

Eyelid and Blink Tracking in an Animal Model of Facial Palsy

Guy Tsrer

Master of Engineering

Biological and Biomedical Engineering Master's Program

Biomedical Engineering Department

McGill University

Montreal, Quebec

20-09-2018

A thesis submitted to McGill University in partial fulfillment of the requirements
of the degree of Master of Engineering.

© Guy Tsrer, 2018

Table of Contents

Abstract.....	3
Abrégé.....	4
Acknowledgments.....	5
Contribution of authors.....	6
Chapter 1: Introduction.....	7
Chapter 2: Background of Facial Palsy and Neural Regeneration	10
2.1 Background	10
2.2 Current treatments.....	10
2.3 Assessment of Neural Regeneration in Animal Models of Facial Palsy.....	12
Chapter 3: Review of Blink Tracking Methods.....	17
3.1. Blink tracking methods	17
3.1.1. Physical methods	17
3.1.2. Physiological signals.....	18
3.1.2. Light-based methods.....	20
3.1.3. Video-based methods.....	20
3.2. Thesis Rationale.....	23
Chapter 4: Methods.....	24
4.1. Requirements.....	24
4.2. Experimental Setup	25
4.3. Pilot Study Analysis Methods.....	29
4.3.1. Red Pixel Tracking	29
4.3.2. Edge Tracking.....	33
4.4. Final Algorithm: The active-contour Method	38
4.4.1. Rationale	38

4.4.2.	Method Background.....	38
4.4.3.	Problems and Solutions.....	40
4.4.4.	Eye Tracking Algorithm	43
4.5.	Validation.....	48
Chapter 5:	Results.....	50
5.1.	Introduction	50
5.2.	Pilot Studies.....	51
5.2.1.	Red Pixel Detection	51
5.2.2.	Edge detection.....	55
5.3.	Active-contour Algorithm	59
5.4.	Eye Area vs. Minor Axis Length	63
Chapter 6:	Discussion and Conclusions.....	65
6.1.	Discussion	65
6.1.1.	Summary	65
6.1.2.	Experimental Setup.....	66
6.1.3.	Limitations	67
6.1.4.	Future Work	69
6.2.	Conclusions	71
7.	References:.....	72
8.	Appendix.....	79
8.1.	Project code	79
8.2.	Code manual.....	79

Abstract

Facial palsy (FP) is a clinical condition caused by damage to the facial nerve, resulting in paralysis of facial muscles that may affect patients' speech, eyesight and social interactions. We hypothesize that activity can be restored in the injured hemiface, by electrical stimulation of its muscles, using the activity of muscles on the healthy hemiface as a control input. To explore this hypothesis, our group uses a rat model of FP, which treats blinking and whisking as the features of interest in facial movement. This thesis describes the development of an experimental system and a methodology for the automatic measurement of eyelid displacement and detection of blinks in video recordings of the rat. Specifically, we used an active-contour approach to localize and track rodent eyes in a head-fixed video. The algorithm is initialized by manually marking the eye contour in the first frame of the video; subsequent frames are analyzed automatically based on an energy function that depends on image features in the region of interest. Our results demonstrate that this technique detects blinks in video recordings with 100% success rate, and that there is a high correlation between algorithm output and manual outlining.

Abrégé

La paralysie faciale (PF) est un état clinique résultant de dommages au nerf facial, entraînant la paralysie des muscles faciaux qui peuvent affecter la parole, la vision et les interactions sociales des patients. Nous émettons l'hypothèse que l'activité peut être restaurée dans la moitié du visage atteinte au moyen de la stimulation électrique de ses muscles, en utilisant l'activité des muscles de la moitié saine du visage comme une entrée de contrôle. Pour explorer cette hypothèse, nous utilisons un modèle de rat de PF, qui traite le clignotement des yeux et le mouvement des moustaches comme étant les caractéristiques de l'intérêt dans le mouvement facial. Cette thèse décrit le développement d'un système expérimental et d'une méthodologie de mesure automatique du déplacement des paupières et de la détection des clignotements dans les enregistrements vidéo du rat. Plus précisément, nous avons utilisé une approche de mesure aux contours actifs pour localiser et suivre les yeux du rat dans une vidéo à tête fixe. L'algorithme est initialisé en marquant manuellement le contour des yeux dans la première image du vidéo; les images suivantes sont analysées automatiquement sur la base d'une fonction d'énergie qui dépend des caractéristiques de l'image dans la région d'intérêt. Nos résultats démontrent que cette technique détecte des clignotements dans des enregistrements vidéo avec un taux de réussite de 100% et une forte corrélation entre le résultat de l'algorithme et la décrivant manuelle.

Acknowledgments

I would like to thank and express my appreciation to my supervisor, Dr. Robert E. Kearney, for providing me a great opportunity to learn, develop, explore and work under his guidance and for his continuous feedback and comments on this document, which helped me properly express and explain my ideas and my work.

I would like to thank Dr. Nate Jowett and Diego L. Guarin for providing feedback when necessary, and for their time hosting me and helping with the experiments.

Thank you to the administrative staff in the Biomedical Engineering Department, most specifically to Pina Sorrini, for supporting and caring from day one, and providing assistance and a supportive environment throughout my time with the department.

Lastly, I would like to thank my parents who were always there for me at all times during my studies from across the world and made my move to Montreal possible with their endless support.

This work was supported by the National Institutes of Health and the Biomedical Engineering Department at McGill University.

Contribution of authors

Most of the work presented in this thesis is my own work. Dr. Robert E. Kearney provided valuable input and suggestions in the development of the different methods and helped shaping this thesis document. Dr. Nate Jowett provided feedback that helped improve algorithm functionality. I have developed the two initial methods and the final algorithm presented in this thesis, and manually validated the data for 11 video recordings.

All video recordings were made in the Massachusetts Eye and Ear (MEEI) at Harvard University. Diego Guarin and Chris Knox recorded the videos used for the first two studies (red-pixel and edge-tracking), as well as for the active-contour method in setup A. Diego, Chris and I planned, constructed and tested experimental setup B (chamber and backlight). We also recorded the second set of videos, covering whisker pad and eyelids.

Chapter 1: Introduction

Facial palsy (FP) is a serious condition where facial nerves fail to innervate patient's facial musculature resulting in varying degrees of paralysis. The condition may be temporary or last a lifetime. Bell's palsy is the most prominent cause of facial palsy, but there is a variety of other causes including trauma, tumours, infections and post-surgical insults to nerves.

Facial palsy symptoms differ from one patient to another and depend on the degree of injury of each patient. Loss of facial muscle control causes problems with basic voluntary and involuntary activities such as blinking, smiling and talking, as the facial nerve innervates orbicularis oculi muscles and orbicularis oris muscles. This also leads to loss of facial symmetry at rest, which in itself is a problem since it affects social aspects in patients' lives, as it is visible to the naked eye and will often attract unwanted attention.

Our group hypothesizes that facial symmetry and spontaneous symmetrical movements of the eyes and the mouth can be restored by a functional electrical stimulation (FES) system that activates muscles on the injured hemi-face, based on electromyogram (EMG) input from the healthy hemi-face. A first step to establish such system is to understand which EMG patterns represent different facial movements. To map the relationship between EMG and facial movements, we must first quantify facial movements, more specifically blinking and smiling, into measurable signals.

As a first stage, we developed a rat model for FP. In rats, blinking and whisking (the act of repeated retraction and protraction of facial whiskers) are equivalent to human blinking and smiling, the facial movements of interest. Therefore, we need to create a system to track blinking and whisking behavior through time and provide a quantitative measure of both movements. In addition, these measures could be used to assess nerve regeneration, by comparing measures of blinks or smiles

at different points over a long period (several months, for example). Improved measurements will indicate a recovering nerve, as nerve health can be reflected in muscle functionality.

To meet these needs, we decided to use a single high-definition video camera to capture both whisker and eyelid movements. Each video recording leads to four output signals, one for each eye and one for the whisker-pad on each side. This thesis focuses on the acquisition and analysis of blinking signals from the proposed system.

Methods of measuring blinking behavior in animals as well as humans have been developed previously. However, these methods had limitations that made them inappropriate for our study. Some of the main obstacles in existing methods were the system complexity, as they often required infrared light emitters and sensors, and in some cases required mechanical connections to the eyelid that were invasive and complicated to record. In addition, no previously described method allowed recording of both eyelid and whisker movement using the same experimental system in a single recording.

This study aimed to accurately quantify eyelid movement from video recordings of an animal model of facial palsy. We wanted to quantify any eyelid movement, from complete blinks to partial closure. In addition, we aimed to create a new experimental system that is capable of simultaneously recording both eyelid movement and whisker movement.

This thesis is divided into six chapters. Chapter 2 provides background information about facial palsy including its physiology, pathology and common treatment methods.

Chapter 3 reviews relevant methods of quantifying and tracking blinks in animals and in humans and will discuss their advantages and disadvantages.

Chapter 4 presents the experimental setup we developed as well as the methodology we used to record videos of rodents. It will then describe two analysis methods we used to explore properties of the blinking videos. The final method we developed to quantify and track eyelid movement in the videos will then be described, as well as the challenges encountered during development and their solutions. Finally, the technique developed to validate the algorithm is presented.

Chapter 5 presents the results of the automated algorithm we developed with comparison to the manual validation results, with analysis examples of several video files, and an overview of all videos analyzed. In addition, it will present results from the initial methods we used to learn the system's characteristics. In this chapter, we will also explain the choice of metrics we used as the algorithm output.

Finally, chapter 6 discusses the different advantages of the method we developed and the reasons it stands out. In addition, problems that persisted are presented and as are possible directions for future work. Finally, the chapter will finally presents the main conclusions regarding our work and the goals we tried to achieve.

Chapter 2: Background of Facial Palsy and Neural Regeneration

2.1 Background

Facial palsy (FP) is a devastating clinical condition resulting from insult to the facial nerve (FN) and is the most common type of nerve paralysis in humans [1]. The causes of FP can be idiopathic (Bell palsy, responsible for approximately 50% of all FP cases with an incidence rate of 23 per 100,000 people per year), tumours, Lyme disease, trauma, and post-surgical insults [2, 3]. FP is characterized by poor or absent control of facial muscles, which causes patients to experience communication, functional, and aesthetic difficulties. Oral problems include speech articulation hardships and the inability to smile, while eye problems revolve around the inability to blink and protect the cornea [4]. This inability to blink is an important problem for FP patients since incomplete blinks impair the distribution of tear film across the eye, and increase the chances of corneal infections, irritations and inflammation. If left untreated, some of the difficulties can translate in the long term to psychological consequences, such as increased levels of anxiety and depression, avoidance of social contacts, and communicative disorders. Other difficulties may cause permanent physiological damage, including nasal valve dysfunction, vision loss and rupture of the eye globe [5-9].

2.2 Current treatments

Treatment for FP ranges from physical therapy to surgical procedures, depending on its severity. FP due to Bell's palsy is commonly treated with high-dose corticosteroids to shorten recovery time after symptoms begin. A combination of antivirals and corticosteroids may benefit patients whose FP is due to varicella-zoster virus [10]. When the cause for FP is physical trauma, such as post-surgical insult or blunt trauma to the facial nerve, direct nerve repair is usually required.

When FP is long-standing, injections may help manage the condition – injected fillers add volume to improve symmetry while injected botulinum toxin can reduce muscle hyperactivity and synkinesis (involuntary muscle twitches) in targeted areas. However, when traditional methods fail, surgical intervention may be required to attempt either static or dynamic reanimation [5, 10, 11].

Static reanimation, or static reconstruction, is a surgical procedure aimed at achieving a cosmetic correction by restoring facial symmetry and a fixed neutral expression. Examples include [5]:

- Brow position correction which fixes the brow in a position aligned with that of the healthy side;
- Implanting an eyelid weight in an upper eyelid, to ensure full eye closure;
- Use of a static sling, suspended from the zygomatic arch to lift the oral commissure and create a neutral looking smile.

Static reconstruction methods can improve the psychological effects on patients by making their smile and eyes look more natural in a neutral, everyday situation [5]. Static reconstruction is most common in patients with multiple comorbidities or those who are not candidates for complex surgical procedures due to health issues.

In contrast, dynamic reanimation aims to restore some movement to the injured side. The focus is usually on reanimating oral movements rather than blinking, since static reanimation is usually enough to preserve eye health. Transfer of the temporalis muscle is a popular reanimation technique since it is relatively simple to harvest the temporalis muscle. By doing so, the temporalis muscle replaces the functionality of the zygomaticus muscle and enables smiling. An alternative procedure that may be used is nerve grafting, where a donor nerve is used to replace the injured

one. Cross-facial-nerve grafting (CNFG) is a common grafting technique in which the donor nerve branch is harvested from the non-paralyzed side, and implanted in the injured side, to replace the damaged nerve. CNFG allows spontaneous smile (as a result of emotion, for example), while avoiding formation of facial twitches. It may, however, impair the healthy side or cause synkinesis due to nerve harvesting [10].

Dynamic reanimation of the eye using a surgical method was suggested by Frey *et al.*, by transplanting the gracilis muscle to replace the eye sphincter as well as to reanimate the oral region [12]. Alternatively, Frey suggested a procedure to transposition the temporalis muscle using a technique developed by Gillies and Millard to achieve outcomes similar to transplantation [13].

These dynamic reanimation methods are highly invasive, require muscle or nerve transplants or manipulation, and achieve only limited facial movement. An alternative, less invasive solution lies in functional electrical stimulation (FES). FES is a technique that utilizes low energy electrical pulses to artificially activate the sensory motor system to overcome paralysis or disability. Using FES, a specific motor nerve can be stimulated to affect a desired muscle. Utilizing muscle activity signals (EMG) from the healthy hemi-face as control signals, stimulus can be applied to the injured hemi-face with minimal lag, to replicate a movement and create an overall symmetric facial behavior, including both blink and smile. Attiah *et al.* developed and tested a rodent model of closed-loop FES facial reanimation, and showed that whisker protraction could be controlled by varying the number of stimulus pulses and stimulus frequency [14, 15].

2.3 Assessment of Neural Regeneration in Animal Models of Facial Palsy

Animal models are useful to study the pathology and treatment of FP and properties of neural regeneration. Early studies used large animals, such as cats and rabbits, but since these were

expensive and required intensive care, and their anesthesia was not reliable, smaller animals like rats were later introduced [16]. FN innervation in rats is similar to that of humans, and therefore they are a useful model for the study of FP [17, 18].

Assessing neural regeneration is essential in studies of nervous damage and in tracking patient recovery. Both microscopic and behavioral measures are commonly used in facial nerve studies [19]. The most common **microscopic** outcome measure is axon counting; a cross-section of a nerve is cut and extracted, axons are counted, and their myelin is inspected for growth under a microscope to determine whether the nerve is recovering. The advantage of microscopy-based measure is that it provides detailed information, as axons may be counted individually. However, it has several key disadvantages. First, it is highly invasive - to obtain a nerve cross-section, a nerve must be dissected and removed which is not possible in humans. Second, time course studies become expensive and more complex to perform, as they require the dissection and removal of axons at each point in time. Third, this is an anatomical measure that does not necessarily have direct correlation with function [19, 20].

Non-invasive behavioral measures evaluate functional activities (such as movements) that depend on the nerve's health. These are easier to measure noninvasively and may give a higher-level understanding of the nerve's status. These measures differ with the nerve studied and the muscle it innervates [21, 22].

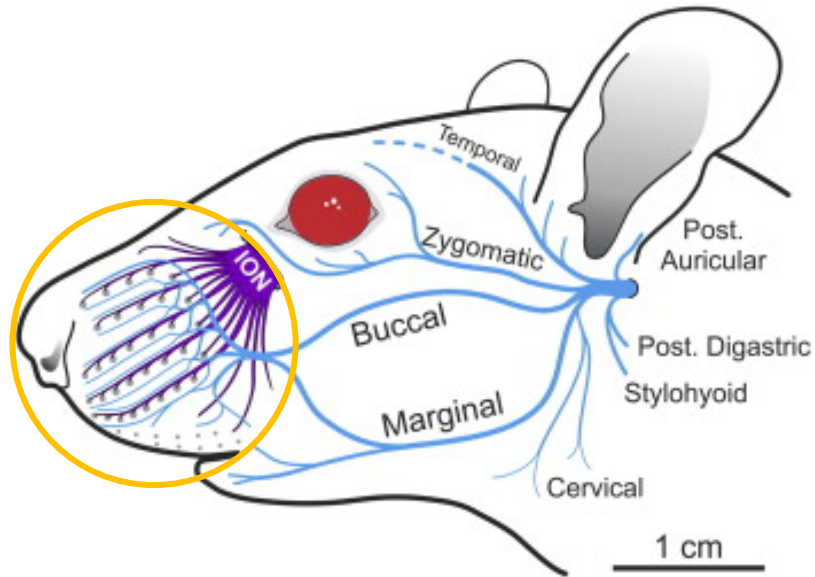


Fig. 1 – Illustration of a rat’s face. The area within the orange circle represents the mystacial pad of the rat. The FN and its branches are shown in blue [16].

Whisking is a commonly used non-invasive assessment metric. A rat’s whiskers are located on both sides of the rodent’s snout, in areas called the mystacial pads (marked in Fig. 1). Each pad contains about 35 large whiskers, used mainly for sensing via whisking movements [23]. The buccal branch of the FN is responsible for whisking in rats, and is equivalent to the buccal nerve in humans, which innervates the oral muscles. Thus, damage to the FN will affect smiling in humans and whisking in rats, highlighting the importance of whisker tracking, as it can help understanding how damage to the buccal nerve affects the human smile. Whisking may be assessed by measuring the frequency of whisker protraction and retraction pre- and post-injury, or by measuring angular velocity and acceleration of forward whisker movement [19].

Blinking is a second means to assess neural regeneration of the FN. To understand why, consider the structure and mechanism of blinking. The orbicularis oculi (OO) muscle, innervated by the FN, is responsible for closing the eyelid, while the levator palpebrae (LP), innervated by the third

cranial nerve, is responsible for opening the eyelid (see Fig. 2). A blink occurs when the LP is turned off and the OO is switched on, resulting in a rapid lowering of the upper eyelid.

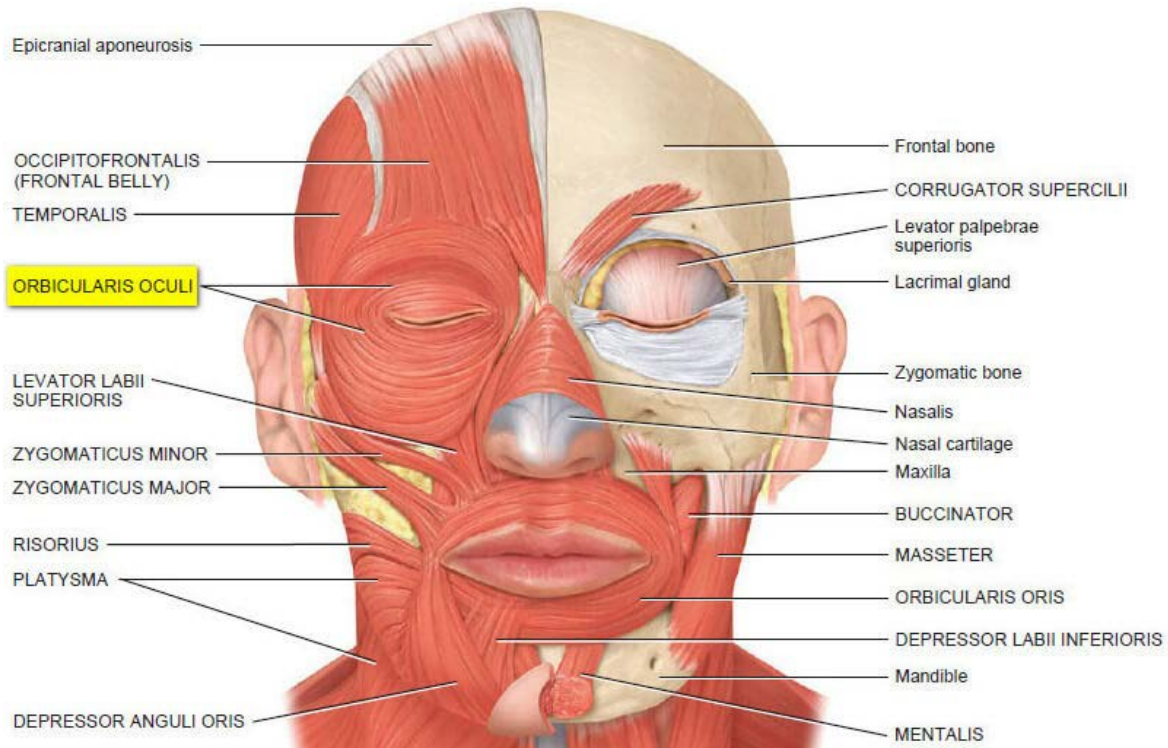


Fig. 2 - Illustration of a human's eye and eyelid musculature. The orbicularis oculi is innervated by the FN and responsible for eye closure, while the levator palpebrae is innervated by the third cranial nerve [25].

In patients with FP, the zygomatic branch of the FN may be injured so that the innervation of the OO is not adequate to fully close the eyelid during a blink and so the cornea is not fully covered (a condition known as lagophthalmos). In rats, as in humans, the zygomatic branch is affected when FP is induced, thus blinking behavior is important to assess as it provides a functional measure of nerve damage. Measuring blinking, or more specifically eye closure, provides valuable information on the facial nerve by quantifying lagophthalmos [26]. Full eye closure indicates full

neural function, while the degree to which the eye does not fully close provides a measure of the extent of neural damage.

Chapter 3: Review of Blink Tracking Methods

To fully understand FP in human patients, it will be important to measure both smiling and blinking. Thus, in the rat model of FP, it will be important to measure the analogues - both whisking and blinking. This thesis deals with the development of a noninvasive, quantitative method to measure blinking and eye closure. The following section will review existing methods of blink tracking.

3.1. Blink tracking methods

Blinks are fast movements of the eyelids over the ocular surface, whose frequency and duration vary from one blink to another and from one person to another. Blinks may be completed in less than 100 ms, so a high temporal resolution is required to measure them. Blinks were first tracked in 1928 by Kennedy and Ponder [27], who counted blinks manually to understand their frequency characteristics and their relation to mental tension. Since then, numerous techniques have been developed to detect and track blinks. These include approaches requiring physical attachments to the eyelids; methods based on physiological signals; methods using light emitters and sensors; and camera-based methods.

3.1.1. Physical methods

The first methods developed to measure blink were physical in nature. Pennypacker *et al.* described an eyelid recording apparatus in which the arm of a micro-torque potentiometer was connected to a primate's eyelid using surgical wire. Tension on the potentiometer's spring was adjusted to permit the animal to blink with minimal impairment of movement. Eyelid motion moved the potentiometer arm, resulting in a signal proportional to eyelid movements [28].

Another approach used a search coil in a magnetic field. The subject was placed inside an oscillating magnetic field. A small, light circular coil (5mm diameter weighing 0.1g) was attached to the eyelashes of the upper eyelid. Movement of the eyelid, as it opened or closed, induced a current in the coil, proportional to the eyelid velocity, due to electromagnetic induction [29, 30].

Marcelli *et al.* used a miniature gyroscope attached to the upper eyelid to measure its angular velocity, as it opened and closed. Angular displacement was calculated by integration [31].

There are several issues associated with these methods. First, any method requiring a physical attachment to the eyelid increases stress on the subjects. This is particularly important in behavioral studies where it is essential to minimize external stress and keep the subject's conditions as close as possible to natural [32]. Secondly, setting up each experiment is cumbersome and requires delicate manual labor (e.g. connecting the eyelids to different mechanisms), making recordings time-consuming and resource-expensive. Furthermore, such methods cannot be used for chronic measurements over prolonged periods due to their complexity and invasiveness.

3.1.2. Physiological signals

Different physiological signals have been used to detect blinks. In 1984, Stern *et al.* found that blinks are associated with changes in the vertical electrooculogram (EOG), which measures the potential between the front and back of the eye [33]. EOG is most commonly used to record movements of the eye globe. However, if one EOG electrode is placed above the eye, and another under, such that they form a virtual vertical line together with the pupil, their potential changes will be related to movement of the eyelid [34].

Electromyography (EMG) of the orbicularis oculi muscle has also been used to study blinks by using filtering methods to exclude irrelevant information (for example, peaks as a result of smiling)

[35-37]. However, this approach does not measure blink magnitude, but rather provides a binary output each time a blink occurred. Methods that combine EOG with EMG to study differences between spontaneous, voluntary and reflex blinks have also been proposed. Using both signals allowed different types of blinks (e.g. reflex blinks as a result of an outside stimulation, and spontaneous blinks that occur naturally) to be distinguished [38, 39].

Blinks produce a transient intraocular pressure (IOP) rise of more than 10mmHG, thus an IOP sensor has been suggested as a mean to measure blinks [40, 41]. IOP can be accurately measured using a contact lens sensor (CLS); for example, Leonardi *et al.* developed a CLS based on a soft contact lens with a strain gauge embedded in it. In the IOP signal extracted from the CLS, spikes were classified as blinks provided they had a specific structure: a local minimum followed by a local maximum followed by a final local minimum [42].

Methods based on physiological signals, like IOP, EMG and EOG, require less physical manipulation and are easier to set up than mechanical methods, since they require only a few, easily placed electrodes or sensors. This makes long-term tracking possible. However, although less invasive than physical methods, the electrodes might alter the behavior of the test subject, as it is not in its natural mode anymore, and might be limited in movement, or respond to the attachment of electrodes in an unwanted manner, possibly complicating the recordings. Another limitation is that physiological signals do not provide a direct measure of eyelid movement. Instead, it must be extracted from specific patterns in the signals and does not provide a quantitative measure of eye closure, but rather a binary output indicating whether the eye is open or not.

3.1.2. Light-based methods

Light-based methods were developed to sense and track blinks with no physical contact. Thompson used a light-emitting diode (LED) to shine a beam of low-intensity infrared (IR) light onto the corneal surface and captured the reflection with a phototransistor. The eyelid's status (open vs. closed) was estimated from the amount of light received, since the surface of the cornea reflects less light than the eyelid [43]. Thompson's method was accurate, and others have improved upon it. Thus, Ryan *et al.* introduced a longer-distance system, that handles IR noise better; and Caffier and Frigerio mounted the IR apparatus on an eyeglass frame to continuously track and analyze eye blink behavior [44-46].

Heaton suggested a system combining two devices to allow tracking of eyelid and whisker movement simultaneously [17]. Like previous methods, the system used a device similar to that described by [43] and added a whisker tracking apparatus capable of tracking whisker movement using laser micrometers [47]. This enabled the recording of two facial movements, but the device had to be placed in close proximity to the animals and blocked their field of vision, possibly affecting their behavior.

A major limitation of IR methods is that the setup must be precise so that the light beams are aimed directly at the eyes. Since the visible surface area of the eye is small, the smallest misalignment in IR beam direction may lead to wrong recordings, as the relevant reflected light must come from the surface of the cornea and the eyelid, and not from the skin surrounding the eye.

3.1.3. Video-based methods

Video-based systems provide a less invasive and more natural way to record eyelid movement, with less obstruction to the animals' vision. These methods consist of video recorders focused on the patients' eye, recording for a given period of time, and using different approaches to image

processing to detect the eye structure within each frame. Subjects are not physically limited as in physical methods and the experimental setup is, in most cases, less cumbersome than light-based methods, physiological methods and physical methods, as recordings are done with cameras that are easy to set up and manage.

Early methods of image processing to track movement used facial markers. Isono *et al.* introduced a system that used 24 markers fixed to the subject's face in pre-determined locations. Recording at 30fps, the direction and distances of the markers were traced between consecutive frames, to create a map of facial movement, with a focus on eye closure [48]. While useful for tracking overall improvement in facial paralysis, this method did not capture single blinks, but rather the magnitude of marker movement. Somia and Bracha replaced the facial markers with eyelid-specific markers and a lightweight helmet with a camera mounted on it, to track the eyelid velocity and position. This method had higher temporal resolution and did track smaller changes compared to previous methods, since the camera was mounted on the helmet and maintained a nearly-fixed position relatively to the markers. However, the setup required many manual adjustments of the marker locations, and placement of a specially designed helmet, meaning it is not easily reproducible, as other researchers would have to reconstruct a similar helmet to achieve similar results, and adjust the algorithm to match their own version of the helmet. [49, 50].

Subsequently, other video methods that do not require markers have been developed. One group of methods combines video recordings with background IR light for better detection of the eye. Thus, Lalonde created a dark recording environment, illuminating the subject's face directly with near IR light, while recording with a black and white camera. Each pair of consecutive frames was differenced to detect areas of significant motion to locate the region of the eye. Once the region of interest (ROI) was set, an optical flow field was calculated and blinks were detected when there

was a downwards movement followed by upwards movement of both eyes [51, 52]. In a different study, the subject's head was fixated to the recording chamber and filmed with a simple IR camera to generate a brightness matrix for each frame. Based on pixel intensity and the number of dark pixels in each frame, the algorithm determined whether the eye was open or closed. This method is very close to automatic; however, user intervention is still required to pick the ROI around the eye and manually define a pixel intensity threshold (changing from one recording to another), to differentiate an open eye from a closed eye. [53].

Another group of methods uses standard video recording with no IR gear. Pardas *et al.* analyzed the video images using a method based on the Snake algorithm (also known as active-contour), which, following an initial stage of detecting the eyes in the first frame, tracked a contour surrounding the eye as it changed throughout the video. The contour was moved around to fit the eye boundaries in each frame by minimizing an energy function depending on the mean value of the image around the contour and the smoothness of the contour. The study focused on tracking the location of eyelids, but did not attempt to extract any metric from the video or to produce a signal that quantified eyelid velocity or displacement [54].

Several template-matching methods that have also been applied to track blinks from simple video recordings [55-58]. These methods are based on the simple observation that, as the eye closes it looks less like an open eye, and as it opens, it looks more like an open eye. Based on this observation, two-dimensional cross-correlation scores between a given frame and an open-eye template are calculated for each frame, after the eye itself was detected. This correlation is high when the eye is open and decreases during a blink. The time-course of the correlation function was then analyzed to measure blink duration. The method's accuracy in binary blink detection was

high at 95.3% and it has the great advantage of requiring no human intervention; the whole process is completely automatic and can run in real time [57].

Most video-based methods were designed to provide only binary detection of blinks. Eye closure is not quantified as a continuous signal but rather as a binary signal specifying whether the eye is open or closed. This is not suitable for the assessment of neural regeneration, since it does not provide a measure of how open an eye is or how well it is innervated.

3.2. Thesis Rationale

As we saw, while there are many methods to detect blinks in humans and animals, most have significant limitations, such as invasiveness, high experimental setup complexity and binary-only output. These limitations make them unsuitable for our purposes. However, the literature review suggested we could overcome these limitations by recording video of the eye, using a single camera, and applying image-processing methods to extract blink information.

This thesis focuses on the development and validation of image-processing methods using the active-contour method to obtain a quantitative measure of eyelid coverage of the eye.

Our thesis objectives were therefore as follows:

- Acquire video recordings to allow both whisker and eyelid tracking.
- Track eyelid movement in a video recording of an animal model.
- Detect blinks and quantify eye closure continuously.

Chapter 4: Methods

4.1. Requirements

The desired characteristics for our algorithm and experimental setup came from the review in chapter 3, as well as from requirements defined by our team, to ensure the algorithm could be used in the facial reanimation project. These include:

- **Non-intrusive recording apparatus:** No physical contact (such as connecting a lever to the eyelid) should be used, since these are too intrusive for the animal during recording. In addition, the animal's field of vision should remain unobstructed.
- **Low recording setup complexity:** No facial or eyelid markers should be used, to reduce manual labour in setting up the experiments, when recording eyelid movement.
- **Reproducible and cost-effective:** To ensure others can easily reproduce the experiment, we would like to record using a single camera, with no additional IR equipment that may bring costs up and complicate the process.
- **Whisker tracking ready:** since this thesis is part of a wider project researching FP, we would like the system to be capable of simultaneously recording both eyelid and whisker movement. This will allow a whisker tracking algorithm (developed by other team members) to function in parallel with the blink tracking algorithm (developed by myself). The recording must be done with a frame rate and resolution high enough to capture minor whisking movements.
- **Continuous output:** On top of binary blink detection, we would like to capture continuous eyelid movement. This will help in assessment of neural regeneration.

4.2. Experimental Setup

The subjects were adult female Lewis rats weighing between 200 and 250 grams, housed individually in a 12h light-dark cycle with ad libitum food and water. All procedures were conducted in the Surgical Photonics and Engineering Laboratory (SPEL) of the Massachusetts Eye and Ear (Boston, MA), and were approved by the Institutional Review Board of the Massachusetts Eye and Ear.

Each animal was fitted with a lightweight, titanium head implant (as seen in Fig. 3A-C) that provided four external attachment points for a head fixation device [59]. The device was mounted on a 75mm diameter polyvinyl chloride (PVC) half-pipe. The animal was strapped to the PVC half-pipe using a cloth sleeve and multiple fabric straps, and fixed to the head fixation device using screws, as shown in Fig. 3 [17].

Experiments were performed using two recording setups. In **setup A**, videos were recorded with a Yi Action Camera (Xiaomi, Beijing), at 240 fps and a resolution of 720x480. The recordings were done with normal ambient lighting. With the rat fixed in place, the camera was placed above the rat's head, and video recorded for up to one minute with room lighting.

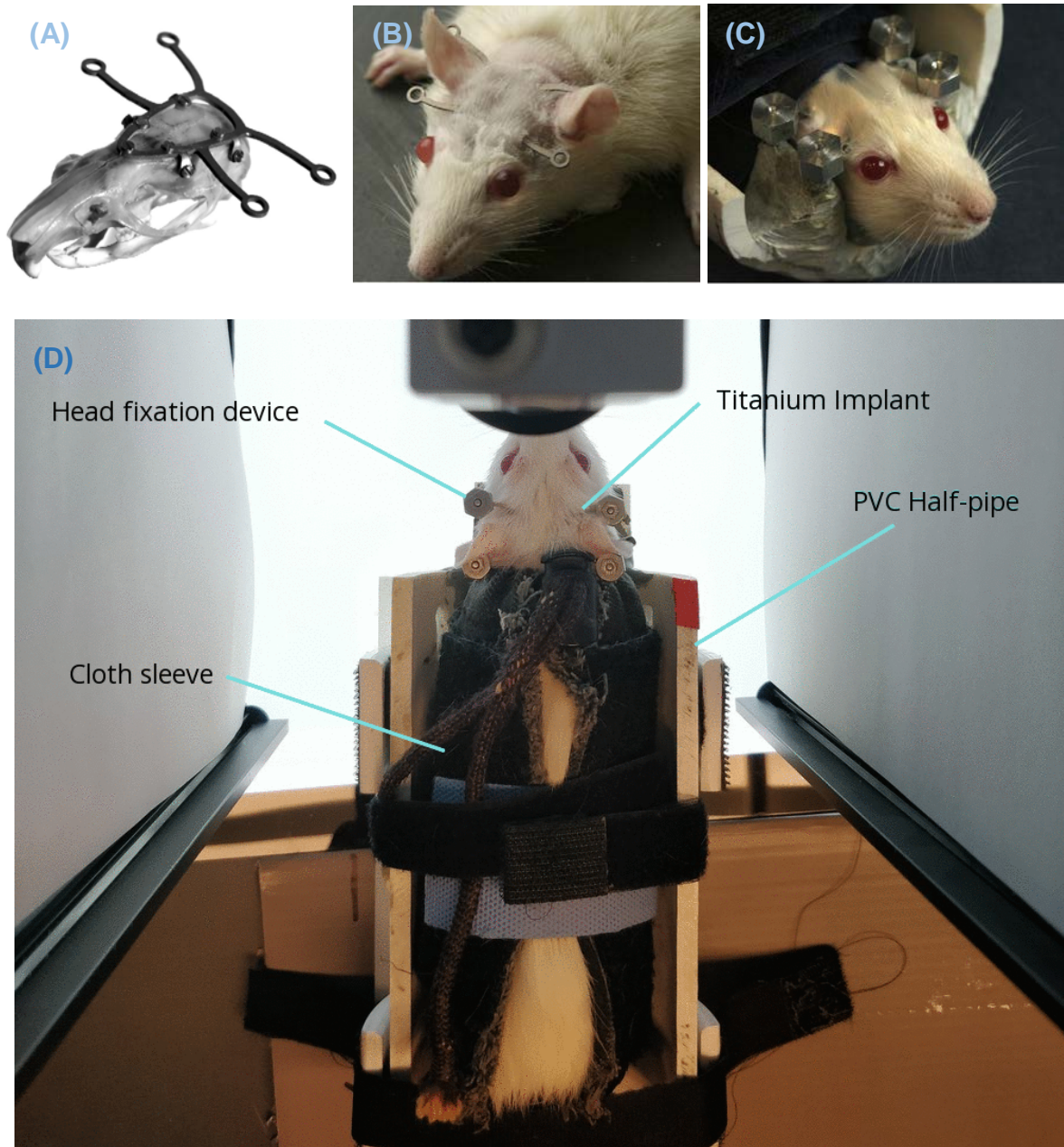


Fig. 3 – Head fixation and half pipe. A) The titanium implant attached to a rat’s skull; B) The titanium implant post-surgery; C) The titanium implant attached to the head fixation device; D) The rodent is placed inside the PVC half-pipe, where it is strapped and fixed to prevent movement using the titanium implant and the head fixation device. Images A-C obtained from [59].

A second recording setup (**setup B**) built later, contained both whiskers and eyelids within the frame. In this setup, a recording chamber created a more controlled environment. The chamber

comprised a rectangular box, height 376 mm, width 158 mm, made of four 2.5mm thick aluminum plates, with an opening (80 mm high by 110 mm wide) through which the animal and half-pipe could be inserted. A fifth plate, mounted directly above the animal, contained a 50.8mm diameter hole for the camera. Ambient lights were turned off during recordings, and the chamber was illuminated with a A4 Portable LED Light Box (LitEnergy) mounted on the chamber base. The light-box was covered with a 200x200 mm White Diffusing Glass (Edmund Optics, Barrington, NJ) to ensure equal distribution of the light. The chamber walls were covered with plain white A4 paper to increase reflection. An ace acA800-510uc HD camera (Basler, Ahrensburg, Germany)) with a 6mm UC Series Fixed Focal Length lens (Edmund Optics, Barrington, NJ) was fixed in place above the light-distributing plate using a metal rod, recording at 500 fps, f2.8 aperture and a resolution of 720x500p. The camera was focused on the rat's snout. Fig. 4A shows the recording chamber and its main features, while Fig. 4B shows an animal inserted into the chamber.

Prior to each recording, animals were brought to the lab for an adjustment period of at least 30 minutes. The animal was placed in a cloth sleeve, to which it had been previously conditioned and placed in the half-pipe, and had its head attached to the head fixation device. The half-pipe was inserted into the recording chamber and recording was begun. Recordings lasted up to 60 seconds. Up to five recordings were made for each placement of the animal. Video images were transferred to a workstation via USB connection and saved in an uncompressed "TIFF" format with 96x96 dpi. SPEL shared the images with our lab at McGill for analysis using a secured Dropbox account.

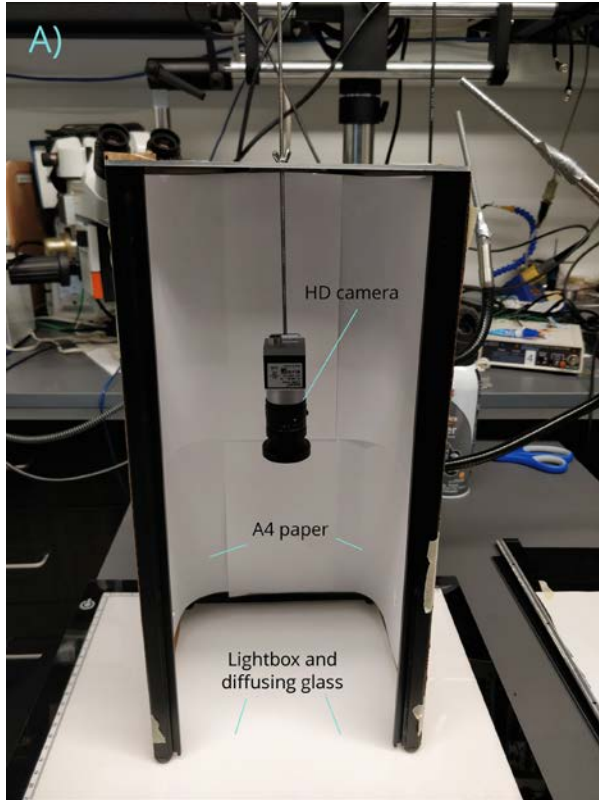
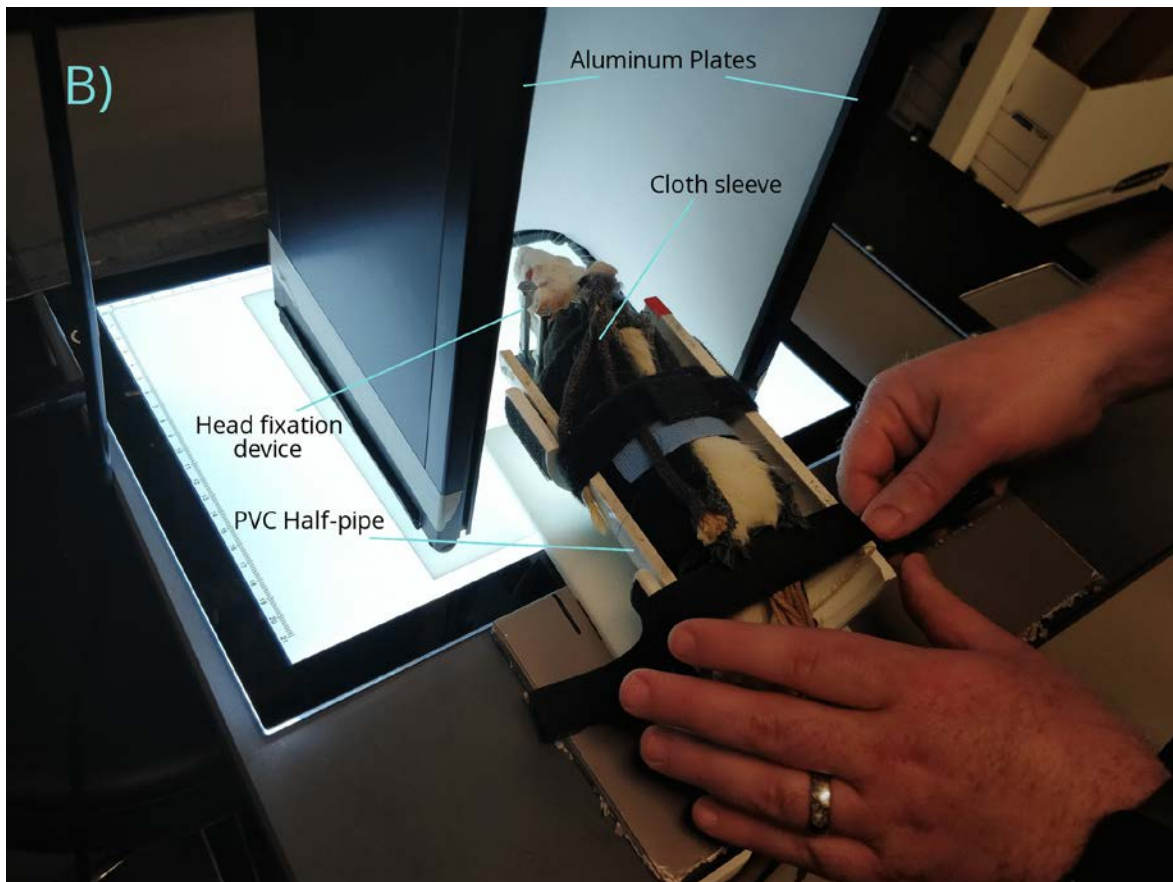


Fig. 4 – Recording chamber and main features.

A) A front view of the recording chamber. A HD camera was fixed from the top plate; the base of the chamber comprised a diffusing glass placed on light emitting plate; the chamber is covered with plain A4 paper to increase indirect illumination.

B) Rodent inserted into the chamber prior to recording, while placed in a cloth sleeve, inside a PVC half-pipe. The rodent's head was fixed in place using the head fixation device.



4.3. Pilot Study Analysis Methods

Two pilot analysis studies were conducted to determine the properties of the video recordings and the challenges associated with tracking and detecting blinks. To quantify eye closure, we used the minor axis of an ellipse fitted to the eye contour. This choice was made since the eye has the general shape of an ellipse and using any other shape to fit the eye contour was less reasonable. The minor axis was used rather than the more intuitive measure, eye area, since blinks occur along the minor axis of the eye, so that any opening and closing movements of the eyelid are best captured in this direction. This was decided after inspecting both output measures and comparing the algorithm results to validation data, as will be discussed in section 5.4. The first study attempted to identify red pixels within each frame to define the eye area, fitted an ellipse to them, and used the minor axis length as a measure of how open the eye was. The second study explored the use of edge detection to establish a contour around the eye in each frame, fitted an ellipse to it, and again used the minor axis length as an output measure.

4.3.1. Red Pixel Tracking

This study attempted to track eye closure using a simple pixel count. The animals used in the experiments were Lewis rats, which have red eyes and white fur, as seen in Fig. 5. Consequently, we thought that it might be straightforward to separate the eyes from the rest of the body purely based on color.



Fig. 5 - Lewis rat used in the recordings. The rat has white fur and red eyes that are easily distinguishable.

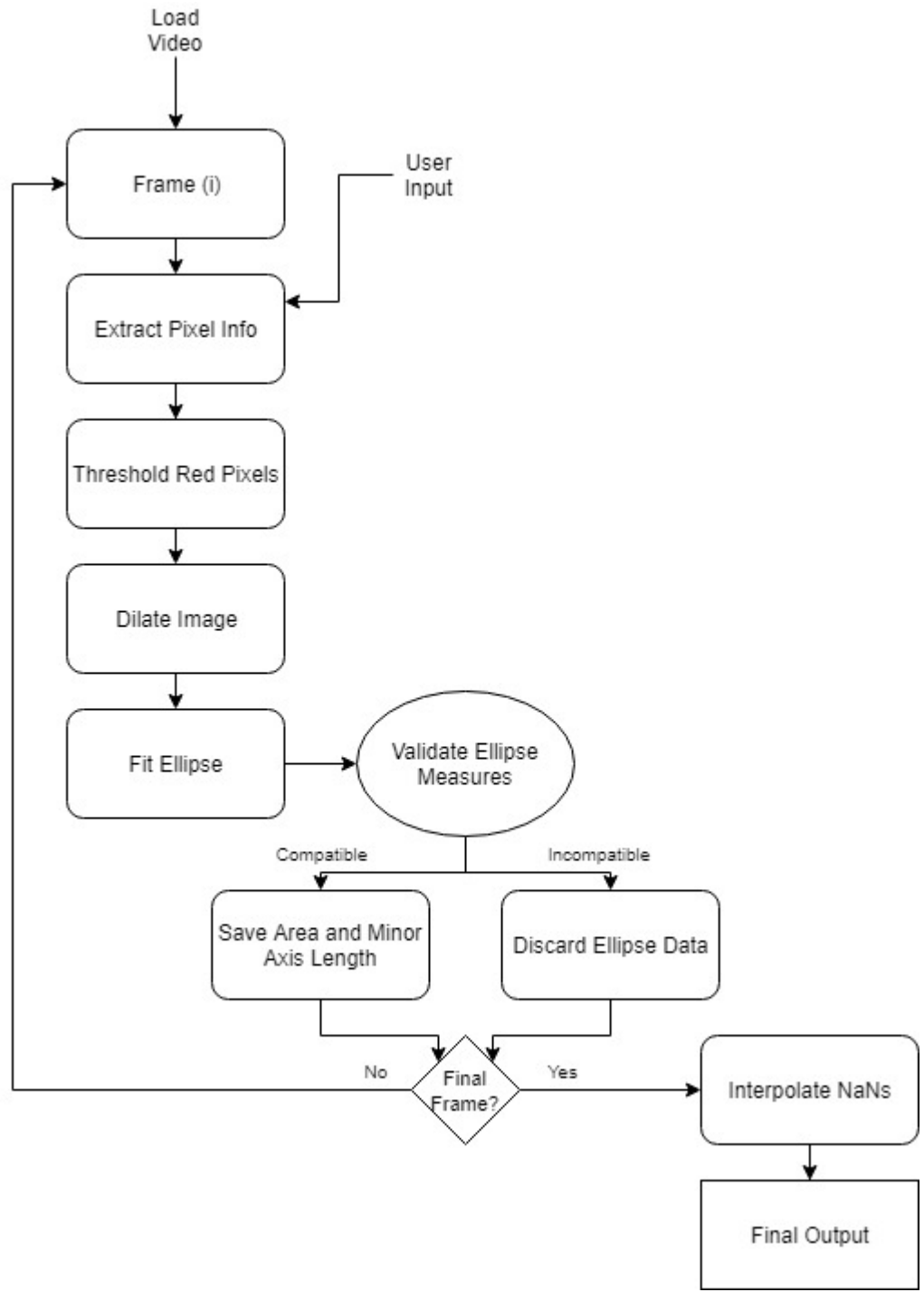


Fig. 6 - Red-pixel tracking algorithm flowchart.

Fig. 6 shows the flowchart of the method we developed to measure eye closure from red pixel tracking. Fig. 7 shows examples of key steps in the procedure:

1. Manually outline the eye area in the initial frame (Fig. 7A).
2. Extract pixel information from the initial area (using MATLAB's `rgb2hsv`) (Fig. 7B):
 - a. Average and standard deviation of hue (H).
 - b. Average and standard deviation of saturation (S).
 - c. Average and standard deviation of brightness (V).
 - d. Length of the minor axis of the eye.
 - e. Angle of the eye with respect to the horizon.
3. Threshold the HSV values to keep pixels based on the following conditions (Fig. 7C):
 - a. Pixel hue to be smaller than average hue, minus its standard deviation, or larger than the average hue, plus its standard deviation.
 - b. Pixel saturation to be within one standard deviation of the average saturation.
 - c. Pixel brightness to be within one standard deviation of the average brightness.
4. Dilate image to remove small gaps between individual pixels using MATLAB's `imdilate` (Fig. 7D).
5. Fit an ellipse to the detected red area using MATLAB's `regionprops` (Fig. 7F).
6. Validate the fitted ellipse by checking that:
 - a. Ellipse angle must be within 10 degrees from that of the initial eye outline.
 - b. Area is no larger than 120% of that of the initial outlined eye outline.
7. If validation fails, set the current frame value to null (in MATLAB, NaN) Go to the next frame, and repeat from step 2.
8. If validation is successful, save the minor axis length. Go the next frame and repeat from step 2.

9. After all frames have been processed, interpolate across all nulls using MATLAB's `interp1` function, to close gaps due to skipped frames.

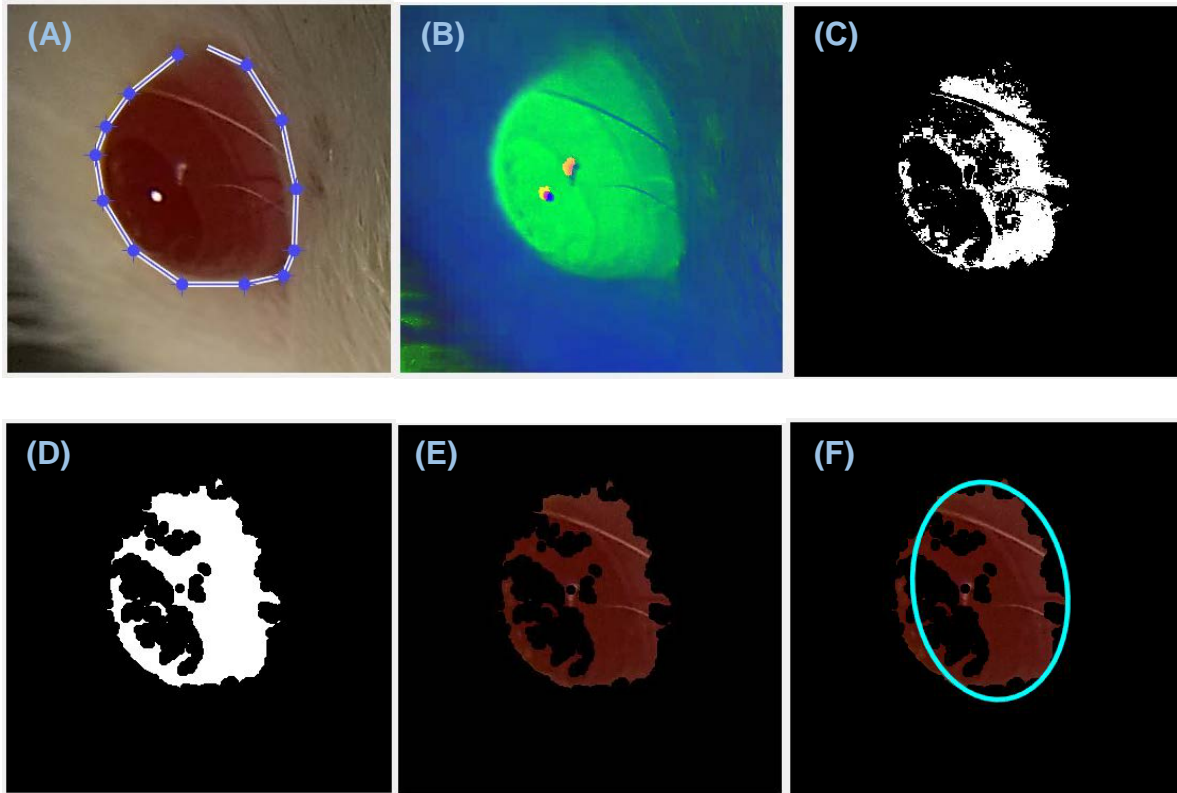


Fig. 7 - Red pixel tracking algorithm, key steps. A) User selects eye area during the initial run; B) Frame in HSV color space prior to applying threshold; C) Frame in binary view, after threshold of irrelevant pixels; D) Dilation reduces gaps between individual pixels; E) Eye pixels highlighted in their original color; F) Eye pixels with the fitted ellipse.

4.3.2. Edge Tracking

A second study attempted to track blinks using edge detection based on the image intensity gradient. To highlight drastic changes in pixel intensity and uncover edges, we used a common filter, known as the Sobel operator, which computes the gradient of intensity changes in both horizontal and vertical directions at each point in the frame. The Sobel operator is frequently used

for edge detection problems, and it is more noise resistant than other common methods, like the Prewitt filter [60]. Moreover, it is supported in MATLAB's image processing toolbox.

The kernels used in the Sobel operator are:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

where S_x represents the kernel used for edges in the X direction, and S_y for edges in the Y direction.

The kernels are convolved with each frame (Fr), and to generate the edge images, G_x and G_y :

$$G_x = S_x * Fr; \quad G_y = S_y * Fr$$

These are then combined to form the overall edge image:

$$G_{tot} = \sqrt{G_x^2 + G_y^2}$$

which accounts for all edges in both directions.

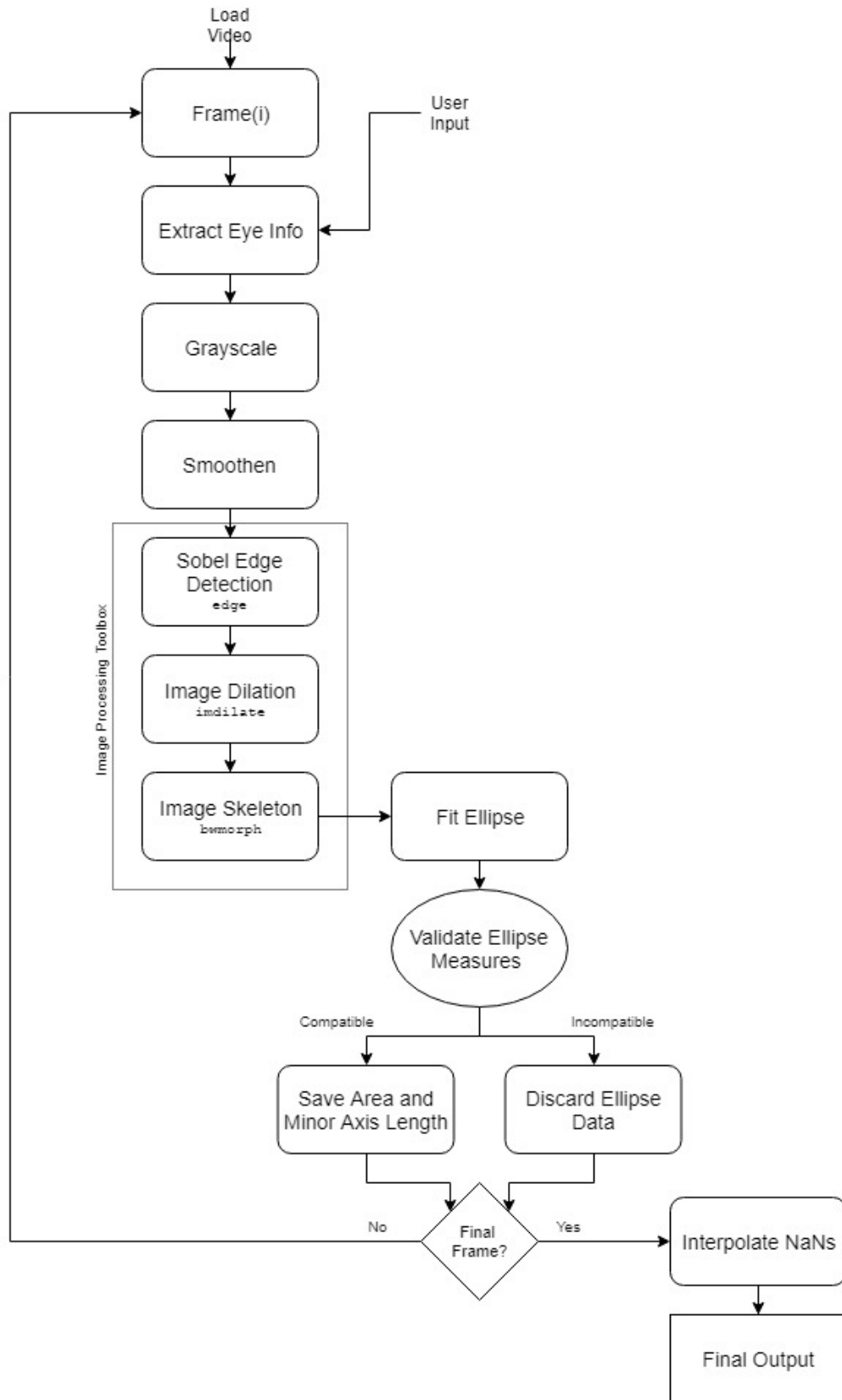


Fig. 8 - Edge-detection tracking algorithm flowchart.

Fig. 8 shows a flowchart for the method we developed for blink tracking with edge detection. It comprised the following steps:

1. Define the eye location, size and orientation from the first frame (Fig. 9A).
2. Pre-process the image to:
 - a. Convert it from RGB to grayscale using `rgb2gray` (Fig. 9B).
 - b. Filter with a Gaussian kernel with $\sigma = 6$, selected after several trials using different values (Fig. 9C).
3. Apply Sobel edge detection, using the `edge` function from the MATLAB Image Processing Toolbox (Fig. 9D).
4. Dilate the image to connect nearby disconnected edges (non-continuous lines), using `imdilate` function (Fig. 9E).
5. Extract the image skeleton by removing pixels on the boundaries of objects, using `bwmorph` function (Fig. 9F).
6. Fit an ellipse to the resulting skeleton around the original eye location and calculate the following parameters:
 - a. Minor axis length.
 - b. Ellipse area.
 - c. Ellipse angle.
7. Validate that the detected ellipse matches the eye by testing:
 - a. Ellipse angle must be within 10 degrees from the original eye angle extracted in the first frame.
 - b. Area is no larger than 120% of the original eye area.
8. If validation fails, set the output value is to null (NaN), go to the next frame.

9. If validation succeeds save the minor axis length as output for the current frame.
10. Move to the next frame and repeat from step 2.
11. After all frames have been processed, interpolate across all nulls using MATLAB's `interp1` function, to close gaps due to skipped frames.

Fig. 9 illustrates the key steps of this process.

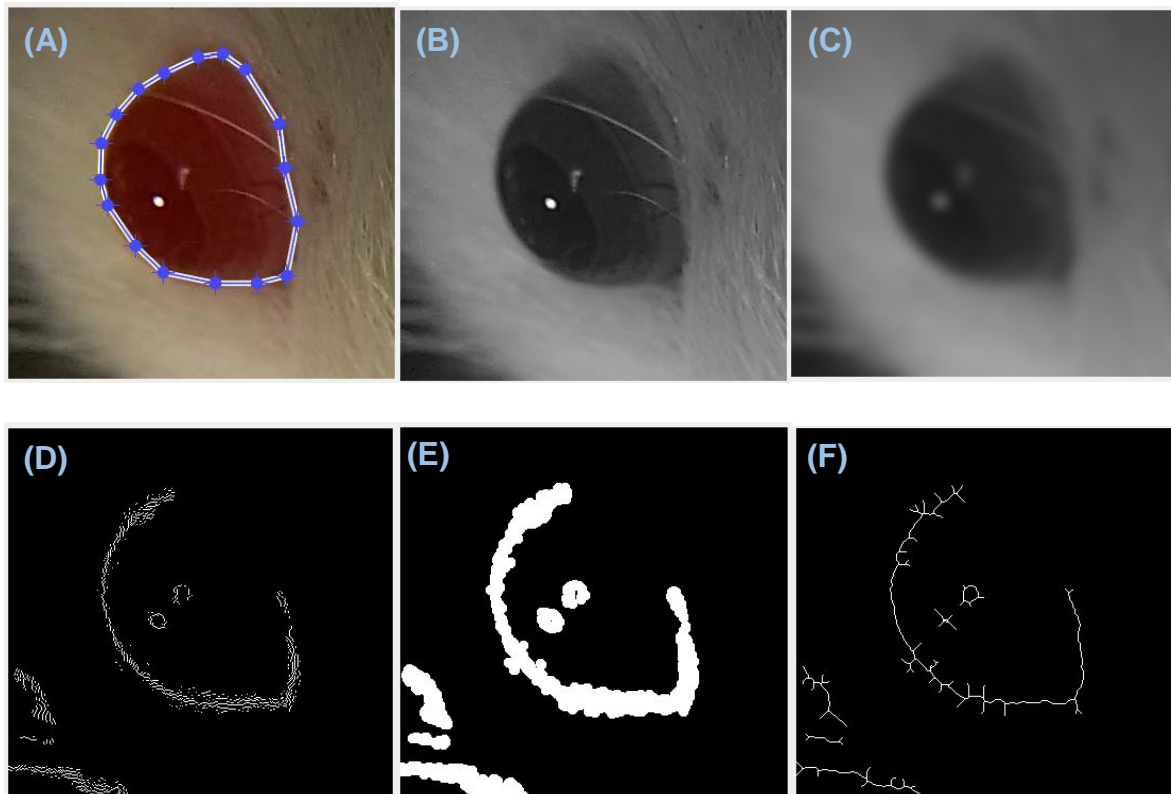


Fig. 9 - Edge tracking, key steps. A) User selects eye area for the first frame B) Frame converted to grayscale; C) Frame is smoothed using a Gaussian filter; D) Edges are detected using the Sobel operator; E) The image is dilated to close gaps between edges; F) Image skeleton is extracted to define the edges.

4.4. Final Algorithm: The active-contour Method

The initial studies paved the way to the choice of the final algorithm. We learned that using the natural properties of the rat's appearance, such as fur and eye color, could assist in tracking the eye surface. In addition, we saw that eye edges change with eyelid movement; thus, edges can be valuable features to use when measuring eye closure. These takeaways, as well as methods we reviewed in chapter 3, led us to the active-contour (or snake) approach, similar to that suggested by [54].

4.4.1. Rationale

The active-contour method tracks contours in an image using an iterative process that minimizes an energy function based on features in the image and the contour. We used the implementation of MATLAB's image processing toolbox of this algorithm since it had several advantages for robust tracking of the eye in a video recording. In particular, it is not color dependent, its dynamic characteristics support tracking over multiple frames, and it is resilient to minor head movements that might occur [61].

4.4.2. Method Background

The active-contour method starts with an initial contour or snake, C_0 , defined by the user or the program, near a contour of interest, γ_0 , in the image. The position of C_0 can be described parametrically as:

$$\mathbf{v}(\mathbf{s}) = (\mathbf{x}(\mathbf{s}), \mathbf{y}(\mathbf{s})), \mathbf{s} \in [0, 1] \quad (1)$$

where $x(s)$ and $y(s)$ are coordinates along the contour. The snake's position is changed iteratively to minimize an energy function E_{snake} given by:

$$\mathbf{E}_{snake} = \int_0^1 [\mathbf{E}_{int}(\mathbf{v}(s)) + \mathbf{E}_{ext}(\mathbf{v}(s))] ds \quad (2)$$

where \mathbf{E}_{int} is an internal energy component defined by:

$$\mathbf{E}_{int} = \frac{1}{2} [\alpha |\mathbf{v}'(s)|^2 + \beta |\mathbf{v}''(s)|^2] \quad (3)$$

and \mathbf{E}_{ext} is an external energy component defined by:

$$\mathbf{E}_{ext} = -|\nabla I(\mathbf{x}, \mathbf{y})|^2 \quad (4)$$

We can use discrete approximation to find the derivatives [54]. For example:

$$\mathbf{v}''(s) = \mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1} \quad (5)$$

Equation (3) defines the internal energy, which imposes a smoothness constraint on the contour's curvature. The first derivative of the contour defines the distance between the snake's vertices while the second derivative defines the angles between vertices. The parameters α and β define the smoothness and sharpness of the contour. The relative values of α and β control the smoothness of the contour. As default, both parameters are equal and set to $\alpha = \beta = 1$.

Equation (4) defines the external energy component, which depends on the image gradient ∇I in each frame and will be minimized when the snake is aligned with pronounced image features, such as edges or dark lines.

We force the contour $\mathbf{v}(s)$, to be closed for each frame, by enforcing the condition:

$$\mathbf{v}(\mathbf{0}) = \mathbf{v}(\mathbf{1}), \quad \mathbf{s} \in [\mathbf{0}, \mathbf{1}] \quad (5)$$

The contour movement in a picture is determined by gradient-descent minimization, where each step the energy function is minimized further. The direction of movement is determined by the derivative of the energy function – the contour will move towards the steepest slope, leading to

the local minimum. The process stops when a local minimum is reached, or after a predefined number of iterations. Contour initialization is therefore very important – the contour will converge to a local minimum, and so the initial contour needs to be set in a relevant ROI.

This process is designed to work on a single picture, rather than a video. To enable this process in a video recording, we must make sure each frame produces a contour that is used as a reference contour for the following frame. The reference contour will change its size and location based on image features, to find a new local minimum in the next frame.

4.4.3. Problems and Solutions

While developing the method and testing it on our videos, we faced two significant problems. The first problem caused the contour to disappear when full blinks occurred, and the second problem led to the contour “leakage” from the relevant eye area. The following sections explain these problems and their solutions.

4.4.3.1. Full Blinks

As a full blink occurs, the contour of the eye becomes smaller and eventually disappears completely as Fig. 10 illustrates. This causes a problem for the active-contour method since there is then no contour to use as a reference in subsequent frames. A reference contour is essential for the algorithm to work, since it forms the basis for the updated contour at each step. Without a reference contour the algorithm cannot work.

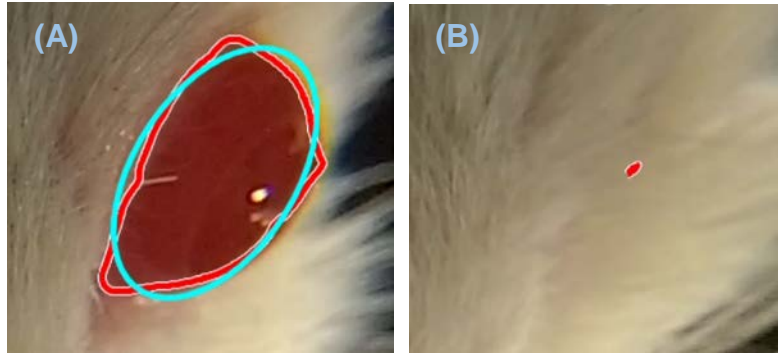


Fig. 10 - Full blink problem. A) A fully open eye, with the detected contour marked in red and the cyan ellipse in cyan. B) An almost fully closed eye, one frame prior to the contour disappearing. The red spot is the contour detected in this frame.

To resolve this, we introduced a procedure to identify when a full blink occurs. During the validation stage (step 9 in the algorithm described in section 4.4.4), the frame is tested to see if the minor axis length is less than two pixels. If so, we classify it as a full blink frame.

When a full blink is detected, the algorithm skips ahead 150 frames (with frame rate of 500fps, this is the duration of an average blink) to a frame where the eye will be fully, or almost fully open. The tracking algorithm is reset using the original contour as reference. Active-contour tracking is then run backwards towards the point of full-closure. Once it reaches the frame where the full blink was detected in the first place, it jumps again 150 frames forward (which will, in most cases, be sufficient for the eye to open to some extent), but this time, continues to run in the original direction, and the process continues as normal. If the eye remained shut for more than 150 frames, this solution might fail, however this is unlikely to happen, since rats' blinks are rare and short [62]. This solution prevented misclassifications of full blinks or sudden stops in the program when a full blink occurred.

4.4.3.2. Contour Leakage

Contour leakage occurs when the borders of the eye in the preprocessed image are blurry, preventing an edge from being detected. This results in the contour “leaking” outside of the eye area, as illustrated in Fig. 11A. This happened most often in rats who had been operated on to crush their facial nerve to mimic facial paralysis. In these surgeries, a portion of the fur under the eye was shaved to expose the skin for operation. Videos of these rats had lower separation between the eye and the background, as there was no fur between them (as visualized in Fig. 12). After preprocessing, this led to border blurring between the eye and background, and a minimal edge.

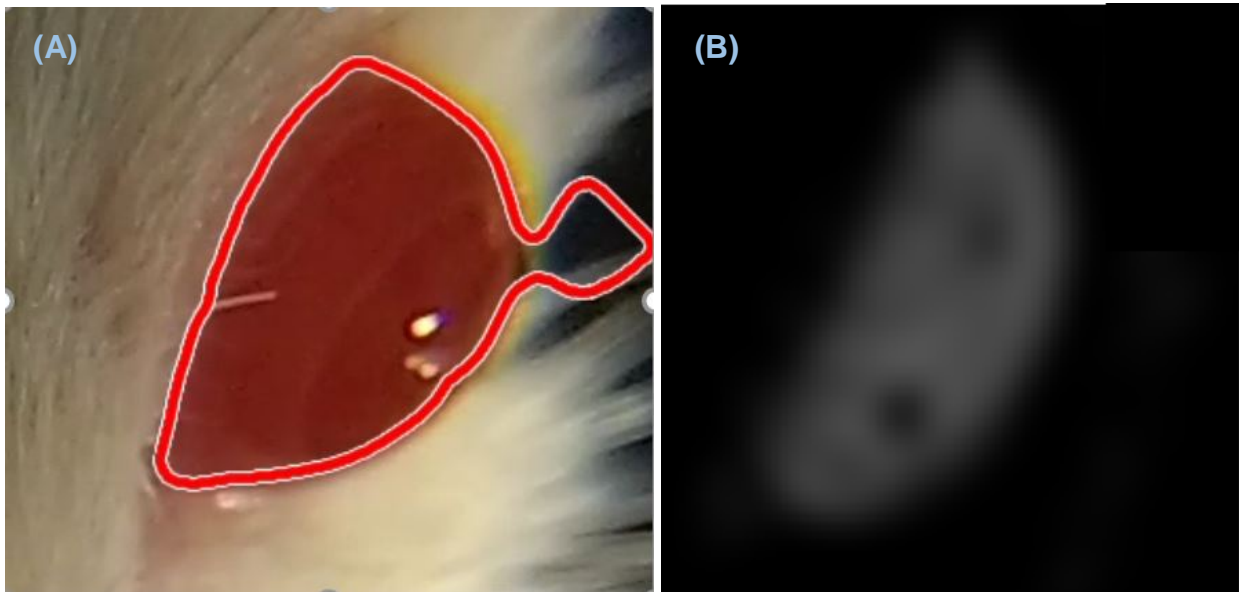


Fig. 11 - Contour leakage and post processing view. A) A contour leaking out of the eye to the background due to bad separation; B) Eye area following HSV threshold and grayscale.

To solve this issue, we used the observation that the eye’s brightness and saturation are considerably different from its fur surroundings. To take advantage of these differences, we converted the frame to HSV color space; we introduced an additional step of conversion to the HSV space and threshold of irrelevant pixels out. The threshold was set on the brightness and

saturation channels, as they distinguished eye from surrounding areas the best. Thresholding was done by picking only pixels that met both the brightness and saturation constraints. After conversion and threshold of pixels is complete, we can see that relevant pixels are within the eye area, rather than in the background (Fig. 11B).



Fig. 12 - Fur separation. On the right side of the frame (left eye), there is a clear separation between they eye area and the dark background. On the left side of the frame (right eye), the separation is not as good, and there is some overlap between the eye and the background.

4.4.4. Eye Tracking Algorithm

The flowchart (Fig. 13) shows the final algorithm developed to track eyelid movement and measure eye closure, using the active-contour method explained above.

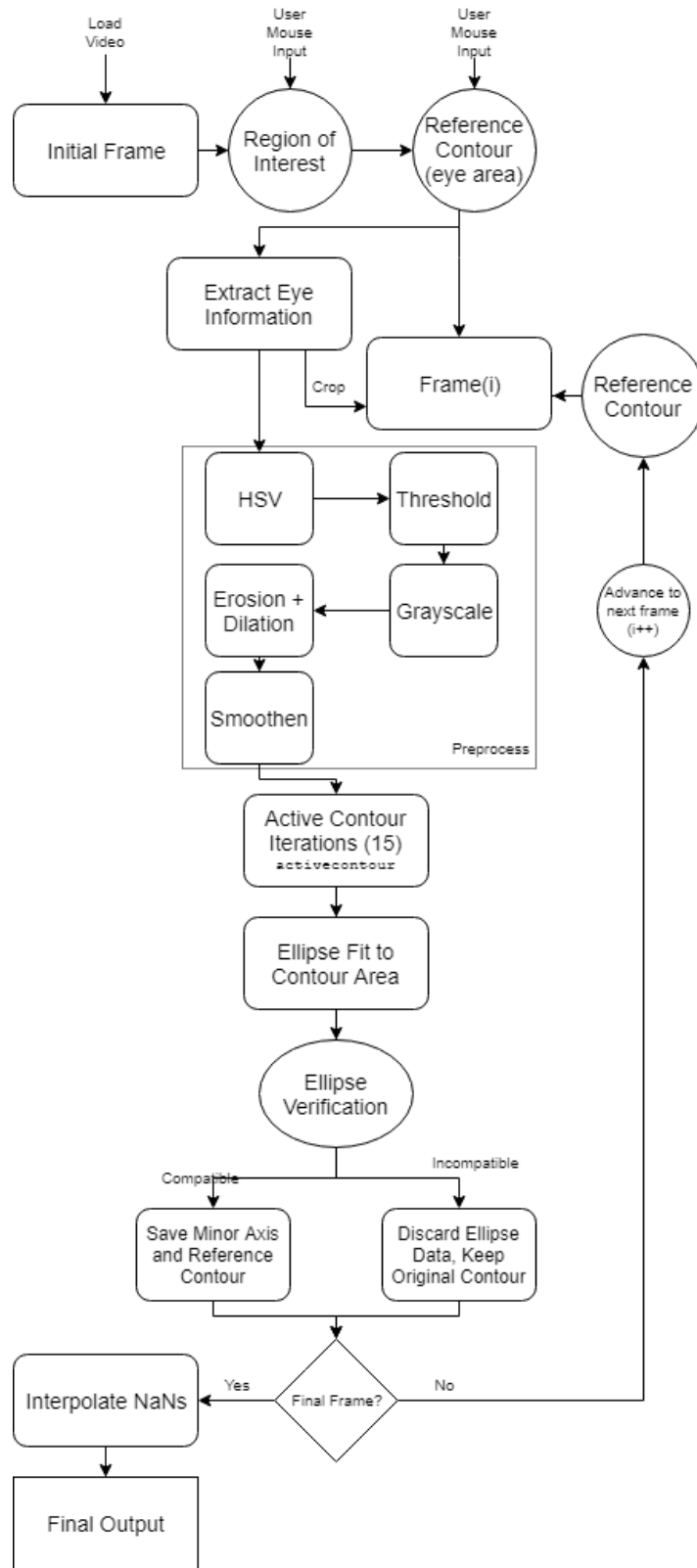


Fig. 13 - Active-contour tracking algorithm flowchart.

The algorithm can be broken down into the following steps:

1. Load the first frame of a video recording.
2. Select a rectangular ROI around the eye area (Fig. 14A).
3. Define an initial contour that around the eye border (Fig. 14B).
4. Extract the following information from the eye area:
 - a. Average and standard deviation (STD) of the hue, saturation and brightness (value) for all pixels within the eye area.
 - b. Eye area.
 - c. Angle between the eye and the horizontal axis.
 - d. Coordinates of the eye area center.
5. Load the next frame of the video and crop it to the ROI selected in the first frame.
6. Preprocess image to:
 - a. Convert from RGB to HSV color space (Fig. 14C).
 - b. Convert irrelevant pixels to black. Relevant pixels are those with:
 - i. Brightness values within two standard deviations from the average eye brightness.
 - ii. Saturation values greater than two standard deviation below average saturation (Fig. 14D).
 - c. Convert to grayscale (Fig. 14E).
 - d. Erode and dilate the frame using MATLAB's `bwmorph`, function to reduce noise from outliers and gaps (Fig. 14F-G).
 - e. Filter the image with a 2-dimensional, low-pass, Gaussian filter, with $\sigma = 6$ (Fig. 14H).

7. Apply the active-contour method (using `activecontour` function) to find the eye boundary (Fig. 14I):
 - a. Use the contour from previous frame as the initial contour.
 - b. Limit the number of iterations to 15.
8. Fit an ellipse to the contour detected around the eye. Extract the following information (Fig. 14J):
 - a. Area within the contour (representing the eye boundary).
 - b. Angle of the ellipse fitted to the eye.
 - c. Center point of the ellipse that fits the eye.
 - d. Minor axis of the ellipse that fits the eye.
9. Validate the output:
 - a. Ellipse angle must be within 10° from the original eye angle.
 - b. Ellipse center points must be inside a 50x50-pixel box centered around the original eye center point.
 - c. Contour area must be less than 120% of the original eye area.
 - d. Minor axis length must be larger than a single pixel.
10. If the validation fails, set the output value is set to NaN, go to step five, and use the contour from the first frame as the reference contour.
11. If validation succeeds, save the minor axis as the output. Go to step five.
12. After all frames have been processed, interpolate across all nulls using MATLAB's `interp1` function, to close gaps due to skipped frames.

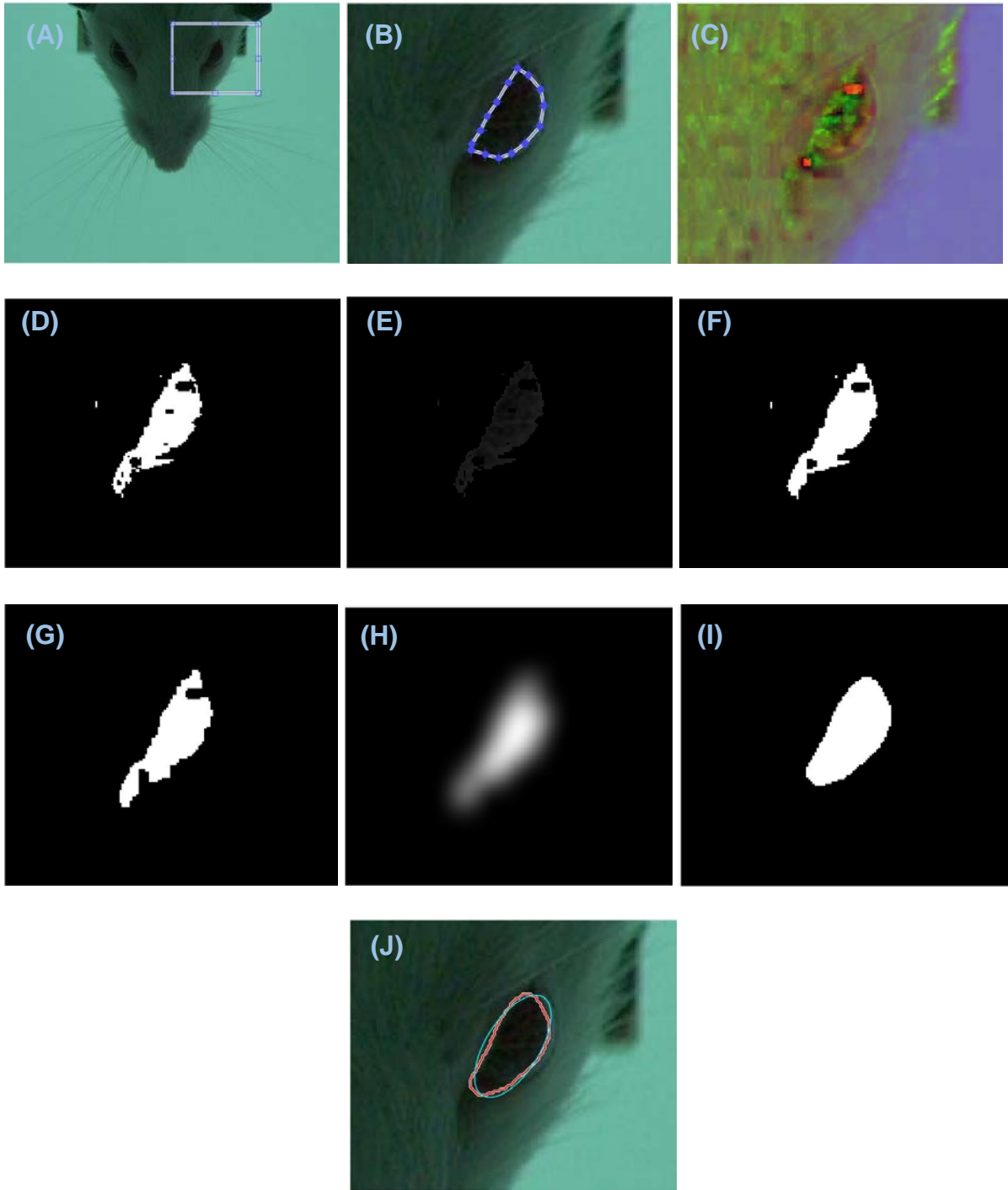


Fig. 14 - Key points in the algorithm to track eye closure using the active-contour method. A) User selects a ROI; B) User selects initial eye contour; C) Frame converted to HSV; D) Relevant pixels after threshold; E) Frame converted to grayscale; F) Image dilated; G) Image eroded; H) Smoothened frame after filtering; I) Following active-contour detection, this is the area within the contour; J) The frame, with the active-contour detected (in red), and the ellipse fitted to it (in cyan).

4.5. Validation

Output validation is an essential part of algorithm development, since we must confirm that the measures it produces are accurate. Such performance evaluation requires a data set accurately describing levels of eye closure throughout videos.

Since no automated method to track eye closure and blink detection was available, we developed a manual analysis method, that required a user to manually go through a video recording and mark the eye contour in a set of frames (as illustrated in Fig. 15).

The method worked as follows:

1. Load a frame and present it on-screen.
2. Present the original contour (that the user set when running the algorithm originally) around eye area, as polygon with its vertices (as seen in Fig. 15A).
3. Allow the user to use mouse pointer to change vertices locations, to match the eye contour (Fig. 15B).
4. Fit an ellipse to the contour.
5. Extract ellipse and contour information for current pixel:
 - a. Contour area (pixels)
 - b. Ellipse minor axis length (pixels)
6. Skip either one or five frames ahead, based on user keyboard input. The choice was made based on the stage of blink in a particular time in the video. In case the rat was not blinking, and the eyes were mostly static, five frames could be skipped at once; if, however, a blink was in progress, the user would skip one frame at a time, to capture the smallest changes in eye closure.

7. Repeat steps 3-6 until reaching the end of the video.
 8. Save all frames information into a .mat file to compare later with signals generated by our algorithm.
- algorithm.

Originally, videos were recorded at high frame rate (500fps) to allow capture of whisker movements. However, since this process was manual we first down-sampled the recording to 250fps (we used this rate in the pilot studies, and it was enough to capture blinks entirely), as this cut manual processing times by half.

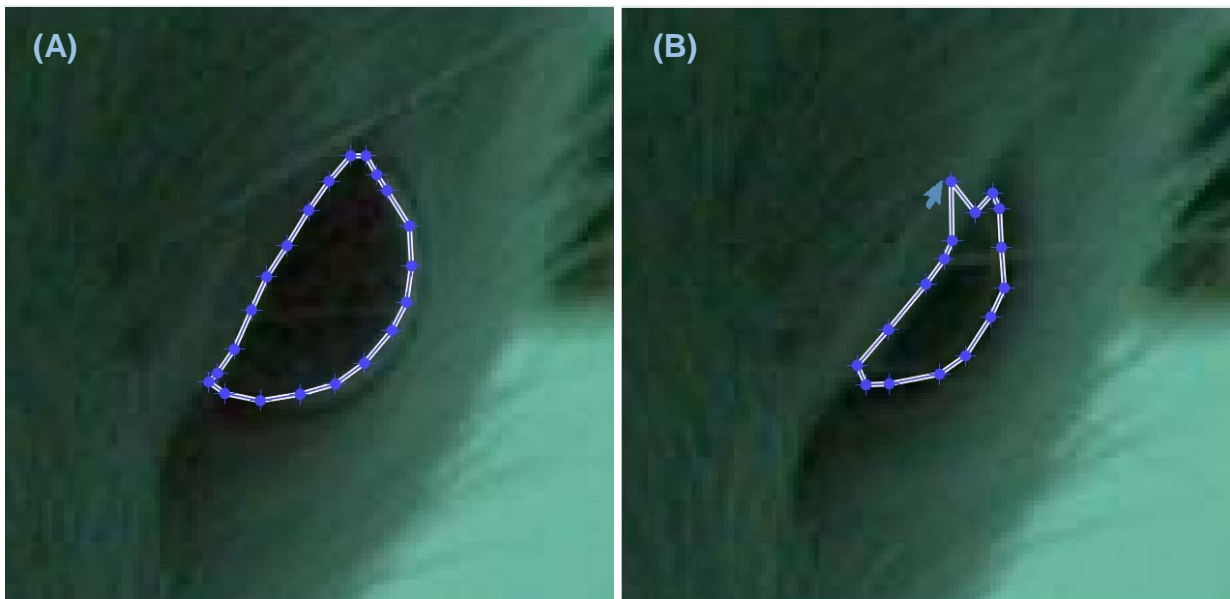


Fig. 15 - Manual validation steps. A) A random frame from the video, with the contour surrounding the eye, where the blue dots mark moveable points in the contour; B) The following frame from the same video, as a blink begins. As can be seen, the eye is smaller and the contour has changed, with one more moveable point to be changed by the user, using the mouse pointer (cyan).

Chapter 5: Results

5.1. Introduction

Chapter 4 explained the methods we used to track eye closure, in both pilot analysis studies and in the developed algorithm. We will now present the results of the different methods to demonstrate their performance and goodness-of-fit to the manually validated data and present the results for both experimental setups using the final, active-contour based, algorithm.

We used three measures to assess algorithm performance:

- Root mean square error (RMSE) was calculated to get the exact error for each video analyzed. It was calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_{out_i} - y_{val_i})^2}{N}}$$

Where y_{out_i} is the i^{th} element in the automated output signal, y_{val_i} is the i^{th} element in the manual validation signal and N is the number of elements in the signals.

- R^2 score – was calculated to estimate the goodness-of-fit of the algorithm output to the manually generated validation data:

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_{val_i} - y_{out_i})^2}{\sum_{i=1}^N (y_{val_i} - \overline{y_{val}})^2}$$

Where $\overline{y_{val}}$ is the mean of the validated data.

- Cross-correlation coefficient function (r_{ov}) was calculated as well for each video analyzed, to demonstrate the correlation between the output and validation data. We begin by finding R_{ov} , the cross-correlation function between validation data y_{val} and algorithm output y_{out} :

$$R_{ov}(\tau) = E[y_{val}(t)y_{out}(t + \tau)]$$

where τ is the lag, in seconds. We then calculate the cross-covariance function between the two:

$$C_{ov}(\tau) = R_{ov}(\tau) - \mu_o\mu_v$$

where μ_o and μ_v are the means of the output and validation data, respectively. Finally, we can find the cross-correlation coefficient function by:

$$r_{ov}(\tau) = \frac{C_{ov}(\tau)}{\sqrt{C_{oo}(0)C_{vv}(0)}}$$

where C_{oo} , C_{vv} are the auto covariance functions of the output signal and validation signal, respectively.

5.2. Pilot Studies

5.2.1. Red Pixel Detection

The red pixel detection method was applied to four short recordings from an injured rat, between 8 and 17 seconds long. For performance evaluation, only videos with blinks were considered, as we would like to measure blink detection rates.

Fig. 16 shows an example where the red-detection algorithm worked well. It is evident by inspection that the two signals were highly correlated and that blinks were captured successfully (Fig. 16A; blinks are marked with arrows across panels). Fig. 16B shows the residuals between the output and validation signal, measured in pixels. During opened-eye periods, these residuals were close to zero, but they increased somewhat during blinks, reaching up to 80 pixels difference. Fig. 16C shows the high correlation between the output and validation data. This is backed up by the R^2 score (0.935). The RMSE for this recording was 20 pixels.

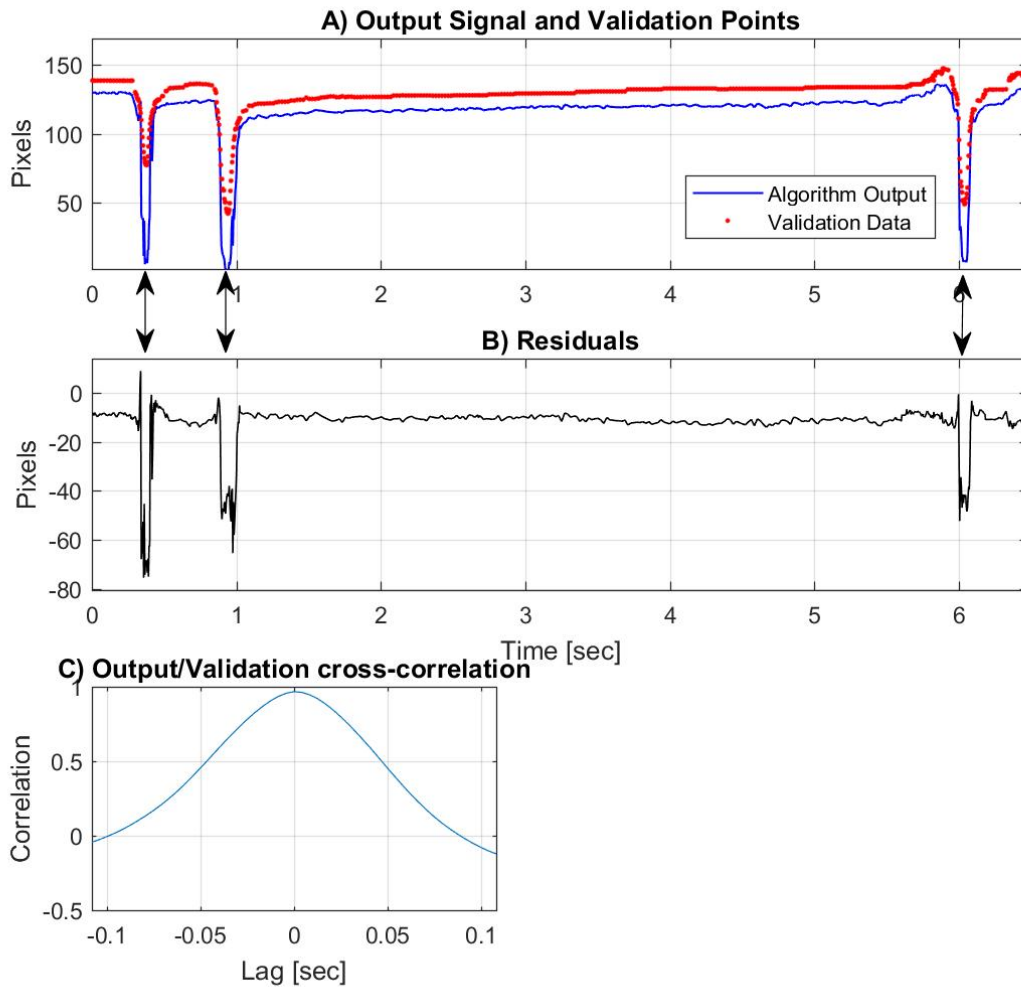


Fig. 16 - Red-Pixel Tracking Results (video recording YDXJ122, left eye). $R^2=0.935$; $RMSE=20$ pixels. Arrows between panels A and B represent blinks. A) The output of the red-pixel detection algorithm output superimposed on the manual validation signal. B) The residuals between the two signals; C) Cross correlation between the two signals.

However, the red-pixel algorithm did not work as well in all other videos. Fig. 17A shows an example where the algorithm's output was noisy and differed substantially from the validation signal. Fig. 17B shows the residuals in this case are larger than before and persist throughout the video. Fig. 17C supports this observation as well, showing lower correlation between the two

signals. Fig. 18 shows a snapshot of the erroneous red-pixel detection from the same recording. As seen in the snapshot showing the processed image (Fig. 18B), most of the red pixels of the eye were not captured and the ellipse was therefore only fitted to the main bulk of pixels that was detected. This caused a misidentification of the ellipse and its minor axis.

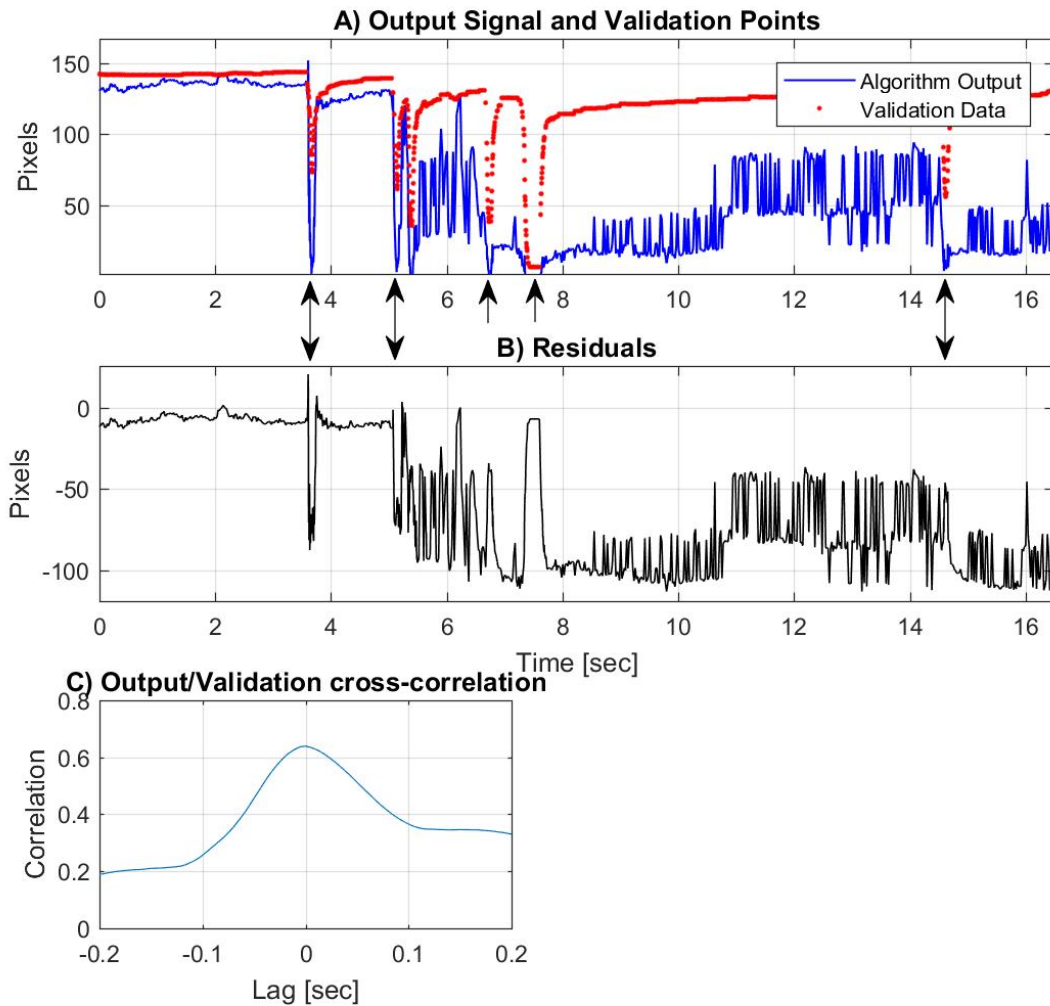
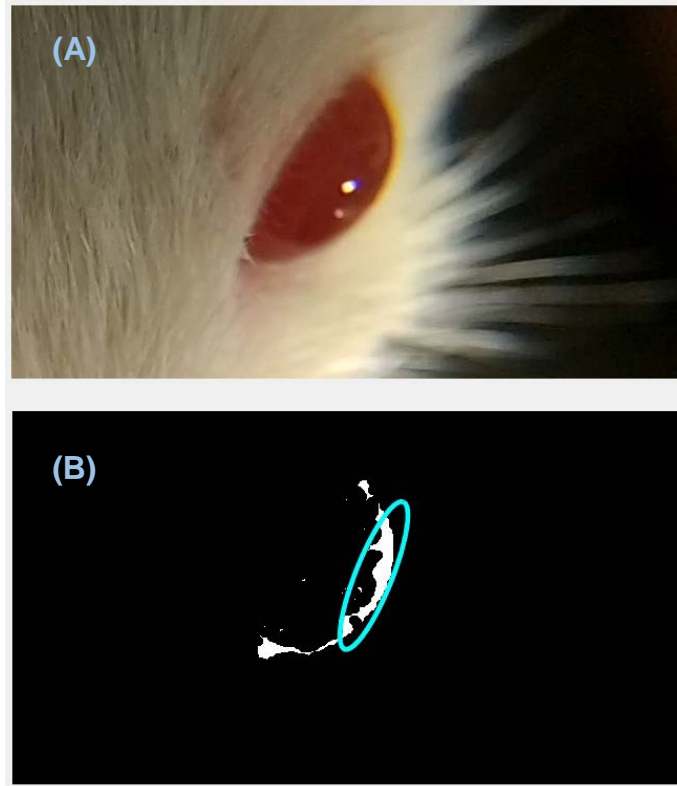


Fig. 17 - Red-Pixel Tracking Results (video recording YDXJ121, left eye). $R^2=0.408$; $RMSE=69.2$ pixels. A) The output of the red-pixel detection algorithm output superimposed on the manual validation signal. B) The residuals between the two signals; C) Cross correlation between the two signals.

Fig. 18 - Eye misclassification in red pixel tracking.

A) Typical raw frame

B) Processed frame following steps to focus only on red pixels. Clearly, most of the red pixels of the eye were disregarded.



We observed minor head movements in the video presented above (Fig. 17) at a specific point in time (after approximately 7 seconds). We hypothesized that the cause for misclassifications was the head movements, since these can change the angle of reflected light and alter the tones of red in the eye. Fig. 19 shows the eye's ROI's red channel histogram and the significant difference in its location prior to the movement (marked in red), and after (marked in black). Due to this change, pixels that passed the threshold as red pixels before the rodent moved its head, may have not counted as red later on, thus creating a wrong measure of the eye area and affecting the minor axis length and overall output of the method.

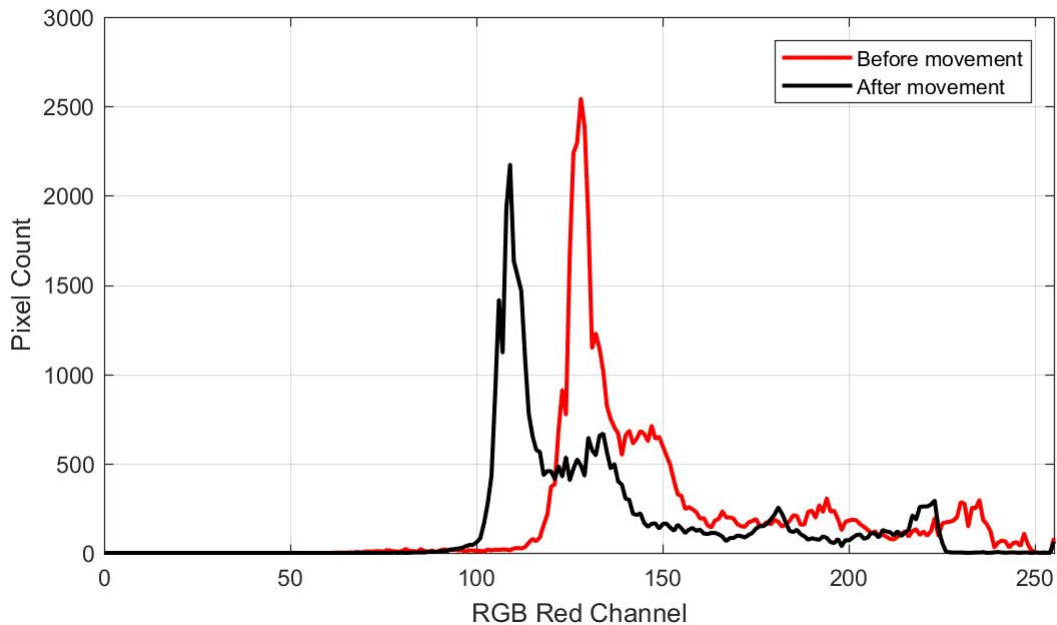


Fig. 19 - Red channel histogram of left eye region of interest, recording YDXJ121. Red histogram indicates pixel count before recorded head movement; Black histogram indicates pixel count after head movement.

5.2.2. Edge detection

Tracking using edge detection was applied to the same four blink recordings. Edge detection by itself was less accurate and noisier than the red-pixel results.

First, full edges could not be detected in all videos. Fig. 20 illustrates a case where fully closed contour could not be generated, due to blurry edges and low gradient of pixel intensity. When the contour between the eye and its surrounding area was undefined, the edge could not be detected (Fig. 20C), which led to wrong output data, because the ellipse is fit accurately to fully closed lines. In this case, ellipses were fitted to parts of the contour, as seen in Fig. 21.

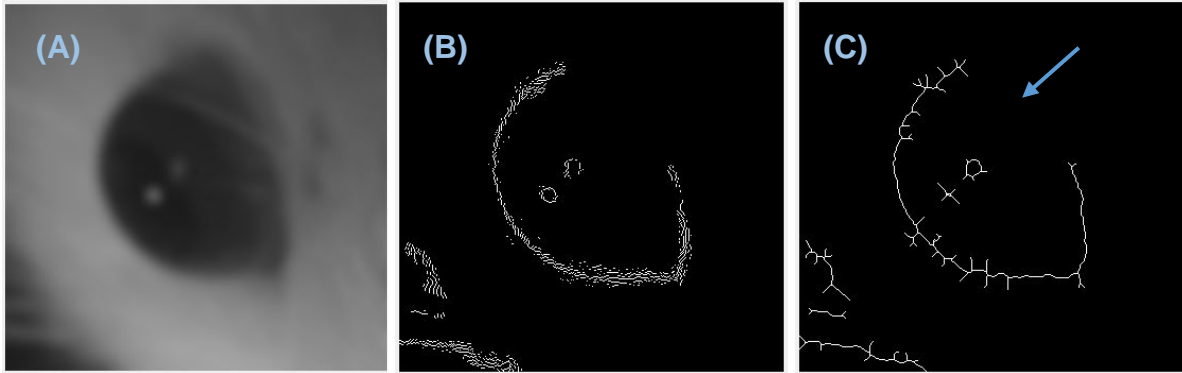


Fig. 20 - Incomplete edges. A) A typical frame where the border between the top of the eye and the fur is blurry and poorly defined; B) Edges detected, prior to any processing; C) Final edges, after applying the edge-detection algorithm. Note the large gap in the edge indicated by the arrow.

Fig. 22 compares the edge-detection algorithm output with the validation signal; clear differences are evident. The edge-detection output signal is very noisy. The noise was introduced when the edge around the eye was not clearly defined, as we see in Fig. 21. Two main problems repeated: First, gaps in the detected edge led to wrong ellipse fit by the program (as seen in Fig. 21A, where the ellipse detected by the algorithm is smaller than the actual eye, due to open edge). Second, fur around the eye led to larger ellipses detected by the program (as seen in Fig. 21B, where the ellipse is larger than the eye contour due to proximity to whiskers and fur).

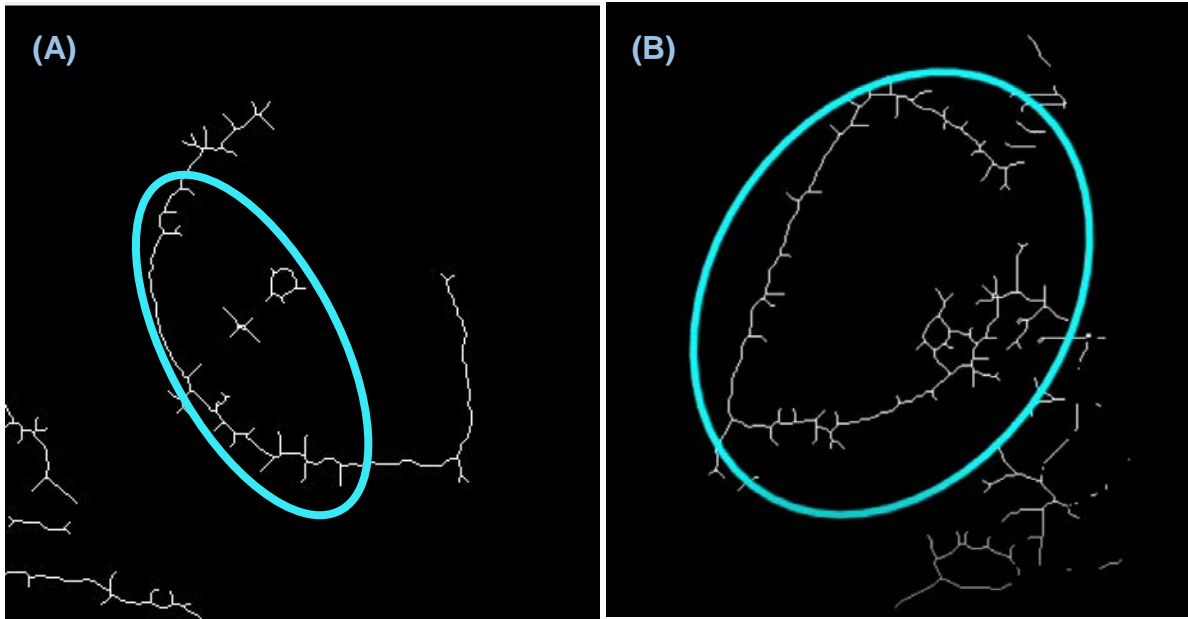


Fig. 21 - Misfitted ellipses due to false edge detection. A) An open edge due to blurry border, and an ellipse incorrectly fitted to a smaller area of the eye leading to wrong measure of minor axis length; B) Incorrect fitting of the ellipse due to irrelevant edges from rat whiskers and fur.

Blinks were clearly evident in the output signal (Fig. 22A) despite the mismatch and low correlation between the output and validation (Fig. 22C), indicating that this algorithm could be used for blink detection but not for eyelid tracking, if a low-pass filter were to be used to remove high frequency changes.

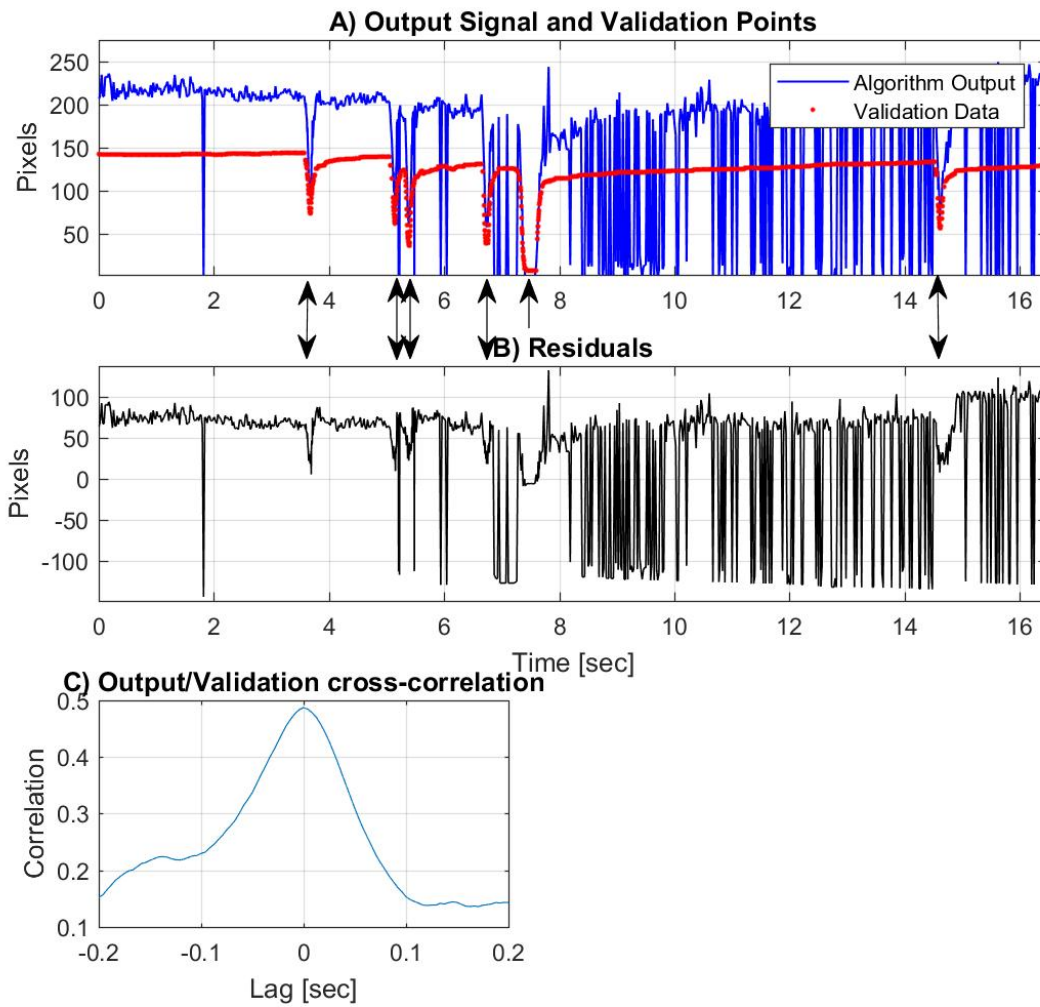


Fig. 22 - Edge-detection tracking results (video recording YDXJ121, left eye). R^2 score was 0.237 and RMSE was 77.4 pixels. Blinks are marked with arrows between panels A and B. A) Edge-detection algorithm output and manual validation data superimposed; B) Residuals between the two signals; C) Cross-correlation between the two signals.

5.3. Active-contour Algorithm

The active-contour method was applied to two sets of videos. The first set comprised six videos obtained with experimental setup A, and had high levels of eye illumination. The second set of videos (five recordings), obtained with the revised experimental setup designed to support accurate whisker tracking (setup B), had higher contrast and higher resolution, but lower eye illumination. In total, the algorithm was run on fourteen eyes from seven recordings. Only eleven were considered, as the other three did not include blinks and so could not be accurately validated. A clear breakdown of the recorded eyes and video files is provided in Table 1 and Table 2.

Fig. 23A shows the output of the active-contour algorithm applied to a 35-second video recording using experimental setup A. It is evident that the two signals are very similar, and all blinks were detected by the algorithm. Fig. 23B shows that the residuals between the two signals were close to zero during open phases but increased somewhat during blinks (marked between panels A and B with arrows). Fig. 23C demonstrates the high correlation between the two signals. The RMSE for this recording was 2.9 pixels, with $R^2=0.967$.

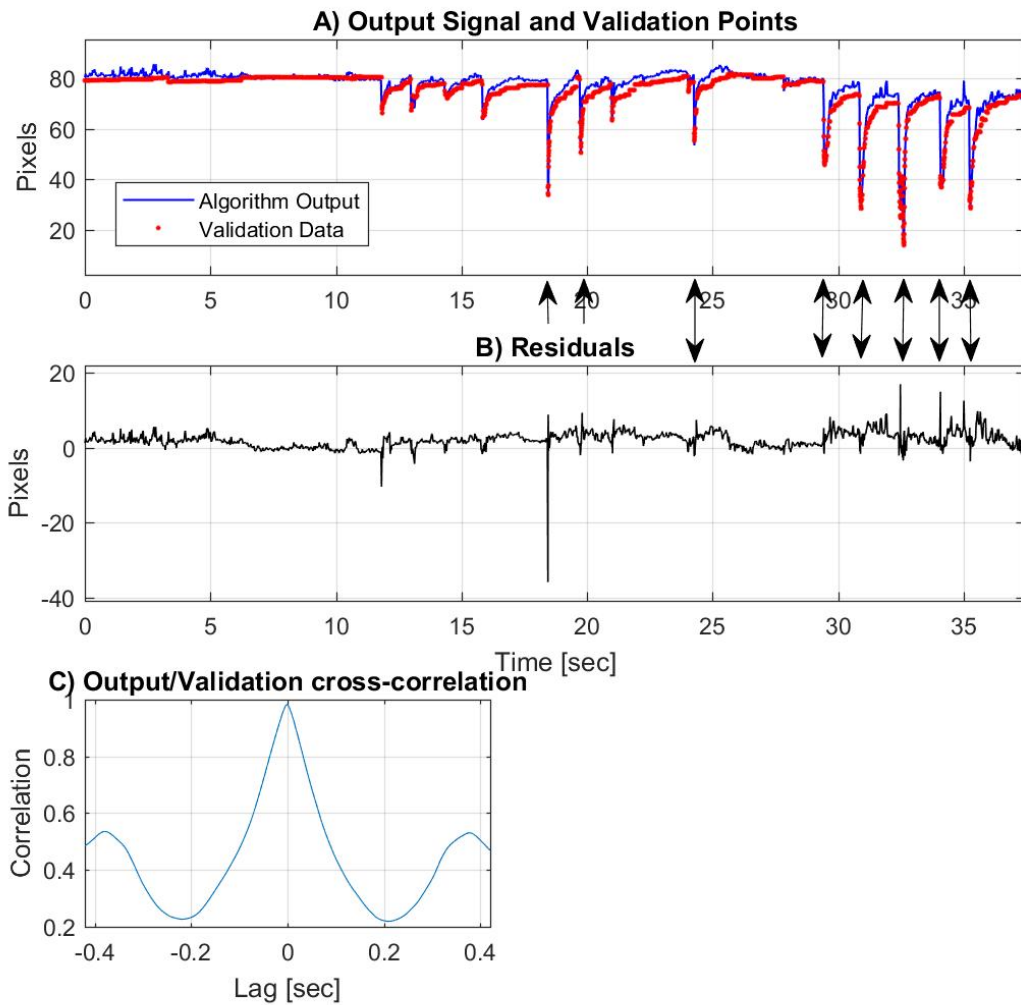


Fig. 23 - Active-contour-based tracking (video recording 6400t1, right eye, setup A). $R^2=0.967$; RMSE=2.9 pixels. A) Automatic algorithm output and manual validation data superimposed; B) Residuals; C) Cross-correlation between the signals.

Fig. 24 shows the result of applying the active-contour method to a video acquired with experimental setup B. Here, we see lower correlation between the two signals (with $R^2=0.895$), as the eye is harder to differentiate from its surroundings, since the images obtained are darker due to the lack of front lighting (see section 4.2). This means that borders between the eye and the fur

are not as clear as in the original experimental setting. However, all blinks are accounted for, and the RMSE remained low at 2.7 pixels.

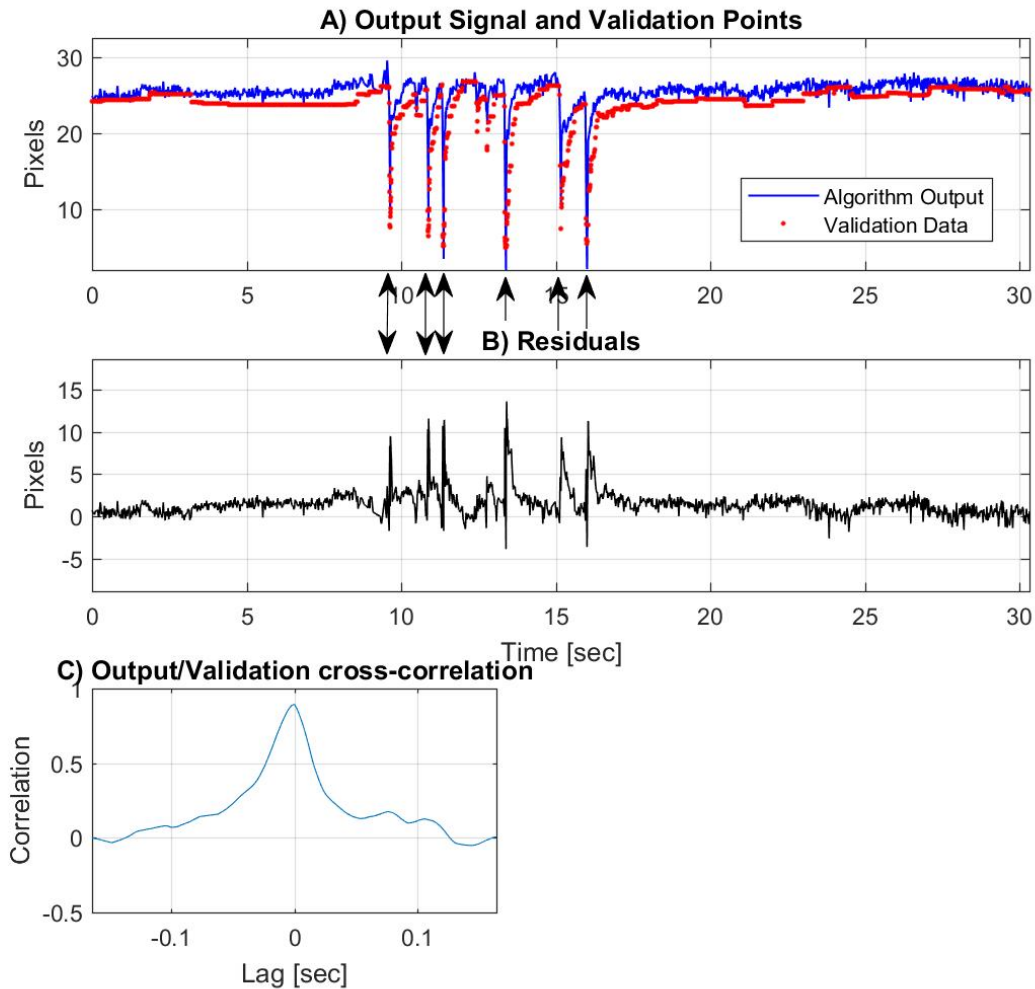


Fig. 24 - Active-contour-based tracking (video recording AN3v2, right eye, setup B). $R^2=0.895$; RMSE=2.7 pixels. A) Algorithm output with manual validation data superimposed; B) Residuals between the two; C) Cross-correlation between the two.

Table 1 summarizes the results of the active-contour method for all videos analyzed using setup A. These yielded very accurate results, with mean R^2 score of 0.981 ± 0.011 , mean RMSE

4.62±2.20 pixels, and all blinks successfully detected. In comparison, results for videos from experimental setup B (summarized in Table 2) demonstrated a lower R^2 score of **0.845±0.055** and mean RMSE of **3.65±1.71** pixels and all blinks were successfully detected as well.

Video File	Experimental Setup	Duration (sec)	Actual Blinks	Detected Blinks	R² Score	RMSE (pixels)
YDXJ121L	Setup A	16.5	6	6	0.995	8.04
YDXJ122L	Setup A	7	3	3	0.995	3.58
6400t1L	Setup A	16	4	4	0.979	7.38
6400t1R	Setup A	38	13	13	0.967	2.95
6400t2L	Setup A	20	12	12	0.968	2.92
6400t2R	Setup A	20	12	12	0.985	2.85
				Mean:	0.981	4.62
				STD:	0.011	2.20

Table 1 - Results for all videos analyzed using the active-contour method in experimental setup A.

Video File	Experimental Setup	Duration (sec)	Actual Blinks	Detected Blinks	R² Score	RMSE (pixels)
AN1V1BR	Setup B	54	4	4	0.878	2.44
AN1V3m8L	Setup B	60	3	3	0.788	7.03
AN1V3m8R	Setup B	50	1	1	0.783	3.08
AN3V2R	Setup B	30	6	6	0.895	2.74
AN3V2L	Setup B	30	6	6	0.884	2.94
				Mean:	0.845	3.65
				STD:	0.055	1.71

Table 2 - Results for all videos analyzed using the active-contour method in experimental setup B.

5.4. Eye Area vs. Minor Axis Length

Eye area and minor axis length were extracted from all recordings as possible eye closure measures. We calculated for each recording the mean relative error (MRE), for both minor axis output and eye area output. Calculation was done as follows:

$$MRE = \frac{1}{N} \sum_{i=1}^N \frac{|y_{out_i} - y_{val_i}|}{y_{val_i}}$$

where N was the total number of frames, y_{out_i} was the algorithm output (minor axis length or eye area) in a given frame i , and y_{val_i} was the manually validated output (minor axis length or eye area) in the same frame.

As seen in Table 3, the MRE for eye area as output was three times higher than the minor axis MRE. This means that the minor axis was closer to the manual output across recordings, thus we chose it as the preferred output measure of our algorithm.

Several factors might lead to the advantage of minor axis over eye area. First, to maintain a reasonable runtime, the algorithm is limited to 15 iterations in each frame, that was determined empirically after comparing with both higher and lower number of iterations. Increasing the runtime to MATLAB's default 100 iterations or above will significantly slow down the process as all calculations reoccur each iteration. This means that the contour might not fully converge to the borders of the eye and include small areas outside of the actual eye area, which will lead to a miscalculated area measurement. Secondly, the borders of the eye are sometimes blurred, and the contour may have trouble finding the exact edge, which will also cause misidentification of eye area. These two factors might affect minor axis measurements as well but the main difference is

that these errors are limited to the minor axis direction only, while in area measurement, any wrong addition to the contour by the algorithm, in any direction, will increase the error in measurement.

Video File	Experimental Setup	Mean Relative Error, Minor Axis	Mean Relative Error, Area
YDXJ121L	Setup A	0.063±0.107	0.056±0.193
YDXJ122L	Setup A	0.002±0.057	0.023±0.103
6400t1L	Setup A	0.061±0.483	0.084±0.516
6400t1R	Setup A	0.029±0.040	0.003±0.113
6400t2L	Setup A	0.035±0.061	0.037±0.072
6400t2R	Setup A	0.041±0.045	0.073±0.063
AN1V1BR	Setup B	0.134±0.354	0.440±0.681
AN1V3m8R	Setup B	0.078±0.284	0.370±0.360
AN1V3m8L	Setup B	0.367±0.315	1.105±0.663
AN3V2R	Setup B	0.114±0.250	0.410±0.330
AN3V2L	Setup B	0.096±0.375	0.386±0.411
	Mean:	0.093	0.271
	STD:	0.099	0.328

Table 3 - Results table showing the mean relative errors for both minor axis and area outputs, for both experimental setups.

Chapter 6: Discussion and Conclusions

6.1. Discussion

6.1.1. Summary

The main goal of this study was to develop a method to quantify rodent eyelid movements from video recordings. We explored algorithms based on red-pixel detection and edge detection, which led to the development of a final algorithm that used an active-contour approach.

We learned that using a simple edge detection approach on an entire frame might introduce noise from irrelevant edges, such as whiskers and fur, so we decided to focus edge detection on a smaller, more relevant ROI around the eye area. This helped us arrive at the conclusion that active-contour could be a possible direction for development, since it focused the effort on the border of the eye and its surroundings. From the red-pixel algorithm, we found that tracking the eye based on color only is ineffective, as recording light may vary and affect the tones of red in the eye, and animal colors vary from one to another. However, we saw that conversion to HSV color space could be very useful in helping separate the eye from its surroundings. This came in handy when we encountered the leakage problem during development of the final algorithm, as we managed to overcome the issue thanks to conversion to HSV and separation based on brightness levels.

While developing the algorithm using the active-contour approach, we faced several problems; the main two were contour leakage and loss of reference contour in full blinks. We managed to overcome the first problem by introducing a concept we used in the red-pixel approach – converting the color space to HSV and setting a threshold to remove irrelevant pixels, only this time we used brightness and saturation values. In addition, we solved the full blinks issue, by introducing a skip-forward approach, where tracking was stopped in case a full blink was detected,

the time marker skipped forward 150 frames and the algorithm was run backwards until reaching the point of full closure.

Our final eyelid-tracking algorithm overcame the initial approaches problems (noisy edges and red-tone inaccuracies) and proved to work with a 100% blink detection rate and high rates of goodness of fit to the actual measures. For setup A, the algorithm presented an average R^2 score of 0.981, and for setup B an average R^2 score of 0.845. Using this method allowed detecting partial blinks as well as full ones, and provided an accurate measure of eye closure, that will be useful in assessing nerve regeneration at a later stage.

Now that we have managed to quantify eye closure, an important piece of the project is completed. The algorithm described in this document will be used to map different EMG patterns from healthy facial EMG into corresponding eyelid movement. In turn, this will be used to understand the required functional electrical stimulation that is required to induce similar movements in the injured side, thus creating the desired symmetrical facial movement.

In its current form, I believe the algorithm is ready to be used by our group. The code was written in MATLAB and is fully documented and explained, in case any parameters will require change (due to changes in recording setup or possible new setups). It proved to be robust enough to work in both setups we used and demonstrated 100% blink detection rate and high correlation to the validation data.

6.1.2. Experimental Setup

Another goal of the project was to create a system that can be easily implemented and not require extensive setup procedures. The experimental systems described in chapter four cover two different setups, one simpler to implement (setup A) but the other capable of recording whisking

as well as blinking. Both setups are easily reproducible in other lab environments that require head fixation, since they require very little equipment – either a camera alone or a camera with a simple chamber and back light - and can be modified depending on research needs without altering the algorithm code.

An important success of the new experimental setup (setup B) is that it can be used for both whisker tracking and eyelid tracking, without any modifications, and with a single recording session. The camera and recording chamber described in detail in chapter 4 were crafted in a trial-and-error process, each trial trying to improve the captured video, so that whiskers can be captured in detail and processed offline, as well as the eyes. Ensuring that both elements of the rodent’s face were captured in sufficient quality differentiates this recording system from any other currently published, as it allows analysis of both whisker movement and eyelid movement from a single video recording, acquired from a single camera. This simplifies an existing process that requires laser measurements and infrared eyelid sensors, as presented in chapter 2 [17]. Our colleagues at MEEI are refining an existing algorithm for whisker tracking to fit to our experimental system, and once this is complete, both eyelid and whisker tracking could be combined into a single program.

6.1.3. Limitations

In its current form, the algorithm still has several limitations:

- **Limited validation set:** So far, a single tester manually validated all videos, as there was no ready-made automatic method to perform validation. Since eye borders are not clearly defined, manual validation could be biased to a single person’s opinion on where borders lie. This means that our performance estimations rely on a single opinion, and high goodness-of-fit errors could represent either a problem in the algorithm workflow or a

problem with the manual process. Since there is only manual data from one tester, we do not have high certainty if the errors that do occur are due to algorithmic or human errors.

- **Analysis speed:** Processing times are currently very long. On average, runtime is 17.6 seconds per 1 second of video. This issue is not crucial right now, since this algorithm is meant to work in an offline environment in the context of our study, and processing time is not the top priority. However, if we would like to have this algorithm used by other teams in different research contexts, we must reduce processing time significantly.
- **Not fully automatic:** Currently, user input is required at two points in the process: picking the ROI around the relevant eye and setting an initial contour circling the eye. While being fully automatic was not one of the requirements of the project, we believe that an automated algorithm has multiple benefits. It can reduce waiting time between steps, as no human interaction will be required, and the process will run on its own. In addition, it will simplify the process significantly, and require only loading a video and waiting for the processing to complete. Consequently, we believe it will be more appealing to other research groups to use this algorithm.
- **Mandatory head fixation:** As mentioned above, both experimental settings are reproducible and can be implemented by other researchers without much effort. Both settings, however, require the animal's head to be fixed in place and implanting the titanium device requires surgical knowledge and expertise. This might limit the method's use to research facilities that can perform such surgery on-site.

6.1.4. Future Work

We must take into account the challenges and whether the algorithm will be used by our group only or also by other research groups, when considering any future work. Some of the future directions of work include the following:

- **Adding human testers for manual validation:** As discussed in the previous section, using a single human tester to validate all videos can be problematic. Introducing additional testers will allow comparison of error rates of different testers. This will help understanding if performance errors are due to problems in the algorithm or in human validation errors.
- **Adding videos for improved performance:** Development of the algorithm was done using a limited number of videos. This provides a proof of principle as the algorithm worked well using the settings provided in our recordings and will behave the same as long as no significant changes are introduced to setup A or setup B, described in chapter 4. However, I believe we have not used enough videos to ensure performance in extreme situations. As we saw earlier, when we introduced new videos, we came across new challenges we had to face, such as contour leakage in shaved animals, or loss of contour in cases of full blinks. Using eleven recordings up until now, it is likely that we have not yet seen all extreme cases. Recording new videos will have to take place in our colleague's lab in MEEI, as that is where the experiment setup is located, together with the animals.
- **Reducing run time:** A first step in improving runtime will be to perform a bottleneck analysis on our code and detect any pain points that drain most of the time. These could then be modified or optimized to reduce lag significantly. Another common method of reducing runtimes in time-consuming algorithms is using parallel computing. Parallel computing breaks down a problem into parts that can be solved concurrently, and uses a

different processor to solve each part, before combining all parts to achieve the final output. Since our method has frame-by-frame dependencies, we cannot implement the algorithm as-is using parallel computing. A possible use of parallel computing would require reprogramming the algorithm to run backwards as well as forward: the video can be processed normally up to a given point halfway through the video, while the rest of the video can be processed from the end to the same halfway point. The process could be done on more than two processors by splitting the video into quarters, eighths or more, depending on the machine.

- **Fully automating the algorithm:** To achieve full automation of the algorithm, we must eliminate the need of user input at the beginning of the process. The ROI can be automatically cropped for the preset area containing the eye in the recording chamber. If the recording setup is final and does not change, this preset area will remain constant between all videos. The initial eye contour could be predefined as well, to an ellipse that is significantly smaller than the eye area and that will expand upon the first iteration of the algorithm, as it will look for the nearest border. By removing these two steps from the workflow and assuming we maintain the same recording apparatus as described in chapter 4, we can eliminate these user-dependent steps.
- **Enabling recording without head fixation:** A possible future version of our algorithm may include eye recognition in a video frame, such that it could process videos recorded while the rodent is free to walk around the cage without any head fixation. This would eliminate the need of a head fixation device and pre-recording surgical procedure, and permit a more natural behavior, ideal for behavioral studies. This work will require

additional resources to develop animal tracking and eye detection algorithms, or use existing work to build upon [63, 64].

- **Introducing graphical user interface:** To run the program in its current form, basic MATLAB understanding is required since the code must run using the MATLAB Editor and Command Window. However, if we wish to make it more accessible to external users that are not part of the current research group, we should consider incorporating a front-end interface. This will allow distribution of our program with non-MATLAB users who seek a simple, ready-made solution for eyelid tracking. Such interface can include an option to load a video recording, select the ROI and initial contour, process the tracking and present the output in same window.

6.2. Conclusions

This thesis has demonstrated that we met our early requirements and achieved eyelid tracking in a non-invasive, easy-to-set-up environment. We built an experimental environment that uses a single camera to record both eyelid and whisker movement, both in sufficient quality and resolution to be analyzed and tracked. We demonstrated that the output of the algorithm detects 100% of blinks in both experimental settings. The output was highly correlated to the manually generated data and presented minimal errors. We believe it is now ready to be used in experiments.

7. References:

1. Otto, R., et al., *Electrical restoration of the blink reflex in experimentally induced facial paralysis*. Vol. 65. 1986. 30-2, 37.
2. Marson, A.G. and R. Salinas, *Bell's palsy*. Western Journal of Medicine, 2000. **173**(4): p. 266-268.
3. Ropper, A.H., M.A. Samuels, and J.P. Klein, *Chapter 47. Diseases of the Cranial Nerves*, in *Adams and Victor's Principles of Neurology, 10e*. 2014, The McGraw-Hill Companies: New York, NY.
4. Coulson, S.E., et al., *Expression of Emotion and Quality of Life After Facial Nerve Paralysis*. Otolology & Neurotology, 2004. **25**(6).
5. Chan, J.Y.K. and P.J. Byrne, *Management of Facial Paralysis in the 21st Century*. Facial plast Surg, 2011. **27**(04): p. 346-357.
6. Dey, J.K., et al., *Seeing is believing: Objectively evaluating the impact of facial reanimation surgery on social perception*. The Laryngoscope, 2014. **124**(11): p. 2489-2497.
7. Lavy, J.A., et al., *Gold weight implants in the management of lagophthalmos in facial palsy*. Clinical Otolaryngology & Allied Sciences, 2004. **29**(3): p. 279-283.
8. Dobel, C., et al., *Emotionale Auswirkungen einer Fazialisparese*. Laryngo-Rhino-Otol, 2013. **92**(01): p. 9-23.
9. Fu, L., C. Bundy, and S.A. Sadiq, *Psychological distress in people with disfigurement from facial palsy*. Eye, 2011. **25**: p. 1322.
10. Jowett, N. and T.A. Hadlock, *A contemporary approach to facial reanimation*. JAMA Facial Plastic Surgery, 2015. **17**(4): p. 293-300.

11. Engström, M., et al., *Prednisolone and valaciclovir in Bell's palsy: a randomised, double-blind, placebo-controlled, multicentre trial*. *The Lancet Neurology*, 2008. **7**(11): p. 993-1000.
12. Frey, M., et al., *Dynamic Reconstruction of Eye Closure by Muscle Transposition or Functional Muscle Transplantation in Facial Palsy*. Vol. 114. 2004. 865-75.
13. Gillies, H.D. and D.R. Millard, *The principles and art of plastic surgery*. *The Principles and Art of Plastic Surgery*. 1957: Little, Brown.
14. Attiah, M.A., et al., *A Rodent Model of Dynamic Facial Reanimation Using Functional Electrical Stimulation*. *Frontiers in Neuroscience*, 2017. **11**: p. 193.
15. Popović, D.B., *Advances in functional electrical stimulation (FES)*. *Journal of Electromyography and Kinesiology*, 2014. **24**(6): p. 795-802.
16. Mattox, D.E. and H. Felix, *Surgical anatomy of the rat facial nerve*. *The American journal of otology*, 1987. **8**(1): p. 43-7.
17. Heaton, J.T., et al., *A system for studying facial nerve function in rats through simultaneous bilateral monitoring of eyelid and whisker movements*. *Journal of Neuroscience Methods*, 2008. **171**(2): p. 197-206.
18. Heaton, J.T., et al., *Rat whisker movement after facial nerve lesion: Evidence for autonomic contraction of skeletal muscle*. *Neuroscience*, 2014. **265**: p. 9-20.
19. Wood, M.D., et al., *Outcome measures of peripheral nerve regeneration*. *Annals of Anatomy - Anatomischer Anzeiger*, 2011. **193**(4): p. 321-333.
20. Pfister, B.J., et al., *Biomedical Engineering Strategies for Peripheral Nerve Repair: Surgical Applications, State of the Art, and Future Challenges*. 2011. **39**(2): p. 81-124.

21. de Medinaceli, L., W.J. Freed, and R.J. Wyatt, *An index of the functional condition of rat sciatic nerve based on measurements made from walking tracks*. *Experimental Neurology*, 1982. **77**(3): p. 634-643.
22. R Bain, J., S. E Mackinnon, and D. Hunter, *Functional Evaluation of Complete Sciatic, Peroneal, and Posterior Tibial Nerve Lesions in the Rat*. Vol. 83. 1989. 129-38.
23. Haidarliu, S., et al., *Muscle Architecture in the Mystacial Pad of the Rat*. *The Anatomical Record*, 2010. **293**(7): p. 1192-1206.
24. Pearson Education, i.p.a.B.C. *Anatomy of eyelid upper structure diagram*. [Web] 2006 [cited 2018 27-08]; Available from: <http://www.lifeinharmony.me/anatomy-of-eyelid/anatomy-of-eyelid-upper-structure-diagram-and-periocular-plasti-on>.
25. Tortora, G.J. and B. Derrickson, *Principles of anatomy & physiology*. 13th ed. / ed. 2012, Hoboken, NJ: Wiley.
26. Bozorg Grayeli, A., et al., *Long-term functional outcome in facial nerve graft by fibrin glue in the temporal bone and cerebellopontine angle*. *European Archives of Oto-Rhino-Laryngology and Head & Neck*, 2005. **262**(5): p. 404-407.
27. Ponder, E. and W.P. Kennedy, *ON THE ACT OF BLINKING*. *Quarterly Journal of Experimental Physiology*, 1927. **18**(2): p. 89-110.
28. Pennypacker, H.S., et al., *An apparatus and procedure for conditioning the eye-blink reflex in the squirrel monkey*. *Journal of the Experimental Analysis of Behavior*, 1966. **9**(5): p. 601-604.

29. Guitton, D., R. Simard, and F. Codère, *Upper eyelid movements measured with a search coil during blinks and vertical saccades*. Investigative Ophthalmology & Visual Science, 1991. **32**(13): p. 3298-3305.
30. Collewyn, H., J. van der Steen, and R.M. Steinman, *Human eye movements associated with blinks and prolonged eyelid closure*. Journal of Neurophysiology, 1985. **54**(1): p. 11-27.
31. Marcelli, E., et al., *A NEW GYRO-BASED METHOD FOR QUANTIFYING EYELID MOTION*. International Journal of Artificial Organs, 2013. **36**(3): p. 195-202.
32. Aring, E., et al., *Visual fixation development in children*. Graefe's Archive for Clinical and Experimental Ophthalmology, 2007. **245**(11): p. 1659-1665.
33. Stern John, A., C. Walrath Larry, and R. Goldstein, *The Endogenous Eyeblink*. Psychophysiology, 1984. **21**(1): p. 22-33.
34. Cruz, A.A.V., et al., *Spontaneous Eyeblink Activity*. The Ocular Surface, 2011. **9**(1): p. 29-41.
35. Kugelberg, E., *FACIAL REFLEXES*. Brain, 1952. **75**(3): p. 385-396.
36. Alice, F., B. Stefano, and C. Paolo, *Surface Electromyography Recording of Spontaneous Eyeblinks: Applications in Neuroprosthetics*. Otolaryngology-Head and Neck Surgery, 2012. **148**(2): p. 209-214.
37. Chen, K., et al., *Eyeblink reanimation prototype for facial nerve paralysis*, in *Neural Computation, Neural Devices, and Neural Prosthesis*. 2014. p. 99-120.
38. Kenichi, K. and S. Kazuyoshi, *Evaluation of Three Types of Blinks with the Use of Electro-Oculogram and Electromyogram*. Perceptual and Motor Skills, 1999. **88**(3): p. 1037-1052.

39. Gehricke, J.-G., E. Ornitz, and P. Siddarth, *Differentiating between reflex and spontaneous blinks using simultaneous recording of the orbicularis oculi electromyogram and the electro-oculogram in startle research*. Vol. 44. 2002. 261-8.
40. Coleman, D. and S. Trokel, *Direct-recorded intraocular pressure variations in a human subject*. Archives of Ophthalmology, 1969. **82**(5): p. 637-640.
41. Gisler, C., et al., *Automated Detection and Quantification of Circadian Eye Blinks Using a Contact Lens Sensor*. Translational Vision Science & Technology, 2015. **4**(1): p. 4.
42. Leonardi, M., et al., *First Steps toward Noninvasive Intraocular Pressure Monitoring with a Sensing Contact Lens*. Investigative Ophthalmology & Visual Science, 2004. **45**(9): p. 3113-3117.
43. Thompson, L.T., et al., *A system for quantitative analysis of associative learning. Part I. Hardware interfaces with cross-species applications*. Journal of Neuroscience Methods, 1994. **54**(1): p. 109-117.
44. Ryan, S.B., et al., *A long-range, wide field-of-view infrared eyeblink detector*. Journal of Neuroscience Methods, 2006. **152**(1): p. 74-82.
45. Caffier, P.P., U. Erdmann, and P. Ullsperger, *Experimental evaluation of eye-blink parameters as a drowsiness measure*. European Journal of Applied Physiology, 2003. **89**(3): p. 319-325.
46. Frigerio, A., et al., *Infrared-based blink-detecting glasses for facial pacing: Toward a bionic blink*. JAMA Facial Plastic Surgery, 2014. **16**(3): p. 211-218.
47. Bermejo, R., D. Houben, and H.P. Zeigler, *Optoelectronic monitoring of individual whisker movements in rats*. Journal of Neuroscience Methods, 1998. **83**(2): p. 89-96.

48. Isono, M., et al., *An objective evaluation method for facial mimic motion*. Otolaryngology - Head and Neck Surgery, 1996. **114**(1): p. 27-31.
49. Somia, N.N., et al., *A computer analysis of reflex eyelid motion in normal subjects and in facial neuropathy*. Clinical Biomechanics, 2000. **15**(10): p. 766-771.
50. Bracha, V., et al., *Video recording system for the measurement of eyelid movements during classical conditioning of the eyeblink response in the rabbit*. Journal of Neuroscience Methods, 2003. **125**(1): p. 173-181.
51. Bhaskar, T.N., et al. *Blink detection and eye tracking for eye localization*. in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*. 2003.
52. Lalonde, M., et al. *Real-time eye blink detection with GPU-based SIFT tracking*. in *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*. 2007.
53. Mitelman, R., et al., *A noninvasive, fast and inexpensive tool for the detection of eye open/closed state in primates*. Journal of Neuroscience Methods, 2009. **178**(2): p. 350-356.
54. Pardas, M. *Extraction and tracking of the eyelids*. 2000. IEEE, PISCATAWAY, NJ, (USA).
55. Grauman, K., et al. *Communication via eye blinks - detection and duration analysis in real time*. in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001.
56. Grauman, K., et al., *Communication via eye blinks and eyebrow raises: video-based human-computer interfaces*. Universal Access in the Information Society, 2003. **2**(4): p. 359-373.
57. Chau, M. and M. Betke, *Real Time Eye Tracking and Blink Detection with USB Cameras*. Vol. 2215. 2005.

58. Awais, M., N. Badruddin, and M. Drieberg. *Automated eye blink detection and tracking using template matching*. in *2013 IEEE Student Conference on Research and Development*. 2013.
59. Hadlock, T., et al., *A Novel Method of Head Fixation for the Study of Rodent Facial Function*. *Experimental neurology*, 2007. **205**(1): p. 279-282.
60. Rafati, M., M. Arabfard, and M. Rafati-Rahimzadeh, *Comparison of different edge detections and noise reduction on ultrasound images of carotid and brachial arteries using a speckle reducing anisotropic diffusion filter*. *Iranian Red Crescent medical journal*, 2014. **16**(9): p. e14658-e14658.
61. Chan, T.F. and L.A. Vese, *Active contours without edges*. *IEEE Transactions on Image Processing*, 2001. **10**(2): p. 266-277.
62. Kaminer, J., et al., *Characterizing the Spontaneous Blink Generator: An Animal Model*. *The Journal of Neuroscience*, 2011. **31**(31): p. 11256.
63. Salem, G.H., et al., *SCORHE: A novel and practical approach to video monitoring of laboratory mice housed in vivarium cage racks*. *Behavior research methods*, 2015. **47**(1): p. 235-250.
64. Rodriguez, A., et al., *ToxTrac: A fast and robust software for tracking organisms*. *Methods in Ecology and Evolution*, 2017. **9**(3): p. 460-464.

8. Appendix

8.1. Project code

The project code is available in the following public repository:

https://github.com/reklab/eyelid_tracking

8.2. Code manual

The manual for running the code is available in the same repository. Link to manual:

https://github.com/reklab/eyelid_tracking