Tracking the Motion of Box Jellyfish

Kjellberg, Tobias; Palmér, Tobias; Oskarsson, Magnus; Åström, Karl

2014

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Tracking the Motion of Box Jellyfish

Tobias Kjellberg  Magnus Oskarsson  Tobias Palmér  Kalle Åström
Centre for Mathematical Sciences, Lund University, Sweden
{magnuso, tobiasp, kalle}@maths.lth.se

Abstract—In this paper we investigate a system for tracking the motion of box jellyfish *Tripedalia cystophora* in a special test setup. The goal is to measure the motor response of the animal given certain visual stimuli. The approach is based on tracking the special sensory structures – the rhopalia – of the box jellyfish from high-speed video sequences. We have focused on a real-time system with simple building blocks in our system. However, using a combination of simple intensity based detection and model based tracking we achieve promising tracking results with up to 95% accuracy.

I. INTRODUCTION

Box jellyfish, or cubomedusae, have very special visual systems, [1], [2]. The visual system is based on four identical sensory structures, which are called rhopalia. Each rhopalia consists of six different eyes: the upper and lower lens eyes, the pit eyes, and the slit eyes [3]–[8]. The lens eyes have image-forming optics and resemble vertebrate and cephalopod eyes [9]–[11].

The role of vision in box jellyfish is known to involve phototaxis, obstacle avoidance, control of swim-pulse rate and advanced visually guided behaviour [12]–[14]. Most known species of box jellyfish are found in shallow water habitats where obstacles are abundant [15]. Medusae of the studied species, *T. cystophora*, live between the prop roots in Caribbean mangrove swamps [16], [17]. They stay close to the surface [16] where they prey on a phototactic copepod that gathers in high densities in the light shafts formed by openings in the mangrove canopy. The medusae are not found in the open lagoons, where they risk starvation [12]. The overall goal of this project is to learn more about the visual system and the neural processes involved. One interesting problem is the study of the connection between visual stimuli presented to the animals and the motor response of the animal. However, tracking free-swimming jellyfish poses a very demanding tracking problem. Instead we look at a special experimental setup where the animals are tethered whilst they are submitted to different light stimuli. The goal is then to track how the animals direct themselves, i.e. how they would like to move. In order to do this, an initial goal which we discuss in this paper is how to track the four rhopalia. These appear as four dark discs, situated on the perimeter of the bell, see figure 2. There have been a number of previous studies on motor response from controlled visual stimuli, see e.g. [18], [19].

II. EXPERIMENTAL SETUP

In this study we used in total 33 animals, with sizes ranging from 0.43 to 0.89 cm. The animals were tethered by the top of the bell during the experiments, using a glass pipette with gentle suction, and placed in a Plexiglas tank with inside dimensions of $5 \times 5 \times 5$ cm. The vertical walls of the tank were covered with diffusing paper and a neutral density filter. Each vertical wall was illuminated from the outside by four blue-green LEDs. The diffuser was used to make a plane light source, while the neutral density filter was used to increase the contrast between lit and dark panels and switching one or more panels off was used as the behavioural trigger. The colour of the LEDs matched the maximum spectral sensitivity of the animals and had a peak emission at 500 nm. During the experiments a box was placed over the set-up in order to eliminate visual cues coming from outside. Image sequences...
were recorded with a high-speed camera operated at 150 frames per second. The dataset consists of 15 video sequences, each with around 100 greyscale frames. Each greyscale frame has a resolution of $800 \times 864$ pixels. Depending on how the jellyfish is set up, the light and shadows form differently which will make the video sequences different from each other, see figure 2. Even though great measure has been done 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. The tethering of the animal is also visible in all sequences. This tether-shadow causes problem when the jellyfish is contracting and the rhopalia is moving over the tether-shadow. Since the box jellyfish is moving in every video sequence some parts of the jellyfish are moving in and out of focus. The physical nature of the jellyfish, i.e. being transparent, also affects the appearance and causes refraction on the surroundings.

The rhopalia in each frame in each video sequence has been manually annotated in order to perform evaluation and testing of algorithms.

**III. SYSTEM OVERVIEW**

In this section we will describe our system. Since the focus is on real-time, the chosen building blocks are quite simple in nature, especially the first steps. In figure 3 an overview of the system is shown. We have divided it into three parts. For every frame $i$, a first detection step – which only uses the local intensity distribution around a point – produces a large number of detections, $X_{di}$. These points are then clustered into a set of points $X_{ci}$, in order to remove multiple detections, as well as for improved positional accuracy in the detected points. Finally these clustered positions are sent to the tracking step, which also gets as input the previous frame’s four detected points, $X_{t_{i-1}}$. The output is the four detected points $X_{t_i}$. In the following subsections we will describe the different steps in more detail.

A. Detection

The rhopalia appear as dark discs on a brighter background in the images that are quite consistent in size and appearance in the images. See figure 4 for a close-up view. For this reason we have tested a number of template based approaches for detection. The template is based on the assumption that we have a number of pixels near the rhopalia that are dark. Further outside the rhopalia we should have pixels that are brighter. Figure 4 shows some example templates that we have tried. For speed we have adopted quite sparse templates. Top row shows pixels that should be inside the rhopalia, and hence should be darker. Bottom row shows the pixels that are assumed to be outside the rhopalia. For each point $X_{di}(j)$ in the input image $I_i$, we can then define a number of inside and outside points, $\Omega_{in}(j)$ and $\Omega_{out}(j)$. Examples of these point sets can be seen in figure 4. We have looked at two types of measures, one absolute and one relative. For the absolute measure we define a threshold for the inner pixels, $t_{in}$, and one for the outer pixels, $t_{out}$. We then count the number of inside and outside pixels that fulfill the constraints, i.e.

$$N_{abs}(j) = \sum \mathbb{1}(\Omega_{in}(j) \leq t_{in}) + \sum \mathbb{1}(\Omega_{out}(j) \geq t_{out}),$$  \hspace{1cm} (1)

where $\mathbb{1}(x) = 1$ if $x$ is true and zero otherwise, and

$$X_{di} = \{X_{di}(j) | N_{abs}(j) > N_{det}\},$$  \hspace{1cm} (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 $\Omega$ we have,

$$N_{rel}(j) = \sum_{k=1}^{n} \mathbb{1}(R(\Omega_{in}(j)) < R(\Omega_{out}(j))),$$  \hspace{1cm} (3)

and

$$X_{di} = \{X_{di}(j) | N_{rel}(j) > N_{det}\}. $$ \hspace{1cm} (4)

We have evaluated our whole system in order to find templates that generate enough possible detections, i.e. that in most cases at least generates the four correct points, but that do not generate excessive amounts of false positives. In figure 5 a typical detection result is shown. Since there are many rhopalia-like structures in the images we get a large number of false positives, but these will be eliminated in the following clustering and tracking steps.

B. Clustering

We take a scale space approach for clustering the detections; by smoothing the input image $I_i$ isotropically with a Gaussian kernel we get a low scale version $I_{sm}$. We then
find all local minima $X_{loc}$ of $I_{em}$. The reason for this is the appearance of the rhopalia as dark spots in the images. We then calculate how many detections we have within a vicinity of each $X_{loc}$,

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

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

$$X_{ci} = \{X_{loc}(j)|N_{loc}(j) \geq N_{min}\}.$$

This gives a fast, accurate and quite robust way of clustering the detections. See figure 6b for an example result of the clustering.

C. Tracking

For the tracking step we have looked at a number of simple algorithms. The input consists of the four points from the previous frame, $X_{t_{i-1}}$ and a number of possible candidate points $X_{ci}$. In our model we could also look at the scenario where the general statistical distribution of the four points is given in some form. Arguably the most simple tracking is just choosing as new points, the four closest points from the candidate points to the points from the previous frame, i.e.

$$X_{t_{i}}(j) = \arg \min_{X_{ci}(k)} ||X_{t_{i-1}}(j) - X_{ci}(k)||_2, \quad j = 1, \ldots, 4.$$  

(7)

We have also looked at learning an affine shape model of the four points. In this case we can fix a canonical coordinate system by placing the first three points at $(0,0), (0,1)$ and $(1,0)$. The final points will then be placed at some point $X_a$. Using a large number of ground truth positions we can estimate the mean $X_m$ and covariance matrix $\Sigma$ for $X_a$. This gives us a way of finding the four points that statistically most resemble a four-point configuration, given affine deformations. For all possible subsets of four points $Y_i$ of $Xd$, we change coordinate system so that the first three points are at $(0,0), (0,1)$ and $(1,0)$ and the fourth point at $Y_a$. If there are $n$ such subsets we find the best subset by

$$k_{opt} = \arg \min_k (Y_a(k) - X_m)^T \Sigma^{-1} (Y_a(k) - X_m),$$

and

$$X_{t_{i}} = Y_i(k_{opt}).$$

(8)

For both equation (7) and (8) we can choose $X_{t_{i}} = X_{t_{i-1}}$ if the optimal value is to large, i.e. if the best clustered points are too far away from the previous frame’s points we choose the previous frame’s point positions for the new frame. See figure 6c for an example result of the tracking.

We have not yet focused on the tracking, and more complex motion models are of course possible and quite easy to implement into the framework, such as e.g. Kalman filters [20] or Particle filters [21].

IV. EXPERIMENTAL RESULTS

In total we have in our test setup 1469 frames, and we have manually marked the true coordinates of the rhopalia in each frame. We have tested our system on the video sequences and compared it to ground truth in the following way. For the detection accuracy we count for each of the four rhopalia in the images if there are $N$ or more detections within a circle with radius of 10 pixels around the true coordinate. We have used $N = 10$ and $N = 20$. For the cluster accuracy we have counted the percentage of rhopalia that have a cluster within a circle of 10 pixels around the correct coordinate, and likewise for the tracking accuracy we count the percentage of rhopalia with a tracked point within 10 pixels. We do this for all the 1469 frames, with four rhopalia in each frame. We have mainly tested different detection parameters and settings. In figure 7 the resulting accuracy percentages can be seen. We see that...
the whole bell of the jellyfish in order to measure the motor should be used in order to learn more about the visual system we can get very good performance on a varying dataset. The detection method in combination with clustering and tracking can be constructed. We have shown that using a low-level detection of the special eyes – the rhopalia – of box jellyfish the best performing systems have a tracking accuracy of 95%.

Fig. 6. The figure shows the output of the three steps of our system for an example input image. Left all the detected points are shown in red. These are fed into the clustering which outputs the resulting yellow clustered points in the middle. Finally the tracking outputs the four points to the right, depicted in green.

G. Laska and M. Hündgen, “Morphologie und ultrastruktur der lichtsin-


