Where is my mate? - Real-time 3-D fish tracking for interactive mate-choice experiments

Klaus Müller[1], Stefanie Gierszewski[2], Klaudia Witte[2], and Klaus-Dieter Kuhnert[1]

[1]Institute of Real-Time Learning Systems, University of Siegen, Germany
Email of corresponding author: klaus.mueller@uni-siegen.de

[2]Research Group of Ecology and Behavioral Biology, Institute of Biology, University of Siegen

***Abstract*—Virtual 3-D fish stimulus animations are a powerful tool in fish behaviour research. Besides the option to increase standardisation of such experiments it opens the possibility to change stimulus' morphology or colouration. The next evolution step in virtual stimulus animations is the ability of interaction between live and virtual fish. Since the virtual stimulus fish can not see its live counterpart and consequently can not interact with it, such systems require a reliable real-time tracking of the live fish. In this paper we present an easy-to handle method for fully-automatic real-time 3-D tracking of fish. The method was already tested in practice successfully.**

I. Introduction

In order to increase standardization and reproducibility of fish behaviour experiments, the use of live animals was partially substituted by artificial virtual screen-based stimuli. Additionally virtual stimuli offer more benefits: researchers have the option to change the morphology and appearance of the stimulus and can even define its behaviour. In contrast to most virtual stimuli experiments in the past, in which recorded fish animations were used, we developed a system which overcomes one of the biggest limitations of such systems: the missing interaction between virtual and live fish. Since the virtual fish has no "eyes" to track its live counterpart and therefore can not react to it, a real-time 3-D fish tracking system is necessary. For this purpose, a tracking system has to meet task specific demands: it has to track the live fish reliably, even under hindered conditions like waved and rough water (see III), in real-time.

Fish tracking has often been subject of research. It was e.g. used for counting fish [1], analysing fish behaviour [2], reconstructing swimming kinematics [3] or even monitoring aquatic life at coral reefs [4]. Nowadays also commercial 3-D tracking solutions are available [5]. There is less work in terms of fish-tracking for real-time applications. A 2-D real-time tracking system was used to trigger interactive fish stimulus animations in fish behaviour experiments [6] and to control a robotic fish, swimming with a live fish swarm [7], in order to study social behaviour of fish. A real-time 3-D fish-tracking system was used in a virtual reality framework for fish experiments [8]. The position of the tracked fish was used to update the screen-based animation of a circular disc dynamically in order to investigate startle-response behaviour.

In this paper we present a robust fully-automatic 3-D-tracking system for interactive fish animations. Main features of the system are (1) fully-automatic initialisation and reinitialisation and (2) applicability also under rough conditions like waved water surfaces.

II. Background

A. Fish behaviour research: mate-choice experiments

The here presented method is used in a mate-choice experiment with sailfin mollies (*Poecilia latipinna*), which have a size of 4 to 10 cm. The aim of this experiment is to find out if interactive fish stimuli are as useful as the commonly used live fish stimuli in two-choice experiments. The original experimental set up is performed with real fish, e.g. a female test fish which can choose between two male stimulus fish. The female test fish is inside a big tank (1000 mm x 500 mm x 400 mm), and a stimulus male is inside a small tank (200 mm x 400 mm x 400 mm) adjacent to each end of the large tank. In general the stimuli at the left and right side differ e.g. in morphology, colouration or behaviour. The live female test fish in the middle tank can observe the other fish through the tank windows. If one of the stimuli fish seems more interesting for the female, it will show its preference by staying on the side of the correspondent fish inside a preference zone for a longer time than in front of the non-preferred male stimulus.

The experiment follows a strict sequence: after the test fish is set into the fish tank in the middle, it first can (1) acclimate for around 10 minutes. During this time, the two stimuli tanks are covered, so that the test fish can not see the fish inside these tanks. In the second step, the test fish is put inside a plexiglass cylinder (2), which is placed in the middle of the tank. At the same time the covers of the stimulus tanks are removed, so that the test fish can observe both stimuli. After 5 minutes of observing the stimuli, the experimenter released the test female and the first trial of the preference test starts (3). The trial lasts 10 minutes. In order to avoid a side preference of the fish (left or right) the stimuli were swapped and after another observation phase (4) another preference test trial starts (5).

For virtual fish stimuli experiment, the left and right fish tank are exchanged by screens, showing an interactive virtual fish stimulus animation. The virtual stimulus is swimming randomly through the virtual fish tank. If the live fish swims closer to the screen and enters the preference zone, the virtual

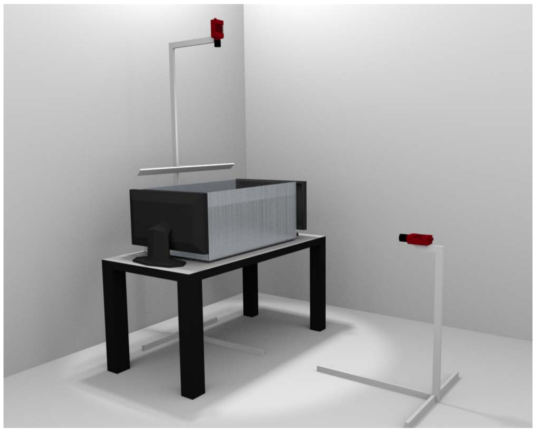

Fig. 1. **Experiment setup.** The screens with the virtual stimuli on it stand on the left and right side of the tank. The cameras are mounted in front and above the fish tank. The figure just shows one of the three light sources, placed in 45 degree angle above the rear tank edge.

stimulus will automatically follow it along the screen window and perform courtship display. Since the two methods should be proved regarding equality, all other steps of the sequence with virtual stimulus males have to be performed in the exact same way as the original experiment with live stimulus males.

B. System infrastructure

The here presented tracking system is a subsystem of the interactive fish animation system. Besides the tracking software, the system consists of two game engine instances, which render the two fish animations to the screens, a fish controller system, which takes care of the fish movement and behaviour, and a graphical user interface, which informs the experimenter during experiment with specific data specific data (e.g. real-time position of test fish and virtual fish and time spent in preference zone). The animation system and the tracking system run on different computers, since both components requires high computational power. All these components have to work together in real-time. In order to simplify the communication between the components, we employ a middleware called ROS (Robot operation system - *www.ros.org*). ROS is mainly used for robotic purpose and provides many different libraries and components like sensor drivers, an Ethernet-based communication infrastructure or developer tools, which are also very helpful in projects like the here presented.

C. Vision system

In the past, different kind of vision systems were used to extract the 3-D position of fish. Besides systems, which take the shadow of the fish into account [9], most systems use multiple views to calculate the exact 3-D position. These include systems with one camera and several mirrors [10] [8] or systems with multiple cameras [3] (see also [11]). In the here presented work, a setup with two orthogonal placed cameras (Allied Vision Technologies, Prosilia GT1910c, 1920 x 1080, RGB, up to 57 fps, 12 mm focal length) is used. A two-camera solution causes more effort in synchronizing and calibrating but is less invasive to experiment than a camera-mirror solution. Since the left and right window of the tank is covered with screens and the ground of the tank with sand, the cameras are placed 1400 mm in front and above the fish tank (see Fig. 1). The cameras are synchronized with the help of hardware triggers.
For extrinsic camera calibration we glue four red triangles at the upper fish tank corners and four to the front corners. At the beginning of each experiment, the system segments the red corners out of the images (top corners - camera above, front corners - camera in front) by color and extracts the outer corner pixel coordinate of the triangle, which is congruent with the fish tank corner. We defined four pairs of 2-D pixel- (previous defined corner of triangle) and 3-D world-coordinate (coordinate of each tank corner - left, upper, front corner is the origin of the coordinate system) for each camera and use the algorithm presented in [12] to calculate the extrinsic calibration matrix.

III. AUTOMATIC FISH TRACKING

In contrast to the real-time 3-D fish tracking system of [8] in the here presented project, we face additional problems: rough and waved water surface and consequently undefined refraction artefacts. These water waves arise at the beginning of each preference test, when the experimenter removes the plexiglass cylinder from the fish tank. As a consequence of this, background subtraction, what is mainly used for segmentation in these kind of projects [11], cannot be used, since the water waves changes the background continuously. In order to show this effect, we conducted a previous experiment, in which we tried to segment a fish with the help of a codebook based background subtraction method after the experimenter removed the plexiglass cylinder from tank. It took around 22 s until the waves calmed down and the background subtraction returned to fault-free operation (see Fig. 2). This effect also affects other methods like optical-flow. For that reason we apply a color-based histogram back-projection combined with the widely used *Continuously Adaptive Mean Shift (CAMShift)* tracking-algorithm (III-1) to track the fish in each of both video streams. In an additional step, the tracking results of both trackers are combined to the final 3-D-position of the fish (III-3). Since one of the project aims is the development of a fully-automatic tracking system, we also add a method for initializing the CAMShift-tracker (III-2) and for re-initializing it (III-4) in case, the tracked fish get lost.

1) CAMShift and histogram backprojection: The CAMshift algortihm was introduced by Bradski [13] as an extension of the MEANshift algorithm [14]. The MEANshift algorithm iteratively shifts a previous defined tracking window to the local maximum of a probability density function. Here this function is represented by a color-histogram back projection. This method assumes that the color of the fish is unique in the recorded scene. For initializing the histogram back projection a color-histogram (hue-channel) of the tracking-object is created. In the following all pixels of the incoming images are arranged to this histogram and replaced by the referred histogram bin's probability. The resulting image is the source for the MEANshift algorithm. In order to reduce computing time, the histogram back projection is just applied to the region around the last tracked location, since the tracked object can just cover a limited distance between two frames. At the beginning of the MEANshift algorithm, the initial point has to be set manually. In contrast to MEANshift, the CAMshift tracker continuously adapts the tracking window size and orientation to the object's size and orientation. The tracker returns the 2-D center coordinate of the tracking window as a result.

2) Initialisation: In order to initialize the CAMshift tracker the color-histogram of the fish has to be calculated and the initial position and size of the tracking window has to be set. This can be done manually by selecting the fish on the image. In the interest of making the whole tracker automatic, we add an algorithm to select the fish area initally. Therefore we employ a background-substraction mechanism, which is applied during the acclimation phase of the experiment, in which the water surface is calm. A detailed description of the background substruction method can be found in [15]. Based on the selected fish area the histogram is calculated and the initial position and size of the tracking window is set.

3) 3-D-position calculation and filtering: Based on the two resulted 2-D pixel coordinates (p_1', p_2' - front camera, top camera) of the CAMshift trackers, the 3-D position P of the fish can be calculated as following:

- The pixel coordinates are transformed to rays by multiplying it with the inverse projection matrix of the correspondet camera. The resulted ray starts and the camera projection center and points in direction of the fish. Since these rays cross the air-water-boarder and consequently get refracted there, we also have to take the refraction into account. Therefore we first calculate the intersection points of the rays with the window (front camera) or with the water surface (top camera). After calculating the angle between ray and window- or water- plane we rotate them around the refraction angle. The final refractred rays start at the intersection point and point in direction of the fish. More details can be found in [15].
- Since the tracking-window's center coordinate of the top and front camera does not compulsory describe the exact same three-dimensional point, the probability that the two rays directly cross each other is low. By that reason we approximate the 3-D point, by calculating the position of the shortest line segment between the two rays. This line segment starts at a point C_1 on the ray of the front camera and ends at a point C_2 on the ray of the top camera. Finally the approximated 3-D position P of the fish is defined by $P = \frac{C_1+C_2}{2}$.
- In order to reduce the influence of noise and to stabilize the 3-D tracking position we apply a Kalman filter with a six dimensional state consisting of the three dimensional velocity and the three dimensional position of the fish.

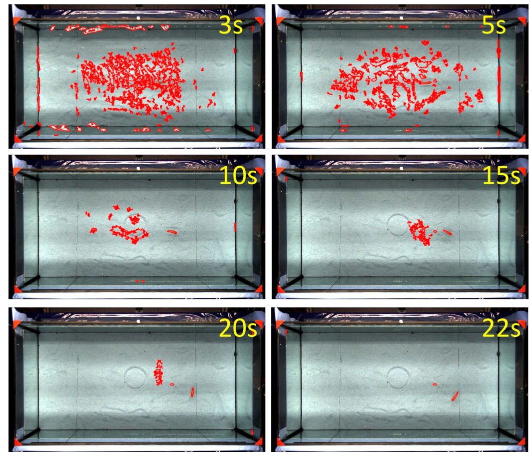

Fig. 2. **Background subtraction while waves occur.** The red marked areas are detected as foreground by the background subtraction algorithm. The yellow time counter refers to the time passed since the experimenter removed the plexiglass cylinder from the fish tank.

4) Re-initialisation: It could happen, that one or even both CAMshift trackers, track a wrong object, which has the same color as the fish (e.g. mirrored fish in the window of the fish tank) or even lose the fish in case of occlusion (e.g. caused by the experimenter while removing the plexiglass cylinder). In case one of the trackers tracks a wrong object, at first this failure has to be detected. In the current algorithm two detecting strategies are implemented: 1) In theory the pixel rays of p_1' and p_2' have to cross each other. As already mentioned in III-3 this is not always the case here, but they should at least nearly cross each other. Based on this fact, every frame we check the distance between C_1 and C_2. If it is larger than a threshold, it is detected as failure. 2) Since the CAMshift algorithm adapts the tracking window size every frame regarding the objects size, we defined minimal and maximal thresholds regarding the fish's size for the tracking window. By this wrong objects, which are bigger or smaller than the fish can be identified. In case one of the strategies detects a failure the tracking algorithm get reset, which requires re-initialization. The re-initialization is done by refreshing 2-D coordinates of both tracking windows. These new positions are found with the following methods: at first, a histogram back projection is done for both camera images. Since the whole image is used, his takes much longer than the back projection during CAMshift. By thresholding the resulted images, segments are defined. If more than one segment per image is found, the pixel rays of the segment's center will be calculated and the shortest distance of all possible pairs (top camera segments - front camera segments) will be searched. The segment pair with the shortest distance is used to refresh the 2-D coordinates of the tracking-window.

IV. Experiments and evaluation

The here presented method was tested in several trials at the laboratory of the Research group of Ecology and Behavioral Biology at the University of Siegen. The used computer was

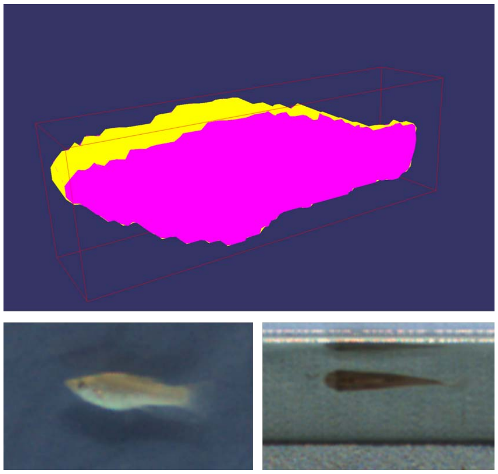

Fig. 3. **3-D ground truth *edge-fish*.** The figure at top shows the *edge-fish* and its red bounding box. The *edge-fish* results from the intersection set of the fish's contour observed from above (right image at the bottom) and from in front (left image at bottom). The yellow part of this 3-D mesh model is derived from the front camera, while the pink part is derived from the top camera.

equipped with a intel i7-3770 (4 x 3.4 GHz) CPU and 16 GB of memory. The experiments were conducted with female sailfin mollies of size between 4 and 5.5 cm.

A. Precision of the tracking result

In order to test the precision of fish localisation, we recorded a random swimming sequence of a female sailfin molly. The sequence consisted of 2580 frames (20 fps) and the fish tank had a size of 1000 mm x 500 mm x 400 mm. We generated ground truth data out of the sequence with the help of the method described in [15]. For every frame, the method calculated a 3-D "edge-fish" mesh shown in Fig. 3 with an absolute position precision better than 2.1 mm (see [15]). Since our method does not track a specific landmark on the fish's body, but the area of maximum pixel density in the histogram back projection image, which is roughly the center of the fish's body, we measure the distance between the center of the *edge-fish* and the resulted coordinates of our method. As the center of the *edge fish* we took the center of its bounding box. In our test 99.5% of the calculated coordinates lay inside the bounding box. The mean distance between *edge-fish's* bounding box center and the calculated result was 4.3 mm with a standard deviation of 1.9 mm.

B. Runtime

We measured the runtime of our method per frame during the sequence of 2580 frames used in section IV-A. On average the whole process took 14.1 ms with a standard deviation of 3.02 ms and therefore a theoretical frame rate of 70 fps.

C. Re-initialisation

As described in section III, especially the time after the experimenter removes the plexiglass cylinder is critical. In all eight runs of the fish oracle, described in the following section, the tracking system relocated the fish immediately after the cylinder was removed.

D. Application: The fish oracle

During the European football championship in 2016 we diverted the whole experiment setup from its intended use and built up a fish oracle with the aim to predict the result of the matches. Therefore we added a national flag to the textures of the virtual stimuli (left e.g German stimulus, right e.g. France stimulus) and conducted a preference test with a live female sailfin molly named Molly. The tracking system worked fine and passed the test very well. Unfortunately Molly made just one right prediction. All videos and a more detailed description can be found under *http://virtualfishproject.wixsite.com/em2016-fisch-orakel*.

V. Conclusion

In this paper we presented a novel method for automatic 3-D fish tracking in real-time under hindered conditions like waved water. In a test we showed that the calculated 3-D position had a mean deviation of 4.3 mm in reference to the center of *edge-fish's* bounding box (ground truth), what is precise enough for the here presented purpose. We also showed that the method is fast enough to process up to 70 fps running on a standard computer and is consequently real-time capable.

Acknowledgment

The presented work was developed within the scope of the interdisciplinary, DFG-funded project "virtual fish" (KU 689/11-1 and Wi 1531/12-1) of the Institute of Real-Time Learning Systems (EZLS) and the Research group of Ecology and Behavioral Biology at the University of Siegen.

REFERENCES

[1] E. F. Morais, M. F. M. Campos, F. L. Padua, and R. L. Carceroni, "Particle filter-based predictive tracking for robust fish counting," in *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*. IEEE, 2005, pp. 367–374.

[2] H. AlZu'bi, W. Al-Nuaimy, J. Buckley, L. Sneddon, and I. Young, "Real-time 3d fish tracking and behaviour analysis," in *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*. IEEE, 2015, pp. 1–5.

[3] S. Butail and D. A. Paley, "Three-dimensional reconstruction of the fast-start swimming kinematics of densely schooling fish," *Journal of the Royal Society Interface*, vol. 9, no. 66, pp. 77–88, 2012.

[4] R. B. Fisher, Y.-H. Chen-Burger, D. Giordano, L. Hardman, and F.-P. Lin, *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. Springer, 2016, vol. 104.

[5] Noldus. (2015) Track3d. Noldus Information Technology. [Online]. Available: http://www.noldus.com/innovationworks/products/track3d

[6] T. Butkowski, W. Yan, A. M. Gray, R. Cui, M. N. Verzijden, and G. G. Rosenthal, "Automated interactive video playback for studies of animal communication," *JoVE (Journal of Visualized Experiments)*, no. 48, pp. e2374–e2374, 2011.

[7] T. Landgraf, D. Bierbach, H. Nguyen, N. Muggelberg, P. Romanczuk, and J. Krause, "Robofish: increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live trinidadian guppies," *Bioinspiration & biomimetics*, vol. 11, no. 1, p. 015001, 2016.

[8] S. Butail, A. Chicoli, and D. A. Paley, "Putting the fish in the fish tank: Immersive vr for animal behavior experiments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5018–5023.

[9] B. J. Laurel, C. J. Laurel, J. A. Brown, and R. S. Gregory, "A new technique to gather 3d spatial information using a single camera," *Journal of Fish Biology*, vol. 66, pp. 429–441, 2005.

[10] L. Zhu and W. Weng, "Catadioptric stereo-vision system for the real-time monitoring of 3d behaviour in aquatic animals," *Physiology & Behavior*, vol. 91, pp. 106–119, 2007.

[11] J. Delcourt, M. Denoël, M. Ylieff, and P. Poncin, "Video multitracking of fish behaviour: a synthesis and future perspectives," *Fish and Fisheries*, vol. 14, pp. 186–204, 2012.

[12] X. S. Gao, X. R. Hou, J. Tang, and H. F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(8), pp. 930–943, 2003.

[13] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.

[14] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[15] K. Müller, J. Schlemper, L. Kuhnert, and K.-D. Kuhnert, "Calibration and 3d ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time." in *VISAPP (3)*, 2014, pp. 626–634.