

**Bachelorarbeit
in Bioinformatik**

**STATISTICAL ANALYSES OF DROSOPHILA
BEHAVIOR UPON MANIPULATION OF DEFINED
NEURONS**

Jessika Binder



LUDWIG-MAXIMILIANS-UNIVERSITÄT
MÜNCHEN
TECHNISCHE UNIVERSITÄT MÜNCHEN



Bachelorarbeit
in Bioinformatik

Statistical analyses of *Drosophila* behavior upon manipulation of defined neurons

Jessika Binder

Aufgabensteller: PD Dr. Thomas Wachtler
Betreuer: Dr. Hiromu Tanimoto
Abgabedatum: 15. November 2012

Selbstständigkeitserklärung

Ich versichere, dass ich diese Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. November 2012

Abstract

This bachelor thesis deals with the analysis of *Drosophila* appetitive behavior after blocking sugar sensory neurons. In particular, the sugar sensory neurons that express the sugar receptor Gr5a were blocked and the effect on behavior was studied by using a simple sucrose preference assay. Videos of fly behavior in the setup were the starting material for this work.

To quantify fly behavior, I developed a high-throughput pipeline that allows automated analysis of the previously acquired videos. This software makes it possible to calculate the sucrose preference of flies over time. Furthermore, a measure of fly activity can be determined. Using the developed software, I showed that blocking the Gr5a-expressing sugar sensory neurons impairs sugar preference and causes an increase in overall activity.

Diese Bachelorarbeit befasst sich mit der Analyse des Appetenzverhaltens von *Drosophila*-Fliegen nach der Blockade von Zucker-Sensorneuronen. Insbesondere jene Zucker-Sensorneuronen, die den Zuckerrezeptor Gr5a exprimieren, wurden blockiert und der Effekt auf das Verhalten mit Hilfe eines einfachen Saccharose Präferenz-Tests untersucht. Das Ausgangsmaterial hierzu waren Videos von Fliegenverhalten innerhalb dieses Aufbaus.

Um das Fliegenverhalten zu quantifizieren, wurde ein High-Throughput-System entwickelt, das eine automatische Analyse der vorher erhaltenen Videos ermöglicht. Dieses System macht es möglich, die Saccharose-Präferenz der Fliegen über die Zeit zu berechnen. Außerdem kann ein Maß für die Fliegenaktivität eingeführt werden. Mit Hilfe der entwickelten Software kann gezeigt werden, dass eine Blockade des Gr5a-exprimierenden Zucker-Sensorneurons die Zucker-Präferenz vermindert und einen Anstieg in der gesamten Aktivität verursacht.

Acknowledgements

I would like to thank my supervisor Dr. Hiromu Tanimoto for his support and for giving me the opportunity to write my thesis in the context of this interesting but also challenging project.

I also want to thank Vladimiro Thoma for taking his time to help whenever I needed it. Thanks for reading my thesis, introducing me to fly biology and his valuable assistance during the writing time.

Without their unbroken encouraging and suggestions on the right time I might have given up on the project.

Christian Garbers often had the saving ideas when I was stuck in my source code. Without his help I would have despaired.

Furthermore, I would like to thank PD Dr. Thomas Wachtler for accepting to supervise my thesis without knowing what to expect.

Last but not least, I am deeply grateful to my friends and family who never stopped motivating me. Thanks for their patience and believing in me.

Contents

1	Introduction	1
1.1	Quantification of appetitive behavior in the fruit fly	1
1.1.1	From neurons to behavior	1
1.1.2	The fruit fly as a model organism	1
1.1.3	The GAL4/UAS system	2
1.1.4	Appetitive behavior of the fruit fly	3
1.1.5	The fly gustatory system	4
1.1.6	Studying fly appetitive behavior	5
1.1.7	Questioned addressed in this work	7
1.2	Implementing machine vision for analyzing fly behavior	7
1.2.1	Definition and advantages	7
1.2.2	Already known approaches for analyzing fly behavior	8
1.3	Automatic analysis and quantification of fly appetitive behavior	9
2	Materials and methods	11
2.1	Video description	11
2.2	Step 1 - Processing with Avidemux	12
2.3	Step 2 - Processing with Matlab	12
2.4	Step 3 - Analyzing with ImageJ	13
2.4.1	Finding first flies	13
2.4.2	Border detection	14
2.4.3	Fly detection and counting	16
2.4.4	Detection of fly movement	23
3	Results	29
3.1	Optimizing video processing	29
3.2	Optimizing border detection	29
3.3	Optimizing fly detection	30
3.3.1	MaximumFinder as chosen method	30
3.3.2	Further improving of the fly detection	31
3.4	Preference indices	32
3.5	Fly movement results	37
4	Discussion	41
4.1	Automatic analysis of fly behavior	41
4.2	Blocking the neurons in Gr5a-GAL4 impairs sucrose response	42
4.3	Blocking the neurons in Gr5a-GAL4 increases locomotion in the sugar arena setup	42
	List of Figures	43
	List of Tables	44
	References	45

1 Introduction

1.1 Quantification of appetitive behavior in the fruit fly

1.1.1 From neurons to behavior

One of the major challenges of modern biology is to understand the organization and functions of the central nervous system. Animals can perceive their environment through sensory neurons. The information is relayed to interneurons, where it is processed. Finally, motor neurons can activate a set of appropriate behavior. Sensory neurons, interneurons and motor neurons are connected through synapses, forming organized networks that are called neuronal circuits.

The study of such complicated networks is a field of intensive research in neurobiology. Nevertheless, despite these efforts, neuronal circuits can be extremely complex and are only beginning to be understood.

1.1.2 The fruit fly as a model organism

The human brain is believed to have approximately 10^{11} neurons, which make an estimated 10^{14} connections (synapses). Due to this complexity, studying the neuronal basis of behavior in humans can be challenging. Additionally manipulative experiments often cannot be performed. For these purposes, simpler organisms, which can serve as models are commonly used.

The fruit fly *Drosophila melanogaster* is a well-established model organism in neurobiology. This is because of several advantages. *Drosophila* has only an estimated number of 10^5 neurons in its brain and has thus a numerical simplicity compared to the human brain. Second, it has a short life cycle and it can be kept in the lab relatively easy. Third and most importantly a flood of genetic tools are available in the fruit fly. Such genetic tools allow the functional manipulation of genes and cells (including neurons) in this model organism.

One of the most powerful genetic tools in *Drosophila* is the GAL4/UAS system [11]. This system allows targeted gene expression. In other words, the GAL4/UAS system can be used to express any gene in a defined population of *Drosophila* cells.

1.1.3 The GAL4/UAS system

The GAL4/UAS system consists of two components: a GAL4 line and a UAS line.

GAL4

The GAL4 line is a fly line containing the GAL4 gene, which is a transcription activator from yeast. The expression of GAL4 in this line is regulated by neighboring DNA sequences within the fly genome, which are called enhancers. These enhancers determine where the GAL4 gene is expressed.

One approach to target GAL4 to specific cells is to use the enhancers of a gene with known expression pattern. By cloning these enhancers and inserting them upstream of the GAL4 sequence, the GAL4 expression pattern is expected to recapitulate the gene's expression pattern.

UAS

In yeast, the product of the GAL4 gene is a protein (also called GAL4) which binds to specific DNA sequences. Such a sequence is called Upstream Activating Sequence (UAS). Normally, *Drosophila* does not possess such sequences. Therefore, GAL4 expression alone usually has no effect in *Drosophila* cells. However, when a UAS is introduced in *Drosophila*, GAL4 drives the expression of any gene lying downstream of the UAS (Figure 1). Using this approach, the expression of virtually any gene can be directed at the GAL4-expressing cells.

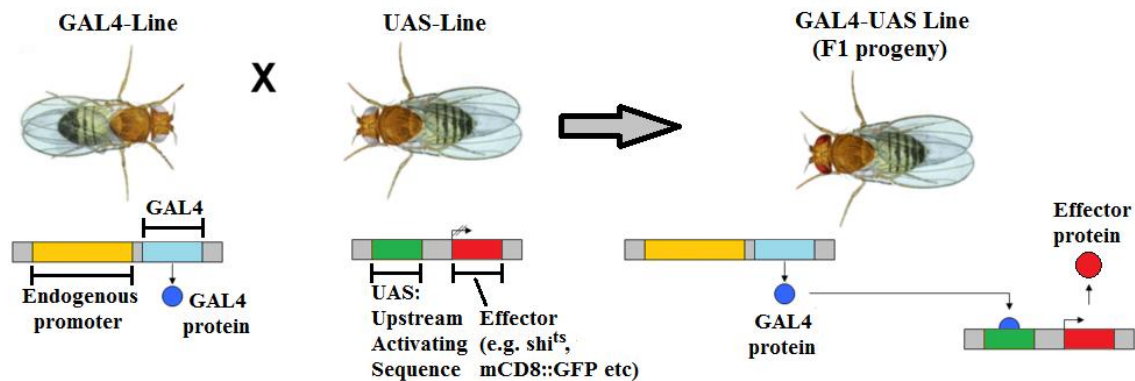


Figure 1: The GAL4/UAS system

Effectors and reporters

Several UAS-genes are of particular use in *Drosophila* neurobiology. Such genes are often called effectors, because they can have a particular effect on the neurons that express them. Some effectors can be used to activate neurons (e.g. TrpA1 [21]), while others can block them (e.g. shibire [28], Kir [10]). By expressing such genes, the effect of activating or blocking a particular group of neurons on behavior can be directly investigated. A

popular blocker of neuronal activity is Kir. Kir is an inward rectifying potassium channel that decreases the excitability of neurons.[10]

Apart from effectors, other UAS-lines can be used to visualize the GAL4-expressing neurons. Such lines are often called reporters. A popular transgene for labeling the GAL4-expressing cells is the green fluorescent protein (GFP). If a UAS-GFP line is used with a GAL4-line, the GAL4-expressing cells will be fluorescently labeled, and their anatomy can be studied with fluorescence microscopy.

The existence of an arsenal of genetic tools in *Drosophila* is particularly helpful in understanding functional neuronal circuits. In particular, specific neurons can be blocked or activated and the effect of this manipulation on the animal's behavior can be studied. If a neuron's activity is necessary and/or sufficient for a specific behavior, then this can be taken as an indication of the neuron's involvement in that behavior. While there are other approaches to the analysis of neuronal circuits (e.g. [36]), this strategy has been repeatedly shown to be extremely effective. In particular, using this approach, neurons involved in circadian rhythms [35], courtship [43][50], temperature regulation [21], feeding [20][31] and learning and memory [29] have been identified, to name only a few examples.

1.1.4 Appetitive behavior of the fruit fly

The detection of food, the evaluation of its quality and eventually its ingestion, are essential behaviors for the survival of all animal species. Therefore, such behaviors are conserved throughout the animal kingdom.

Appetitive behavior in the fly is subject to motivational control. Starvation is known to exert this kind of control over appetitive behavior. While fed flies are less likely to display appetitive behavior after detecting palatable substances, starved flies respond to such stimuli with higher probability. The neuronal basis of this control is partially understood [32][23].

Fruit flies can detect a variety of palatable stimuli. Sugars often can have a high caloric content and are therefore excellent sources of energy. In *Drosophila*, they can be detected by a number of sweet taste receptor neurons that are present in the fly's labellum and tarsi [51]. Besides sugars, other compounds, such as amino acids, can drive appetitive responses, although the underlying cellular and neuronal mechanisms are less understood [48]. Interestingly, recent work suggests that the ingestion of food is driven not only by taste, but also by caloric content [41]. The mechanisms by which caloric content is sensed are still poorly understood [41].

Appetitive stimuli produce a sequence of stereotyped behaviors in the fruit fly. Food detection initially takes place through gustatory sensory neurons (GRNs) in the distal-most parts of the flies' legs (tarsi). Upon detection, the flies stop walking. They extend their proboscises and then begin ingesting the food. The neurons driving these behaviors have been identified in some cases [51][31][20].

Sensory systems are the best studied, and will be developed below.

1.1.5 The fly gustatory system

In insects, sugars, salts, bitter compounds and several other chemicals are detected by gustatory receptor neurons (GRNs) that are distributed in different organs. The main gustatory organs of flies are the proboscis and the tarsi, but other body parts such as the wings and the female genitalia also express gustatory receptors (GRs)(Figure 2A). GRNs

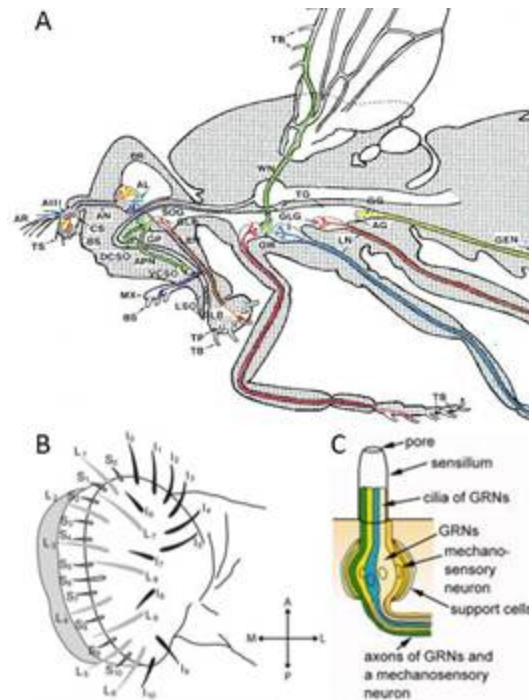


Figure 2: The fly gustatory system (Sources: A [42], B [52], C [33])

are either housed in taste bristles (sensilla) or in taste pegs. Each taste peg contains one chemosensory and one mechanosensory neuron, while each bristle is typically innervated by two to four chemosensory and one mechanosensory neuron [19][14]. GRNs are best characterized in the external mouthpart (the labellum of the proboscis), where the taste sensilla are categorized into three classes by size, position, and composition of GRNs. The positions of the sensilla on the tarsi seem to be similarly stereotyped [34]. However, except for their special requirement for pheromone perception [44][47][45] not much is known about the GRNs housed in tarsal sensilla. GRNs are bipolar, extending their dendrites into the lumen of the sensilla and their axons to the central nervous system (Figure 2C). Their cell bodies are located in the base of the sensilla. GRNs in the proboscis project to the primary gustatory center (PGC): the subesophageal ganglion (SOG)/tritocerebrum. The PGC receives gustatory input through three peripheral nerve tracts. GRNs in the labellum and the oesophagus project through the labial and pharyngeal nerves, and terminate in the SOG and the tritocerebrum (anterior to the SOG), respectively. GRNs from the wings and legs are connected to the PGC through the cervical connective. Interestingly, some tarsal GRNs innervate the thoracic ganglia, whereas others appear to project directly to the SOG and terminate in the most posterior part of the SOG [51][16]. GRNs of different

taste modalities (e.g. tuned to sweet or bitter tastes) project to spatially segregated regions in the PGC, thereby creating a spatial map for taste qualities [51][46][33]. GRNs expressing the same GR in different body parts also project to distinct locations in the PGC [51]. Such spatially segregated gustatory inputs might provide means for different behavioral outputs in response to the same tastant.

The *Drosophila* GRs constitute a family of 60 genes, that are predicted to encode 68 seven-transmembrane receptors through alternative splicing [14][18][39][37]. Eight of these receptors (Gr5a, Gr61a, Gr64a-f) form a predicted subfamily of sugar receptors [37]. Each sensillum contains a sweet receptor neuron that expresses subsets of these receptors in a combinatorial manner. Gr64f acts as a general co-receptor for sugar detection with a few exceptions (e.g. fructose) [26]. The other putative GRs for sweet taste are ligand-specific. For example, Gr5a is tuned to a small group of sugars, including trehalose [24][15], whereas Gr64a is tuned to a broad range of sugars including fructose, sucrose and maltose [16][27]. Strikingly, flies lacking both Gr5a and Gr64a do not show electrophysiological or behavioral responses to any of the tested sugars [16].

1.1.6 Studying fly appetitive behavior

The complete sequence of the *Drosophila* genome identified the genomic structure of 60 GRs [9]. This information promoted generation of dozens of transgenic GAL4 drivers under the control of virtually all GR promoters [51][16][46][52]. Driving transgenic fluorescent proteins such as GFP with this collection of GAL4 lines revealed the expression patterns of specific GRs in the gustatory organs and their projections into the central nervous system [51][16][46][52]. These GAL4 drivers have been employed to analyse the role of specific GRNs by expressing transgenes that manipulate neuronal activity. Some prominent examples for such effectors are UAS-shibirets1, UAS-Kir2.1 or UAS-TrpA1 which can be used for silencing or activating neurons, respectively. Targeting expression of such effectors to specific GRNs revealed their functions in distinct gustatory behaviors. In this way, receptor neurons for sugars [51][16], bitter compounds [52], glycerol [53] and other modalities such as water [13] or sex pheromones [47][45] have been identified.

A number of assays have been developed to study fly appetitive behavior in the laboratory. The most important of these are the proboscis extension reflex assay, the two-choice preference test and the capillary feeder assay.

In the proboscis extension reflex (PER) assay, a single, immobilized fly is used. An appetitive stimulus, such as a droplet of sucrose solution, is applied on the fly's tarsi [51] or labellum [40]. The fly typically responds by extending its proboscis. By measuring the responses of a number of flies, the probability of proboscis extension to a given stimulus can be calculated.

In the two-choice preference test [22], several flies are given a choice between two appetitive stimuli. Each stimulus additionally contains a tasteless dye and it is presented as an aqueous solution upon filter paper. Two dyes of different color are used for the two stimuli. Flies are allowed to feed for a given time, before they are observed under a

microscope. Flies that ingested a stimulus can be easily followed due to the color of the dye. Therefore, by counting the flies with different colors, their preference can be quantified.

Finally, for the capillary feeder (CAFE) assay [25], a number of flies are introduced into a closed container. The palatable stimulus is a solution contained within a glass capillary. One end of this capillary is introduced into the fly-containing compartment, allowing the flies to drink the substance. By measuring the amount of solution consumed within a given time, the appetitive response can be quantified. Additional capillaries containing solutions of other stimuli can be introduced, to calculate preferences in a manner similar to the two-choice preference test.

To analyze fly appetitive behavior, a novel behavioral assay has been introduced in Dr. Tanimoto's laboratory, by modifying the arena for appetitive visual learning [38]. Flies are introduced in a circular arena and given a choice between two sides (Figure 3). The bottom of the arena is covered with two halves of dried filter papers soaked with a sucrose solution and water. The distribution of the flies is taken as a measure of appetitive responses to sugar and quantified by calculating a preference index

$$PI = \frac{\# \text{ of flies on sugar side} - \# \text{ of flies on water side}}{\text{total } \# \text{ of flies in arena}}.$$

This assay, which we call the sucrose preference assay, offers several advantages. It is simple and rapid, allowing the measurement of a great number of fly lines in little time. It utilizes freely moving flies and thus allows quantification of sugar detection in a more natural setting. Importantly, all experiments are video-recorded. This allows the extraction of information about different variables of feeding behavior, such as how the PI evolves over time and how fly movement is affected by sucrose.

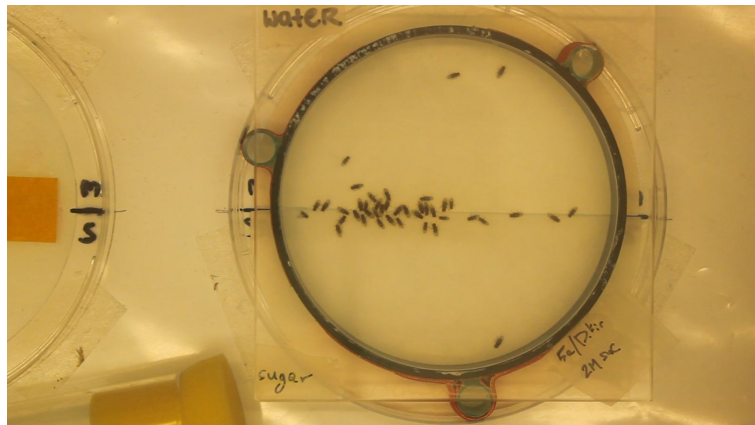


Figure 3: Original video

1.1.7 Questioned addressed in this work

The detection of sugars in *Drosophila* by sweet receptor neurons has been extensively studied. In a landmark paper [51], researchers used the PER assay and Gr5a-GAL4 to study sugar perception in *Drosophila*. Gr5a-GAL4 is a fly line that drives GAL4 expression to several sweet receptor neurons that express Gr5a, a gene that codes for a sugar receptor. These sweet receptor neurons are present in both the fly's labellum and tarsi. The authors showed that blocking the cells labeled by Gr5a-GAL4 results in impaired behavioral responses to sugars. As a result, Gr5a-GAL4 is considered to label most, if not all, sugar receptor neurons in the fruit fly. However, blocking these neurons impairs, but does not completely abolish, responses to sucrose.

To better understand the role of these sensory neurons in sugar response, the sucrose preference assay, which resembles the natural setting of fruit flies more closely than the PER assay, was used. The Gr5a-GAL4 cells were blocked with UAS-Kir [10] and the behavior of these flies was video-recorded. In addition, the respective genetic controls, namely Gr5a-GAL4 on its own or UAS-Kir on its own, were also analyzed in a similar manner (Table 1). The videos of these experiments were already acquired. To analyze the behavior of the flies in greater detail, I developed a high-throughput pipeline for processing these videos. In addition, I developed software for the automatic quantification of two aspects of fly sucrose response; namely, the PI and fly movement.

1.2 Implementing machine vision for analyzing fly behavior

1.2.1 Definition and advantages

“Machine vision [...] is a possibility of a machine [...] to obtain an information about surrounding environment for further analytical treatment.” [30] It combines hardware (such as camera) and different information processing methods for the desired analysis [30]. Some of these information processing methods are used during this thesis. They can be summarized as image processing. Image processing uses different algorithms for analyzing images, e.g. pixel counting, thresholding and edge detection.

Machine vision has different advantages for analyzing fly behavior. The automatic analysis allows to analyze fly behavior without human bias. Furthermore the otherwise too labor-intensive manually working time for analyzing hundreds of videos can be skipped and several aspects of fly behavior can be analyzed simultaneously.

1.2.2 Already known approaches for analyzing fly behavior

There is already existing software in this region of research. Most of this software though is only developed for the utilization of one particular scenario. Many existing softwares are using tracking for an analysis of the fly behavior. Tracking allows to locate moving flies over time and analyze the resulting trajectory in order to get information about the behavior. Three approaches that use tracking are briefly discussed below.

FTrack

The software FTrack that is based on Matlab uses background subtraction to find the remaining pixels that represent flies. After this, the maxima in the images are detected and assigned to individual flies. Thus the trajectory of a fly can be analyzed. Unfortunately the setup has to contain only one fly for this program to be applicable [49]. This makes FTrack completely inappropriate for the analysis of our videos because they contain multiple flies.

Using a Gaussian Mixture Model

Another method also uses background subtraction and works with the foreground image afterwards. By fitting a Gaussian mixture model (GMM) with three different brightness levels the orientation of the fly is determined and fitted by an ellipse. This method is designed for a video with only two flies and thus also not applicable for our videos. Furthermore a higher resolution than in our videos is needed for useful results of the GMM [17].

Ctrax

A recently published algorithm is CTrax, which allows to identify and determine the trajectory of more than two flies in a closed setup and automatically quantify their social and individual behavior [1].

Initially a static background image is subtracted from all frames. The remaining groups of pixels represent flies. All images are thresholded to yield binary images where flies are represented by circular objects. An ellipse is fitted to each object in order to determine fly position, size and orientation. By analyzing consecutive frames the movement of flies can be extrapolated to the following frames, thus resulting in the fly trajectory.

Ctrax is extremely useful for extracting a large amount of information on fly behavior. In particular, several aspects of fly movement such as speed, trajectory, walking distance and turning can be quantified. Additionally the user can define new behaviors which Ctrax can consequently detect and quantify.

However Ctrax can only be adopted under specific conditions. Firstly, the setup has to be a large circular arena with strong homogeneous back lighting. Secondly, the recording conditions are a spacial resolution of 1280x1024 pixels and temporal resolution of 20 frames per second (fps) and require a still camera. Movements of the arena or shadows over the flies can easily lead to large errors in the subtraction. Thirdly, a small fly number is used (20 flies) and flies have clipped-off wings to prevent jumping and flying. In addition, errors

in fly identity and therefore tracking can occur and require user interaction for correction. Finally a long processing time is required for each video [12]. Therefore Ctrax would be challenging to apply for our data collection.

1.3 Automatic analysis and quantification of fly appetitive behavior

Unfortunately, all of the available programs are not general enough to be directly applied for the analyses of our videos. Therefore a new software package for quantifying fly behavior had to be developed. As explained above, in the recorded videos flies are given a choice between a sugar and a non-sugar side. Thus different variables of the fly feeding behavior can be detected and analyzed with the help of this software package. A key parameter to quantify is the preference index (PI) of flies and its change in time. Manual observation of the videos reveals that flies slow down on the sugar side and therefore the fly movement is being quantified as an additional parameter.

Genotype	Genotype abbreviation	Effect
w ⁻ ; Gr5a-GAL4/+; UAS-Kir2.1::eGFP/+	5a/Kir	Gr5a-GAL4 cells blocked
w ⁻ ; Gr5a-GAL4/+; +	5a/+	No effect
w ⁻ ; +; UAS-Kir2.1::eGFP/+	Kir/+	No effect

Table 1: Tested genotypes

2 Materials and methods

2.1 Video description

The starting material is a collection of approximately 1000 videos showing fly behavior in the arena setup. These videos have the features shown in table 2.

Figure 4 summarizes the processing procedure, starting from the raw videos and resulting

Parameter	Value
format	RGB, .MOV
spacial resolution	1920x1088 pixels
temporal resolution	20 fps
length	approx. 1:15 min

Table 2: Features of original videos

in the data. Each step in the processing procedure is explained in further details below.

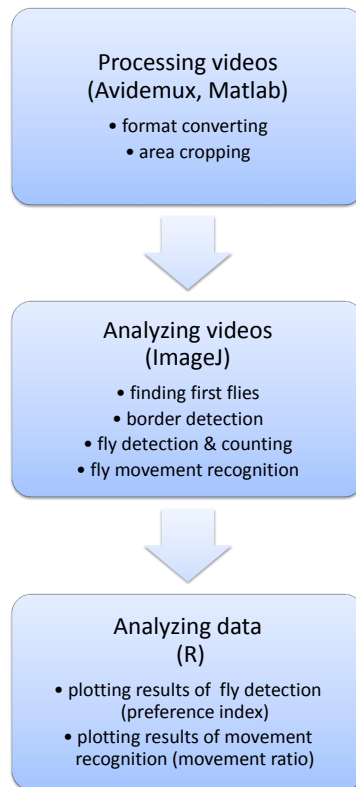


Figure 4: General workflow for each video

2.2 Step 1 - Processing with Avidemux

The processing starts with Avidemux [2], a free and open source video editing and processing program. This program was used to change the format to .AVI and to change the RGB to grayscale (Figure 5). This change was necessary, because ImageJ which is used for the analysis of the videos does not support the .MOV-format and grayscale pixel values can be evaluated better than RGB values. Moreover, Avidemux was initially used to downsize the videos by decreasing the spatial resolution to 720x408 pixels and the temporal resolution to 5 fps. The latter was done to decrease the size of videos in an effort to keep processing time and computation as low as possible. However, decreasing the spatial resolution lead to noise which caused high false positive error (Section 3.1). In addition, decreasing the temporal resolution caused problems in fly movement detection (Section 3.5). Thus downsizing was abandoned.

Because of the large amount of data this processing steps were automatized with Javascript and batch processing. Therefore this step is done with maximal efficiency and does not require user input.

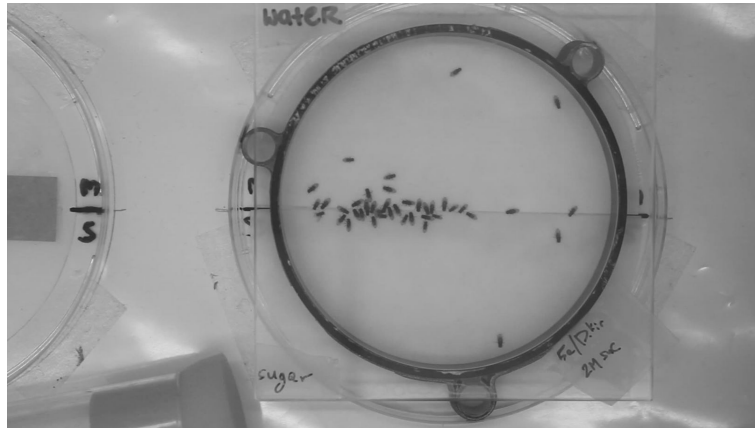


Figure 5: Converted video

2.3 Step 2 - Processing with Matlab

In addition everything except the arena has to be excluded, so that walking flies on the outside do not influence the analysis. The image processing toolbox of Matlab [3] is used for this purpose. A circular mask was applied to select the region of interest (ROI) for the arena and to display the rest of the video white. For this step the user has to input the diameter of the arena.

The final product (Figure 6) of the two processing steps with Avidemus and Matlab can be directly analyzed with ImageJ to yield data on fly appetitive behavior.



Figure 6: Final processed video

2.4 Step 3 - Analyzing with ImageJ

ImageJ [4] is an image processing program in which own plugins can be written in Java. It was used to quantify two aspects of fly appetitive behavior. Firstly, the preference index of the flies was calculated for every frame of the videos. Secondly, fly movement was assessed with a strategy that did not require tracking. These results were plotted with help of the programming language and software R [5].

2.4.1 Finding first flies

For the further analysis the frames without flies are not needed and a starting point from which all videos can be compared has to be defined. For these reasons, each frame from the beginning on is checked for presence of a certain amount of fly-like particles. The characteristics of these particles are defined by their size and circularity.

Due to a manual analysis of different videos a range of both parameters could be set. If more than two of these particles are present in the current frame, it is considered as a frame with flies. Less of these particles are not treated as flies, as they could be noise particles.

As soon as the first frame with the defined characteristics is found the border detection starts two frames before this starting frame. The frame direct before can not be used for the detection, because there could already be some flies in it. The characteristics indeed define the shape of a fly, but when the flies are injected they are blown towards one side of the area and thus the shape could be deformed and not identified as a fly.

2.4.2 Border detection

The ROI of the arena can be subdivided into two roughly equal semicircles. One side is covered with sucrose, while the other is covered with water. To calculate the flies' preference towards the sugar side, flies on both sides need to be counted. Therefore the border between these two sides has to be precisely identified.

When observing the videos the border could easily be recognized as a dark straight line near the middle of the arena. I took advantage of this observation and developed three different approaches to automatize border detection. Frames which did not contain flies were used to more correctly implement the border detection.

Approach 1 - Region of Interest (ROI) and ParticleAnalyzer

The easiest approach to find the border is to use its described darkness compared to the rest of the arena. With the help of thresholding these parts of the arena inside a predefined ROI can easily be detected.

Because the border is not completely horizontal but shifted a bit up or down, at least two y-coordinates are needed to draw a border line between them. Thus two ROIs were used - one on the left side of the arena and one on the right side (Figure 7).

After starting with a predefined threshold, the ImageJ ParticleAnalyzer [6] checks,

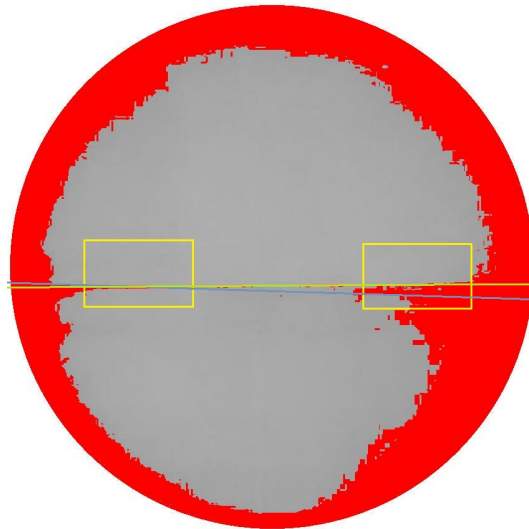


Figure 7: Detected border with approach 1 (blue) and real border (green)

whether the thresholded area in the ROI is too big or too small (based on a predefined size). In both of these cases the threshold is either decreased or increased until the area is in the right size range. The ParticleAnalyzer then provides the center of mass for this thresholded area, which can be used as a y-coordinate for the border.

Approach 2 - ROI and CurveFitter

A way to improve this approach is using a single ROI instead of two (Figure 8). This needs to be selected in the middle of the frame, so that it covers the border region. Additionally to this, also the method for defining the border coordinates changes. Not the area as one particle is used to define the coordinates, but its containing pixels. The coordinates of these pixels are the input for the ImageJ CurveFitter [7], which uses these to fit a straight line. Since most thresholded pixels should be near the border, the line matches the actual border quite well.

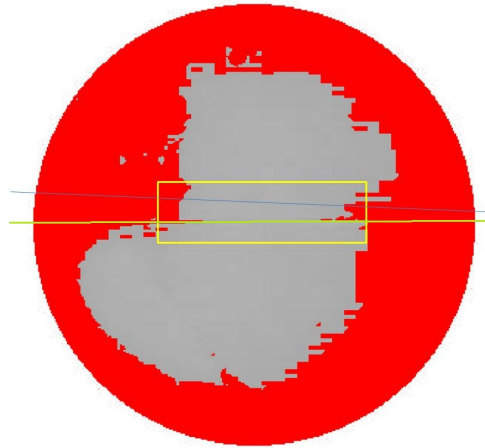


Figure 8: Detected border with approach 2 (blue) and real border (green)

Approach 3 - EdgeFinder, ROI and CurveFitter

In cases where the contrast between the border and the surrounding area is poor, finding the edges can be useful to enhance the contrast. Otherwise the other approaches would not allow to fit the border according to the real border position. The EdgeFinder is a built-in plugin of ImageJ that uses a Sobel operator to detect sharp changes in pixel intensity.

An appropriate ROI near the middle of the arena is selected and the EdgeFinder is then applied. After that the CurveFitter is adopted like in approach 2 to fit a line which accurately matches the border even in difficult cases (Figure 9).

Section 3.2 shows the reasons why approach 3 is used in the final implementation of the software.

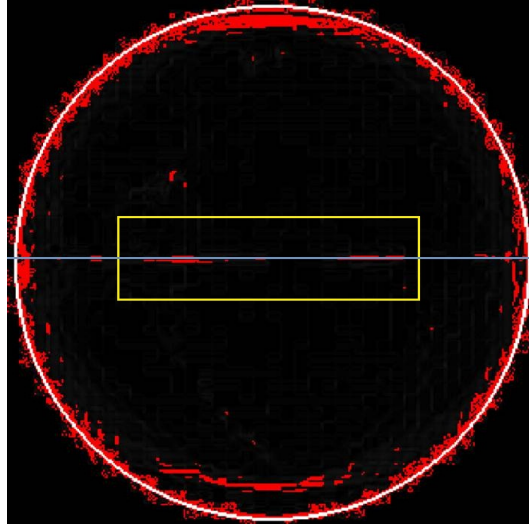


Figure 9: Detected border with approach 3 (blue)

2.4.3 Fly detection and counting

I used four different approaches (Table 3) to detect and count the flies and selected the best. Even in the best case there were errors so a post filtering step had to be applied.

Approach	Approach description
1	manual counting
2	thresholding
3	find maxima
4	find edges & thresholding
5	find edges & find maxima

Table 3: Approaches for fly detection and counting

Chosing test frames

The four approaches have all been tested at a selection of 20 frames from 20 different videos. These frames contain different cases of fly behavior in order to test the performance of the methods in all possible scenarios.

Table 4 contains the different cases that are included in the analysis. Figures 10, 11, 12, 13, 14, 15 and 16 show the used frames to the different scenarios. Each of the frames contains of about 30 flies.

The flies in these 20 frames are manually counted and afterwards compared to the results of the four different approaches of the fly counting to find the amount of false negatives (FN) and false positives (FP). FN are in this case the flies which are not detected as flies and FP are noise particles which are falsely detected as flies or double counted flies.

Situation
separate flies
border accumulation
jumping flies
boundary crawling
circular accumulation
jumping and crawling flies
border and circular accumulation

Table 4: Situations present in the used test-frames

Testing methods on frames

The thresholding method (Figure 17(a)) uses a simple threshold for all frames, that defines all particles darker than a certain pixel value as a fly.

The next method (Figure 17(b)) uses the former described EdgeFinder to find the edges between the flies and the background. On this new created picture a threshold can be applied like in the thresholding method.

The MaximumFinder (Figure 17(d)) finds all local maxima in a picture based on a certain noise tolerance level. A low noise tolerance level leads to more found maxima than a high level. Flies are the darkest spots inside the arena and thus have a greater difference to their neighbor pixel values. Therefore flies can be considered as the maxima.

Another method (Figure 17(c)) uses both the EdgeFinder and the MaximumFinder.

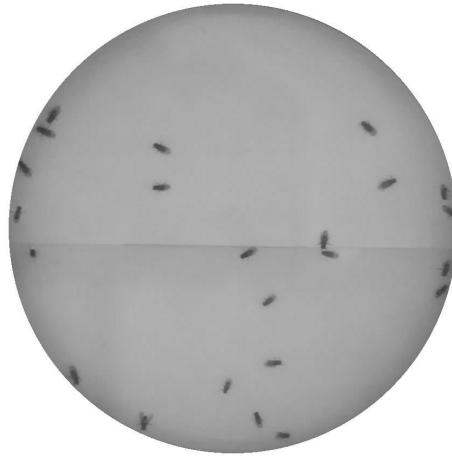
The best of the tested methods is the MaximumFinder. The detailed results of the tests can be found in the results section 3.3.1.



(a) Frame 7



(b) Frame 8



(c) Frame 13

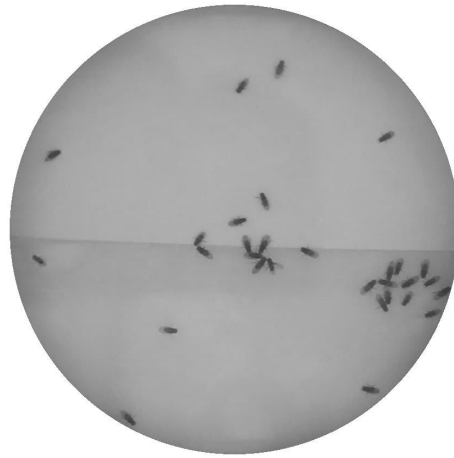
Figure 10: Testframes with separate flies



Figure 11: Testframe with border accumulation (Frame 11)



(a) Frame 1



(b) Frame 16

Figure 12: Testframes with circular accumulation

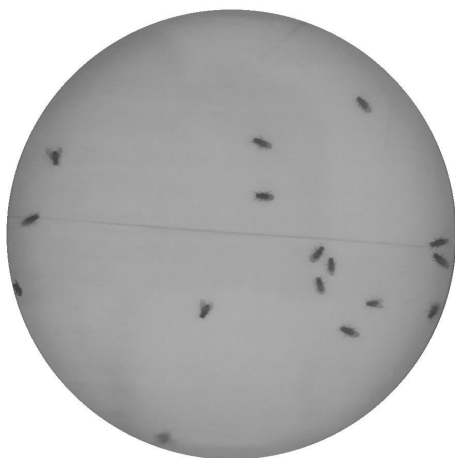


(a) Frame 10



(b) Frame 20

Figure 13: Testframes with border and circular accumulation



(a) Frame 3



(b) Frame 6



(c) Frame 14



(d) Frame 17



(e) Frame 19

Figure 14: Testframes with jumping flies



(a) Frame 2



(b) Frame 5



(c) Frame 9



(d) Frame 12



(e) Frame 18

Figure 15: Testframes with flies crawling up the boundary

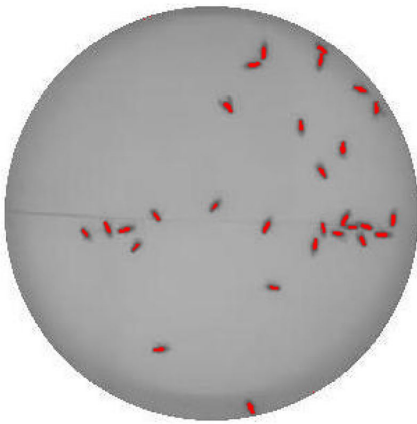


(a) Frame 4

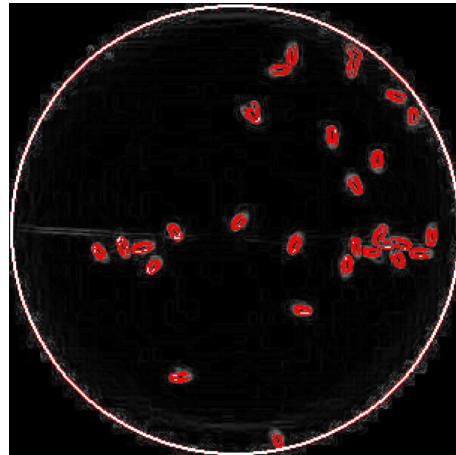


(b) Frame 15

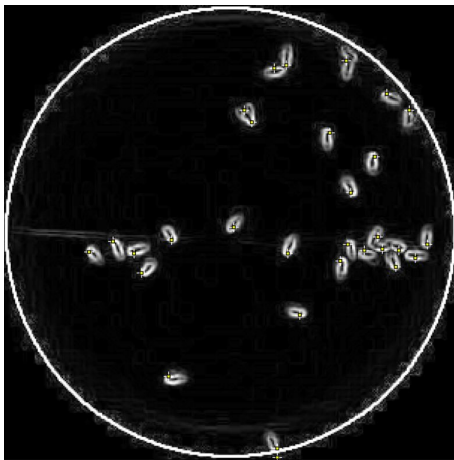
Figure 16: Testframes with jumping flies and flies crawling up the boundary



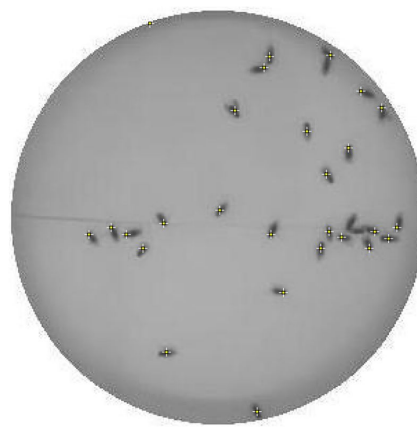
(a) thresholding



(b) find edges & thresholding



(c) find edges & find maxima



(d) find maxima

Figure 17: Different approaches for fly detection and counting

Post-filtering

However, the application of the MaximumFinder is not sufficient enough, therefore a post-filtering method was designed to filter out remaining FP. For this reason the coordinates of all found maxima were saved and used for the filtering. Two steps are performed for each of the saved maximum.

Firstly, the maximum is only considered as a possible fly, if its pixel value is lower than a certain cutoff value. If this is true, a second step is applied on this maximum and its eight neighboring pixels. This step calculates the difference between the pixel itself and its eight neighboring pixels. The former found maximum is only counted as a fly if for more than five of these nine checked pixels all calculated differences are lower than another cutoff value.

Hence, the post-filtering introduces two new parameters that need to be optimized together with the noise tolerance level. This results also can be found in the results section 3.3.1.

Calculation of the preference index

After using the coordinates of the border detection to define two ROIs, the amount of flies both on the sugar- and the water-side of the arena can be counted. These two numbers are used to calculate the preference index PI for each frame of the video. This index is defined as

$$PI = \frac{\# \text{ of flies on sugar side} - \# \text{ of flies on water side}}{\text{total \# of flies in arena}}.$$

The preferred side of the flies in the arena can be detected (Table 5) and analyzed (Section 3.4) with the help of this index.

PI value	preferred side
$0 < PI \leq 1$	sugar
$-1 \leq PI < 0$	water
$PI = 0$	equally preferred

Table 5: Values for preference index PI

2.4.4 Detection of fly movement

While the MaximumFinder can successfully detect the general position of the flies, it cannot be used for detecting fly movement. This is because the local maximum can shift within a fly between frames even if the fly is not moving.

One approach to measure fly speed is using tracking to define the trajectory of individual flies. As this approach is difficult to implement under the current experimental setup, a different strategy to assess fly movement was used.

The approach is based on the differences and similarities of two consecutive frames t and

$t + 1$. These measures are assigned to a new image. A moving fly should change position between consecutive frames. If a fly moves less than its body length, there is partial overlap between the particles in the resulting image (Figure 18(a)). If the fly moves a distance bigger than its body length, the objects in the new frame have no overlap (Figure 18(b)). Finally, if a fly does not move at all, there is full overlap (Figure 18(c)). The non-overlapping pixels between the consecutive frames is a measure of the moving parts of the flies while the overlapping pixels indicate the parts of flies that are not moving.

The two numbers of changed and unchanged pixels in the two consecutive frames are

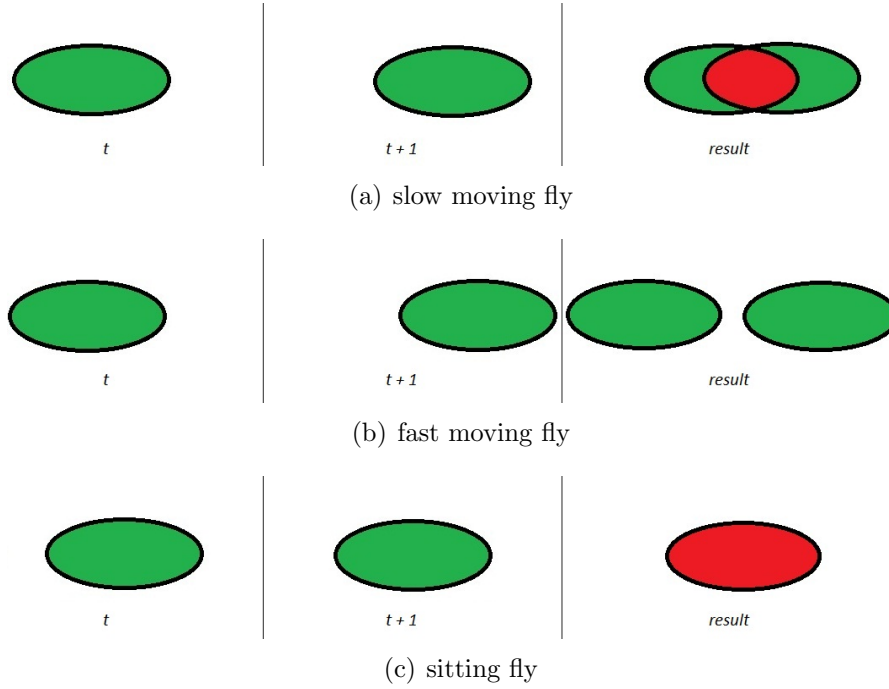


Figure 18: Flies moving with different speed

used to calculate a ratio r for the movement. It is defined as

$$r = \frac{\# \text{ of moving pixels}/2}{(\# \text{ of moving pixels}/2) + \# \text{ of sitting pixels}}.$$

This formula defines the fly movement as a ratio r between 0 and 1. If all flies are moving more than one body length r should be 1, if all flies are staying then r should be 0. If the the amount of moving and sitting particles is exactly the same, then r is 0.3. Doing this analysis over the whole video provides a time course over the ratios, that can be evaluated.

ImageJ has already implemented functions that can be used to calculate the described differences and similarities of two consecutive frames (Figure 19(a) and (b)).

The first function is the difference operator. This operator creates a new image that shows the differences between the two frames. That means, if two pixels that are in the same position in both frames have the same value, the pixel value in the resulting image is 0 (thus black). The bigger the difference of the pixel values is, the higher gets the pixel value in the resulting image. The bright parts of the resulting image (Figure 19(c)) thus

are matching the green (moving) parts of figure 18.

The second operator (maximum operator) uses a similar approach to identify the static pixels. A new image is created that shows the respective maximum pixel value of two corresponding ones of the consecutive frames. That means, always the brightest of the two pixel values will be assigned to the new image. If a fly moves, it will leave a brighter spot (background) in the first frame, which is then assigned to the new image. Only the pixels that still indicate a fly will be seen in the new image as this pixel then is still darker than the background. The inversion of this image (Figure 19(d)) is thus matching to the red (unchanged) parts of figure 18.

The two resulting images are mutually exclusive. A threshold is applied in both images, in order to get the amount of pixels that represent flies. After the conversion to binary data is still noise at the rim of the arena visible that leads to an inaccurate ratio for the movement. For filtering this noise, the ImageJ open operator, consisting of two separate steps, can be applied. Firstly, an erosion filter replaces each pixel with the lightest value of its eight neighbor pixels to remove the isolated pixels of the noise. However, this removes also pixels at the edges of flies, which is why the following dilation filter is afterwards applied. This inversion of the erosion filter replaces each pixel with the darkest value of the eight neighbors. Thus some black pixels are added again to the edges of the flies so that they are smoothened.[8] The flies have then the same size as before but no noise is remaining (Figure 19(e) and (f)). Figure 20 shows an overlay of the two resulting images. The green parts show the fly parts that are moving between the two frames, the red parts show the flies that are sitting.

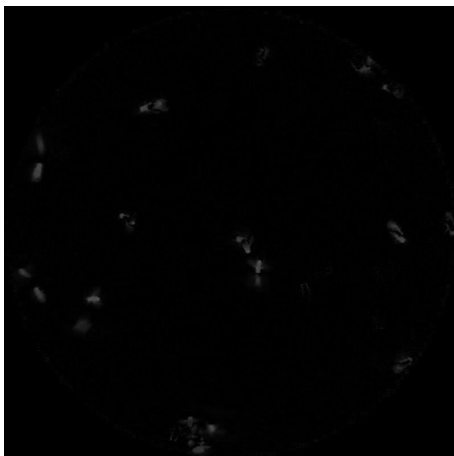
After counting the black pixels (flies) in the two images, the movement ratio can be calculated and analyzed (Section 3.5).



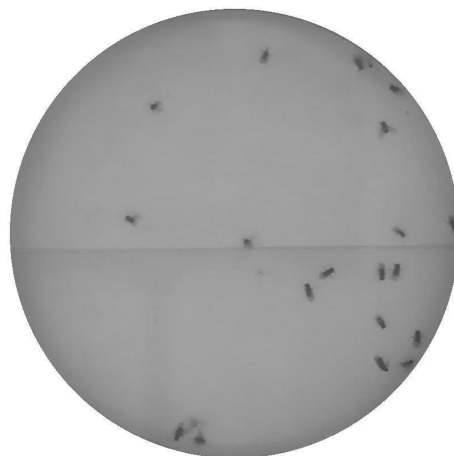
(a) Frame 1



(b) Frame 2



(c) Difference operator: moving parts



(d) Maximum operator: sitting parts



(e) applied threshold and open filter on (c)



(f) applied threshold and open filter on (d)

Figure 19: Detection of fly movement

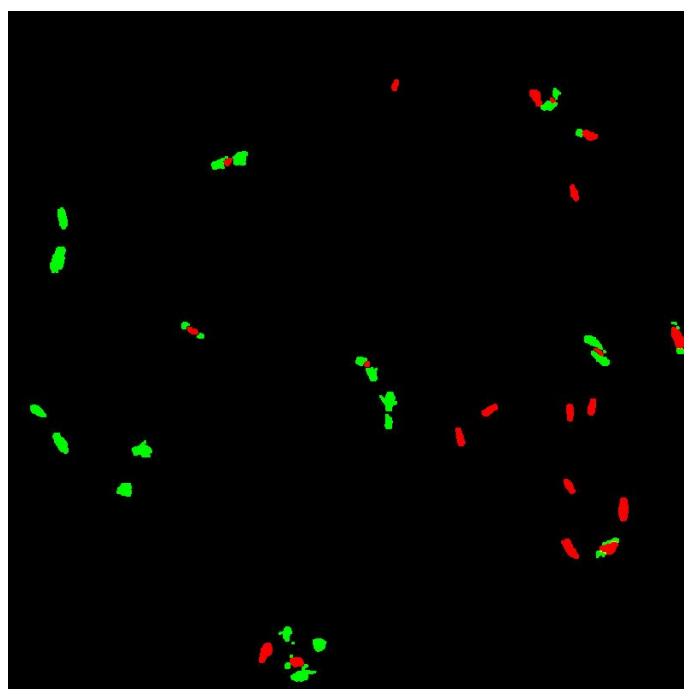


Figure 20: Overlay of figures 19(e) and (f)

3 Results

3.1 Optimizing video processing

Figure 21 shows the same section of a downsized (Figure 21(a)) and a non-downsized frame (Figure 21(b)). Downsizing changes the pixel values compared to the original video. Firstly, this leads to noise at the rim of the arena that can cause FP when counting the flies. Secondly, the downsizing also induces a less visible and thus harder to detect border than in the original video. Thirdly, the original resolution allows distinguishing the flies better from the background than the reduced resolution. These three reasons imply that for the final analysis the original video size is used.

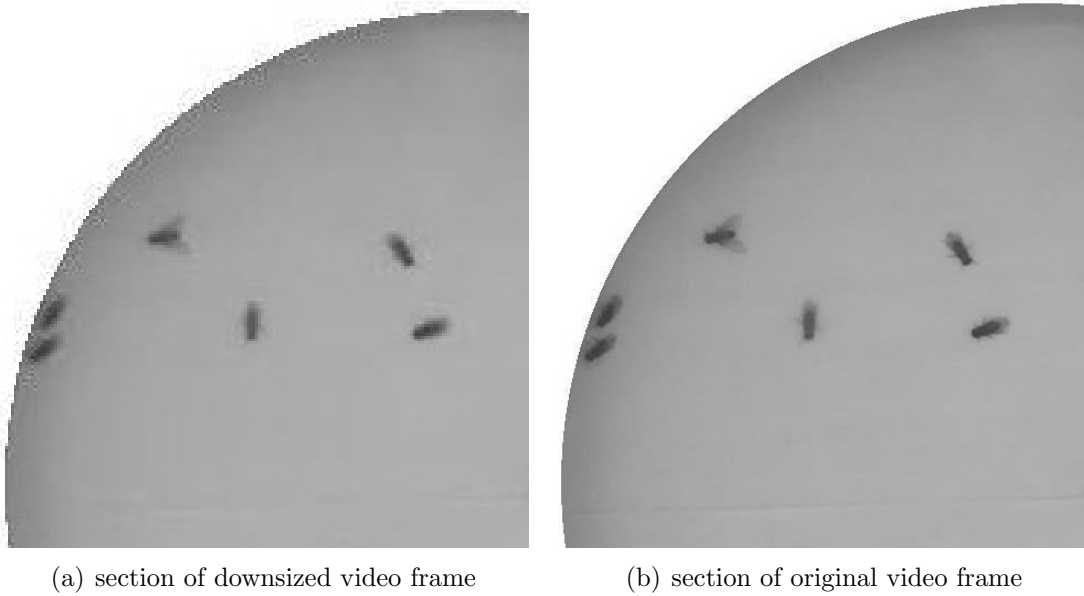


Figure 21: Noise in Video frames

3.2 Optimizing border detection

Approach 1 for the border detection uses a threshold applied on two ROIs to get the y-coordinates of the border. This method uses only two points to define the border and can therefore lead to errors. If one center of mass of the thresholded area is not exactly on the real border, the detected border does not match the real border (Figure 7). Hence this method is not optimal.

Approach 2 uses one ROI and each thresholded pixel inside to fit a straight line that matches the real border. However, in some videos the border is difficult to detect due to its low contrast with the background. Additionally, the experimenter that introduces the flies into the arena casts shadows that decrease the contrast between the border and the surrounding region. As a result the fitted line does not match the real border (Figure 8). Approach 3 applies the ImageJ function EdgeFinder before applying approach 2. With

this pre-filtering the shadows are eliminated, because their pixel-intensities are too consistent to be identified as edges. In the resulting picture all pixels at the edges are displayed white and the inner pixels as black. Using this approach, the border was accurately detected in all the analyzed videos.

3.3 Optimizing fly detection

Table 3 summarizes the methods tested for fly detection. As mentioned in section 2.4.3, the optimal method was the MaximumFinder with post-filtering.

3.3.1 MaximumFinder as chosen method

Comparing different detection approaches

All four methods were tested on the same set of 20 heterogeneous frames. This collection of frames was visually inspected and a single appropriate cutoff value was selected for each of the four different detection approaches. Figure 22 shows the percentage error of each of the methods compared to the number of manually counted flies numbered as in table 3. The error bars show the standard error of the mean (SEM).

The thresholding method has an error of more than - 15%. With a fixed threshold but differing lighting conditions in the videos not each fly can be detected individually. Flies sitting close to each other are often mistakenly merged.

The EdgeFinder has an error of nearly + 10%. The application of the EdgeFinder gives each fly a sickle shape. This shape is often divided into more than one particle and thus leads to an overestimation of flies in each frame.

The MaximumFinder approach leads to under + 1% error. This rate is much lower than the other approaches. Therefore, this method was selected for the analyses.

The method using first the EdgeFinder and later the MaximumFinder has an error of about -6%. In contrast to the former method that also uses the edges to define flies are less flies found than manually counted. Flies that are close to each other could be fused on particular edges and decrease the detected fly amount. These flies are still separated with a threshold, but not with the MaximumFinder.

Comparing different parameter settings - parameter fitting

After deciding that the MaximumFinder is the best method, further tests were carried out to analyze how tolerant the error rate is towards different noise tolerances. These tests can be seen in figure 23. The error rates are extremely high for all tested noise tolerance levels except the formerly tested tolerance of 45 (Figure 23(a)). A lower noise tolerance led to a higher positive error rate and a higher level resulted in fewer flies being identified than the manually counted.

To better understand the sources of the error we examined the contributions of FP and FN to the average error rate. These analyses revealed that even in the case of very low average error (noise tolerance level 45) the FP and FN rates are in fact quite high (10%)

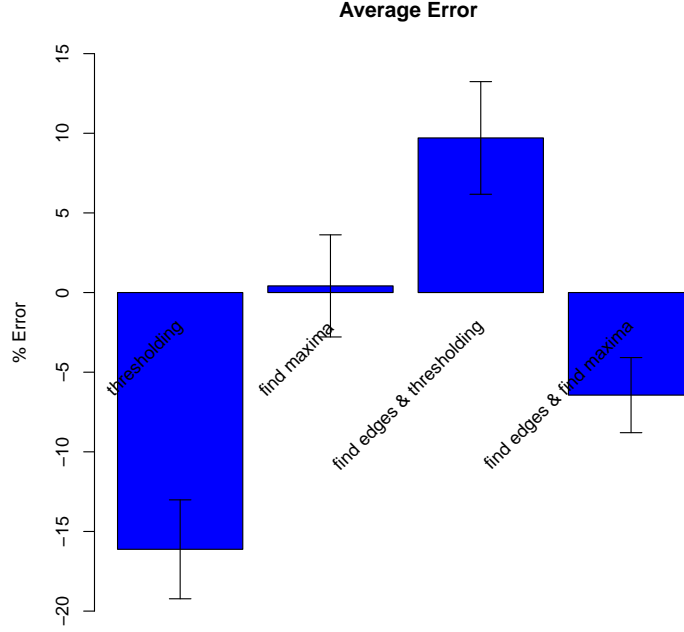


Figure 22: Errors in tested methods

but they cancel each other out (Figure 23(b)). This is not optimal and thus it was decided to apply postfiltering to eliminate as many FP and FN as possible.

3.3.2 Further improving of the fly detection

Eliminating FP

Measuring the pixel values of the flies in the 20 representative test frames, showed that their absolute values in all frames are lower than 105. Therefore, the threshold for the absolute value was set at a constant value. However, the threshold for the difference between the pixel values had to be tested because of the different light conditions and differences between the flies and the background in the videos.

Figures 24(a) and (b) show that the FP rate can nearly be eliminated through the filtering regardless of the parameter for the difference threshold. The FN and the total error percentage however are increased. A threshold of 70 was the best compromise between the FP, FN and the overall error rate.

After these changes is the FN rate still extremely high (17%). Therefore further improvements were necessary.

Eliminating FN

Using the MaximumFinder with the mentioned noise tolerance level 45 filtered out most FP but also led to non-detected flies and the high FN rate. Decreasing the noise tolerance led to more detected maxima. In this way the FN could be decreased without changing

the FP. A noise tolerance level of 10 yields the best possible rates for these 20 downsized frames. However, the FN (5.0%) and FP (4.8%) rates were still very high. Hence it was decided not to use the downsized videos but the original sizes (Section 3.1). The differences in the pixel values between flies and background stayed the same (cutoff level 105) but the parameter for the noise tolerance had to be further adjusted. Applying a tolerance level of 17 and the post-filtering led to lower FP and FN rates (0.5% and 3.7%, respectively) that are acceptable.

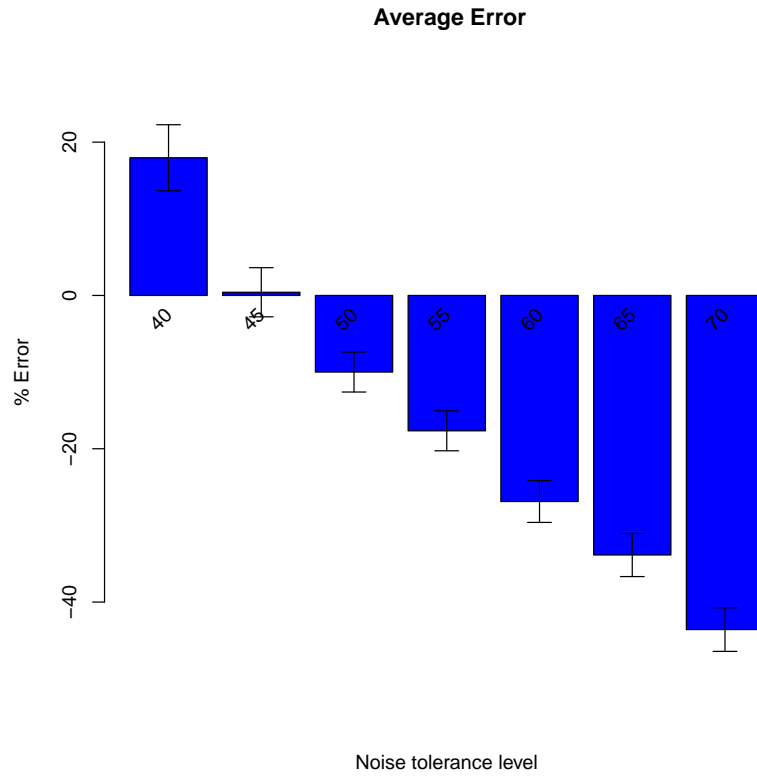
3.4 Preference indices

The preference index can be calculated with the previously mentioned equation (Section 2.4.3). Five videos for each of the three fly lines 5a/+, Kir/+ and 5a/Kir were analyzed. The average preference index over each time point as well as over the whole first 60 seconds of the videos was calculated for each fly line.

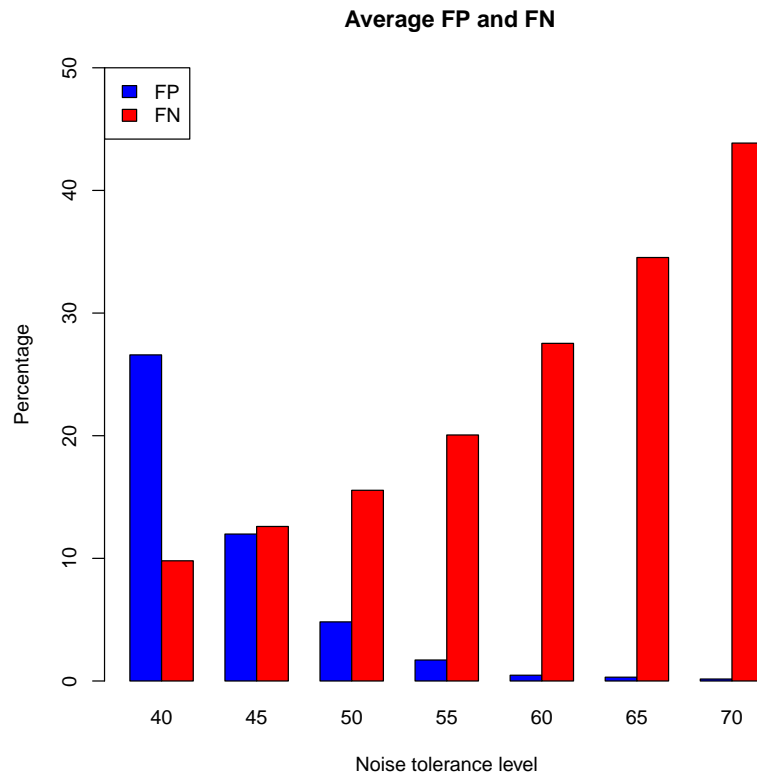
Figure 25 shows the course of the average *PI* over the first 60 seconds of the videos. The *PI* increases during the first ten seconds (200 frames) in all three fly lines and then nearly stays the same in the rest of the videos. In the first seconds the flies are distributing over the whole arena from the region on which they were blown in the beginning. After they spread and detect the sugar, they typically remain on the sugar side of the arena.

The two decreases of the preference index in line 5a/+ between frame 800 and 1000 are due to digital noise in some of the original videos that led to an erroneous counting of the flies.

The difference between the single fly lines is thus not the time in which the flies decide for one side of the arena but the extent to which the flies are choosing the sugar- or water-side (Figure 26).

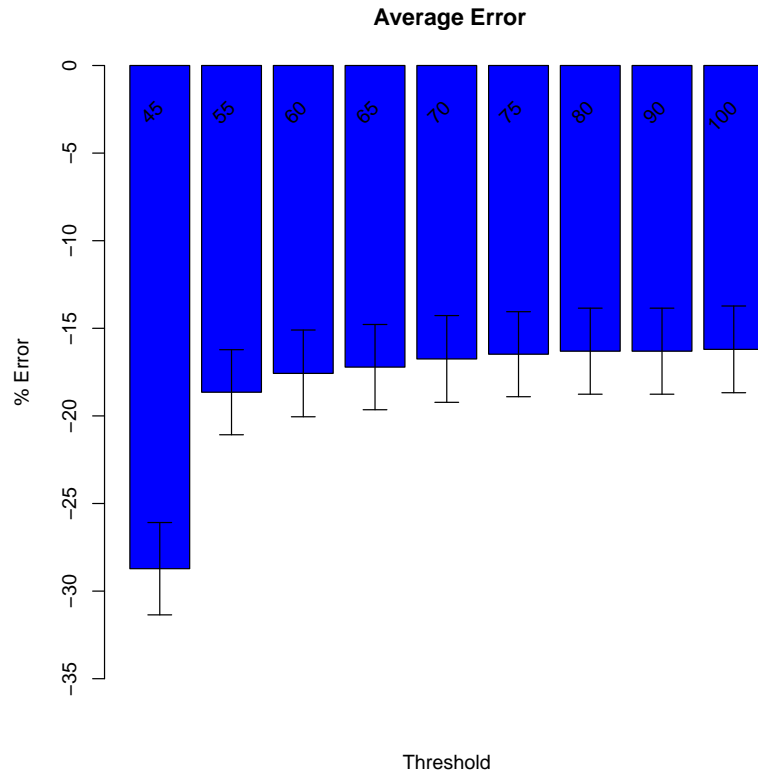


(a) % error using different noise tolerances

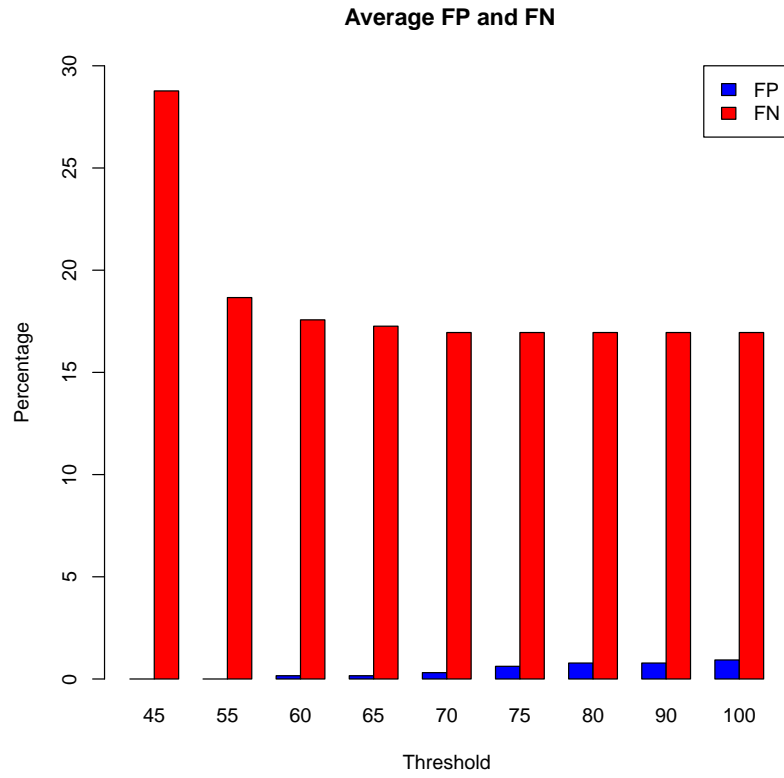


(b) average percentage of FN and FP per frame

Figure 23: Error rates at different noise tolerances



(a) % error using different maximal differences



(b) average percentage of FN and FP per frame

Figure 24: Error rates at different levels of maximal difference

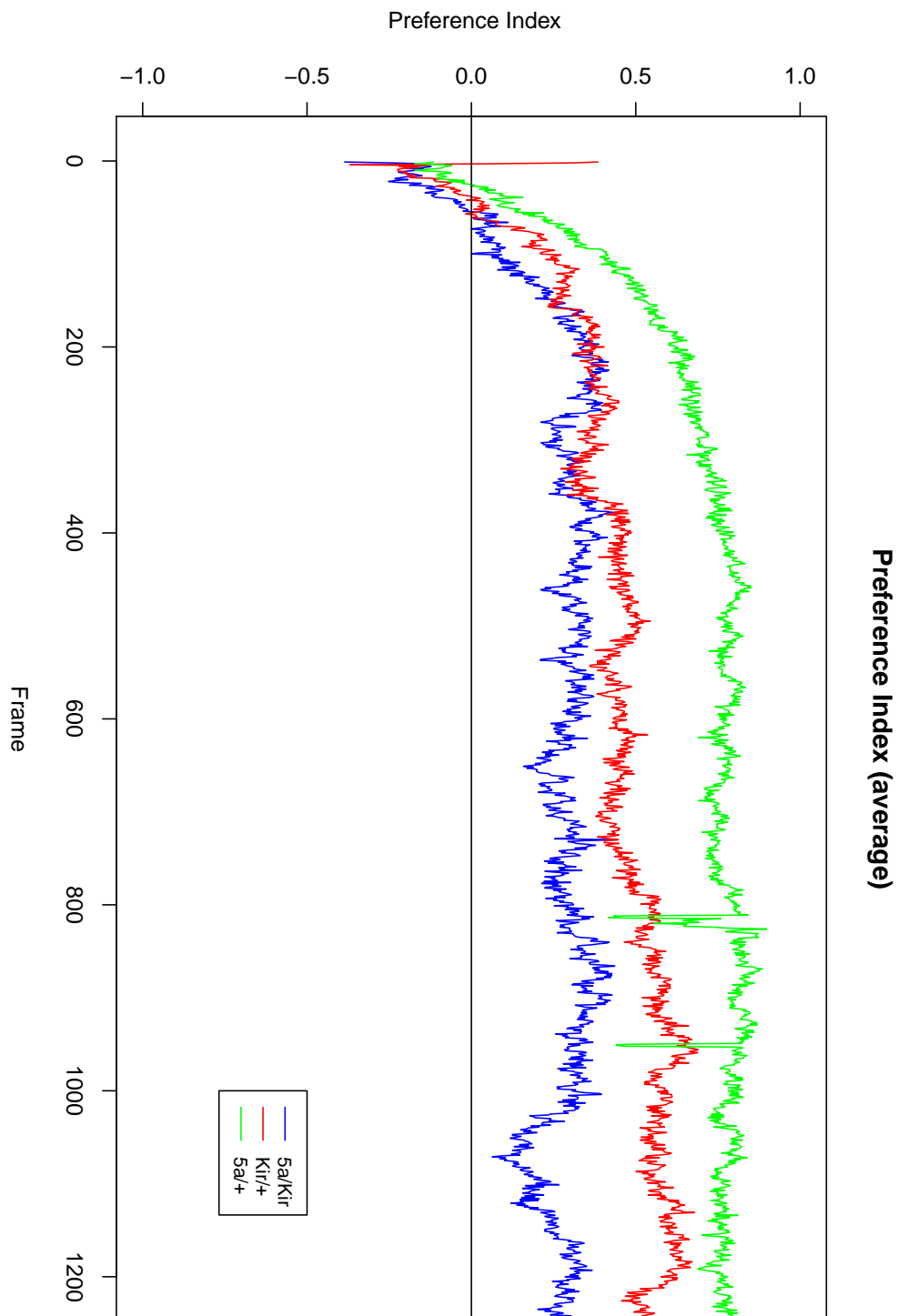


Figure 25: Preference indices for the fly lines over time

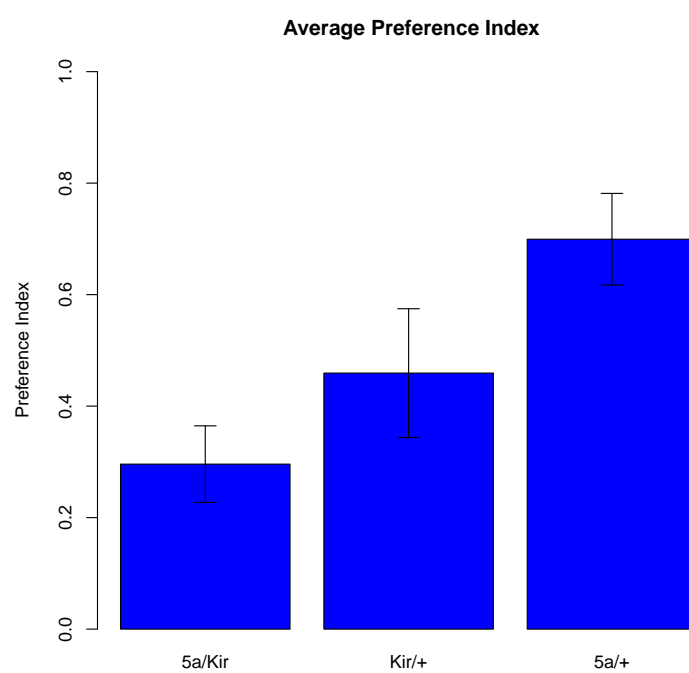


Figure 26: Average preference indices

3.5 Fly movement results

The movement of the different fly lines was analyzed both for the sugar- and water side of the arena and compared.

Testing the settings for the movement detection in a 5 fps video, detected many flies that moved more than one body length during the time between two frames. As the calculated ratio (Section 2.4.4) just gives a movement measure between 0 and 1, fly movements greater than one body length can't be distinguished. To minimize these cases the frame rate was increased. The 20 fps original frame rate led to an improvement in the quantification. Nevertheless, flies that moved more than one body length were at times detected.

The denominator of the equation used to calculate the ratio of movement indicates the amount of changed and unchanged particles, meaning all pixels that stand for flies. Thus this number should remain constant over the video, as the fly number also stays the same. Figure 27 shows the time course of the denominator. The 5a/+ line again has an error between frame 800 and 1000, due to the digital noise. However, in other cases, the denominator was essentially constant.

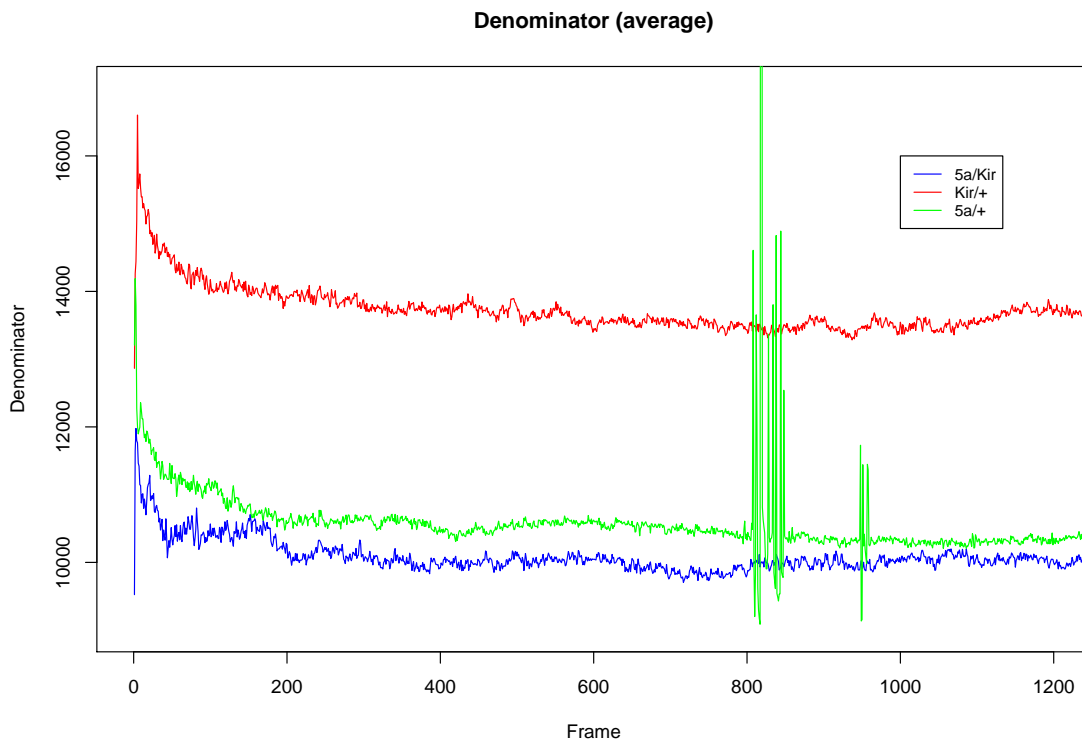


Figure 27: Average denominator of the fly lines

Comparing the movement of the flies over time between the different lines (Figures 28, 29 and 30) showed that all lines need about 20 seconds (400 frames) until the ratio stops decreasing. During these seconds the flies are leaving the region in which they were blown

in and are exploring the arena. As the flies in all lines prefer the sugar side (Figure 26), the movement on the water side decreases after the flies go over the border.

Flies normally stop moving when eating and thus the movement on the sugar side for 5a/+ (Figure 28) and Kir/+ (Figure 29) is nearly zero from the beginning on. In the 5a/Kir-line (Figure 30) the flies are not able to properly detect the sugar and thus the movement on the sugar side is higher than in the other two lines (Figure 31).

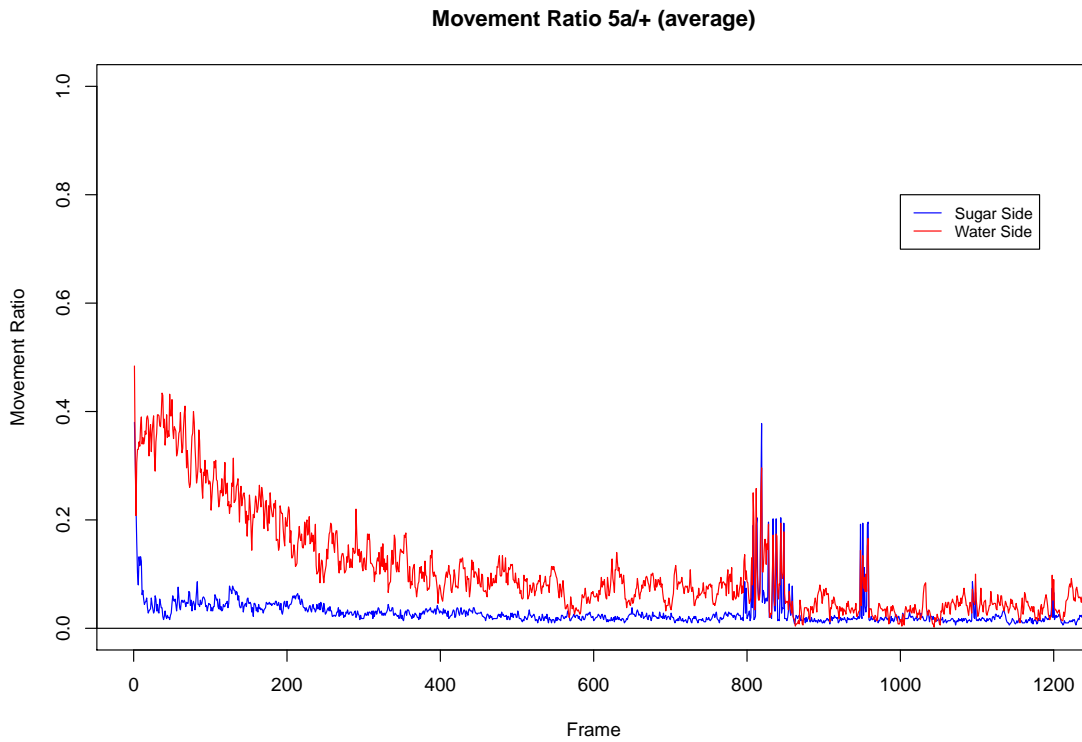


Figure 28: Movement of 5a/+ line

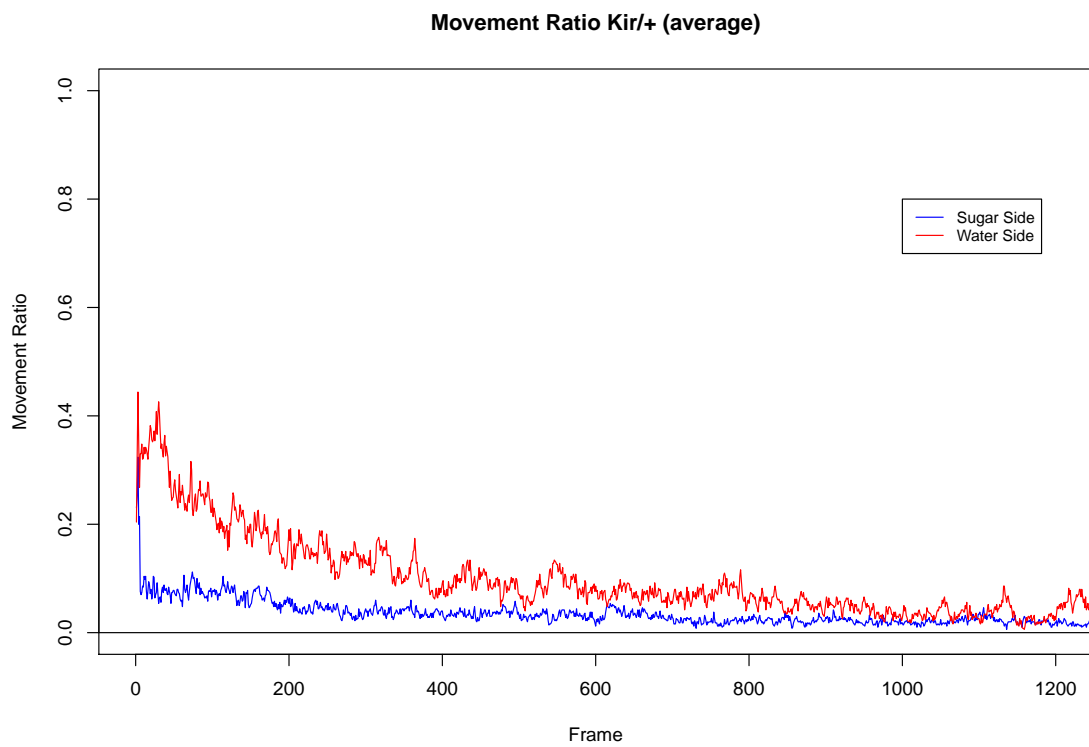


Figure 29: Movement of Kir/+ line

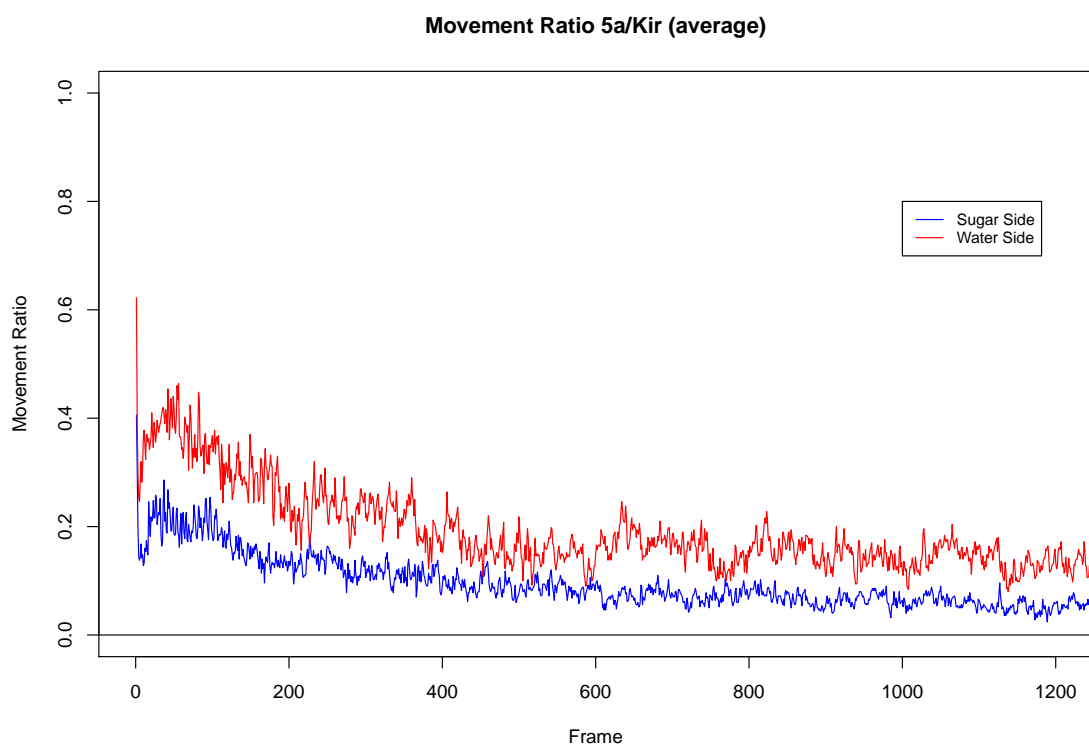


Figure 30: Movement of 5a/Kir line

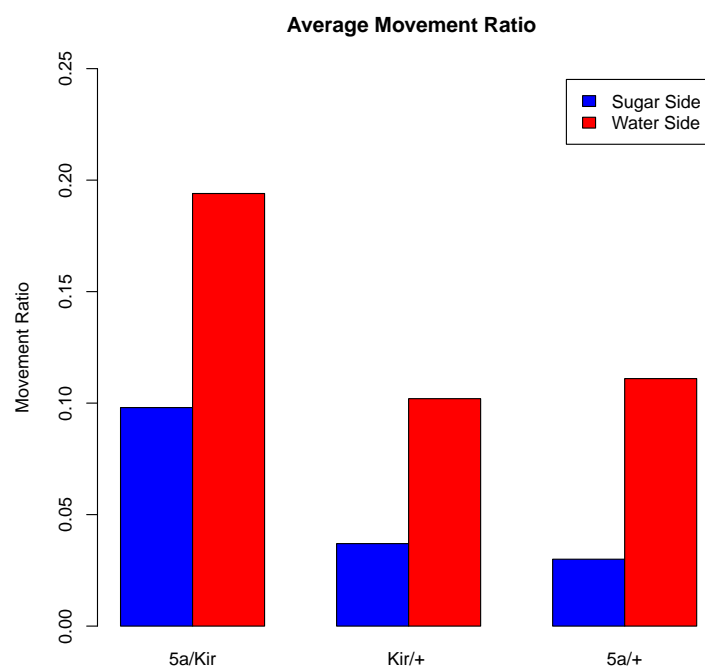


Figure 31: Average movement on different sides of arena of the fly lines

4 Discussion

4.1 Automatic analysis of fly behavior

For the project described here, I developed a high-throughput pipeline that allows automated analysis of fly appetitive behavior. In particular, two aspects of fly behavior in the sugar arena setup can be quantified. Firstly, the preference of the flies towards the sucrose side can be calculated. Secondly, a measure of fly activity can be determined for both the sucrose and the water side.

Making the analyses automatic is crucial, since Dr. Tanimoto’s laboratory has a collection of more than a thousand videos of behaving fruit flies in the sugar arena setup. While manual analysis is possible, it is extremely labor-intensive. Thus, the algorithms described here make the analyses far more efficient.

In addition, the described approach allows the quantification of an index that reflects fly activity. Importantly, this does not require individual fly tracking, which is not only computationally demanding, but also very labor-intensive.

Despite these important advantages, there are still aspects of the algorithms that can be improved. For example, while the average FN rate is acceptable (3.7%), a closer look revealed that individual FN rates are highly variable (0.0% - 13.3%). The FN rate should therefore be further decreased by applying another filtering or post-filtering step.

In addition, fly accumulation at the border often causes an underestimation of the PI. This is because flies stop at the border, with their heads facing towards the sucrose side but their abdomens remaining in the water side. These flies can detect sugar, but since the find maxima algorithm may detect pixels in their abdomen as maxima, they are counted on the water side. An appropriate shift in the border could be useful in fixing this issue. By now, the steps are still separated like shown in figure 4. With some further improvements and combinations of the algorithms, the analyses can be completely automatic. Also the current runtimes (Table 6) of the several steps for each video could be improved by other algorithms.

Step	Runtime
Processing with Avidemux	1 min
Processing with Matlab	3 min
Border detection & preference index calculation	5 min
Border detection & movement calculation	8 min
Border detection & both calculations	12 min

Table 6: Runtime of the several processing steps for one video

4.2 Blocking the neurons in Gr5a-GAL4 impairs sucrose response

Out of all three groups, 5a/Kir showed the worst performance (i.e. lowest PI). This observation is consistent with the fact that the neurons labeled in Gr5a-GAL4 are important for sugar perception, as previously shown [51]. In accordance with that study, the sugar response was impaired, but not completely abolished (i.e. $PI \neq 0$).

The two genetic controls, namely 5a/+ and Kir/+, performed better than the experimental group, since no sweet receptor neurons were blocked in these lines. However, it is worth noting that Kir/+ performed worse than 5a/+ (Figures 25 and 26). This may be due to technical reasons, such as subtle differences in the extent of starvation of between the two lines or an insufficient amount of experiment replications.

4.3 Blocking the neurons in Gr5a-GAL4 increases locomotion in the sugar arena setup

Using my algorithm for quantifying fly activity, I was able to show that blocking the neurons in Gr5a-GAL4 with Kir results in increased activity on both the sucrose and water side. The two genetic controls showed remarkably lower activity on both sides of the arena (Figure 31).

In the genetic controls, no neurons are blocked and therefore the sucrose response should be normal. These analyses showed that flies show lower activity on the sucrose side. This is consistent with our observations, since flies inhibit their locomotion, and can even stop, on the sucrose side. The activity on the water side is higher, since there is no gustatory stimulus on that part of the arena.

Interestingly, blocking the Gr5a-GAL4 neurons impairs, but does not abolish, the decrease in locomotion that is associated with the sucrose response. Indeed, these flies are still slower on the sucrose side than the water side. A possible interpretation is that blocking the Gr5a-GAL4 cells impairs the flies' ability to detect sucrose, but does not abolish it. This observation confirms our findings regarding the flies' preference and is also consistent with the literature [51]. Another possible interpretation is that blocking the Gr5a-GAL4 cells simply results in more active flies. This is indeed a possibility, since these flies showed higher overall activity on both sucrose and water. While the data presented here cannot be used to exclude this scenario, additional experiments can be of aid in clarifying this. For example, an arena where both halves are covered with water-soaked filter paper can be used to assess activity, using the same algorithm.

List of Figures

1	The GAL4/UAS system	2
2	The fly gustatory system (Sources: A [42], B [52], C [33])	4
3	Original video	6
4	General workflow for each video	11
5	Converted video	12
6	Final processed video	13
7	Detected border with approach 1 (blue) and real border (green)	14
8	Detected border with approach 2 (blue) and real border (green)	15
9	Detected border with approach 3 (blue)	16
10	Testframes with separate flies	18
11	Testframe with border accumulation (Frame 11)	18
12	Testframes with circular accumulation	19
13	Testframes with border and circular accumulation	19
14	Testframes with jumping flies	20
15	Testframes with flies crawling up the boundary	21
16	Testframes with jumping flies and flies crawling up the boundary	22
17	Different approaches for fly detection and counting	22
18	Flies moving with different speed	24
19	Detection of fly movement	26
20	Overlay of figures 19(e) and (f)	27
21	Noise in Video frames	29
22	Errors in tested methods	31
23	Error rates at different noise tolerances	33
24	Error rates at different levels of maximal difference	34
25	Preference indices for the fly lines over time	35
26	Average preference indices	36
27	Average denominator of the fly lines	37
28	Movement of 5a/+ line	38
29	Movement of Kir/+ line	39
30	Movement of 5a/Kir line	39
31	Average movement on different sides of arena of the fly lines	40

List of Tables

1	Tested genotypes	10
2	Features of original videos	11
3	Approaches for fly detection and counting	16
4	Situations present in the used test-frames	17
5	Values for preference index PI	23
6	Runtime of the several processing steps for one video	41

References

- [1] <http://ctrax.sourceforge.net/>. Last retrieved 2012-11-13.
- [2] <http://www.avidemux.org>. Last retrieved 2012-11-13.
- [3] <http://www.mathworks.de/products/matlab/>. Last retrieved 2012-11-13.
- [4] <http://rsb.info.nih.gov/ij/>. Last retrieved 2012-11-13.
- [5] <http://www.r-project.org/>. Last retrieved 2012-11-13.
- [6] <http://rsbweb.nih.gov/ij/developer/api/ij/plugin/filter/ParticleAnalyzer.html>. Last retrieved 2012-11-13.
- [7] <http://imagej.nih.gov/ij/developer/api/ij/measure/CurveFitter.html>. Last retrieved 2012-11-13.
- [8] <http://imagejdocu.tudor.lu/doku.php?id=gui:process:binary>. Last retrieved 2012-11-13.
- [9] ADAMS, M. D., ET AL. The genome sequence of drosophila melanogaster. *Science* 287, 5461 (March 2000), 2185–2195.
- [10] BAINES, R. A., UHLER, J. P., THOMPSON, A., SWEENEY, S. T., AND BATE, M. Altered electrical properties in drosophila neurons developing without synaptic transmission. *J Neurosci* 21, 5 (March 2001), 1523–1531.
- [11] BRAND, A. H., AND PERRIMON, N. Targeted gene expression as a means of altering cell fates and generating dominant phenotypes. *Development* 118 (1993), 401–415.
- [12] BRANSON, K., ROBIE, A. A., BENDER, J., PERONA, P., AND DICKINSON, M. H. High-throughput ethomics in large groups of drosophila. *Nature Methods* 6, 6 (June 2006), 451–457.
- [13] CAMERON, P., HIROI, M., NGAI, J., AND SCOTT, K. The molecular basis for water taste in drosophila. *Nature* 465, 7294 (May 2010), 91–95.
- [14] CLYNE, P. J., WARR, C. G., AND CARLSON, J. R. Candidate taste receptors in drosophila. *Science* 287, 5459 (March 2000), 1830–1834.
- [15] DAHANUKAR, A., FOSTER, K., VAN DER GOES VAN NATERS, W. M., AND CARLSON, J. R. A gr receptor is required for response to the sugar trehalose in taste neurons of drosophila. *Nat Neurosci* 4, 12 (December 2001), 1182–1186.
- [16] DAHANUKAR, A., LEI, Y.-T., KWON, J. Y., AND CARLSON, J. R. Two gr genes underlie sugar reception in drosophila. *Neuron* 56, 3 (November 2007), 503–516.

- [17] DANKERT, H., WANG, L., HOOPFER, E. D., ANDERSON, D. J., AND PERONA, P. Automated monitoring and analysis of social behavior in drosophila. *Nature Methods* 6, 4 (April 2009), 297–303.
- [18] DUNIPACE, L., MEISTER, S., MCNEALY, C., AND AMREIN, H. Spatially restricted expression of candidate taste receptors in the drosophila gustatory system. *Curr Biol* 11, 11 (June 2001), 822–835.
- [19] FALK, R., BLEISER-AVIVI, N., AND ATIDIA, J. Labellar taste organs of drosophila melanogaster. *J Morphology* 150, 2 (October 1976), 327–341.
- [20] GORDON, M. D., AND SCOTT, K. Motor control in a drosophila taste circuit. *Neuron* 61, 3 (February 2009), 373–384.
- [21] HAMADA, F. N., ROSENZWEIG, M., KANG, K., PULVER, S. R., GHEZZI, A., JEGLA, T. J., AND GARRITY, P. A. An internal thermal sensor controlling temperature preference in drosophila. *Nature* 454 (July 2008), 217–220.
- [22] HIROI, M., MEUNIER, N., MARION-POLL, F., AND TANIMURA, T. Two antagonistic gustatory receptor neurons responding to sweet-salty and bitter taste in drosophila. *J Neurobiol.* 61, 3 (December 2004), 333–342.
- [23] INAGAKI, H. K., DE LEON, S. B.-T., WONG, A. M., JAGADISH, S., ISHIMOTO, H., BARNEA, G., KITAMOTO, T., AXEL, R., AND ANDERSON, D. J. Visualizing neuromodulation in vivo: Tango-mapping of dopamine signaling reveals appetite control of sugar sensing. *Cell* 148, 3 (February 2012), 583–595.
- [24] ISHIMOTO, H., MATSUMOTO, A., AND TANIMURA, T. Molecular identification of a taste receptor gene for trehalose in drosophila. *Science* 289, 5476 (July 2000), 116–119.
- [25] JA, W. W., CARVALHO, G. B., MAK, E. M., DE LA ROSA, N. N., FANG, A. Y., LIONG, J. C., BRUMMEL, T., AND BENZER, S. Prandiology of drosophila and the cafe assay. *PNAS* 104, 20 (May 2007), 8253–8256.
- [26] JIAO, Y., MOON, S., WANG, X., REN, Q., AND MONTELL, C. Gr64f is required in combination with other gustatory receptors for sugar detection in drosophila. *Curr Biol* 18, 22 (November 2008), 1797–1801.
- [27] JIAO, Y., MOON, S. J., AND MONTELL, C. A drosophila gustatory receptor required for the responses to sucrose, glucose, and maltose identified by mrna tagging. *PNAS* 104, 35 (August 2007), 14110–14115.
- [28] KITAMOTO, T. Conditional modification of behavior in drosophila by targeted expression of a temperature-sensitive shibire allele in defined neurons. *J Neurobiol* 47, 2 (May 2001), 81–92.

- [29] LIU, C., PLAÇAIS, P.-Y., YAMAGATA, N., PFEIFFER, B. D., ASO, Y., FRIEDRICH, A. B., SIWANOWICZ, I., RUBIN, G. M., PREAT, T., AND TANIMOTO, H. A subset of dopamine neurons signals reward for odour memory in drosophila. *Nature* 488 (August 2012), 512–516.
- [30] LOPEZ, M. R., SERGIYENKO, O., AND TYRSA, V. Machine vision: Approaches and limitations, computer vision, xiong zhihui (ed.). *InTech* (2008).
- [31] MANZO, A., SILIES, M., GOHL, D. M., AND SCOTT, K. Motor neurons controlling fluid ingestion in drosophila. *PNAS* 109, 16 (April 2012), 6307–6312.
- [32] MARELLA, S., MANN, K., AND SCOTT, K. Dopaminergic modulation of sucrose acceptance behavior in drosophila. *Neuron* 73, 5 (March 2012), 941–950.
- [33] MIYAZAKI, T., AND ITO, K. Neural architecture of the primary gustatory center of drosophila melanogaster visualized with gal4 and lexa enhancer-trap systems. *J Comp Neurol* 518, 20 (October 2010), 4147–4181.
- [34] NAYAK, S. V., AND SINGH, R. N. Sensilla on the tarsal segments and mouthparts of adult drosophila melanogaster meigen. *Int J of Insect Morphology Embryology* 12, 5-6 (1983), 273–291.
- [35] NITABACH, M. N., AND TAGHERT, P. H. Organization of the drosophila circadian control circuit. *Current Biology* 18, 2 (January 2008), R84–R93.
- [36] OLSEN, S. R., AND WILSON, R. I. Cracking neural circuits in a tiny brain: new approaches for understanding the neural circuitry of drosophila. *Trends Neurosci* 31, 10 (October 2008), 512–520.
- [37] ROBERTSON, H. M., WARR, C. G., AND CARLSON, J. R. Molecular evolution of the insect chemoreceptor gene superfamily in drosophila melanogaster. *PNAS* 100, Suppl 2 (November 2003), 14537–14542.
- [38] SCHNAITMANN, C., VOGT, K., TRIPHAN, T., AND TANIMOTO, H. Appetitive and aversive visual learning in freely moving drosophila. *Front. Behav. Neurosci.* 4 (March 2010), 10.
- [39] SCOTT, K., BRADY, R., CRAVCHIK, A., MOROZOV, P., RZHETSKY, A., ZUKER, C., AND AXEL, R. A chemosensory gene family encoding candidate gustatory and olfactory receptors in drosophila. *Cell* 104, 5 (March 2001), 661–673.
- [40] SHIRAIWA, T., AND CARLSON, J. R. Proboscis extension response (per) assay in drosophila. *J Vis Exp.*, 3 (2007), 193.
- [41] STAFFORD, J. W., LYND, K. M., JUNG, A. Y., AND GORDON, M. D. Integration of taste and calorie sensing in drosophila. *J Neurosci.* 32, 42 (October 2012), 14767–14774.

- [42] STOCKER, R. The organization of the chemosensory system in drosophila melanogaster: a review. *Cell Tissue Res* 275, 1 (January 1994), 1–26.
- [43] STOCKINGER, P., KVITSIANI, D., ROTKOPF, S., TIRIÁN, L., AND DICKSON, B. J. Neural circuitry that governs drosophila male courtship behavior. *Cell* 121, 5 (June 2005), 795–807.
- [44] TETSUYA, M., AND HUBERT, A. Suppression of male courtship by a drosophila pheromone receptor. *Nat Neurosci* 11, 8 (August 2008), 874–876.
- [45] THISTLE, R., CAMERON, P., GHORAYSHI, A., DENNISON, L., AND SCOTT, K. Contact chemoreceptors mediate male-male repulsion and male-female attraction during drosophila courtship. *Cell* 149, 5 (May 2012), 1140–1151.
- [46] THORNE, N., CHROMEY, C., BRAY, S., AND AMREIN, H. Taste perception and coding in drosophila. *Curr Biol* 14, 12 (June 2004), 1065–1079.
- [47] TODA, H., ZHAO, X., AND DICKSON, B. J. The drosophila female aphrodisiac pheromone activates ppk23(+) sensory neurons to elicit male courtship behavior. *Cell Rep* 1, 6 (2012), 599–607.
- [48] TOSHIMA, N., AND TANIMURA, T. Taste preference for amino acids is dependent on internal nutritional state in drosophila melanogaster. *J Exp Biol.* 215, 16 (August 2012), 2827–2832.
- [49] VALENTE, D., GOLANI, I., AND MITRA, P. P. Analysis of the trajectory of drosophila melanogaster in a circular open field arena. *PLoS ONE* 2 (October 2007), e1083.
- [50] VON PHILIPSBORN, A. C., LIU, T., YU, J. Y., MASSER, C., BIDAYE, S. S., AND DICKSON, B. J. Neuronal control of drosophila courtship song. *Neuron* 69, 3 (February 2011), 509–522.
- [51] WANG, Z., SINGHVI, A., KONG, P., AND SCOTT, K. Taste representations in the drosophila brain. *Cell* 117, 7 (June 2004), 981–991.
- [52] WEISS, L., DAHANUKAR, A., KWON, J., BANERJEE, D., AND CARLSON, J. The molecular and cellular basis of bitter taste in drosophila. *Neuron* 69, 2 (January 2011), 258–272.
- [53] WISOTSKY, Z., MEDINA, A., FREEMAN, E., AND DAHANUKAR, A. Evolutionary differences in food preference rely on gr64e, a receptor for glycerol. *Nat Neurosci* 14, 12 (November 2011), 1534–1541.

