepak M Sampathu, Linda K Kwong, Adam C Truax, Matthew C Micsenyi, Thomas T Chou, Jennifer Bruce, Theresa Schuck, Murray Grossman, Christopher M Clark, et al. Ubiquitinated tdp-43 in frontotemporal lobar degeneration and amyotrophic lateral sclerosis. *Science*, 314(5796):130–133, 2006.

[169] Choon-Ching Ng, Moi Hoon Yap, Nicholas Costen, and Baihua Li. Automatic wrinkle detection using hybrid hessian filter. In *Asian Conference on Computer Vision*, pages 609–622. Springer, 2014.

[170] Choon-Ching Ng, Moi Hoon Yap, Nicholas Costen, and Baihua Li. Wrinkle detection using hessian line tracking. *IEEE Access*, 3:1079–1088, 2015.

[171] Yuman Nie, Idaku Ishii, Kenkichi Yamamoto, Kensuke Orito, and Hiroshi Matsuda. Real-time scratching behavior quantification system for laboratory mice using high-speed vision. *Journal of real-time image processing*, 4(2):181–190, 2009.

[172] Lucas PJJ Noldus, Andrew J Spink, and Ruud AJ Tegelenbosch. Ethovision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods*, 33(3):398–414, 2001.

[173] Joseph Nunez. Morris water maze experiment. *Journal of visualized experiments: JoVE*, (19), 2008.

[174] D.H. O'Connor, N.G. Clack, D. Huber, T. Komiyama, E.W. Myers, and K. Svoboda. Vibrissa-based object localization in head-fixed mice. *The Journal of Neuroscience*, 30(5):1947–1967, 2010.

[175] Kohei Ogawa, Yasuaki Ito, and Koji Nakano. Efficient canny edge detection using a gpu. In *Networking and Computing (ICNC), 2010 First International Conference on*, pages 279–280. IEEE, 2010.

[176] TG Ohm and H Braak. Olfactory bulb changes in alzheimer's disease. *Acta neuropathologica*, 73(4):365–369, 1987.

[177] John O'keefe and Lynn Nadel. *The hippocampus as a cognitive map.* Oxford: Clarendon Press, 1978.

[178] David S Olton. Mazes, maps, and memory. *American psychologist*, 34(7):583, 1979.

[179] Wikipedia Online. List of programming languages, 2015.

[180] L. Pammer, D. H. OConnor, S. A. Hires, N. G. Clack, D. Huber, E. W. Myers, and K. Svoboda. The mechanical variables underlying object localization along the axis of the whisker. *The Journal of Neuroscience*, 33(16):67266741, 2013.

[181] TYC Pang, NC Stam, J Nithianantharajah, ML Howard, and AJ Hannan. Differential effects of voluntary physical exercise on behavioral and brain-derived neurotrophic factor expression deficits in huntingtons disease transgenic mice. *Neuroscience*, 141(2):569–584, 2006.

[182] Sonam Parakh and Julie D Atkin. Protein folding alterations in amyotrophic lateral sclerosis. *Brain research*, 1648:633–649, 2016.

[183] Terence John Parr. Enforcing strict model-view separation in template engines. In *Proceedings of the 13th international conference on World Wide Web*, pages 224–233. ACM, 2004.

[184] Katrina L Paumier, Stacey J Sukoff Rizzo, Zdenek Berger, Yi Chen, Cathleen Gonzales, Edward Kaftan, Li Li, Susan Lotarski, Michael Monaghan, Wei Shen, et al. Behavioral characterization of a53t mice reveals early and late stage deficits related to parkinsons disease. *PLoS One*, 8(8):e70274, 2013.

[185] Igor Perkon, Andrej Košir, Pavel M Itskov, Jurij Tasič, and Mathew E Diamond. Unsupervised quantification of whisking and head movement in freely moving rodents. *Journal of Neurophysiology*, 105(4):1950–1962, 2011.

[186] Igor Perkon, Andrej Košir, Pavel M Itskov, Jurij Tasič, and Mathew E Diamond. Unsupervised quantification of whisking and head movement in freely moving rodents. *Journal of Neurophysiology*, 105(4):1950–1962, 2011.

[187] Thomas Philips and Jeffrey D Rothstein. Rodent models of amyotrophic lateral sclerosis. *Current Protocols in Pharmacology*, pages 5–67, 2015.

[188] Lawrence Pinto and Christina Enroth-Cugell. Tests of the mouse visual system. *Mamm Genome*, 11:531–536, 2000.

[189] Mike Potel. Mvp: Model-view-presenter the taligent programming model for c++ and java. *Taligent Inc*, page 20, 1996.

[190] HS Prashanth, HL Shashidhara, and Balasubramanya Murthy KN. Image scaling comparison using universal image quality index. In *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on*, pages 859–863. IEEE, 2009.

[191] T.J. Prescott, B. Mitchinson, and R.A Grant. Vibrissal behavior and function. *Scholarpedia*, 6(10):6642, 2011.

[192] Tamara Pringsheim, Katie Wiltshire, Lundy Day, Jonathan Dykeman, Thomas Steeves, and Nathalie Jette. The incidence and prevalence of huntington's disease: A systematic review and meta-analysis. *Movement Disorders*, 27(9):1083–1091, 2012.

[193] G Scott Ralph, Pippa A Radcliffe, Denise M Day, Janine M Carthy, Marie A Leroux, Debbie CP Lee, Liang-Fong Wong, Lynsey G Bilsland, Linda Greensmith, Susan M Kingsman, et al. Silencing mutant sod1 using rnai protects against neurodegeneration and extends survival in an als model. *Nature medicine*, 11(4):429–433, 2005.

[194] Frank Restle. Discrimination of cues in mazes: A resolution of the" place-vs.-response" question. *Psychological review*, 64(4):217, 1957.

[195] Jacopo Rigosa, Alessandro Lucantonio, Giovanni Noselli, Arash Fassihi, Erik Zorzin, Fabrizio Manzino, Francesca Pulecchi, and Mathew E Diamond. Dye-enhanced visualization of rat whiskers for behavioral studies. *Elife*, 6, 2017.

[196] Jason T Ritt, Mark L Andermann, and Christopher I Moore. Embodied information processing: vibrissa mechanics and texture features shape micromotions in actively sensing rats. *Neuron*, 57(4):599–613, 2008.

[197] Edward Rockenstein, Leslie Crews, and Eliezer Masliah. Transgenic animal models of neurodegenerative diseases and their application to treatment development. *Advanced drug delivery reviews*, 59(11):1093–1102, 2007.

[198] Daniel R Rosen, Teepu Siddiquef, David Patterson, Denise A Figlewicz, Peter Sapp, Afif Hentatif, Jeremiah P O'Regan, Han-Xiang Dengf, Diane McKenna-Yasek, Annarueber Cayabyabi, et al. Mutations in cu/zn superoxide dismutase gene are associated with familial amyotrophic. *Nature*, 362:4, 1993.

[199] A Sajjadian, RL Doty, DN Gutnick, RJ Chirurgi, M Sivak, and D Perl. Olfactory dysfunction in amyotrophic-lateral-sclerosis. *Neurodegeneration*, 3(2):153–157, 1994.

[200] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[201] Nicolas Seiller, Nitin Singhal, In Kyu Park, et al. Object oriented framework for real-time image processing on gpu. *Multimedia tools and applications*, 70(3):2347–2368, 2014.

[202] Peter Sestoft. Numeric performance in c, c# and java. *IT University of Copenhagen*, 2010.

[203] L.G. Shapiro and G.C. Stockman. *Computer Vision.* Prentice Hall, ISBN: 0-13-030796-3, 2002.

[204] T. Sherwood, E. Ahmad, and M.H. Yap. Formulating efficient software solution for digital image processing system. *Journal of Software: Practice and Experience.*, 2015.

[205] Hirotaka Shoji, Hideo Hagihara, Keizo Takao, Satoko Hattori, and Tsuyoshi Miyakawa. T-maze forced alternation and left-right discrimination tasks for

assessing working and reference memory in mice. *Journal of visualized experiments: JoVE*, (60), 2012.

[206] Josh Smith. Patterns-wpf apps with the model-view-viewmodel design pattern. *MSDN magazine*, page 72, 2009.

[207] Susan E Smittkamp, Jordan W Brown, and John A Stanford. Time-course and characterization of orolingual motor deficits in b6sjl-tg (sod1-g93a) 1gur/j mice. *Neuroscience*, 151(2):613–621, 2008.

[208] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. 1968.

[209] N.J. Sofroniew, J.D. Cohen, A.K. Lee, and K. Svoboda. Natural whisker-guided behavior by head-fixed mice in tactile virtual reality. *The Journal of Neuroscience*, 34(29):9537–9550, 2014.

[210] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision.* Cengage Learning, 2014.

[211] AJ Spink, RAJ Tegelenbosch, MOS Buma, and LPJJ Noldus. The ethovision video tracking systema tool for behavioral phenotyping of transgenic mice. *Physiology & behavior*, 73(5):731–744, 2001.

[212] A. D. Steele, W. S. Jackson, O. D. King, and S. Lindquist. The power of automated high-resolution behavior analysis revealed by its application to mouse models of huntington's and prion diseases. *PNAS*, 104:1983–1988, 2007.

[213] A Jon Stoessl. Neuroimaging in the early diagnosis of neurodegenerative disease, 2012.

[214] B. Stroustrup. C++ applications, 2014.

[215] Bjarne Stroustrup. *The C++ programming language.* Pearson Education India, 1986.

[216] Matthis Synofzik, Rubén Fernández-Santiago, Walter Maetzler, Ludger Schöls, and Peter M Andersen. The human g93a sod1 phenotype closely resembles sporadic amyotrophic lateral sclerosis. *Journal of Neurology, Neurosurgery & Psychiatry*, pages jnnp–2009, 2010.

[217] J Paul Taylor, John Hardy, and Kenneth H Fischbeck. Toxic proteins in neurodegenerative disease. *Science*, 296(5575):1991–1995, 2002.

[218] Katy Taylor. Alzheimers disease, 2013.

[219] Tonya N Taylor, W Michael Caudle, Kennie R Shepherd, AliReza Noorian, Chad R Jackson, P Michael Iuvone, David Weinshenker, James G Greene, and Gary W Miller. Nonmotor symptoms of parkinson's disease revealed in an animal model with reduced monoamine storage capacity. *Journal of Neuroscience*, 29(25):8103–8113, 2009.

[220] George Teodoro, Rafael Sachetto, Olcay Sertel, Metin N Gurcan, Wagner Meira, Umit Catalyurek, and Renato Ferreira. Coordinating the use of gpu and cpu for improving performance of compute intensive applications. In *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*, pages 1–10. IEEE, 2009.

[221] Kim Tieu. A guide to neurotoxic animal models of parkinsons disease. *Cold Spring Harb. Perspect. Med.*, 1, 2011.

[222] R Blythe Towal, Brian W Quist, Venkatesh Gopal, Joseph H Solomon, and Mitra JZ Hartmann. The morphology of the rat vibrissal array: a model for quantifying spatiotemporal patterns of whisker-object contact. *PLoS Comput Biol*, 7(4):e1001120–e1001120, 2011.

[223] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261. IEEE, 1999.

[224] John W Tukey. Exploratory data analysis. 1977.

[225] Akinori Ueki, Kyoko Goto, Noriko Sato, Hiroyuki Iso, and Yoshio Morita. Prepulse inhibition of acoustic startle response in mild cognitive impairment and mild dementia of alzheimer type. *Psychiatry and clinical neurosciences*, 60(1):55–62, 2006.

[226] Richard F Uhlmann, Eric B Larson, Thomas S Rees, Thomas D Koepsell, and Larry G Duckert. Relationship of hearing impairment to dementia and cognitive dysfunction in older adults. *Jama*, 261(13):1916–1919, 1989.

[227] Bayani Uttara, Ajay V. Singh, Paolo Zamboni, and R.T Mahajan1. Oxidative stress and neurodegenerative diseases: A review of upstream and downstream antioxidant therapeutic options. *Current Neuropharmacology*, 7(1):65–74, 2009.

[228] F Josef van der Staay, Saskia S Arndt, , and Rebecca E Nordquist. Evaluation of animal models of neurobehavioral disorders. *Behavioral and Brain Functions*, 5(11), 2009.

[229] Subramaniam Venkatraman, Ken Elkabany, John D Long, Yimin Yao, and Jose M Carmena. A system for neural recording and closed-loop intracortical microstimulation in awake rodents. *IEEE Transactions on Biomedical Engineering*, 56(1):15–22, 2009.

[230] Carmine Vitale, Vincenzo Marcelli, Roberto Allocca, Gabriella Santangelo, Pasquale Riccardi, Roberto Erro, Marianna Amboni, Maria Teresa Pellecchia, Autilia Cozzolino, Katia Longo, et al. Hearing impairment in parkinson's disease: expanding the nonmotor phenotype. *Movement Disorders*, 27(12):1530–1535, 2012.

[231] Jakob Voigts, Bert Sakmann, and Tansu Celikel. Unsupervised whisker tracking in unrestrained behaving animals. *Journal of neurophysiology*, 100(1):504–515, 2008.

[232] Rainer Von Coelln, Bobby Thomas, Joseph M Savitt, Kah Leong Lim, Masayuki Sasaki, Ellen J Hess, Valina L Dawson, and Ted M Dawson. Loss of locus coeruleus neurons and reduced startle in parkin null mice. *Proceedings of the National Academy of Sciences of the United States of America*, 101(29):10744–10749, 2004.

[233] PME Waite and DJ Tracey. Trigeminal sensory system. in: Paxinos g (ed) the rat nervous system. *Academic Press, San Diego*, pages 705–724, 1995.

[234] EL Walker, WN Dember, RW Earl, and AJ Karoly. Choice alternation: I. stimulus vs. place vs. response. *Journal of Comparative and Physiological Psychology*, 48(1):19, 1955.

[235] Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

[236] WI Welker. Analysis of sniffing of the albino rat 1. *Behaviour*, 22(3):223–244, 1964.

[237] Patrick Weydt, So Yon Hong, Michel Kliot, and Thomas Möller. Assessing disease onset and progression in the sod1 mouse model of als. *Neuroreport*, 14(7):1051–1054, 2003.

[238] WHO. Neurological disorders: Public health challenges, 2006.

[239] Paul Willner. *Behavioural models in psychopharmacology: theoretical, industrial and clinical perspectives*. Cambridge University Press, 1991.

[240] Andrea Wolf, Björn Bauer, Erin L Abner, Tal Ashkenazy-Frolinger, and Anika MS Hartz. A comprehensive behavioral test battery to assess learning and memory in 129s6/tg2576 mice. *PloS one*, 11(1):e0147733, 2016.

[241] Jason Wolfe, Dan N Hill, Sohrab Pahlavan, Patrick J Drew, David Kleinfeld, and Daniel E Feldman. Texture coding in the rat whisker system: slip-stick versus differential resonance. *PLoS biology*, 6(8):e215, 2008.

[242] Philip C Wong, Huaibin Cai, David R Borchelt, and Donald L Price. Genetically engineered mouse models of neurodegenerative diseases. *Nature neuroscience*, 5(7):633–639, 2002.

[243] Christine M. Wooley, Rodger B. Sher, Ajit Kale, Wayne N. Frankel, Gregory A. Cox, and Kevin L. Seburn. Gait analysis detects early changes in transgenic sod1(g93a) mice. *Muscle Nerve*, 32:43–50, 2005.

[244] Thomas A Woolsey and Hendrik Van der Loos. The structural organization of layer iv in the somatosensory region (si) of mouse cerebral cortex: the description of a cortical field composed of discrete cytoarchitectonic units. *Brain research*, 17(2):205–242, 1970.

[245] Thomas A Woolsey, Carol Welker, and Richard H Schwartz. Comparative anatomical studies of the sml face cortex with special reference to the occurrence of barrels in layer iv. *Journal of Comparative Neurology*, 164(1):79–94, 1975.

[246] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[247] Mu Yang and Jacqueline Crawley. Simple behavioral assessment of mouse olfaction. *Current Protocols in Neuroscience*, page Unit 8.24, 2010.

[248] Xin-Ran Zhu, Lyutha Maskri, Christina Herold, Verian Bader, Christine C Stichel, Onur Güntürkün, and Hermann Lübbert. Non-motor behavioural impairments in parkin-deficient mice. *European Journal of neuroscience*, 26(7):1902–1911, 2007.