Tracking the Motion of Box Jellyfish

Tobias Kjellberg

Master's thesis
2014:E26

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Lunds Teknologiska Högskola

Master Thesis

Tracking the Motion of Box Jellyfish

Author:
Tobias Kjellberg

Supervisor:
Kalle Åström

A thesis submitted in fulfilment of the requirements for the degree of Master of Science in Engineering, Computer Science and Engineering

June 2014

Abstract

Lunds Tekniska Högskola

Master of Science in Engineering, Computer Science and Engineering

Tracking the Motion of Box Jellyfish

by Tobias Kjellberg

This master's thesis investigates the possibilities of detecting the rhopalia of box jellyfishes, i.e the eyes of the box jellyfish. Each box jellyfish has four rhopalia and each of the rhopalium consists of six light sensing eyes which they use to interpret the surroundings. In this project we have worked with fifteen film sequences all consisting of different box jellyfishes and recorded under different light settings. The framework for detecting the rhopalia is divided into three parts, in order; detection, clustering and tracking. The input in the detection step is the grayscale image of the box jellyfish and in the output possible rhopalia are marked as detections. These detections are then sent into the clustering step which filters out the noise in the picture and saves clusters of detections. The reason for saving these clusters is because a rhopalium appears as a dark disc with a radius of around 13 pixels and thus should produce many detections. In the final step, four clusters are selected from all clusters as the correct rhopalia. The choice has been made to focus on the detection step in this thesis, leaving the cluster and tracking step with only one algorithm each. A combination of a detection method, a clustering- and a tracking algorithm is called a system.

To be able to detect a rhopalium, a set of data points, a pattern, is used to compare values (light intensity) in order to capture the appearance of a rhopalium. The more data points the longer the execution time so this needs to be done with minimal amount of data points but still enough to capture the visual aspect of the rhopalium.

The best performing systems have patterns of a small disc inside the rhopalium and a larger circle just outside. The values of the pixels outside should be greater (brighter) than the ones inside. But because of some artefacts and noise in the pictures not all need to be darker, instead, a threshold is used, e.g 78 % of the pixel values outside should be greater than the ones inside. This results in a accuracy of 98%.

Acknowledgements

This report is my master thesis for the degree in Master of Science in Engineering, Computer Science and Engineering. This project was performed within the Mathematical Institution at Lund Institute of Technology (LTH). My supervisors for this project were Kalle Åström, Magnus Oskarsson and Tobias Palmér who all belong at the Centre for Mathematical Sciences. I would like to thank everybody who helped me with this project. Especially my supervisors who kept me on track and helped me with whatever problem I was facing. I would also like to thank Ronald Petie for providing the data and making this project even possible and also a special thanks to Kalle Åströms son, Oskar Åström, who took the time and had the patience to create an ground truth for all the pictures.

Contents

Chapter 1

Introduction

The road to understanding the human brain is a very long and crooked one. There are many ways to approach this problem and some of them is more plausible than others. Since not many human beings would volunteer to be experimented on and maybe even die in the name of science, the majority of the experiments is performed on animals. But still there are ethical guidelines to follow when experimenting on animals, the potential gain of the experiment is set in relationship to the suffering of the animal[1]. This also implies that the more primitive an organism is, the easier it is to get permission to experiment on it.

The brain can be found in all vertebrate and in almost all invertebrate except for sponges, adult sea squirts, starfish and jellyfishes. But even though these animals don't have any brain, there are nerve signals that can be studied. This is also done much easier than if the animal had a brain. This is the reason why the vision of the box jellyfish, Tripedalia Cystophora, has been examined by Ronald Petie at the institue of Biology. He did this by tethering the animal in a box and turning on and off the light in different quadrants to see how the movement of the animal changed. In this thesis, many of the film sequences has been gathered and different algorithms for detecting the placement of the four rhopalia has been created and evaluated.

1.1 Similar work

The are different companies in the field of studying animal behavior via image analysis. One well known company is Noldus who has developed an array of different software. One of their software is EthoVision® XT which detects and tracks a wide array of smaller animals (rodents, spiders and fish for example) in a three-dimensional space. The only thing that is required is a camera to be mounted above or under the animal

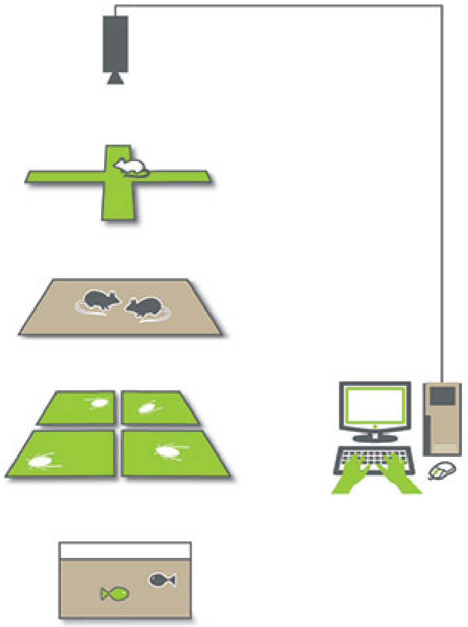


Figure 1.1: Different simple setups to track smaller animals by Noldus software EthoVision® XT.

to be studied. The software then take care of the rest, i.e no sensors is needed to be attached to the animals. Some applications for this is to study some behavioral responses due to odors such as pheromones and plant volatile compounds, characterization of the swimming behavior in fish and horizontal and vertical movement of animals in a cage, see figure 1.1[2]. In 2011, 60 % of all publications (more than 3000 worldwide) on video tracking used EthoVision XT[3]. The reasons for studying smaller animals by image analysis are many and can be used in many different areas. EthoVision XT and similar software fully autmates the research, increasing productivity and efficiency, while reducing human error, giving the researcher more time for other studies.

Another company in the same field is Biobserve and their equivalent of Noldus EthoVision XT is called Viewer. The differance between EthoVision XT and Viewer is that the latter doesn't only track the center of the animal but also the nose and tail. Thus Viewer can provide a much more detailed behavioral analysis and automate complex experiments like basic object recognition and social interaction. This doesn't mean that the Viewer is better than EthoVision XT, they simply fill different purposes. The fact that Viewer knows the x- and y-coordinate of three points (the nose, body and tail) the software can automatically detect and count behaviors like head wagging, head stretches and tail moves. This also entails that freezing behavior and movements are automatically scored. These parameters and others can be adjusted to make the system score according to your preference. In Viewer you can create different zones if you would like to observe, quantify or compare the behavior of an animal in one or more defined areas of the experiment arena, see figure 1.2[4].

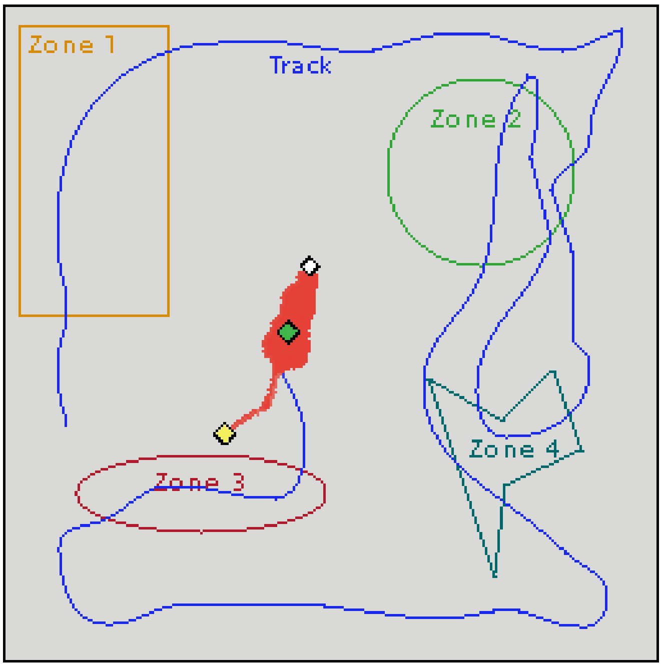


Figure 1.2: Biobserves animal tracking software, Viewer. As seen in this image, not only the center of the animal is tracked but also the nose and tail. The zones are created manually to record specific behavior in these areas.

1.2 Image analysis

The definition of image analysis is the subject concerned with extraction of meaningful information from images, digital images specifically[5]. Today this is highly used in barcode reading as in the supermarket. These bar codes is very feasible for a computer to interpret but extremely hard for a human being. The reason for this being easy for computers is due to the binary nature of the barcodes; it only uses black and white and represent data by varying the widths and spacings of the parallel lines (one dimensional) [6].

More advanced image analysis, compared to barcode reading, and widely used today is face recognition which extracts features from an image of the subjects face. The features extracted can be the relative postion, size, and/or shape of the nose, jaw, cheekbones etc[7]. These features are then used when searching for other images with matching features[8]. An example where this technique was used was at Super Bowl XXXV in January 2001 where the police in Tampa Bay used facial recognition software to identify potential criminals and terrorits attending the event[9, 10]. Face recognition today is widely used in camera software/apps and webpages were people can be tagged in pictures, e.g Facebook and Google+.

Although facial recognition has come a long way since its beginning, the human brain is still much better. The reason for this is that the human visual cortex is an

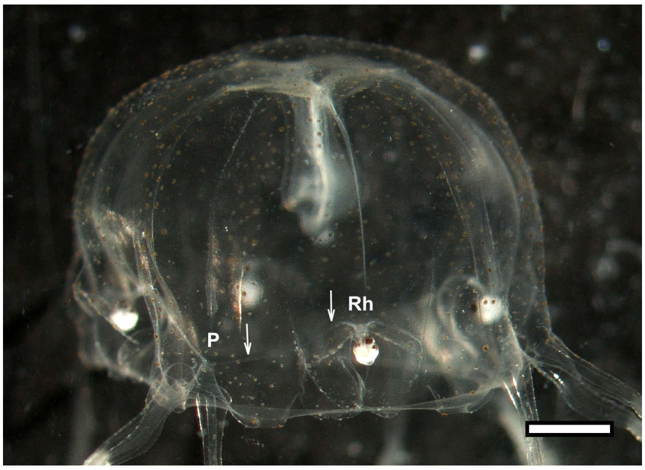


Figure 1.3: The Tripedalia Cystophora viewed from the side where *Rh* marks the location of one rhopalium

excellent image analysis apparatus. This makes the brain very formidable of extracting higher-level information and is also the reason why computers haven't completely replaced humans in areas such as security, remote sensing and medicine, as of today that is[5]. The reason for computers still struggling with this is because of vast amount of data that is needed to be processed and analyzed in pictures. In a colored .bmp picture each pixel has 3 values (one for each color channel; Red, Green, Blue) that ranges between 0-255 which means that a single pixel can have 256^3 (about 16.5 million) different values (colors) . This variation combined with the values of neighboring pixels in order to detect a specific pattern makes it impossible to construct a generic image analysis technique for a wide range of tasks. This means that every image analysis technique is used for a small range of tasks which is the case of this thesis.

1.3 Objective

The objective of this thesis to detect the rhopalia of the box jellyfish, *Tripedalia cystophora*, in video recordings, see figure 1.6 for an example. Each rhopalium consists of six light sensing eyes which the jellyfish use to interpret the surroundings. In each image in the video sequence the rhopalium has the visual appearance of a small black disc. Each box jellyfish has four rhopalia distributed evenly along the velarium, the edge of the body seen in fig 1.3. In this thesis we have worked with are 15 different film sequences, each with different box jellyfishes, light conditions and artefacts, forcing the solution to be more of a generic detection-algorithm. The method utilized to find the rhopalia in the film sequences is divided into three steps; detecting, clustering and selection which will be described thoroughly later in the thesis. Similar work on motor response from

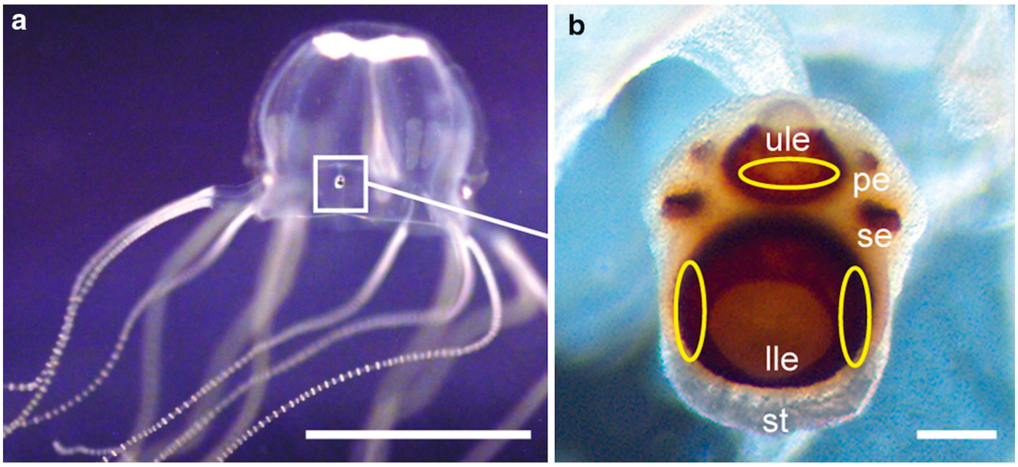


FIGURE 1.4: The *T. cystophora* (**a**) with a zoomed in image of the rhopalium (**b**). In **b** the six eyes are visible where **ule** - upper lens eye, **lle** - lower lens eye, **se** - slit eye and **pe** - pit eye.

controlled visual stimuli can be seen in e.g [11] and [12]. The aim of this master's thesis is to:

- Find all the rhopalia in every film sequence.
- Focus on evaluating different detection-methods since this is the crucial step.
- Performing the above in almost-realtime.

1.4 The jellyfish - *Tripedalia cystophora*

The box jellyfish, *T. cystophora*, is a roughly 1 cm sized box jellyfish that have a very special visual system,[13, 14], and whose habitat is in the mangrove swamps. It preys on small copepods that swarm between the roots of the mangrove trees. The copepods gather in light shafts created by the canopy above. The box jellyfish uses its visual system to detect these light shafts but it cannot see the copepods themselves. The interesting part in this thesis is their visual systems which is distributed at four sensory organs, the rhopalia. Each rhopalium consists of six different eyes: the upper and lower lens eyes, the pit eyes, and the slit eyes [15–20], see figure 1.4 [21]. The ones looking downwards are also directed inwards towards the bell which results in the box jellyfish to "look through" its own bell. This unique visual system enables the box jellyfish to display visually guided behaviour that appear remarkable for such "simple" animal[22–24].

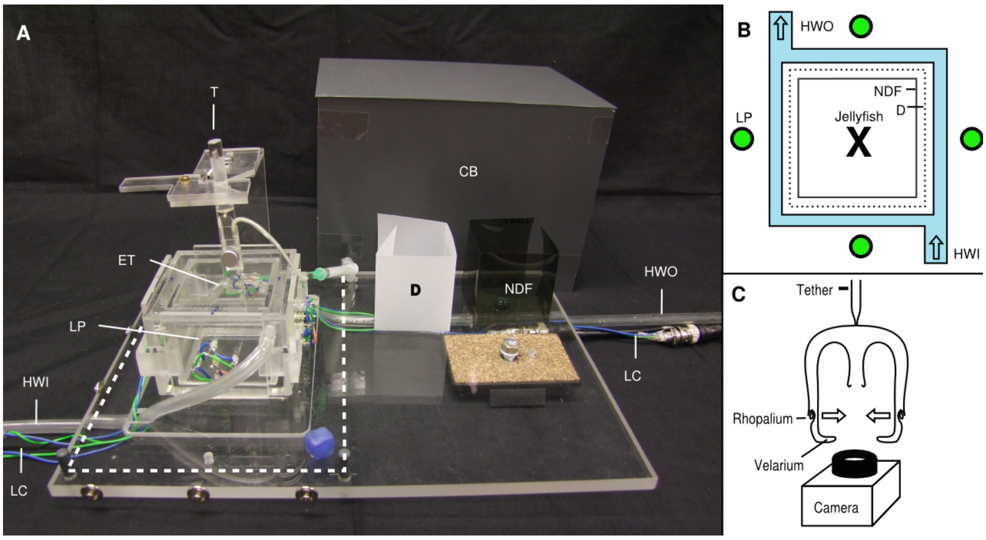


FIGURE 1.5: Experimental setup. The blue-green LEDs can be seen at each vertical wall (**LP**)and the paper diffuser (**D**) and the density filter (**NDF**) can be seen next to the setup. The placement of *T. Cystophora* in the box together with the LEDs can be seen in **B** and in **C** the tethering and the location of the animal is presented relative to the camera.

1.5 Experimental setup

To record each film sequence the jellyfish must be set up in a special environment, see figure 1.5 . This environment was a plexiglass tank with inside dimensions of $5 \times 5 \times 5$ cm filled with suitable water and the right temperature for the box jellyfishes. Each wall then was fitted from the outsude by four blue-green LEDs to provide with the light stimuli and a paper diffuser was used to make the illumination more even. To eliminate visual cues coming from outside and thus making sure that the eyes looking up through the water surface do not recieve direct visual stimulation, a box was placed over the set-up. To hold the box jellyfish in place and thus being able to study its movement the *T. Cystophora* was tethered in the back of the tank, using a glass pipette with gentle suction, opposite of the camera. To increase the contrast between the dark and lit panels a neutral density filter was placed in front of the walls and the diffuser. This was done as a precaution so that the jellyfish wouldn't respond to the reflections that would otherwise appear. Switching these panels on or off was the behavioural trigger. The colour of the LEDs matched the maximum spectral sensitivity of the animals and had a peak emission at 500 nm. The image sequences were recorded with a high-speed camera operated at 150 frames per second.

1.6 Data set

In this section we will present detailed information of the data used in this thesis.

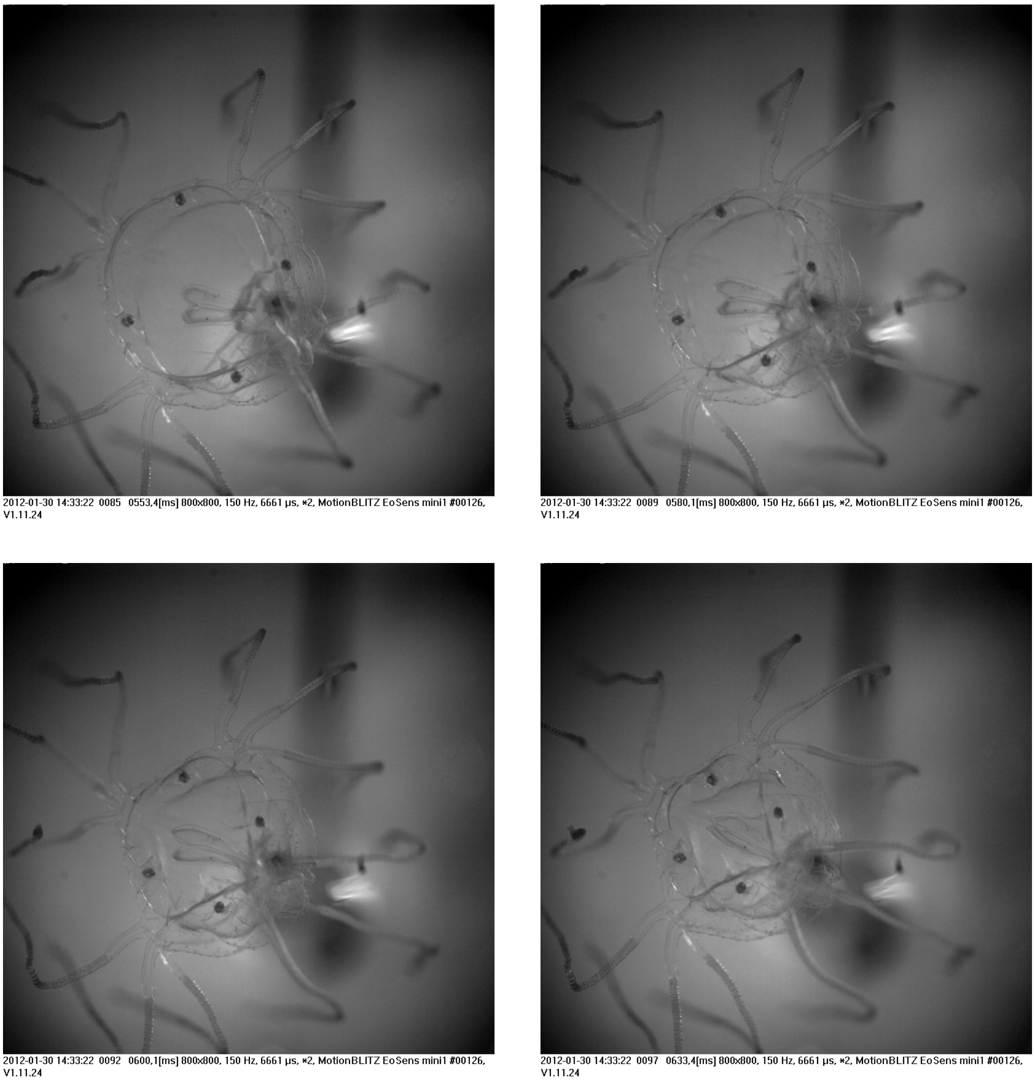

2012-01-30 14:33:22 0085 0553,4[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
2012-01-30 14:33:22 0089 0580,1[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
2012-01-30 14:33:22 0092 0600,1[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
2012-01-30 14:33:22 0097 0633,4[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24

Figure 1.6: A contracting box jellyfish from a film sequence.

1.6.1 The image sequences

Each of the 15 film sequence consists of about 100 grayscale images in an uncompressed format (.bmp). A selection of images can be seen in figure 1.6. Each grayscale picture is 800 × 864 pixels in size, where each pixel has one value between 0 and 255. The 15 film sequences have a lot of features in common but each film sequence still differentiates from the others in one or more aspects. Depending how the jellyfish is set up the light and shadows form differently which will make each film sequence different from the each other. Some typical differences can be observed in 1.7. Even though great measure has been taken in order to minimize artefacts in the film sequences the difference between them can be quite large. Some of the film sequences are brighter, making it easier to find the rhopalia while some are darker and thus making it hard to distinguish the rhopalia from the background (figure 1.8 and 1.9). An artefact that is visible in all film sequences is a smudge on the camera lens that looks like an elongated rhopalium (A in figure 1.10). The large resemblance between the smudge and a rhopalium makes it very hard for the detection algorithm to separate them, making the smudge a false positive, incorrectly regarded as a rhopalium, in every picture. The tethering of the animal is also visible

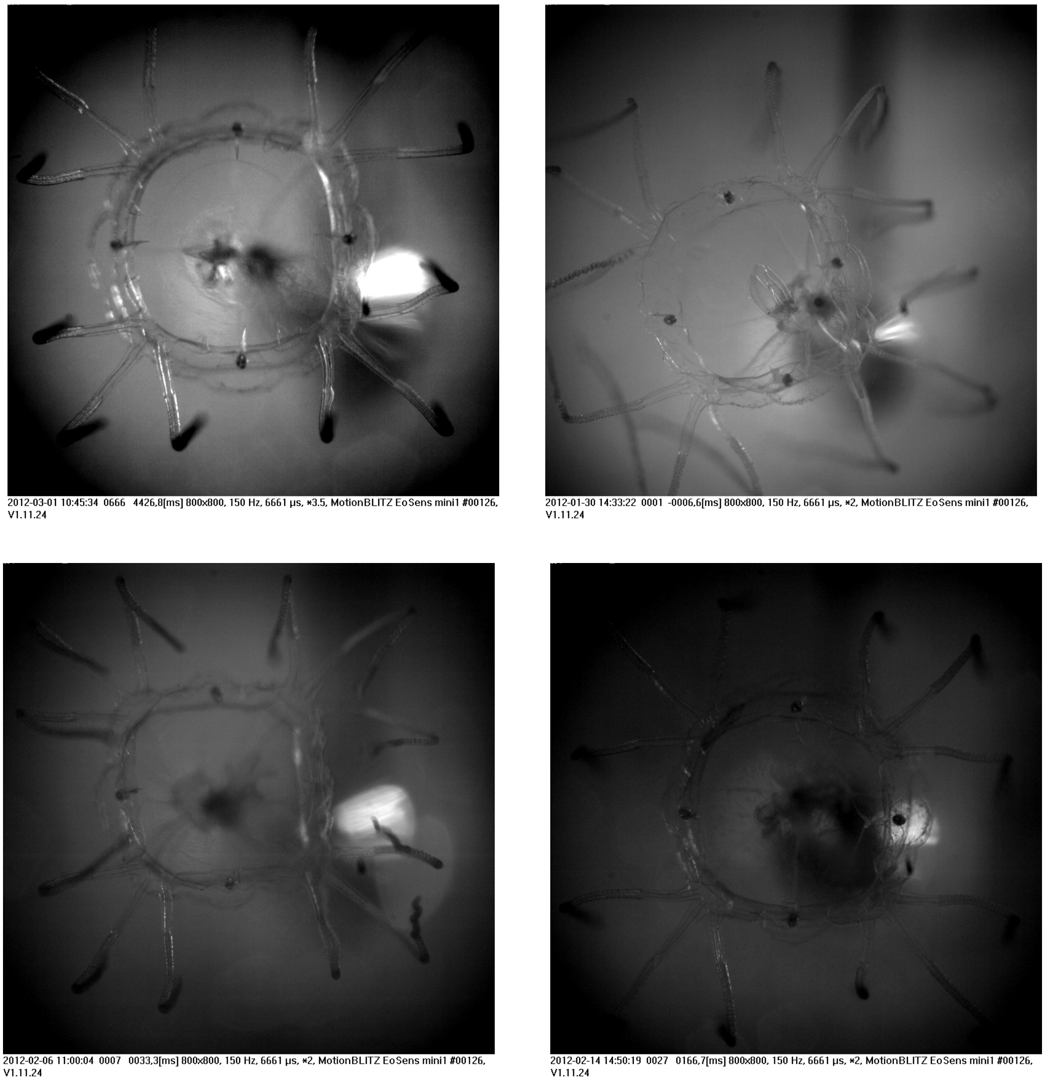

2012-03-01 10:45:34 0666 4426,8[ms] 800x800, 150 Hz, 6661 µs, *3.5, MotionBLITZ EoSens mini1 #00126, V1.11.24
2012-01-30 14:33:22 0001 -0006,6[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
2012-02-06 11:00:04 0007 0033,3[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
2012-02-14 14:50:19 0027 0166,7[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24

Figure 1.7: Example input frames from a number of different film sequences. Notice the high variance in lightning conditions. In some frames the rhopalia are barely discernible and in many frames there are structures that have an appearance very similar to the rhopalia.

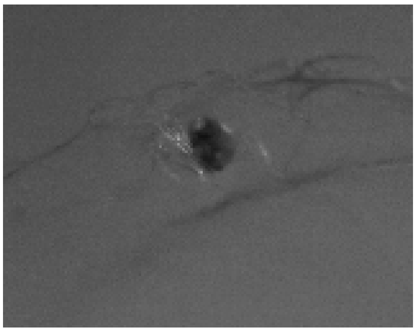


Figure 1.8: Bright surroundings.

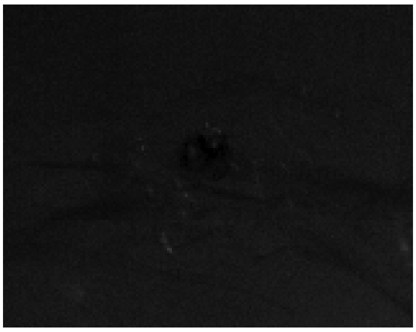


Figure 1.9: Dark surroundings.

in all film sequences. It is the big black shadow in the middle of the jellyfish(B in figure 1.10). This tether-shadow causes problems when the jellyfish is contracting and the rhopalia is moving over the tether-shadow. Since every rhopalium is rather dark compared to the rest of the jellyfish the rhopalium blends with the tether-shadow and thus making it almost impossible to make out the rhopalium from the shadow.

Since the box jellyfish is moving in every film sequence, some parts of the jellyfish is moving in and out of focus. This makes the rhopalia sometimes shift out of focus and thus making it hard to differentiate from the surroundings which can be seen in figure 1.11. The physical nature of the *T. Cystophora*, i.e being transparent, also affects the images in some meaning. The light bounces off and in the animal and creates refraction on the surroundings. Refraction is a phenomenon that often occures when waves, light waves in this case, travel from a medium with a given refractive index, the water, to a medium with another, the box jellyfish, at an oblique angle. At the edge between

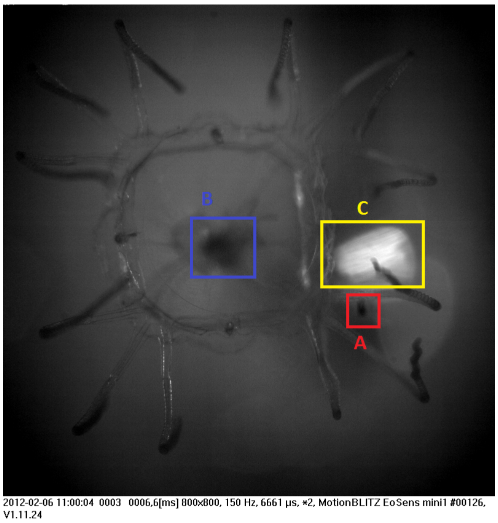

B
C
A
2012-02-06 11:00:04 0003 0006,6[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126,
V1.11.24

Figure 1.10: Different artifacts seen in every picture. A is the smudge, B is the tether shadow and C is the glare.

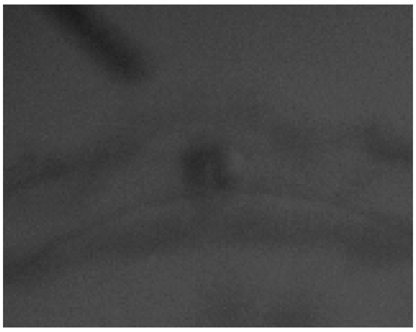


Figure 1.11: An unfocused rhopalium.

the water and the *T. Cystophora* the wave's phase velocity is altered usually causing a change in direction[25]. This change of direction will cause the light to focus at some points and deter from other, causing spots with greater intensity and shadows, much like a magnifying glass. These shadows can end up with the same characteristics as a rhopalium but mostly don't. Another minor problem with the transparent jellyfish is when some of the body overlaps the rhopalia and thus making it less visible. This is only a problem when the light is forced to go through the whole length of the jellyfish.

In all film sequences we can see a reflection, almost looking like a glare, located to the right of the center(C in figure 1.10). This glare is very visible but the effect on the rhopalia is pretty much non-existent. The reason why is that it's located behind the box jellyfish, and the rhopalia is dense enough to not let any light through, making it still available for detection.

1.6.2 The ground truth

To be able to evaluate the tracking quality and thus compare different algorithms we need an answer sheet. The answer sheet consists of the correct coordinates for each rhopalium, that is the x- and y-coordinate, which means four pairs of coordinates per picture. These were obtained by manually clicking on each and every rhopalium throughout the film sequences. Since the rhopalium isn't distinguishable in some pictures the position was, in some cases, predicted depending on its previous location and the position of the other three rhopalia.

Chapter 2

The framework

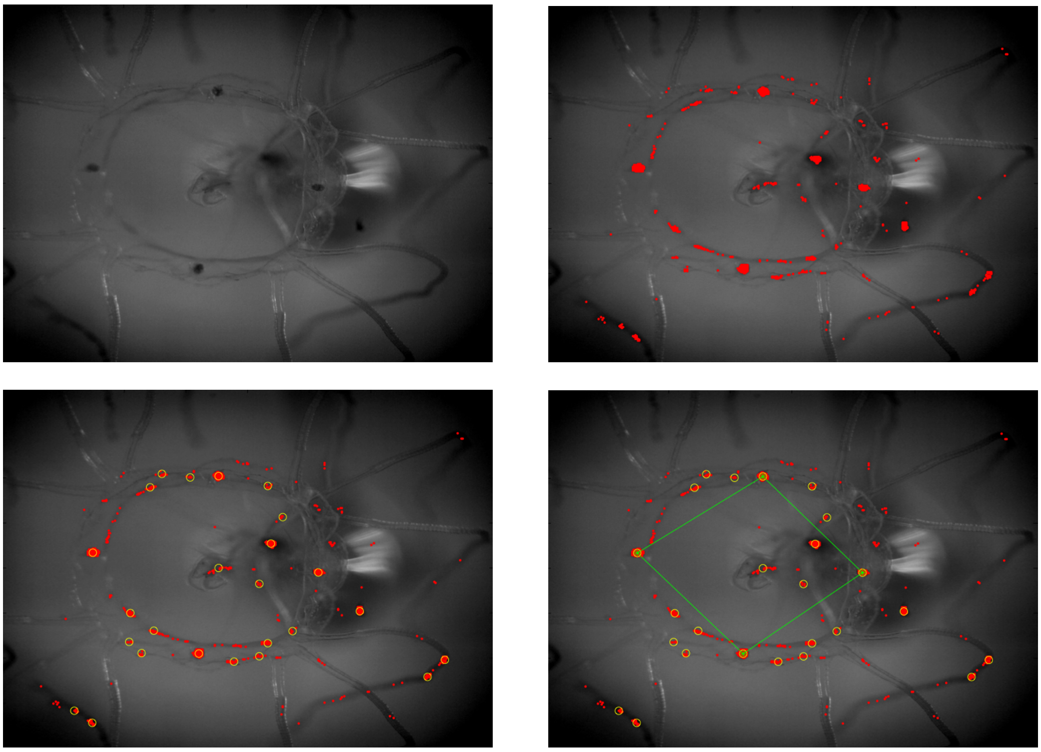


Figure 2.1: The different stages in the framework

2.1 System overview

The framework in this thesis consists of three separated stages in which the output from the previous stage is the input in the following stage until the last stage is reached, see figure 2.2 for a system overview. The stages are, in order; detection, clustering and tracking seen in figure 2.1. The first input, the input to the detection step, is an image,

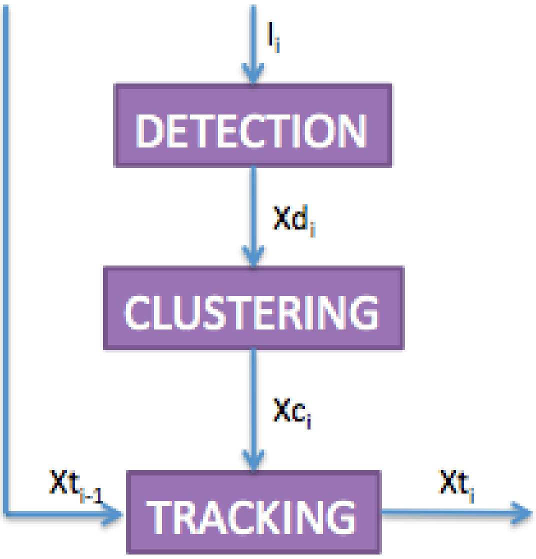


Figure 2.2: An overview of the system. The detection algorithm is executed for each frame I_i. The current detection algorithm will then produce a number of points Xd_i which are then sent further down the pipeline to the clustering algorithm. The clustering step will reduce the number of detections and group these in to clusters, Xc_i. In the last step, the tracking algorithm takes the clusters togehter with the four tracked positions in the previous frame, Xt_{i-1} to produce the final four tracked points, Xt_i.

i from the film sequence. Here, based on the current detection-algorithm, the image is processed and coordinates of detections, Xd_i, are sent to the next step, the clustering step. In the clustering step the detections are clustered into a set of points, Xc_i, in order to remove alone detections to improve positional accuracy in the detected points. When the clustering is done, the list of cluster coordinates is sent to the tracking step together with the previous frame's four tracked points, Xt_{i-1}. Here the tracking step decides which four coordinates are the correct ones i.e the coordinates corresponding to the four rhopalia in the picture, Xt_i.

In this thesis the focus will be on the detection step. This means that the remaining steps, clustering and tracking, will only have one method each, and we evaluate a number of different detection algorithms. This means that the only difference between one set up of a system (the combination of one detection method, one clustering method and one tracking method) from another is the detection step. Still this leads to entirely different results as you will see.

The reason for focusing on the detection step and not on one of the other is that the detection step is the first step and thus any detections missed here will not magically appear in the clustering or tracking step, making this step more crucial. Also since the input size in the following steps is always less or equal to the current step we lose more and more information the further down the pipeline we travel, limiting the amount of possible outputs in the following steps. The clustering and tracking step will be described in detail below but the detection step will get a more general description only to get into more detail when the different systems are explained.

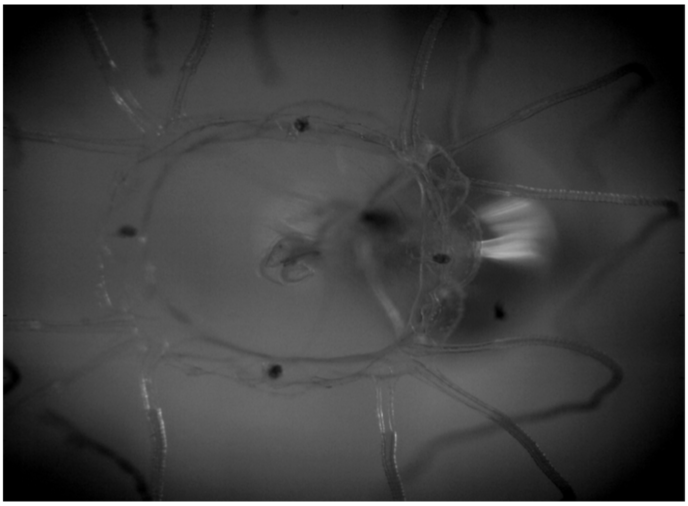


FIGURE 2.3: Original picture.

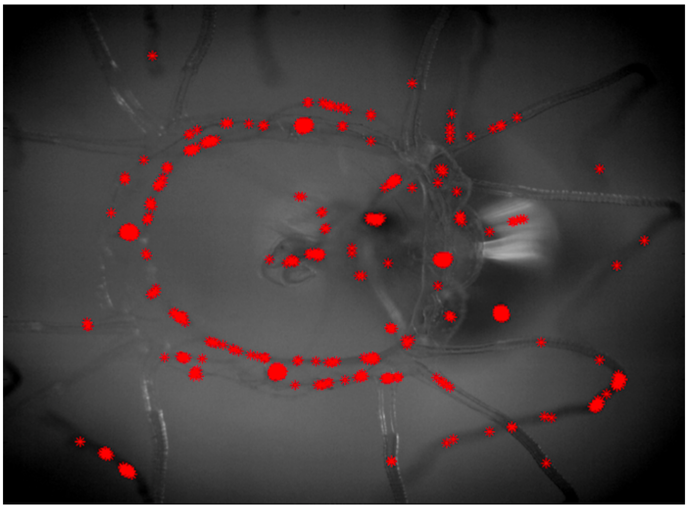


FIGURE 2.4: The detections.

2.2 Detection step

There are many different methods to use when trying to find specific objects in images. Two groups of detection method is appearance-based and feature-based detection methods. Appearance-based methods uses templates (example images) of the objects desired to detect and then compare these templates to images to find any resemblance. An example of appearance-based detection method is edge detection where the edges in the image and template is compared. The feature-based detection methods searches in images to find feasible matches between object features and image features . One constraint is that the object from where the features is extracted should account for all the feasible matches[26]. An example of feature-based detection is SIFT (Scale invariant feature transform) where keypoints of the object of intrest is extracted and stored in a database. An object is then recognized in an image by individually comparing the features from the new image to this database and finding candidate matches based on some distance measure, e.g. the Euclidean distance of their feature vectors[27].

The detection methods in this thesis are more appearance-based than feature-based, mostly because we're using templates, patterns, when deciding if current pixel should be a dection or not, but this will be discussed in greater detail in chapter 3. As seen in figure 1.8 the rhopalium is roughly a dark disc with brighter surroundings. Here the idea is to use this information to extract each rhopalium from each image but since the rhopalia could differ in size and in brightness/darkness this task isn't that straight forward and results in many false positives which can be seen in figure 2.4. Due to these large differances between each film sequence our detection algorithm needs to have a more breadth to it and thus not only finding the rhopalia, but false positives as well. So the goal here is to create a detection method which minimizes the false positives but still manages to find all the rhopalia and does this within reasonable time, i.e minimize the amount of data to check.

2.3 Clustering step

The task of a clustering algorithm is to group a set of objects in such manner that the objects in the same group (called cluster) have more in common than those in other groups. This task has a wide range of usability, such as data mining and statistical data analysis including machine learning, information retrieval, bioinformatics and in this case; pattern recognition and image analysis[28]. There are various algorithms to use when solving the clustering task and they differ significantly in their notion of what constituates a cluster. One example is the k-means algorithm that aims to partition n observations into k clusters (the number k is chosen arbitrarily) in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. When an observation has been added to a cluster, the mean of that cluster is recalculated to include the new observation. This makes this clustering algorithm sensitive to outliers[28]. The aim of the clustering step in this thesis is to reduce the amount of detections to a much smaller amounts of clusters. This particular clustering method first smoothens the image, I_i by using Gaussian blur [29] to obtain I_{sm} and get rid of the most of the noise and then correlating the detections with the local minima,

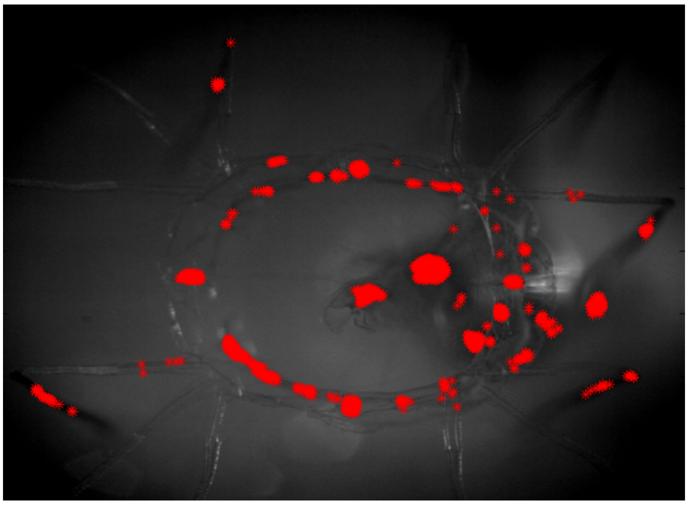


Figure 2.5: The detections.

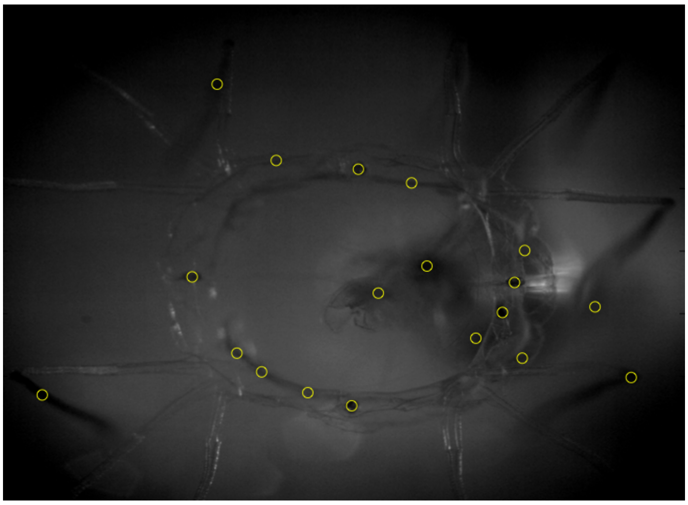


Figure 2.6: The clusters formed from the detections.

X_{loc} in the smoothened image. The reason for correlating the detections with the local minima is because a rhopalium should present itself as a dark disc in the blurred image, i.e a local minima. We then calculate the number of detections within a vicinity of each X_{loc},

$$N_{loc}(j) = \sum_{k=1}^{N_d} (||X_{loc}(j) - Xd(k)||_2 < \epsilon_{cluster}), \tag{2.1}$$

and if there are a minimum number N_{min} of detections, then we add this local minimum to our clustered points Xc_i,

$$Xc_i = \{X_{loc}(j) | N_{loc}(j) \geq N_{min}\}. \tag{2.2}$$

The resulting clusters from detections can be seen in figure 2.6.

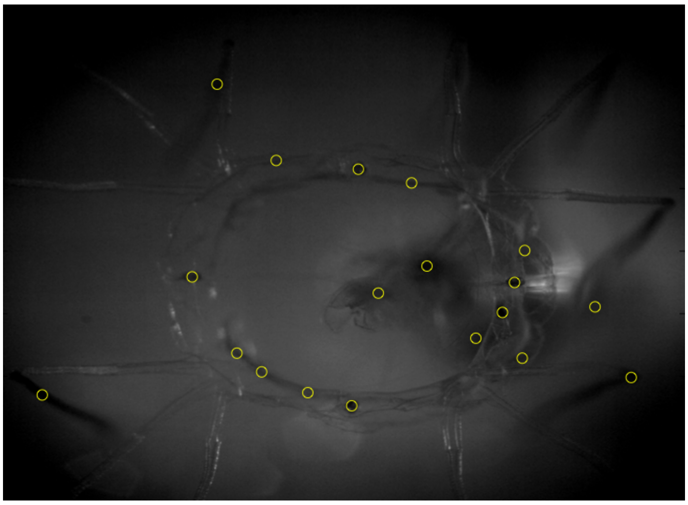


Figure 2.7: The clusters.

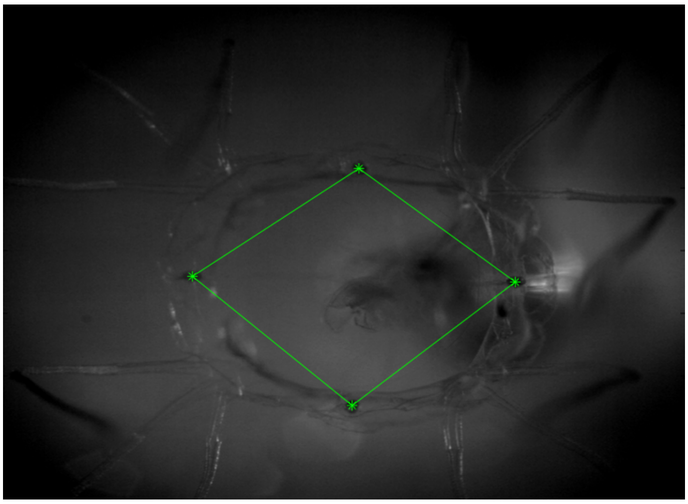


Figure 2.8: The final four coordinates.

2.4 Tracking step

Tracking, or more specifically, video tracking, is the process of locating target objects in consecutive video frames. This technique has various uses: human-computer interaction, security and surveillance, traffic control, medical imaging etc.[30]. This task can be rather difficult when the objects are moving fast in relation to the frame rate. Another aggravating situtation is when the object changes orientation over time. For these situations video tracking systems must utilize a motion model which account for how the appearance of the target might change. There are a variety of tracking algorithms, each with different strenghts and weaknesses. For this reason it's very important to consider the intended use when selecting the tracking algorithm. In the field of visual tracking systems there are two major approaches: *target representation and localisation* and *filtering and data association*. The differance between those two are that the former is a bottom-up process that gives a variety of tools for identifyng the moving object and its tracking performance is very dependent on the algorithm. The latter is mostly a top-down process and more complex which involves prior knowledge about the scene or object, dealing with dynamics of the object and different hypotheses evaluation. These type of tracking algorithms allow the tracking of complex objects and the movement behind obstructions of these objects. These type of algorithms also suit the situations when the camera isn't mounted on rigid foundation but on a moving ship for example. Example of this type of tracking algorithm is Kalman filter or Particle filter[31].

In this thesis the *target representation and localisation* is utilized and the tracking algorithm has a rather low complexity. Here, the tracking step is where the final four coordinates are chosen from the clusters. The result can be seen in figure 2.7 and 2.8. The input to the tracking algorithm for each film sequence is the ground truth for the first image. In the following frames the input consists of the four tracked points from the previous frame, Xt_{i-1} and a number of possible candidate points Xc_i. The four closest clusters within pixels will be the next tracked coordinates, i.e.

$$Xt_i(j) = \arg \min_{Xc_i(k)} ||Xt_{i-1}(j) - Xc_i(k)||_2, \quad j = 1, \ldots, 4. \tag{2.3}$$

Chapter 3

Detection methods

3.1 Disclaimer

In each of the following detection methods some arrangements have been made in order to make the runtime shorter. The first one is to crop the image so that we only use the area in the image where the rhopalia exists. This is also done to get rid of the lower text field where the image information (i.e time, date, camera settings etc) is and only acts as a place for possible false positives. Another measure taken in all detection methods except one is to subsample the image with a factor of 2, i.e omit each other pixel in both x-axis and y-axis. Doing this greatly reduces the time consumed by the detection step and, in our experience, without loosing too much information in order to detect the rhopalia.

3.2 General

As stated earlier, we're using template-based detection methods and all detection methods basically has the same course of action. The template is based on the notion that a rhopalium is a dark disc with brighter surroundings. This means that the pixels in the template near the rhopalium should be dark and brighter further away. Due to the goal of running this program in realtime, the templates has a rather sparse appearance, but enough to capture the visual appearance of a rhopalium. Among the detection methods there are basically two types of measures, one absolute and one relative. For the absolute measure we define an intensity threshold for the inner pixels, t_{in}, and one for the outer pixels, t_{out}. These thresholds, are obtained by iterating through all images and taking samples inside and outside of each rhopalium in the image. The mean intensities of both inside and outside are then stored in the thresholds respectively. For each pixel

$X_i(j)$ in the input image, I_i we can then define a number of inside and outside points, $\Omega_{in}(j)$ and $\Omega_{out}(j)$. The number of inside and outside pixels that fulfill the constraints are then counted, i.e

$$N_{abs}(j) = \sum(\Omega_{in}(j) \leq t_{in}) + \sum(\Omega_{out}(j) \geq t_{out}), \tag{3.1}$$

and

$$Xd_i = \{X_i(j) | N_{abs}(j) > N_{det}\}, \tag{3.2}$$

where N_{det} is some bound. For the relative measure we randomly compare n inside and outside pixels, and count how many of the inside pixels are darker than the outside pixels. So if we let $R(\Omega)$ denote a function that randomly chooses a point from the set Ω we have,

$$N_{rel}(j) = \sum_{k=1}^{n}(R(\Omega_{in}(j)) < R(\Omega_{out}(j)), \tag{3.3}$$

and

$$Xd_i = \{X_i(j) | N_{rel}(j) > N_{det}\}. \tag{3.4}$$

All detection methods will use either the absolute measure or the relative measure and how each differ from each other will be described in greater detail below.

3.3 Detection method 1

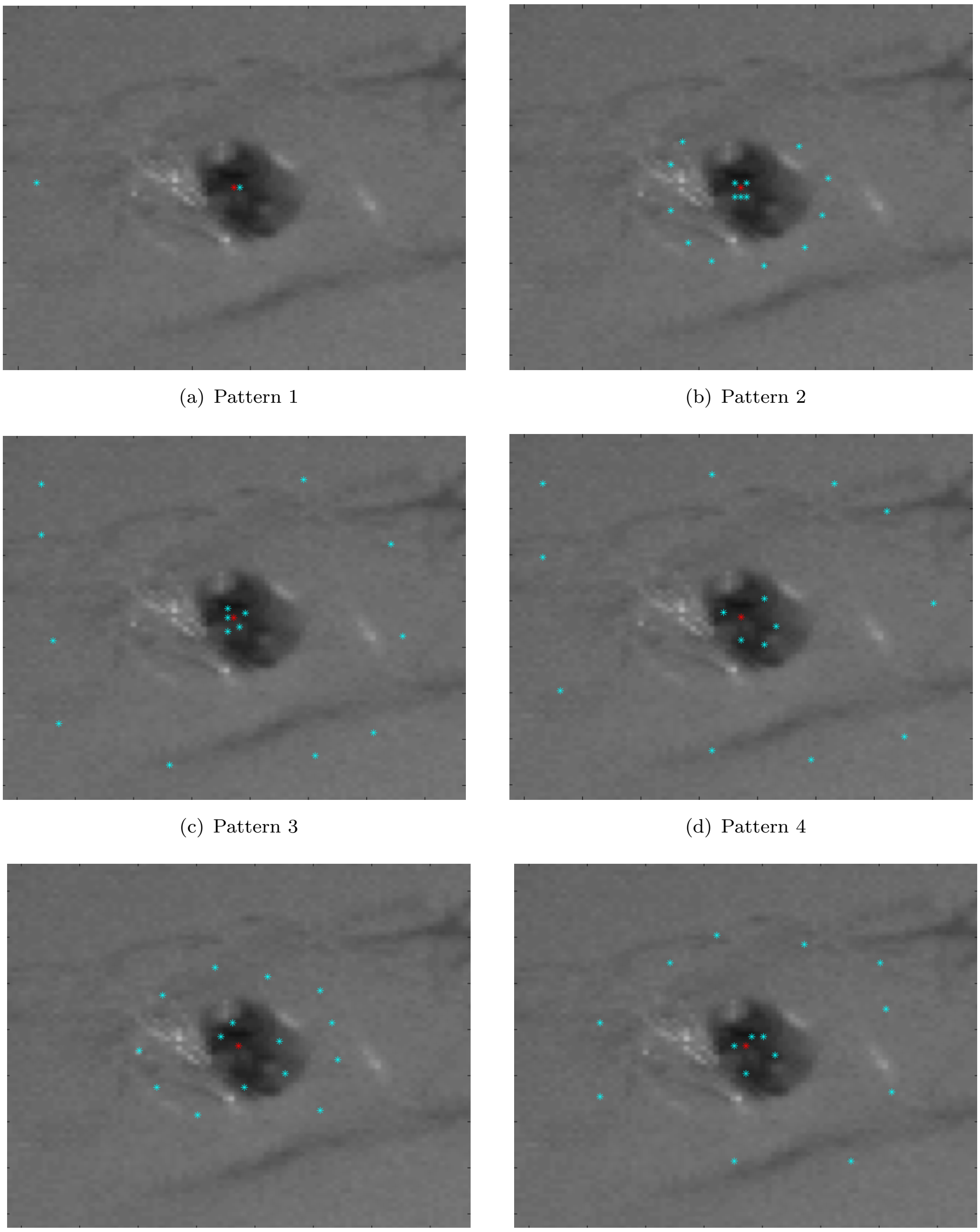
This detection method is based on the examination of 15 data points. These 15 data points are relative positions to the pixel currently being examined. Five of those data points are close to the current pixel, typically inside the rhopalium ($\Omega_{in}(j)$), whereas the remaining ten are further away ($\Omega_{out}(j)$), i.e outside the rhopalium. This detection method and the two following uses six different distributions of data points, i.e six different patterns which can be seen in figure 3.1. When comparing the pixel intensities in the templates, the absolute measure is applied, see equation 3.1. But instead of using the two thresholds, the mean of both t_{in} and t_{out} is used, t_{mean}. We then get,

$$N_{abs}(j) = \sum(\Omega_{in}(j) \leq t_{mean}) + \sum(\Omega_{out}(j) \geq t_{mean}), \tag{3.5}$$

where

$$t_{mean} = (t_{in} + t_{out})/2 \tag{3.6}$$

If every comparison is ok, $N_{det} = 15$ in equation 3.2, the evaluated pixel is marked as a detection.



(a) Pattern 1

(b) Pattern 2

(c) Pattern 3

(d) Pattern 4

(e) Pattern 5

(f) Pattern 6

Figure 3.1: The different patterns used in detection method 1,2 and 3.

3.4 Detection method 2

This detection method is based on the first detection method but it differs in the comparison step. It's still fifteen data points where five are inside and ten are outside but it utilizes the relative measure 3.3 instead of the absolute. Here we want the ones inside to be less than the ones outside and if each of the fifteen comparison fulfills the requirement, it is marked as a detection, again $N_{det} = 15$. As stated earlier, this method uses the same patterns used in the first detection method.

3.5 Detection method 3

This one is almost exactly like the previous detection method. Instead of having the requirement of every comparison to be true to be marked as a detection this only needs thirteen of the fifteen comparisons to be true to be marked as a detection, $N_{det} = 13$.

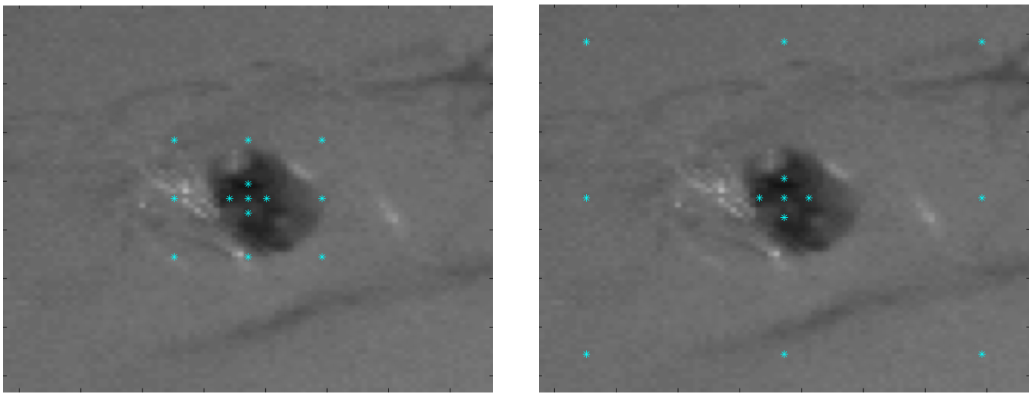


FIGURE 3.2: The pattern used in detection method 4. This particular detection methods utilizes 5 data points inside the rhopalium and 8 outside. The only differances between each pattern is the distances between the two sets of data points

3.6 Detection method 4

This detection method is slightly different from the others in the sense that it got a precomputation step. This step was introduced because there can be a big difference between the film sequences. Depending on where the data points are located and the different light settings it can generate good results in one film sequence but a poor performance in another. In the precomputation step the first picture from each film sequence is analyzed. Prioritizing finding all four rhopalia with a minimal amount of false positives the precomputation step iterates through four variables. These variables are length, l, a factor, f, and two different thresholds, $N_{det,in}$ and $N_{det,out}$. The position of the *inside points* are the center pixel $(0, 0)$ and l from the center of the rhopalium in both positive and negative vertical and horizontal axis, i.e $(0, l), (l, 0), (0, -l)$ and $(-l, 0)$. To get the position of the *outside points* you use the inside points, excluding the center point, and multiply with f and add the corner points to form a square, i.e

$(0, l \cdot f), (l \cdot f, 0), (0, -l \cdot f), (-l \cdot f, 0)$ and the corner points $(-l \cdot f, -l \cdot f), (-l \cdot f, l \cdot f), (l \cdot f, l \cdot f), (l \cdot f, -l \cdot f)$, see figure 3.2. This means that the only thing that differs between two patterns are the distance between the two sets of data points.

The thresholds, $N_{det,in}$ and $N_{det,out}$ are related to the two set of data points, i.e the inner and outside pixels. They are simply the thresholds of how many comparisons that need to be true in the inside and outside respectively in order to mark current pixel as a detection. The comparison in this method is done with fixed values which are extracted from the first picture in the film sequence. If k denotes the film sequence we then transform the thresholds t_{in} and t_{out} to $t_{in,k}$ and $t_{out,k}$ and get,

$$N_{abs,in}(j,k) = \sum(\Omega_{in}(j) \leq t_{in,k}) \tag{3.7}$$

and

$$N_{abs,out}(j,k) = \sum(\Omega_{out}(j) \geq t_{out,k}) \tag{3.8}$$

and thus

$$Xd_i = \{X_i(j) | N_{abs,in}(j,k) > N_{det,in} \wedge X_i(j) | N_{abs,out}(j,k) > N_{det,out}\}, \tag{3.9}$$

3.7 Detection method 5

After evaluating the detection methods above it was clear that depending on variations in the amount of data points, position of those and the threshold, the results could be very different. This is why following detection method was created. This method could be seen as many detection methods since they differ in number of data points, pattern and threshold but the principle is the same for every variation; you compare the intensity in data points further away to closer ones and depending on the threshold the pixel is set as a detection or not. In other words, the relative measure seen in 3.3 and 3.4 is utilized here.

Since evaluating systems are rather time consuming, a maximum of 50 different systems are set. First, six different inside-patterns with nine data points each were created together with 13 outside-patterns with $10 - 200$ data points each. The next step was to combine each of these inside-patterns with the outside-patterns to create different systems. But this would create 78 $(6 \cdot 13)$ systems, which is too many. When adding the threshold parameter, which could range between $0.50 - 0.98$ with a 0.01 accuracy, meaning 49 different values, the total number of possible parameter settings would be 3822 $(78 \cdot 49)$. This would not only be too many systems to analyze but would also consist of a lot of redundancy. This is why a selection step was created to decide which of the inside- and outside-patterns to be used. This selection step created systems with

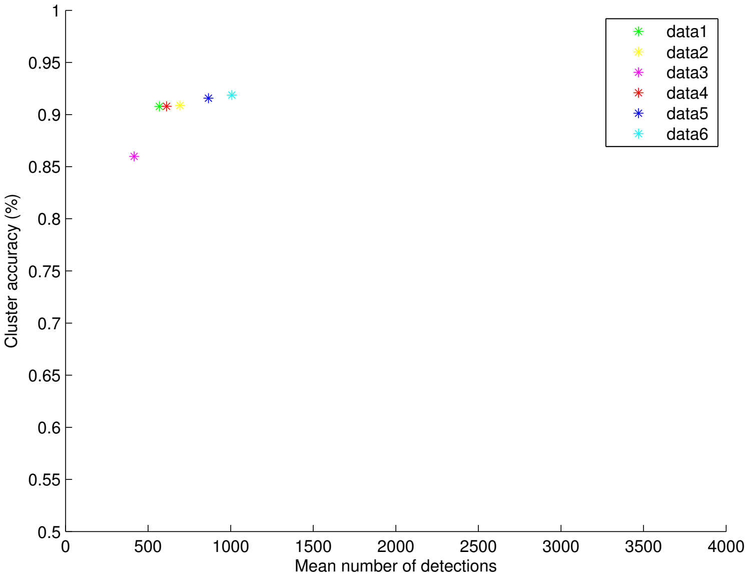


Figure 3.3: The results when among the different inside patterns.

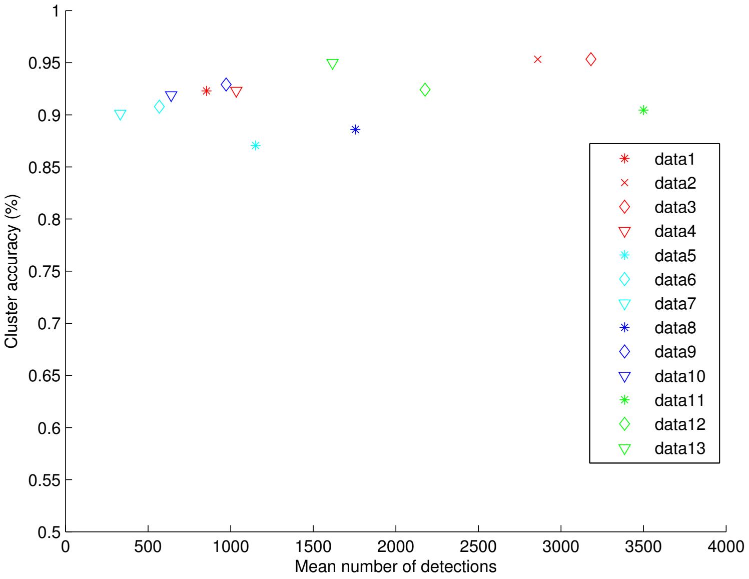
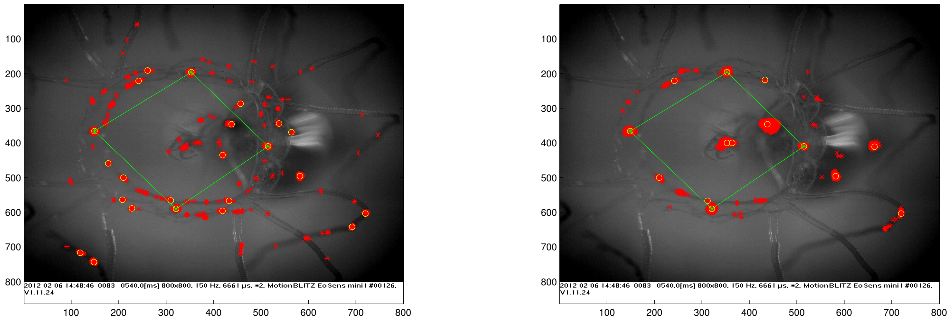


Figure 3.4: The results among the different outside patterns.

all of the inside patterns together with just one of the outside and vice versa, then these systems were tested.

The threshold was set to a fixed value depending on how many data points the pattern had so only the positions of the data points would matter. The test was carried out on all video sequences and the mean amount of detections was saved together with the total accuracy. From those results a plot was created with the number of detections on the x-axis and the accuracy on the y-axis. One plot for when the different inside-pattern is tested and one for the outside-patterns which can be seen in figures 3.3 and 3.4. Since the goal is to have a high accuracy accompanied with few detections the higher up and more to the left in the plot, the better the result. From the inside-patterns number 1, 2 and 5 were chosen (data 1, 2, 5 in the figure). The reason for choosing these was the high accuracy and relatively low amount of detections. The data 6 has marginally better accuracy but almost twice the number of detections which makes this pattern worse than the others.

When choosing the outside-patterns it's not as straight forward as in choosing the inside-patterns. The reason for this is that the number of data points differ in the different outside-patterns. The color of the symbols in the diagram signals that they got the same amount of data points, except the red ones, they got $200, 100, 100$ and 42 data points(data 1 - data 4) each. They are included just to see how much large amounts of data points influence the accuracy and the amount of detections. Note that large amount of data points, 200 and 100 (data 1 - data 3), has a large effect on the running time, making it much longer. That being said, it takes a very high accuracy and a low amount of detections to make up for the extended execution time. As seen in the diagram, data 1 - data 3 ones don't stand out as much as needed to make it a usable pattern. Instead the patterns chosen is data 7, data 9 and data 10. The reason for choosing pattern 7 is because of its low detection count while 9 and 10 are chosen because of higher accuracy but still relatively low detection counts. Looking at the first image in the first video sequence of all the thirteen outside-patterns, the distributions of detections differs. Many are evenly distributed has a more even distribution, i.e more scattered throughout the image (figure 3.5(a)) while some other have less even distribution, i.e the detections are concentrated in larger clusters (figure 3.5(b)). The desire here, is to have large clusters rather than scattered detections since a possible



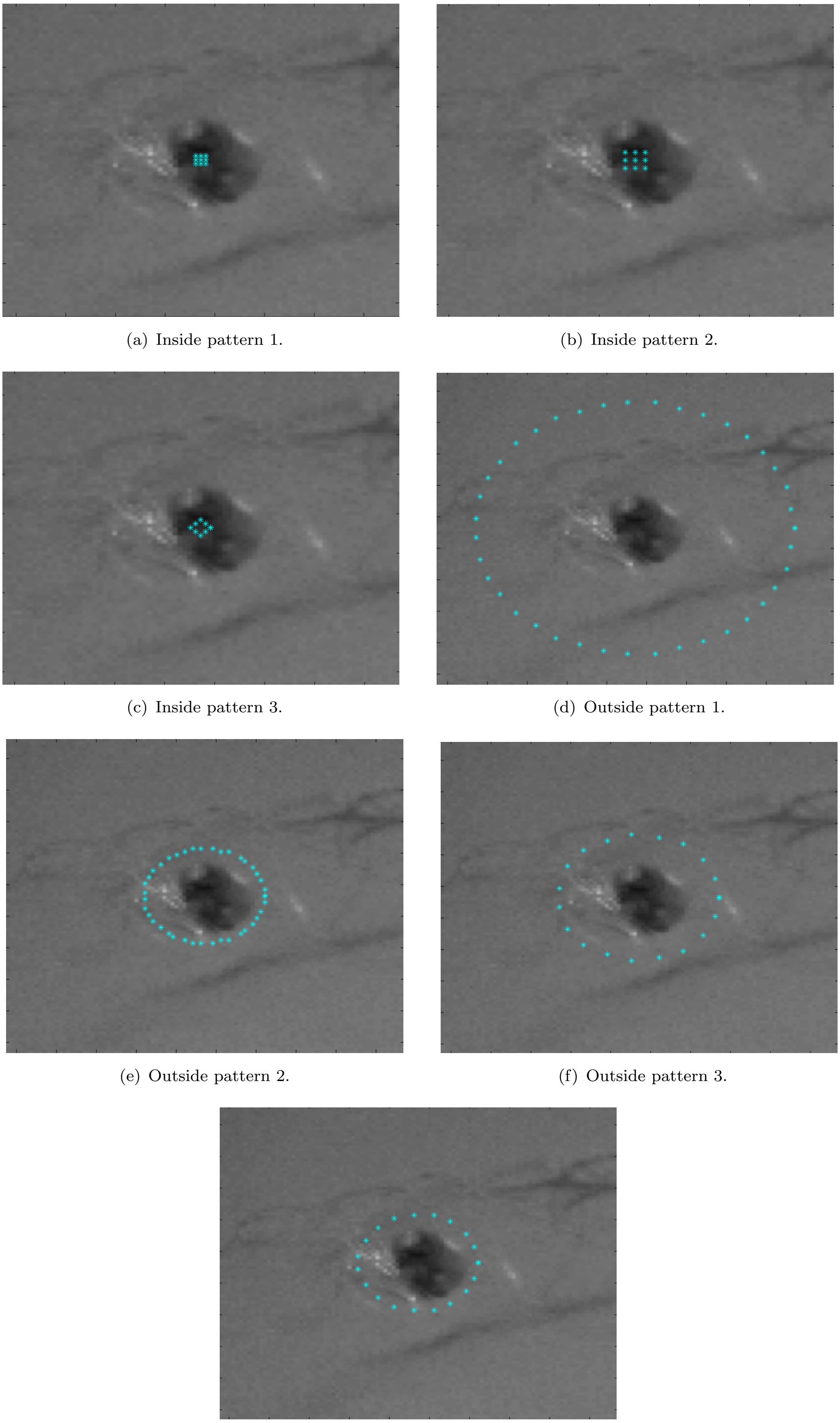
(a) Scattered detections

(b) Clustered detections

FIGURE 3.5: Depending on the pattern used in a detection algorithm, the detections can be more scattered or clustered.

rhopalium should produce a cluster of detections, not a single detection. Since data 5 exhibits clustered detections this outside-pattern is the fourth and final one going to be used. It has considerably lower accuracy and a bit more detections but the cluster-capability of the detections still makes this pattern desirable.

Now we've reach a point where we have three inside-patterns and four outside-patterns, see figure 3.6, which totals of 12 different parameter settings. Next the different thresholds, N_{det} in equation 3.4, is selected for each system. For each of the current systems three different threshold levels are chosen. The first threshold for each system is obtained by iterating the threshold level from 0.95 downwards, decreasing by 0.01 each iteration, and when the amount of detections is greater than 0 the current threshold is saved. The second threshold is selected by iterating the threshold level from 0.5 and upwards and is saved when the number of detections is less than 2150. The third and final threshold for each system is the mean of the first two thresholds. The 12 parameter settings combined with the three thresholds results in 36 new systems. But even though each system is unique with its parameters they can still yield exactly the same result. This is the case when the differance between the thresholds is too small and thus don't affect the end result. Therefore the real number of systems generated from this detection method is 25.



(a) Inside pattern 1.

(b) Inside pattern 2.

(c) Inside pattern 3.

(d) Outside pattern 1.

(e) Outside pattern 2.

(f) Outside pattern 3.

(g) Outside pattern 4.

FIGURE 3.6: The different patterns used in detection method 5.

Chapter 4

System evaluation

4.1 Test environment

The test program runs through every image in each video sequence for every system and saves the most important results. The most vital part here is to obtain the accuracy which is based on four measures. For each of the four measures a rhopalium is marked as found if:

1. there are 10 or more detections within a circle with a radius of 10 pixels around the correct coordinate (from ground truth).
2. there are 20 or more detections within a circle with a radius of 10 pixels around the correct coordinate.
3. there is a cluster within a circle with a radius of 10 pixels around the correct coordinate.
4. there is a tracked coordinate within a circle with a radius of 10 pixels around the correct coordinate.

4.2 Criterias and restrictions for determining the overall best system

When appointing the overall best system there can be different result depending on what aspects you focus on. The following restrictions have been set in order to have a maximum of 20 systems before the final step:

1. **Accuracy.** When the amount of detections is in reasonable range the next important aspect of the overall performance is the accuracy. System with a lower cluster accuracy than 95 % will be omitted.

2. **Reasonable amount of detections.** There shouldn't be too many detections since that will make it harder to distinguish the correct rhopalia. Theoretically the exact number of detections should be around 530 since since the area of a rhopalium is roughly 530 pixels ($13^2 * \pi$) but since we're omitting each other pixel in both axis we must divide it by four. But this division is canceled out due to the fact that there are four rhopalia. But since there is practically no chance of every pixel inside each rhopalium to be detected, this is merely an indicator. After running the different systems, a more suitable number of detections should be around 2500, but no more since most of them will still be false positives. The minimum amount of detections isn't set since few detections yields bad accuracy and thus the accuracy restriction will weed those out.

3. **Execution time.** When you have reasonable amount of detections and a good accuracy you want the system that does this in the least amount of time possible. In order to reach the goal of almost-realtime execution time, a time limit of 400 seconds has been set.

4.3 Relative placement scoring

When these restrictions have been set there are less than 20 detection methods left and a relative placement scoring system is created to determine the overall best result. The score system is divided between each aspect and in the final step they are summarized. Let say there are 10 detection methods left when the restrictions have been done, then the one with the best accuracy will get a score of 10, the next best will get 9 and so on meaning the relative worst detection method will get a score of 1. This is also done for the number of detections and execution time, the more execution time or detections the worse score they get.

Chapter 5

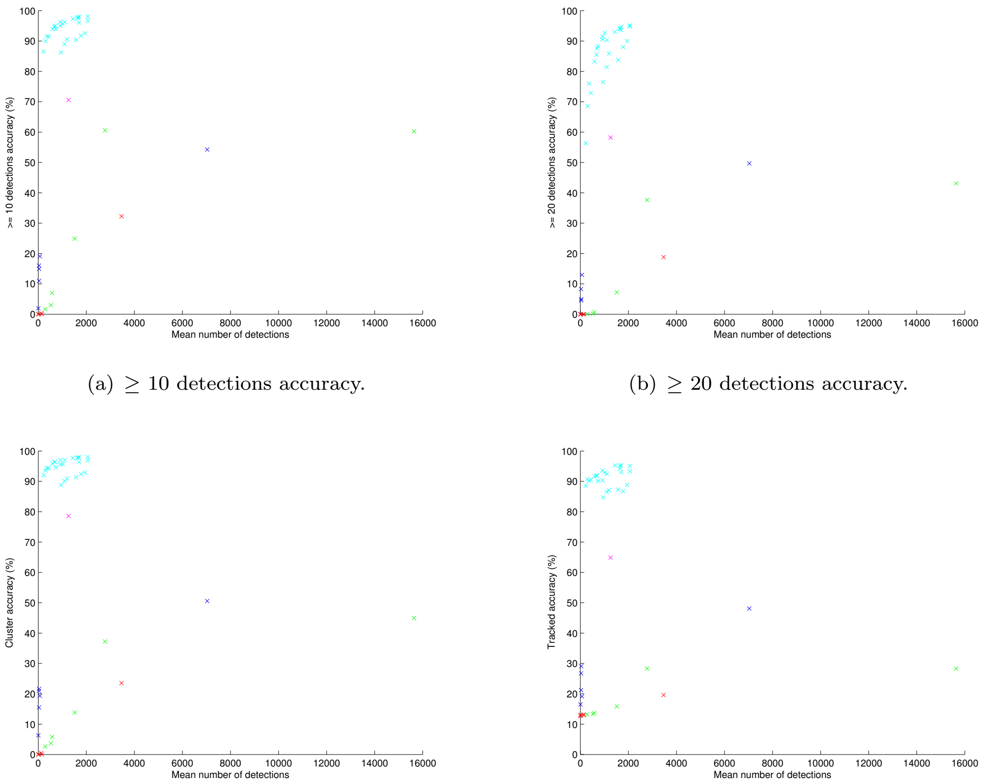
Results

In this section we will visualize the results of the different systems. Each detection method and all their different patterns has been executed in the test program and the results have been saved and then plotted. Each detection method has it's own color and the variation in a color is due to the different patterns. .

There are big differences between the different detection methods but within each method the result is rather concentrated, except for a few outliers. The following subchapters will present a relationship, e.g amount of detections - accuracy, and each result from the detection methods will be presented in regards to the relationship. How the different detection methods are colored together with the number of systems respectively can be seen below:

- Detection method 1 - Blue - 6 systems.
- Detection method 2 - Red - 6 systems.
- Detection method 3 - Green - 6 systems.
- Detection method 4 - Magenta - 1 system.
- Detection method 5 - Cyan - 25 systems.

This totals of 44 systems.



(a) ≥ 10 detections accuracy.

(b) ≥ 20 detections accuracy.

(c) Cluster accuracy.

(d) Tracked accuracy.

FIGURE 5.1: *Mean number of detection in relation to the different measures.* These four diagrams present the number of detections related to the accuracy. The difference between those measures is how it's calculated which can be seen in 4.1.

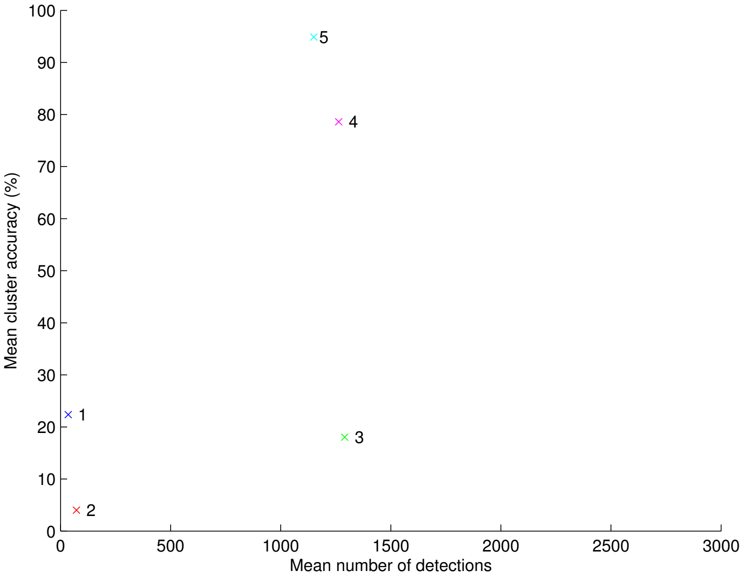


FIGURE 5.2: *Mean cluster accuracy in relation to mean number of detections.* The mean cluster accuracy within each detection method with outliers removed.

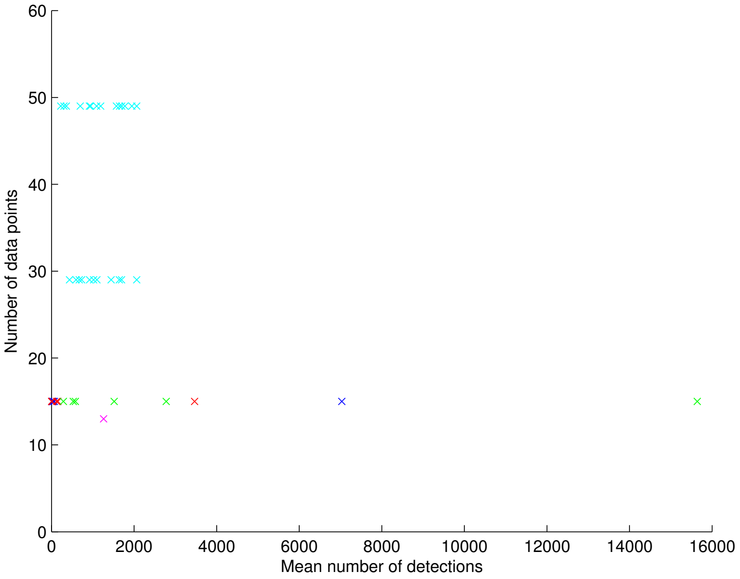
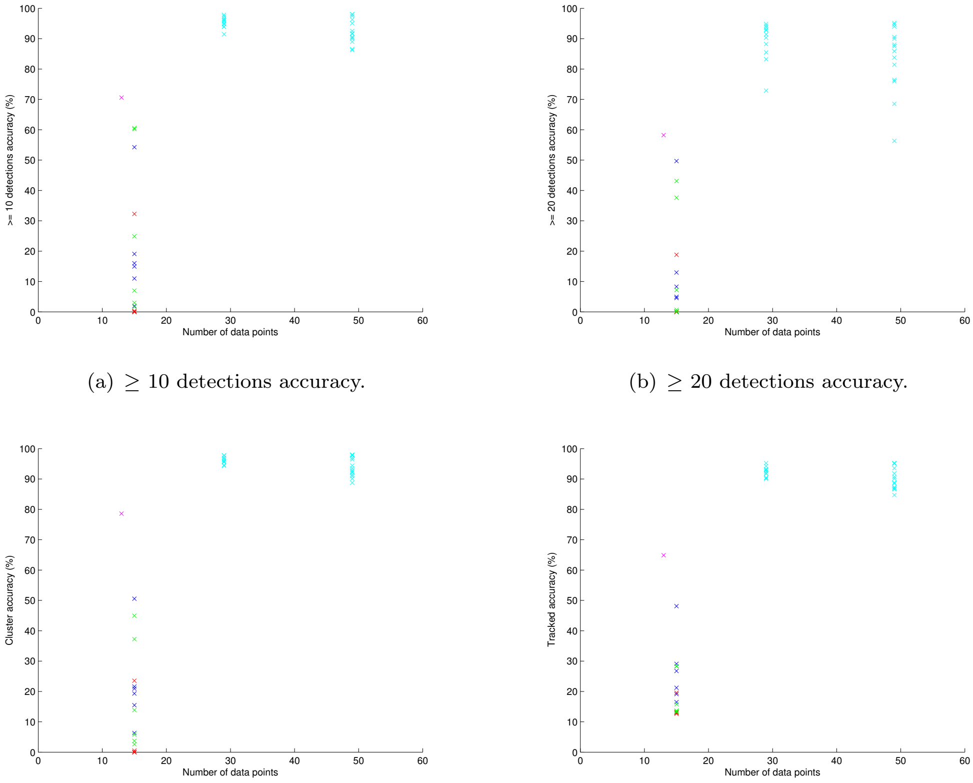


Figure 5.3: *Mean number of detections in relation to data size.* Here the mean number of detections is set in relation to the number of data points in the detection method. The variation within each detection method differ in more in some methods and less in others.



(a) ≥ 10 detections accuracy.

(b) ≥ 20 detections accuracy.

(c) Cluster accuracy.

(d) Tracked accuracy.

FIGURE 5.4: *Data size in relation to the different accuracy measures.* These diagrams present the number of data points in relation to the different measurement criterias.

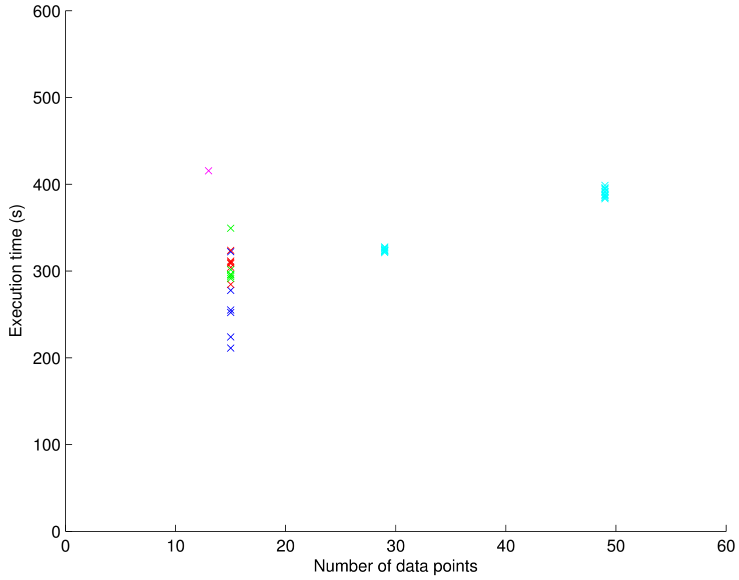


FIGURE 5.5: *Data size in relation to execution time.* To see how the data size correlates to the execution time this diagram is presented.

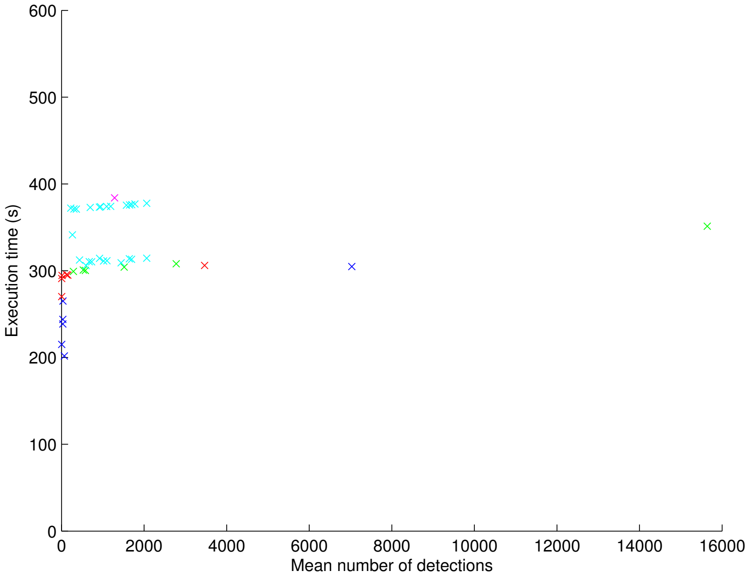


Figure 5.6: *Mean number of detections in relation to execution time.* This diagram is presented to check if the mean number of detections have any impact on the execution time.

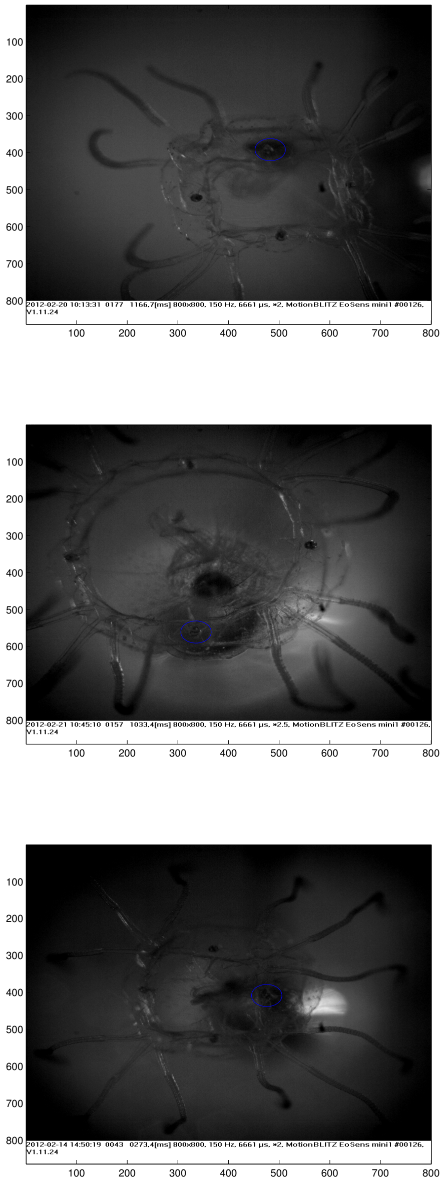
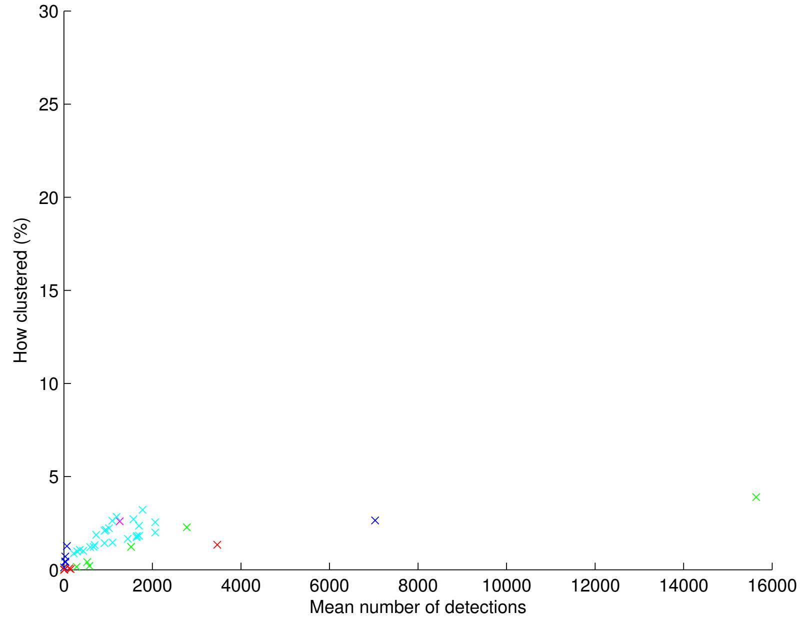

100
200
300
400
500
600
700
800
2012-02-20 10:13:31 0177 1166,7[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
100 200 300 400 500 600 700 800
100
200
300
400
500
600
700
800
2012-02-21 10:45:10 0157 1033,4[ms] 800x800, 150 Hz, 6661 µs, *2.5, MotionBLITZ EoSens mini1 #00126, V1.11.24
100 200 300 400 500 600 700 800
100
200
300
400
500
600
700
800
2012-02-14 14:50:19 0043 0273,4[ms] 800x800, 150 Hz, 6661 µs, *2, MotionBLITZ EoSens mini1 #00126, V1.11.24
100 200 300 400 500 600 700 800

Figure 5.7: *Failure cases: Rhopalia not found.* Because of the many artefacts in the video sequences, every rhopalium not found during the test is saved. It is marked as "not found" by the four different measures discussed in chapter 4, see 4.1. When every test has been executed the rhopalia not found is correlated between all of the systems and saved. By doing this, every undetectable rhopalium is identified and the reason why can easier be established. Using the cluster accuracy, here you can see 3 images where it's impossible to discern the rhopalium from its surroundings.



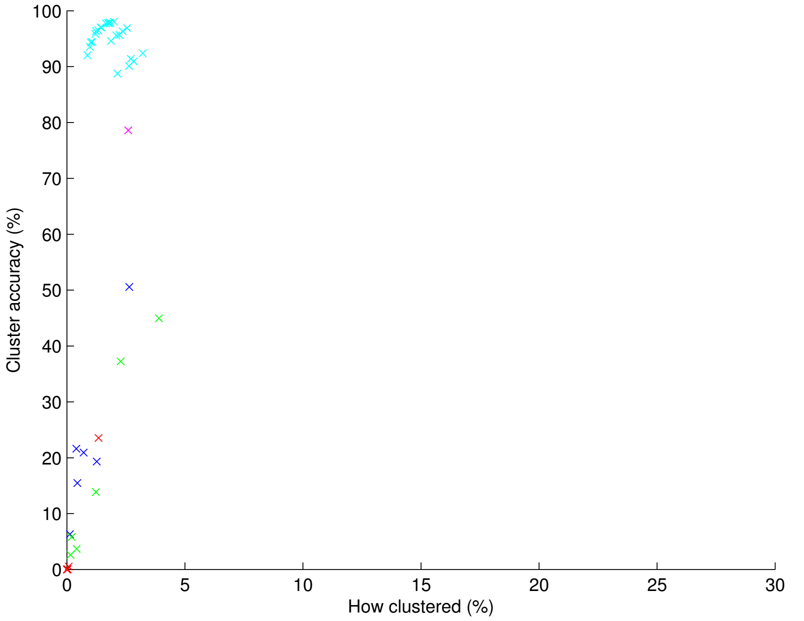
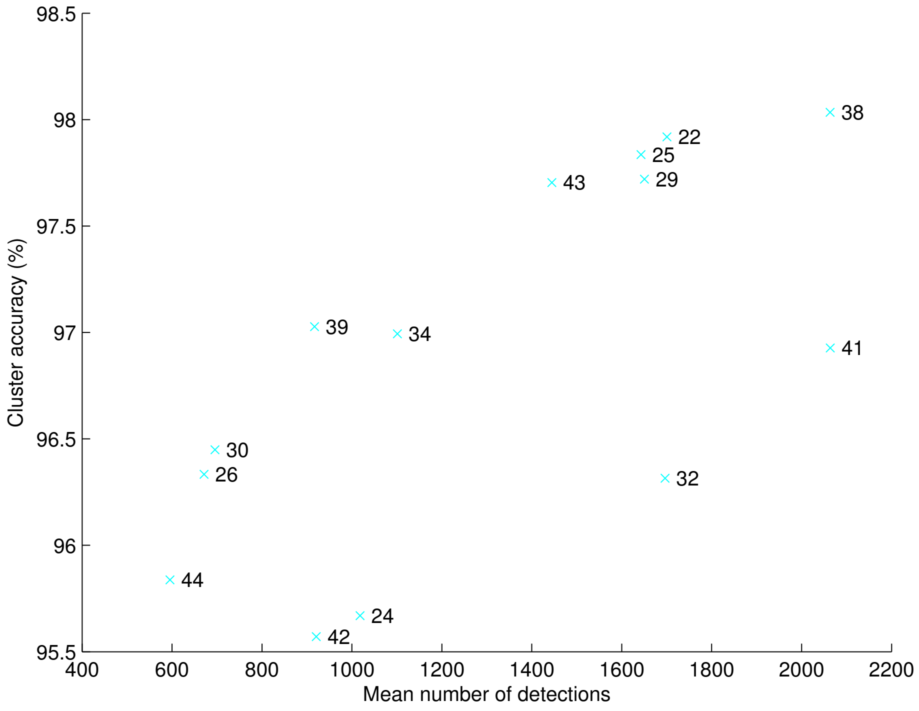


FIGURE 5.8: *How clustered.* To get a good feeling of how clustered the systems are the percentage of detections in a radius of 15 around the rhopalium is calculated. Here it is sufficient to only use one of the accuracy criterias since they have rather similar appearance and we want to see if the clustering affect the accuracy.



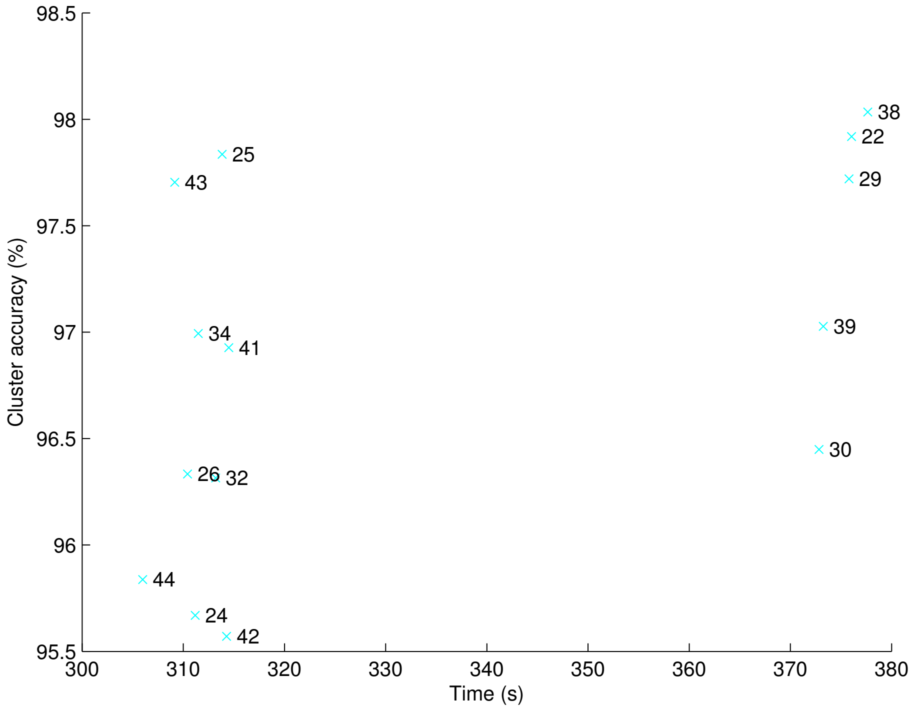
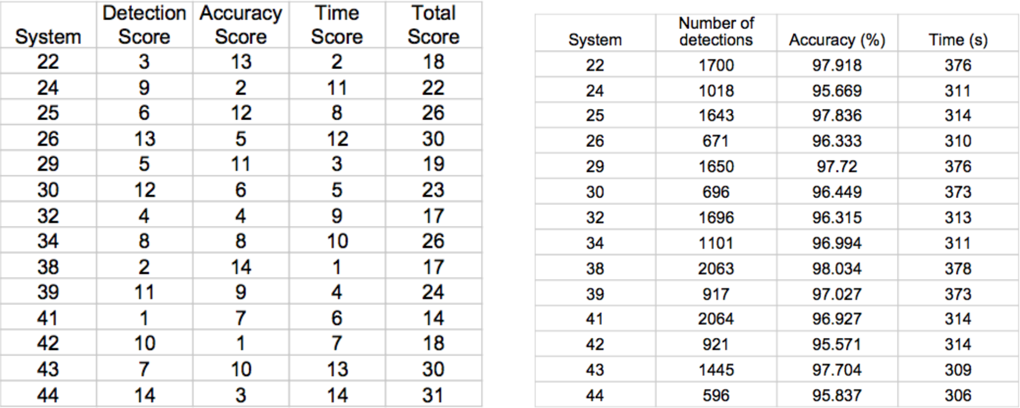


Figure 5.9: *The best performing systems.* When the restrictions have been made there are only 14 systems left and these diagrams present a zoomed in version of some of the previous presented diagrams. The number at each point is the system number.



System	Detection Score	Accuracy Score	Time Score	Total Score
22	3	13	2	18
24	9	2	11	22
25	6	12	8	26
26	13	5	12	30
29	5	11	3	19
30	12	6	5	23
32	4	4	9	17
34	8	8	10	26
38	2	14	1	17
39	11	9	4	24
41	1	7	6	14
42	10	1	7	18
43	7	10	13	30
44	14	3	14	31

System	Number of detections	Accuracy (%)	Time (s)
22	1700	97.918	376
24	1018	95.669	311
25	1643	97.836	314
26	671	96.333	310
29	1650	97.72	376
30	696	96.449	373
32	1696	96.315	313
34	1101	96.994	311
38	2063	98.034	378
39	917	97.027	373
41	2064	96.927	314
42	921	95.571	314
43	1445	97.704	309
44	596	95.837	306

Figure 5.10: *Relative placement scoring.* Using the relative placement scoring system mentioned in 4.3 a table has been created and a score has been distributed to each system depending on the relative placement in each category. The lowest score in each category is 1 and 14 is the highest. A corresponding table but with constant values can also be seen.

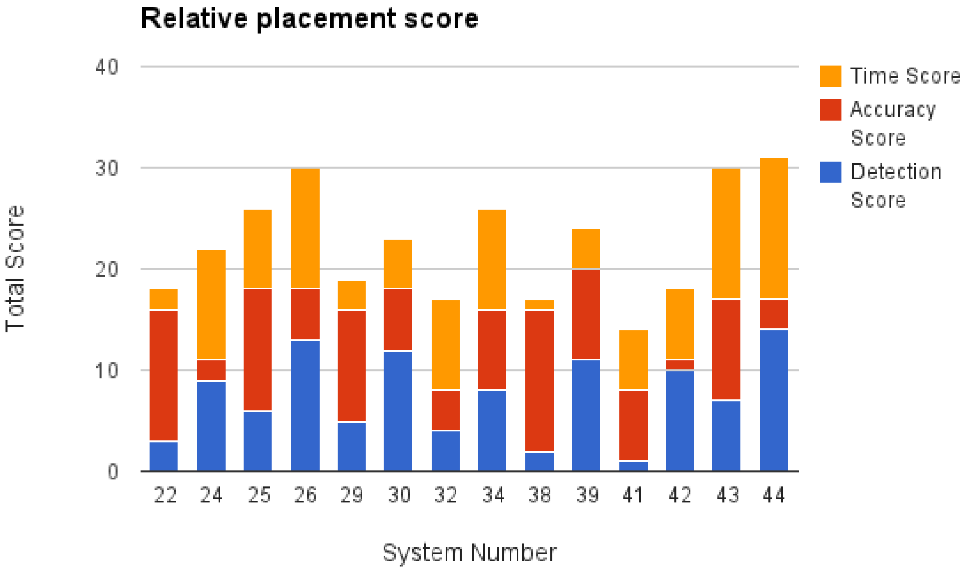


Figure 5.11: A diagram visualizing the score table.

Chapter 6

Discussion

The positions of the tracked coordinates will be static when no detections are found. This will mean that, even though no detections are found, the contracting or expanding rhopalia will eventually overlap with with the static tracked coordinates and thus falsely be marked as found. However, this is only a matter in the detection algorithms with poor performance.

6.1 Accuracy

Looking at the diagrams of the four different accuracy criterias (figure 5.1) we can see that they roughly have the same results. One interesting observation is that detection method 2 and 3 (red and green) have considerably lower accuracy in all but the tracked accuracy. Intuitively that seems a little bit off since the tracking step is done lastly and therefore should have same or worse result but the reason for this is due to the nature of the tracking algorithm. The positions of the tracked coordinates will be static when no or few detections are found. This will mean that, even though no detections are found, the contracting or expanding rhopalia will eventually overlap with with the static tracked coordinates and thus falsely be counted as found. This flaw in the tracking algorithm only affect the worst performing detection methods where, many times, there are no detections.

In detection method 1-3 we can see that each of these has one outlier that has far more mean number of detections, the rightmost ones. The reason for this is that they all share the same pattern which is the one with only two different positions (figure 3.1(a)). This pattern doesn't capture the composition of the rhopalium, being a small dark disc with brighter surroundings. It only detects single pixels with a dark pixel a few pixels away meaning that even noisy areas will be crowded by false positives. Since there is a lot of these areas in the film sequences the detections will shoot through the roof and thus capture much more than just the rhopalia.

Looking at the diagram in figure 5.2 where the mean of each detection method has been visualized (with outliers removed), we see that the detection method 1 has the best performance amongst detection method 1-3. The reason for this is that this method uses fixed values when comparing pixel values. This makes the detection method perform ok in the film sequences that have a light setting close to the fixed values but extremely poor in the others. This is also why it has much fewer number of detections. This is the negative aspect when dealing with fixed numbers as references. The differences between detection method 2 and 3 is pretty obvious since we're lowering the threshold in detection method 3 which will increase the detections as the accuracy improves.

Since detection method 4 is a further development on the first detection method, in regards to using fixed numbers, and detection method 5 is a further development on detection method 2 and 3 one could think they should outperform their precursors, which they do, by a lot.

Detection method 4 has much better results simply because it takes into account all the different light settings and the size of the rhopalium in each different film sequence, and not just the means based on all film sequences like its precursor. Still, this detection method doesn't reach the results of detection method 5 which is the best detection method, with very high accuracy combined with relatively low amount of detections. This method was done lastly and thus with more knowledge and experience. The pattern plays a big role when detecting the rhopalia and comparing to relative values (not fixed) will save both precomputation time and time while executing. The pattern should capture the appearance of the rhopalium and at the same time minimize the risk of detecting something that isn't a rhopalium. This being said, theoretically the best pattern should be a circle lying inside, close to the edge, and another circle just outside the edge of the rhopalium, and all of those outside should be brighter than those inside. But since the appearance of the rhopalia differs between each film sequence and even within, the pattern must have more of a breadth to it.

Since every pattern combined with threshold is unique we can look at the correlation between some other parameters to see how big role the pattern plays.

6.2 Data size and execution time

When looking at the relationship between the number of detections and the data size (figure 5.3) it's rather easy to conclude that they have no correlation since they have a wide range of amount of detections at the same level of data points.

When looking at the data size versus the accuracy (figure 5.4) the observation can be made that detection method 4 has a higher accuracy than detection method 1-3 but with fewer amount of data points and thus the accuracy don't correlate with the number of data points. This notion is further emphasized when looking at the distribution of the two sets of points in detection method 5 where the numbers of data points differ but they roughly have the same accuracy.

If we observe the relationship between the data size and execution time (figure 5.5) one could almost create a line between the two sets of detection method 5 and it would cross the average of detection method 1-3 and one could say that there is a correlation. But this leaves out detection method 4. The reason for detection method 4 to have the longest execution time is because it uses fixed values which have to be loaded and looked up for every picture when comparing. Detection method 1 and 2 also uses fixed values but they are stored in the input, i.e in the system, and thus doesn't need to be loaded. But then again, that is also the reason why detection method 4 has better accuracy than those two.

When looking at detection method 1-3 we see that they have too large range to be able to say that the data size correlate with the execution time but the fact is that the ones that stray away from the 280 seconds mark are due to the amount of detections. The framework has a much shorter execution time if there are no detections because of the cluster-algorithm that iterates through all these. This also means that many detections lead to an extended execution time. And as we see, detection method 1, the one that had very few detections, also has a shorter execution time than the average. The green one with the longest execution time is also the outlier from before with almost 16000 detections. A proof of this can be seen in figure 5.6 where the red, blue and green outliers have an extended execution time compared to the average of the rest of the patterns in their respective detection method. So if we omit detection method 4 and the outliers of detection method 1-3 we have a linear correlation between the data size and execution time which comes naturally because the more data points the more comparisons and thus extended running time. However the correlation between the number of detections and execution time isn't that strong, only when the amount of detections reaches extreme levels.

6.3 Failure cases: Rhopalia not found

The rhopalia not found is different depending on what accuracy criteria you choose. But using the cluster accuracy there are in three film sequences cases where a rhopalium is not detected. As seen in the figure 5.7, the surroundings of the rhopalium not found is somewhat identical to the rhopalium, which is the reason why none of the detection methods could find them. The reason is the tethering shadow coinciding with the rhopalium in a way that makes the rhopalium indiscernible. These correspond to a total of 20 images which is about 1.3 % (20/1469) of all the images and thus don't have a significant impact on the total result.

6.4 How clustered

In figures 5.8(a) and 5.8(b) we see how well the detections are clustered in the respective detection methods. In both diagrams we see that they have quite a uniform distribution and in figure 5.8(a) we see that the more detections, the more clustered in each detection method respectively they are. And as we can see there is no particular system that stands out from the others.

6.5 Determining the overall best system

When determining which of the 44 systems that gives the best overall results it's safe to say that the answer lies in detection method 5. The first step in appointing the best system is to apply the restrictions in section 4.2 and this results in 14 systems left. These system have roughly equal performance in the accuracy category, it only differs 2 percentage points between all. Meanwhile they have a much larger range in the other categories, the number of detections and the accuracy. This could indicate that the accuracy-score shouldn't have the same impact on the total score as the other two categories, e.g having the scores in the accuracy category yield half the score compared to the others. But as we will see, this won't affect our choice of the best system.

When it comes to the patterns used amongst the 14 there's only one pattern that didn't make the restrictions. It's the first outside-pattern (figure 3.6(d)). The reason for this pattern not to qualify is simply because it can't capture the appearance of a rhopalium like the other patterns, simply because the pattern is a circle too far from the edge of the rhopalium.

When looking at the relative placement score diagram (figure 5.11) we see that system 44 has the best result with a total score of 31 and the two systems, 43 and 26, are second best at a score of 30. One thing these three have in common is that they all share the same outside pattern (figure 3.6(g)) and looking at that pattern you can see that it captures the appearance of a rhopalium by being a circle with a radius of 15 pixels close to the edge of the rhopalium. System 44 and 43 also share the same inside pattern (figure 3.6(c)) which is nine data points uniformly distributed in a circle with the radius of 3 pixels. But due to the grid formation of the pixels the exact position needs to be rounded off to the nearest pixel, making the pattern look more like a rhombus.

As said earlier the theoretically best pattern should be two circles close to the edge of the rhopalium, one inside and one outside. But since the size of the rhopalium changes between the different film sequences there need to be more distance between the two circles to be able to capture the different shapes. This is also the reason why the inside patterns with a larger radius didn't perform as well the ones with a smaller one when deciding which inside patterns to be used(figure 3.3).

The only thing that differs between system 43 and 44 is the threshold which can be understood by looking at their detection score in the relative placement score diagram in figure 5.11. System 44 has a higher detection score and thus fewer amount of detections which implies that this system has a higher threshold. Comparing system 43 and 44 further one can see in figure 5.10(b) that the former only has a 2 percentage point advantage on the latter at the cost of more than double the amount of detections which makes system 44 the best overall system and the whole pattern can be seen below in figure 6.1.

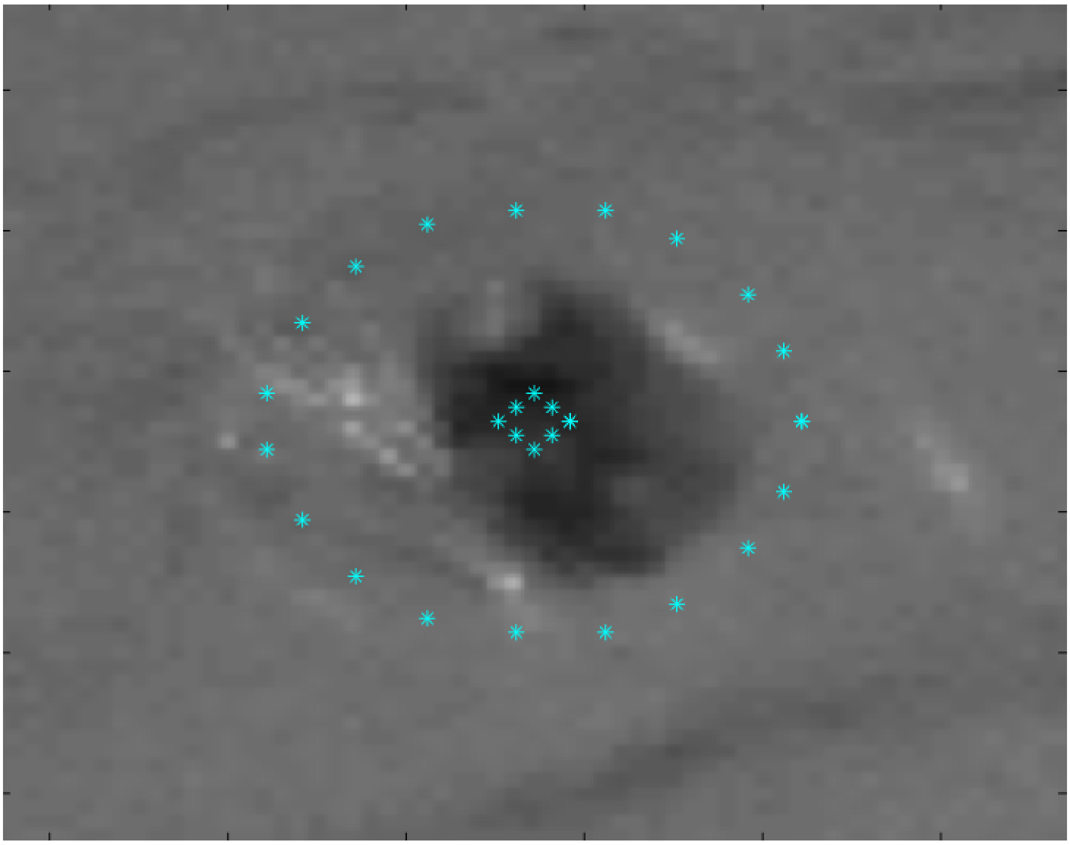


FIGURE 6.1: The pattern of the best performing system

Chapter 7

Conclusion and future work

7.1 Conclusion

When starting this thesis the experience with similar work was fairly limited and there wasn't really only one way to approach it. But after creating detection method 1-4 it was clear that the pattern combined with the number of data points had the most effect on the overall results of a system, which is why detection method 5 was created. Not knowing the appearance of each and every single rhopalia in every film sequence forced the algorithm to be more generic; not using fixed values in the comparison step and trying different inside- and outside-patterns to see which patterns responded best to the film sequences. The different patterns was of course done with the knowledge that the data points should capture the visual aspect of the rhopalium meanwhile using as few data point as possible. Doing so reached a cluster accuracy of around 98 % and considering that 1.3 % was undetectable by every detection method this gives almost a 99.3 % accuracy which can be considered more than enough to pass.

All this entails that to capture a certain visual appearance the goal can be minimized to use very well selected data points at crucial positions and still allow some variance. For instance, if you want to detect small dark quadrants in a picture you should use a template that captures the appearance of the quadrants. This is done by having data points in a quad-shaped pattern that defines the edges, on the inside, of the quadrant as well as data points just outside the quadrant. The data points outside is used to conclude that the inside points differ from the ones outside and thus a certain visual appearance is captured. Depending on if you prioritize accuracy or speed the pattern should have many or few data points. A problem with detecting quadrants and other geometries sensitive to rotation (in contrast to a circle) is that the detection algorithm needs to account for all kinds of rotations and becomes therefore more complex. The variance to be allowed can be decided through the use of threshold as in this thesis.

7.2 Future work

Since the focus in this focus is on the detection step, a natural step would be to do further work with the clustering step but maybe most on the tracking step since its current nature. In its current state it relies on knowing the correct starting position of the rhopalia which is a naive way of tracking. An alternative would be a tracking method that iterates through each cluster, and creates quadrants with other nearby clusters and then chooses the quadrant with most resemblance to the typical positions of the correct rhopalia. But this requires you to have knowledge of these typical squares that the four correct rhopalia form. You could filter out some smaller and larger formations of clusters to minimize the execution time but still this tracking method would probably extend the total running time. When the decision has been made of which four rhopalia that are correct ones, the formation of this quadrant could be saved and in the next image the correct rhopalia should have much resemblance to that since the *T. cystophora* doesn't relocate much between two frames. If, in the next frame, one or more of the correct clusters couldn't be found and thus no similar quadrant could be found, a prediction of the missing coordinates could be done depending on their previous location.

When it comes to further develop the detection methods, the next step could be to merge detection method 4 and 5. Instead of using the crosshair formation seen in figure 3.2 you should use the the best pattern in detection method 5, see figure 6.1. In the precomputing step, instead of iterate through the variables length and factor, you should iterate through the two different radiuses, for the one inside and outside. No fixed values should be used, instead compare the inside and the outside values relatively. This new merged method will handle the pattern problem in detection 4 and introduce some individuality to detection method 5 making the pattern more adapted to each video sequence.

Bibliography

[1] Facial recognition, April 2014. URL http://en.wikipedia.org/wiki/Animal_testing_regulations.

[2] Noldus tracking software, ethovision xt, May 2014. URL http://www.noldus.com/EthoVision-XT/more-about-ethovision-xt.

[3] Noldus tracking software, ethovision xt, May 2014. URL http://www.noldus.com/animal-behavior-research/products/ethovision-xt.

[4] Biobserves tracking software, viewer, May 2014. URL http://www.biobserve.com/products/viewer/.

[5] Image analysis, Januari 2014. URL http://en.wikipedia.org/wiki/Image_analysis.

[6] Barcode, Januari 2014. URL http://en.wikipedia.org/wiki/Barcode.

[7] Facial recognition, April 2014. URL http://www.hrsid.com/company/technology/face-recognition.

[8] Facial recognition, April 2014. URL http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/facial-recognition.htm.

[9] Lisa Greene. Face scans match few suspects. *St. Petersburg Times*, page 798– 803, February 2001.

[10] Is face recognition just high-tech snake oil?, May 2002. URL http://www.enterstageright.com/archive/articles/0102/0102facerecog.htm.

[11] A. Garm R. Petie and D.-E. Nilsson. Velarium control and visual steering in box jellysh. *Journal of Comparative Physiology A*, 199, no. 4:315–324, 2013.

[12] Contrast and rate of light intensity decrease control directional swimming in the box jellyfish tripedalia cystophora (cnidaria, cubomedusae). *Hydrobiologia*, 703, no. 1:69–77, 2013.

[13] E. W. Berger. The histological structure of the eyes of cubomedusae. *Journal of Comparative Neurology*, 8:223–230, 1898.

[14] C. Claus. Untersuchungen uber charybdea marsupialis. *Arb. Zool. Inst. Wien, Hft*, 2:221–276, 1878.

[15] J. S. Pearse and V. B. Pearse. Vision of cubomedusan jellyshes. *Science (New York, NY)*, 199:458–458, 1978.

[16] G. Laska and M.Hündgen. Morphologie und ultrastruktur der lichtsinnesorane von tripedalia cystophora conant (cnidaria, cubozoa. *Zool Jb Anat*, 108:107–123, 1982.

[17] F. Andersson A. Garm and D.-E. Nilsson. Unique structure and optics of the lesser eyes of the box jellyfish *tripedalia cystophora*. *Vision research*, 48:1061–1073, 2008.

[18] J. Piatigorsky and Z. Kozmik. Cubozoan jellyfish: an evo/devo model for eyes and other sensory systems. *Int J Dev Biol*, 48, no. 8-9:719–729, 2004.

[19] J. Ruzickova K. Jonasova V. Paces C. Vlcek Z. Kozmik, S. K. Swamynathan and J. Piatigorsky. Cubozoan crystallins: evidence for convergent evolution of pax regulatory sequences. *Evolution development*, 4, no. 1:52–61, 2008.

[20] F. S. Conant. The cubomedusæ: a memorial volume. *Hopkins press*, 4, no. 1, 1898.

[21] Temporal properties of the lens eyes of the box jellyfish *Tripedalia cystophora*, May 2014. URL http://download.springer.com/static/pdf/364/art%253A10.1007%252Fs00359-010-0506-8.pdf?auth66=1400242043_518a951fbdeb5b34156a8020223bb2d4&ext=.pdf.

[22] E. Buskey. Behavioral adaptations of the cubozoan medusa tripedalia cystophora for feeding on copepod (dioithona oculata) swarms. *Marine Biology*, 142, no. 2: 225–232, 2003.

[23] A. Garm and S. Mori. Multiple photoreceptor systems control the swim pacemaker activity in box jellyfish. *The Journal of experimental biology*, 212, no. 24:3951–3960, 2009.

[24] M. Oskarsson A. Garm and D.-E. Nilsson. Box jellyfish use terrestrial visual cues for navigation. *Current Biology*, 21, no. 9:798– 803, 2011.

[25] Refraction, May 2014. URL http://en.wikipedia.org/wiki/Refraction.

[26] Object recognition, May 2014. URL http://en.wikipedia.org/wiki/Outline_of_object_recognition.

[27] Sift - scale invariant feature transform, May 2014. URL http://en.wikipedia.org/wiki/Scale-invariant_feature_transform.

[28] Cluster analysis, May 2014. URL http://en.wikipedia.org/wiki/Cluster_analysis.

[29] Gaussian blur, Januari 2014. URL http://en.wikipedia.org/wiki/Gaussian_blur.

[30] Danail Stoyanov Peter Mountney and Guang-Zhong Yang. Three-dimensional tissue deformation recovery and tracking: Introducing techniques based on laparoscopic or endoscopic images. *IEEE Signal Processing Magazine*, 27:14–24, July 2010. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5484176.

[31] Video tracking, May 2014. URL http://en.wikipedia.org/wiki/Video_tracking.

Master’s Theses in Mathematical Sciences 2014:E26
ISSN 1404-6342

LUTFMA-3260-2014

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/