INVITED PAPER

Real-Time Feedback-Controlled Robotic Fish for Behavioral Experiments With Fish Schools

Use of cyber–physical robotic fish to study fish-school behavior is proposed in this paper; each robotic fish gets video feedback from the fish group and its environment.

By Daniel T. Swain, *Student Member IEEE*, Iain D. Couzin, and
Naomi Ehrich Leonard, *Fellow IEEE*

ABSTRACT | Integrating robotic agents into animal groups creates significant opportunities for advancing experimental investigations of collective animal behavior. In the case of fish schooling, new insights into processes such as collective decision making and leadership have been made in recent experiments in which live fish were interacting with robotic fish driven along preplanned paths. We introduce a new cyber-physical implementation that enables robotic fish to use real-time feedback to control their motion in response to live fish and other environmental features. Each robotic fish is magnetically connected to, and thus moved by, a wheeled robot underneath the tank. Real-time image processing of a video stream from an overhead camera provides measurements of both the robotic fish and the live fish moving together in the tank. Feedback responses computed from these measurements are communicated to the robotic fish using Bluetooth. We show results of demonstrations and discuss possibilities that our implementation affords for new kinds of behavioral experiments with fish schools.

KEYWORDS | Collective decision making; cyber-physical systems (CPSs); feedback control; fish schooling; real-time; robotics; video processing

I. INTRODUCTION

Fish schools exhibit remarkable collective behavior that reflects a highly efficient group-level capability to forage, migrate, and evade predators [1]–[6]. As a group, fish are observed both to respond quickly to external environmental signals of significance and to move together despite uncertainties and disturbances in the environment [7]. In order to better know and predict these kinds of behaviors, it is critical to understand processes associated with collective decision making and collective motion. This requires investigating the mechanisms that connect the choices that individual fish make in response to what they can sense with the complex behaviors that emerge at the level of the group. Identification of these mechanisms can also provide the means to develop principled design methodologies that enable groups of engineered agents, e.g., robotic mobile sensor networks, to achieve high performance in demanding tasks such as environmental monitoring, search, and exploration [8], [9].

Fish schools are typically leaderless aggregations of selfish individuals where direct access to stimuli from the external environment may be limited to a small number of individuals on the periphery. Fish gain access to information through social cues from their neighbors [10], [11]. For example, an individual fish that does not sense a predator directly may instead sense its neighbor turning fast and make adjustments to turn with it, thus staying with the school and avoiding the predator.

Manuscript received June 30, 2011; accepted August 10, 2011. Date of publication October 19, 2011; date of current version December 21, 2011. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-09-1-1074. All experimental work was carried out in accordance with federal and state regulations, and was approved by Princeton University's Institutional Animal Care and Use Committee.

D. T. Swain and **N. Ehrich Leonard** are with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: dswain@princeton.edu; naomi@princeton.edu).
I. D. Couzin is with the Department of Ecology and Evolutionary Biology, Princeton University, Princeton, NJ 08544 USA (e-mail: icouzin@princeton.edu).

Digital Object Identifier: 10.1109/JPROC.2011.2165449

0018-9219/$26.00 ©2011 IEEE

Investigating mechanisms that explain fish schooling therefore requires an exploration of information passing and decision making at the level of the individuals. Experiments have been run, for instance, to explore the role of different sensing modalities in fish [12], the role of school density on collective response to predation risk [13], and the relationships between group speed, group polarity, and mean nearest-neighbor distances [14]. Additional empirical studies of fish schools are described in [15]–[17].

In recent years, researchers have introduced robots that interact with live animals as a new tool for studying social behavior in animal groups; for a survey, see [18]. The objective is to infiltrate the live animal group with robotic animal replicas that are designed to behave as conspecifics or heterospecifics. By being able to manipulate the behavior of the robotic members of a mixed group of interacting animals and robots, new opportunities are created for inferring mechanisms of collective behavior from observations.

In the experiments of [19], robotic cockroaches moved as conspecifics with live cockroaches in an arena with shelters. The robots could sense shelters and shelter darkness and discriminate between live and robotic cockroaches. Their behavior was programmed according to a parameterized model of shelter seeking; by manipulating the parameters the robotic cockroaches were seen to influence the collective decision making of the mixed robot-animal group. The results yielded new insights on the role of social interaction in collective decision making in cockroaches and demonstrated the possibility of using robots to control animal behavior.

Interactive robots have also been used in behavioral experiments with fish schools. In [20], a pair of remotely controlled dead fish was used to test cooperation in predator inspection. In [1], two remotely controlled robotic fish, moving along guide lines, were used as conspecifics with live fish to investigate quorum responses in collective movement decisions. In [2], a computer-controlled robotic fish was designed to move around a tank as a conspecific with live fish in order to test hypotheses on mechanisms of recruitment and leadership in fish schools. The body of the robotic fish was a replica mounted on a magnetic base that was controlled by an electromagnet underneath the tank. By programming the motors of a pulley system that controlled the electromagnet in two dimensions, the robotic fish was made to follow a specified route at a specified speed.

Aureli *et al.* [21] have developed a free-swimming fish-like robot that is only 13 cm long; while this robot is not meant to function as a conspecific it has been shown to actively engage a school of fish in a tank.

In this paper, motivated by the works of [1], [2], and [19], we present a new cyber–physical implementation of robotic fish with live fish where the robotic fish have available real-time feedback measurements of their environment and can control their own motion in response to these measurements. By facilitating versatile real-time feedback control for the robotic fish, our implementation significantly expands opportunities for manipulating the behavior of the robotic members of a mixed group of robots and fish and as a result expands the scope for new behavioral experiments with fish schools.

Measurements available to the robotic fish can include, for example, relative position, heading, and/or speed of neighbors; this enables a robotic fish to adopt a responsive behavior modeled after a conspecific. Likewise, real-time processing of measurements yields features of the school, such as centroid, boundary, polarity, and momentum; this enables a robotic fish to adopt a responsive behavior modeled after a predator.

Like the robotic fish in [2], each of our robotic fish consists of a replica fish in the tank that is magnetically coupled with an actuator beneath the tank. However, instead of being driven by a pulley system, each replica fish is magnetically connected to a wheeled robot that moves freely underneath the tank. Feedback measurements are provided from real-time image processing of a video stream from a camera overhead the tank. The video stream captures the behavior of the robotic fish, the live fish, and anything else in the tank, and it is linked by a FireWire interface to a computer where real-time tracking is performed. From the tracking data, a range of properties of the mixed robot-fish group are computed in real time and these are used as feedback in the control algorithm implemented on the computer for each of the robotic fish. The resulting actuation signals, which command robot wheel speeds, are sent to the robotic fish through Bluetooth channels.

A critical ingredient and contribution of our work is the design and integration of real-time tracking and computation of features of the fish school. Existing work on offline fish tracking includes [14], [22], and [23]. There has been very recent work on real-time tracking of humans using laser range finders [24] and real-time tracking of flies in three dimensions using multiple cameras [25]. The tracking approach of [25] is similar to ours, although in [25] multiple computers are used to capture motion in three dimensions, whereas we only require a single computer for our 2-D setting. Additionally, segmentation is not a significant issue in [25] since the multiple cameras eliminate most occlusions, whereas we need to implement a real-time segmentation algorithm that can resolve significant and frequent occlusions. The segmentation algorithm we use in real time is similar to the one used offline for flies in a planar arena in [26].

Our approach to tracking and to computing properties of fish schools in real time balances accuracy and computational efficiency. Segmenting the images of individual fish is a particular challenge because conspecifics look nearly identical and they swim very close to one another. We successfully perform segmentation in real time using an expectation–maximization mixture-of-Gaussian (EMMG) algorithm [27], [28]. For data association, we use the Hungarian (Munkres) assignment algorithm [29], [30] to

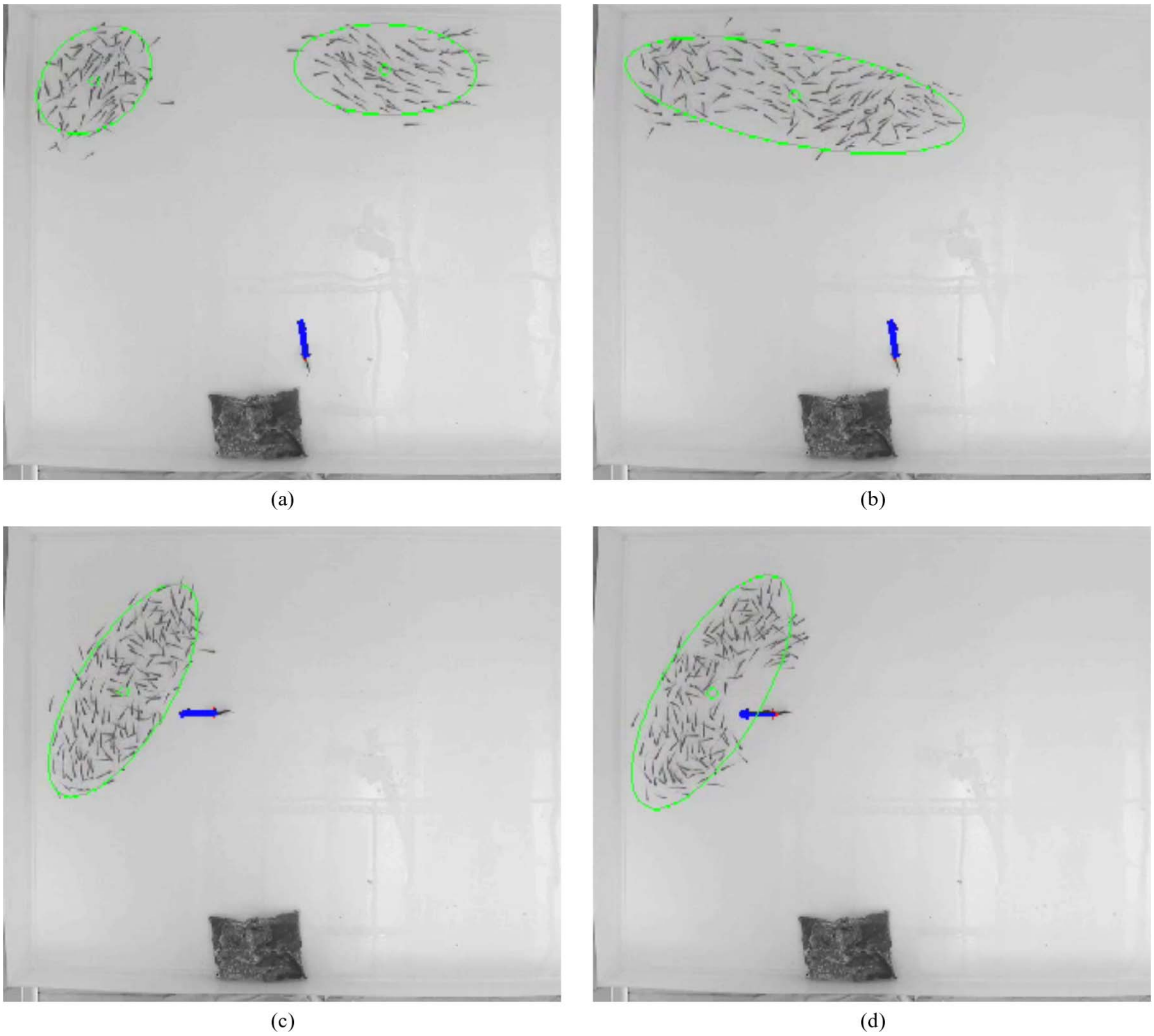


Fig. 1. ***Four frames from the overhead video stream of a robotic predator fish with a live school of golden shiners in the testbed with real-time tracking output overlaid. Frames (a), (b), (c) and (d) are ordered in time with frame (a) the earliest. In each frame, the tracked position and heading angle of the robotic fish are represented by the thick blue line. The green ellipse(s) represents the approximate boundary of the live fish school, and the small green circle(s) represents the location of the centroid of the live fish school. In frame (a), two subgroups of live fish are successfully identified and tracked; the subgroups have joined to make one connected group in frame (b). The robotic predator approaches and intrudes in frames (c) and (d).***

assign an identity to a blob that minimizes distance from the previous tracked location to the centroid of the blob in the current image. Particle filter methods have been used successfully in offline tracking of multiple objects [22], [31], [32], but they proved to be insufficient for our real-time tracking problem. Instead, we use the unscented Kalman filter (UKF) [33], [34]. The UKF provides state estimates from which we compute fish school properties.

Fig. 1 shows four snapshots from the overhead video of our experimental tank with a school of golden shiners (*Notemigonus crysoleucas*) and a robotic fish behaving as a predator. Superimposed on each snapshot are features identified in real time; the thick blue line identifies the computed position and orientation of the robotic fish, a green ellipse (or ellipses) identifies the computed approximate boundary of the school, and a small green circle identifies the computed centroid of the school. The robotic predator can be triggered to approach the school as a function of the time-varying value of these features.

We describe our mixed robot-fish testbed and its components in further detail in Section II. In Section III, we present our approach to real-time tracking and computation of fish school features. In Section IV, we describe two example behavioral experiments with fish schools that were used to demonstrate our real-time feedback-controlled robotic fish interacting with a live fish school. We conclude in Section V and discuss possible future experiments with our testbed.

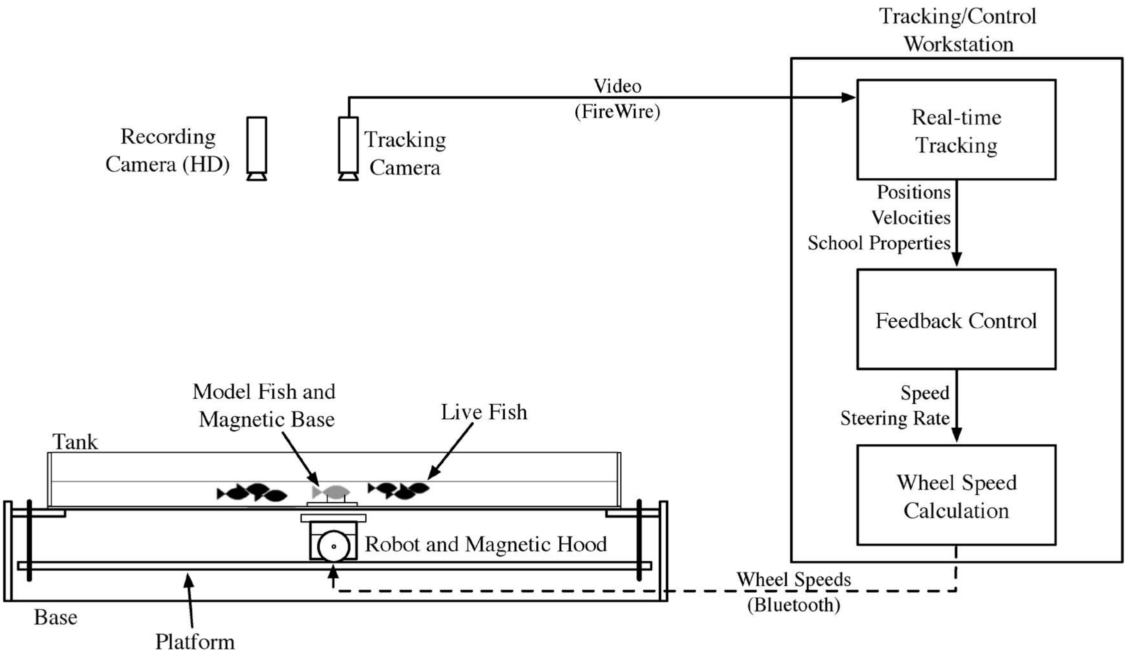


Fig. 2. *Diagram of the mixed robot-fish testbed components.*

II. TESTBED FOR INTERACTING LIVE AND ROBOTIC FISH

A diagram of the mixed robot-fish testbed is shown in Fig. 2. Live and robotic fish swim together in a shallow, freshwater tank. Because the live fish move predominantly in the horizontal plane, the robotic fish are designed for horizontal motion. Each robotic fish is a model fish in the tank magnetically connected to, and thus moved by, a wheeled robot moving freely on a platform underneath the tank. Two cameras are mounted overhead: one FireWire camera acquires real-time video and one high-definition (HD) camera records video for offline analysis after the experiment. Tracking, school features, and feedback control laws are computed on a computer workstation; control inputs are sent from the workstation over Bluetooth channels to each wheeled robot.

In this section, we describe the hardware and software components of the testbed. The real-time tracking is described in more detail in Section III.

A. Hardware

In this section, we describe the hardware associated with the arena, the wheeled robots, the model fish, and the tracking.

1) The Arena: The arena consists of a 4-ft by 5-ft by 12-in-high watertight tank mounted on a wooden base. For the experiments, the water in the tank is typically 2.5–3 in high. The tank is constructed using Garolite G10 fiberglass composite. This material provides superior stiffness and allows the tank base to be only 1/4 in thick and still resist significant bowing from the weight of the water. The thin tank base is important because it ensures the strength of the magnetic coupling between model fish and wheeled robot throughout the tank area.

Securing bolts are placed along the periphery of the tank, fixing it to the support frame. The sides of the tank are 1/8 in thick to keep weight low and 12 in high to prevent fish from jumping out of the tank. There is a 12-in piece placed diagonally across each corner to prevent fish from congregating in the corners. One of these corner pieces has a small hole drilled through the bottom to allow water into that corner where there is a drain hole. A drain valve and attached pump allow the tank to be drained quickly, if necessary. A view from overhead the tank is shown in Fig. 3. In the photo, an underwater camera is clamped to the bottom left corner piece; this is an optional feature.

In addition to providing structural support, the base supports a platform that is suspended below the tank. This platform is the surface on which the wheeled robots move freely. A long access port is cut into one side of the base so that wheeled robots can be inserted and removed. The vertical standoff distance between the platform and the bottom of the tank can be adjusted to accommodate varying degrees of bowing, robot height, and magnet strength. The nominal vertical distance is 4 in.

2) Wheeled Robots: Merlin Systems Corp. MiaBot Pro wheeled robots [35] are used to move the model fish. These wheeled robots were selected because they can be controlled wirelessly with Bluetooth, and they have a high

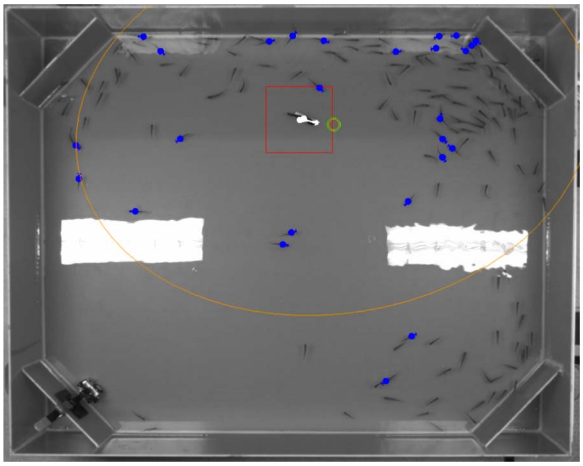


Fig. 3. ***A frame from the overhead video stream of a robotic predator fish chasing the centroid of a live school of golden shiners with real-time tracking output overlaid. The robotic predator's measured and tracked position and heading are each represented by a white dot and arrow and the restricted area for segmenting the robotic predator is outlined in red. The position and heading of each individual segmented live fish are represented by a blue dot and small blue arrow. The threshold τ for segmenting fish is set high, and therefore a relatively small number of individual fish are segmented. The orange ellipse represents the moment-based estimate of the school boundary, and the green circle represents the position of the school centroid. Reflections are present from overhead lights in the laboratory during this demonstration.***

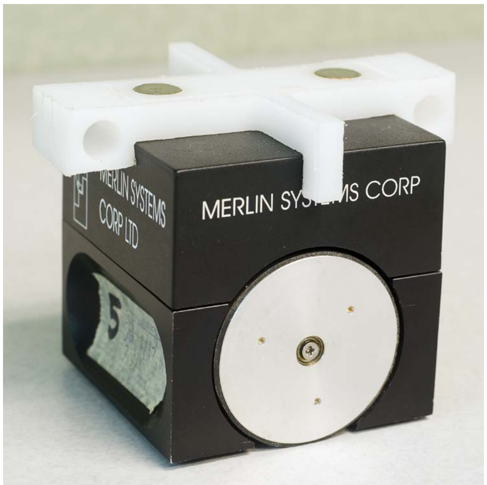


Fig. 4. ***MiaBot Pro wheeled robot with hood and a pair of cylindrical rare-earth magnets press-fit into holes in the hood insert.***

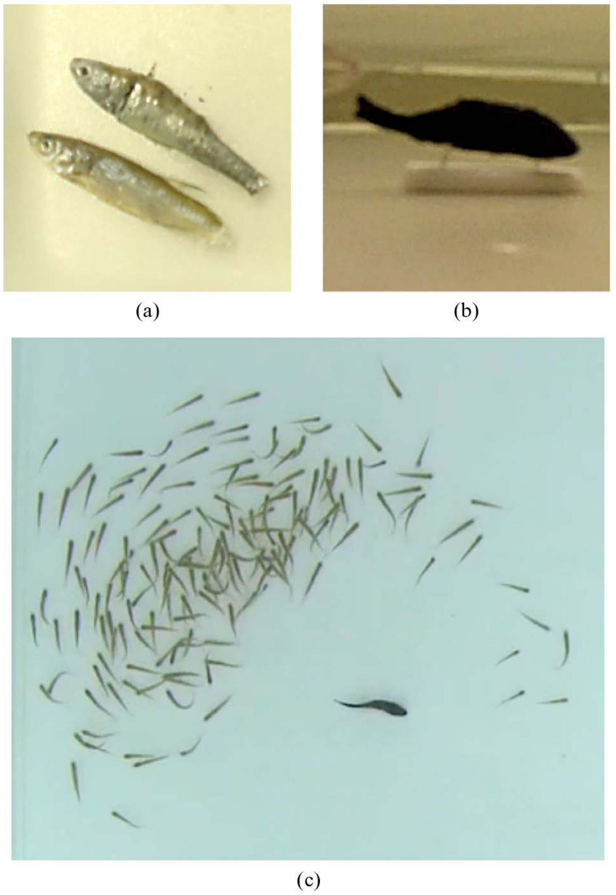
speed and turning rate—up to 3 m/s and 85 rad/s, respectively. Each robot consists of two wheels mounted on either side of a 7.5-cm cube containing a motor for each wheel, batteries, and electronics; see Fig. 4. The wheels are both parallel to the sides of the robot and their spin axis runs right through the middle of the cube. The speed of each wheel can be controlled independently forward and backward so that the robot can be made to translate (forward and backward) and to turn using speed differentials. For each motor there is an encoder, which is used onboard in a feedback loop that controls the wheel speed to a desired speed input to the robot. Plastic nubs prevent the robots from rocking forward or backward on their axles.

The wheeled robots are well suited to emulating the translation and turning movements of fish, although they cannot simulate body contortions and side drifting. Their high speed and turning rate make it possible to model many rapid fish behaviors, including those of a predator darting to pursue its prey. Achievable accelerations are also high with practical limits resulting from motor slip, latency in the closed-loop control, and drag forces acting on the model fish.

A machined plastic hood is mounted snugly to the top of each robot as shown in Fig. 4. Each hood has a milled rectangular slot with a rectangular plastic insert. Magnets are press-fit into holes that are drilled into the rectangular insert. A pair of cylindrical rare-earth magnets, 1/2-in diameter and 1/4-in thick, are separated by 2 in so that they are centered on the robot and aligned front to back with the direction of motion. Alternate magnet sizes and configurations can be accommodated by manufacturing a new insert. The height of the magnets can also be adjusted by inserting shims behind the rectangular insert. The two circular faces of each magnet correspond to its poles; they are inserted into the hood with opposite poles facing up. The opposite pole orientation facilitates torque transfer to the model fish when the robot turns and ensures that the correct mating orientation is maintained. The wheeled robot can sometimes lose traction as it pulls the model fish; however, weights can be added to the hood in order to increase the robot's traction.

3) *Model Fish:* The model fish appearance can be a critical factor in the success of an experiment, particularly if the robotic fish is to be treated as a conspecific by live fish. Different species respond differently to models. For example, in [1] and [2], it was observed that three-spined sticklebacks (*Gasterosteus aculeatus L.*) responded well to robotic conspecifics when realistic eyes were painted on the models. Golden shiners have proved to be more difficult in this regard. One of the golden shiner models in a series under development is shown in Fig. 5(a). This golden shiner model was created from a mold of a deceased golden shiner [also shown in Fig. 5(a)] using a hard plastic; the plastic allowed realistic painting but prevented good reproduction of structural features.

The live golden shiners responded, in the demonstrations described below, to a model koi used to simulate a predator fish. Fig. 5(b) shows an underwater view of the koi model mounted on its base. The model was created by casting from a mold of a deceased koi. The molding and



(a) (b)

(c)

Fig. 5. ***(a) Deceased golden shiner (bottom) and a model golden shiner (top) produced from a mold of the deceased shiner using a hard plastic. (b) Silicone koi predator model mounted on its magnetic base. The photo is taken from an underwater camera in the tank. (c) Overhead snapshot of robotic koi predator with live golden shiner school in the tank.***

model material are both tin-cured silicone rubber (MoldMax 30T and 15T, respectively) distributed by Smooth-On, Inc. [36]. The silicone rubber mold and model create flexible, high-quality replicas of the fish shape and texture. Painting silicone rubber requires specialty paints, and it is difficult to reproduce the markings on a fish. The model koi is painted black as seen in the overhead snapshot of Fig. 5(c) of the model moving with a live golden shiner school.

4) Tracking System Hardware: An Allied Vision Technologies Guppy F-080 grayscale FireWire camera [37] with 1032 × 778 resolution and a 4.5-mm lens is mounted 3 m above the tank. The camera is connected to a computer workstation for tracking, computation, and control. The workstation uses modern commodity hardware for real-time tracking, a FireWire port for the camera, and either built-in Bluetooth or an off-the-shelf USB Bluetooth adapter for communication with the robotic fish.

The camera is mounted so that the image plane is close to parallel with the plane of the tank bottom. As a result, feedback control of the robotic fish from its tracked position provides satisfactory performance without requiring scaling from robot wheel speeds to speed in image coordinates. Correction for refraction due to Snell's law can be neglected due to the shallow depth of water compared to the distance between the camera and the water. Future experiments requiring high precision may necessitate calibration of the camera system to obtain real-world coordinates. Calibration of the HD camera is also necessary for offline analysis so that experimental results can be reported accurately in real-world units.

B. Software

In this section, we give an overview of design of the tracking and control software package that we have developed.

1) The MADTraC Library: The Multi-Agent Dynamic Tracking and Control (MADTraC) library [38] was developed in the Dynamical Control Systems Laboratory (DCSL) at Princeton University, Princeton, NJ, to support several applications for which a high-performance tracking and control framework was needed. MADTraC provides a cross-platform C++ GUI framework and a collection of helper classes that enable rapid application development. The framework is quite flexible, focusing on applications that perform (real-time) tracking on acquired video, provide live visualization and parameter adjustment capabilities, and send control commands to an external device. The video source can be a video file, or one or multiple USB or FireWire cameras. Almost any tracking algorithm can be used through inheritance of the base tracking class. Many of the techniques described in this paper are provided as modules that can be plugged in to customize the tracking algorithm. Other features of the library include parameter persistence (i.e., the values of parameters are saved between successive runs of the program), data output formatting, and the ability to save screen shots and movies directly from within the application.

2) Communication and Control: Communication between the workstation and the MiaBot Pro wheeled robot is established with a virtual serial port over a Bluetooth channel. Multiple robots are easily handled by opening a separate port for each robot, up to seven robots. The number seven is a limitation of the Bluetooth standard and can be overcome with specialty hardware. Wheel speed commands are sent to the robots at each time step using an ASCII string protocol. The robots have onboard proportional–integral–derivative (PID) feedback control of wheel speeds using motor encoder feedback. This inner loop control is sufficiently fast so that robots can be commanded with desired wheel speeds.

Feedback laws that model conspecific or heterospecific behaviors are applied to produce desired steering and speed control for the robotic fish. For example, a feedback law for a conspecific robotic fish will typically require an update on steering in response to measurements of relative position, heading, or speed of near-neighbor fish. A feedback law for a predator robotic fish will typically require an update on steering and/or speed in response to changing features of the live fish school. The desired steering and speed signals are transformed into robot wheel speed commands as follows.

Let $(x(t), y(t))$ define the position of the robot in the plane at time t with speed $s(t)$ and heading direction $\theta(t)$ relative to the x-axis. Let $\omega(t) = d\theta(t)/dt$ be the steering rate. Then, left wheel speed $s_L(t)$ and right wheel speed $s_R(t)$ can be computed from $s(t)$ in meters per second and $\omega(t)$ in radians per second as

$$\begin{bmatrix} s_L \\ s_R \end{bmatrix} = \frac{k_e}{50} \begin{bmatrix} 1 & -\frac{L}{2} \\ 1 & \frac{L}{2} \end{bmatrix} \begin{bmatrix} s \\ \omega \end{bmatrix}.$$

The parameter $L = 0.07$ m is the robot wheelbase, $k_e = 4.0 \times 10^{-5}$ is a scaling factor to convert meters to wheel encoder counts, and 50 is a scaling factor internal to the robot's software.

To illustrate feedback control of the robotic fish, consider a robotic predator seeking to move from its current location to a target location (x', y') with no constraint on the angle of arrival. We prescribe the steering control as

$$\omega(t) = -k \sin(\theta(t) - \psi(t))$$

where $k > 0$ is a constant gain and $\psi(t) = \tan^{-1}((y(t) - y')/(x(t) - x'))$ is the bearing to the target. The controlled heading dynamics have a stable solution at synchronization of heading angle with target bearing $\theta(t) = \psi(t)$ and an unstable solution at antisynchronization of heading angle with target bearing $\theta(t) = \psi(t) + (2n + 1)\pi$, n an integer. We compute the speed control as

$$s(t) = \begin{cases} s_{\max}, & d(t) > d^* \\ s_{\max}\dfrac{d(t)}{d^*}, & d(t) \leq d^* \end{cases}$$

where $s_{\max} < 3$ m/s is the maximum desired speed of approach, $d(t) = ((x(t) - x')^2 + (y(t) - y')^2)^{1/2}$ is the distance between the robot and the target, and d^* is a threshold distance. The speed control ensures that the robot moves with a fixed constant speed at sufficiently large distances from the target and slows linearly with distance for distances smaller than the threshold.

The above simple control law is sufficient when there are few constraints. More sophisticated motion planning and feedback solutions for control of nonholonomic vehicles can easily be substituted.

III. REAL-TIME TRACKING

In this section, we describe our approach and implementation for real-time tracking that enables robotic fish to interact responsively with live fish and other features in the tank environment. In this context, tracking requires extracting from a video sequence time-varying estimates of the state of objects that live conspecific or heterospecific fish would sense on their own.

So that each robotic fish can control its own motion, we estimate its position and velocity. To provide the robotic fish with measurements of the relative position, heading, and/or speed of its neighbors, we estimate the position, heading, and speed of individual neighbors and subtract the position, heading, and speed of the robotic fish. We also extract information about enough of the school so that we can compute estimates of group-level quantities, such as the centroid, boundary, polarity, and momentum of the fish school.

Feedback control of the robotic fish demands a real-time solution at a frame rate of approximately 10 frames/s. This rules out some existing approaches that are computationally costly and motivates us to implement our own computationally efficient solution. To maximize computational performance, our solution is implemented in C++ using the OpenCV computer vision library [39] to perform individual steps wherever possible. Other calculations are implemented in custom modules, many of which we have bundled into a separate library (see Section II-B1). Latency is further minimized by integrating tracking and vehicle control into the same compiled piece of software. In practice, we have been able to track tens of fish in real time.

The first step of the tracking problem is segmentation, which we describe in Section III-A. Segmentation involves extracting measurements for individual objects from the image. Data association is the assignment of each of these measurements to a tracked object, for example, assignment of the center of a blob in the image to the position of a fish that it represents. In Section III-B, we describe data association as part of the state estimation step. In the state estimation step, current measurements are incorporated with previous measurements and a dynamic model of the system in order to estimate the state of the tracked objects; the state of an individual fish includes its position in the horizontal plane, its heading angle, and its speed. Fig. 6 illustrates the steps of segmentation and state estimation where the input is the current image and the output is the estimated current position, heading angle, and speed of each tracked object. From the estimated states, properties of fish schools can be computed as discussed in Section III-C.

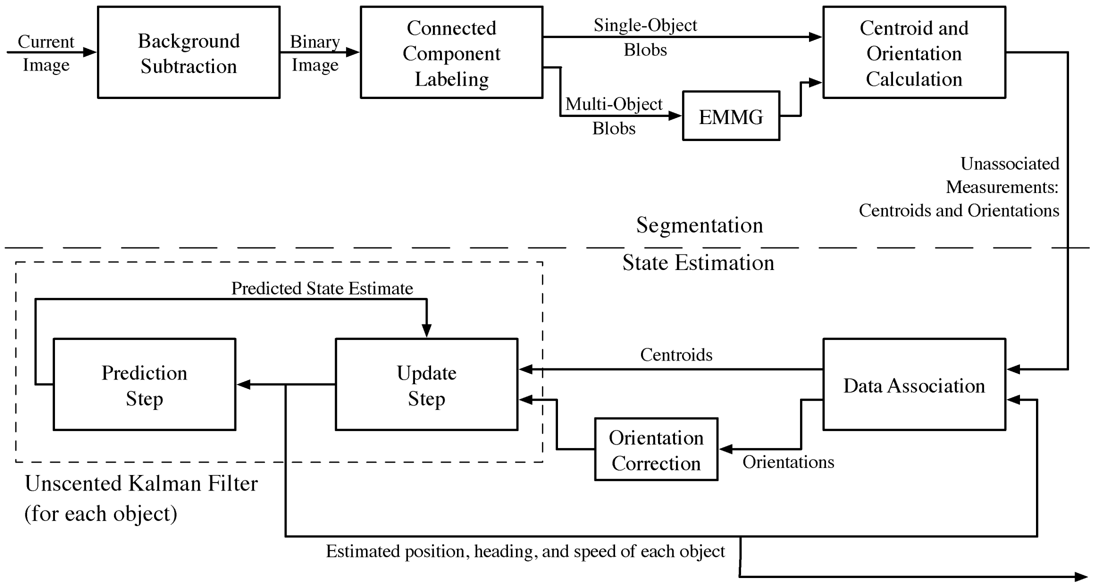


Fig. 6. *Schematic of segmentation and state estimation for tracking. There is a UKF for each tracked object. The output of each UKF provides an estimate of the position, heading angle, and speed of the tracked object.*

A. Segmentation

Segmenting the images of individual fish in a school is particularly challenging; conspecifics look nearly identical, and they swim very close to one another. On the other hand, the high degree of color contrast between the fish and the tank makes it relatively easy to distinguish pixels that belong to the image of a fish from those that belong to the background. The color uniformity among fish and the controlled lighting in the laboratory mean that we do not need a sophisticated color-based segmentation algorithm [40]. Further, since the number of fish in the arena is constant, segmentation does not have to account for fish entering or leaving the scene.

In the first step of segmentation, given the current image, we use a background subtraction algorithm to isolate image pixels that are likely to belong to an object of interest. In the second step, we use a component labeling algorithm to group those pixels into discrete connected components called "blobs." Because of occlusions, a single blob may sometimes represent multiple objects (e.g., multiple fish swimming close); thus, we combine the labeling algorithm with an algorithm that further segments objects within a multiobject blob. Sorting out the identities of the objects is the data association step, which we discuss as part of state estimation in Section III-B.

In the third step, we use image moments to calculate the centroid location (x_{cm}, y_{cm}) and the orientation angle θ_m of each tracked object (e.g., each fish). We assume that the orientation angle is identical to the heading angle. This is true by design in the case of the robotic fish. Live fish do occasionally drift with a velocity component perpendicular to their orientation, but in a tank with no externally induced flow this is usually a minimal effect. The centroid and heading angle of each tracked object as determined by the segmentation step provide the measurements that are input to the UKF for state estimation.

Because the live and robotic fish are significantly darker than the tank, a sign-aware background subtraction algorithm successfully locates objects of interest in the scene. Sign-aware background subtraction can be viewed as a likelihood-ratio test given a background model that has a spatially varying mean (the background image) and constant variance (proportional to a threshold τ). The background frame $\bar{I}$ is calculated by averaging images of the arena prior to experiments. Then, at each time step, the image is subtracted from the background frame. The differences at time t for all pixels are stored in the difference frame $I_D(t)$ with negative values truncated to zero. $I_D(t)$ is therefore zero at pixel locations where the image is brighter than the background. The binary image $I_T(t)$ is computed by applying the threshold τ to $I_D(t)$: $I_T(t)$ is set to 0 at pixel locations where $I_D(t)$ is less than or equal to τ and to 255 otherwise. The value of τ is determined empirically and set manually in the software. It can be adjusted online, although with consistent lighting conditions this is not usually necessary. Using 0 and 255 in the binary image facilitates the display of results without an extra scaling step.

The second step in segmentation groups pixels into blobs and labels each blob. For single objects in uncluttered environments, this is typically addressed with algorithms that find and label connected components within the image. We use an efficient connected-component labeling algorithm based on the one implemented in

OpenCV [39] and first presented by Chang *et al.* [41]. We also apply an algorithm to handle occlusions so that we can further distinguish and label multiple objects that appear together in a single blob. Fish are relatively consistent in their appearance from one frame to the next, except for occasional contortions during startle responses [42], [43]. As a result, the overhead image of a live fish can be modeled effectively as a thin ellipse and this means that we can resolve occlusions using an EMMG algorithm (see [27] and [28] for background on expectation–maximization and mixture models and [44] and [45] for details on implementing the algorithm for a Gaussian mixture model).

EMMG is an iterative algorithm that can be computationally expensive. However, our implementation is made efficient by combining the EMMG with the connected-component labeling algorithm. Area and perimeter thresholds are used to filter the output of the component labeling algorithm and determine which blobs are likely to contain more than one object. The number of objects in a blob is determined based on the blob area and perimeter as compared to the average blob area and perimeter. For those blobs containing more than one object, the EMMG algorithm is used to segment individual objects: each multiobject blob is optimally fitted to a set of Gaussian distributions where there is one distribution for each object. The parameters of the distributions provide estimates of the elliptical shape of each of the segmented objects in a multiobject blob. Likelihoods calculated from these distributions are used to assign individual pixels to objects. A pixel can belong to more than one object when there is overlap of object images and therefore overlap of distributions. Each pixel in the original multiobject blob is assigned to 1) the object with the highest likelihood of having produced that pixel; and 2) any object for which the pixel is located within an ellipse defined by a two standard deviation level set of the object's distribution.

Image moments have been widely used in pattern recognition and computer vision applications [46]; see the review by Prokop and Reeves [47]. We calculate image moments for each labeled object and use these moments to estimate the object's centroid and orientation. The (j,l)th image moment M_{jl} of a labeled object O is defined as

$$M_{jl} = \sum_{x,y \in O} x^j y^l$$

where $x, y \in O$ if the pixel at location (x,y) belongs to the object O. M_{00} is the area (number of pixels) and $(x_{cm}, y_{cm}) = ((M_{10}/M_{00}), (M_{01}/M_{00}))$ is the centroid location of the object. The orientation of an object can be estimated as

$$\theta_m = \frac{1}{2}\tan^{-1}\frac{2(M_{11} - x_{cm}M_{01})}{M_{20} - M_{02} - x_{cm}M_{10} + y_{cm}M_{01}}.$$

However, this estimate is only accurate up to a rotation by π; θ_m could point in the direction of the head or the tail.

The direction of the head can be determined using a third-order projection method that leverages the fact that a fish's head is fatter than its tail. Prokop and Reeves [47] discussed this method in general and DeFroment [48] suggested applying it to head/tail disambiguation for fish. When the pixel distribution is projected onto the direction defined by θ_m, the sign of the skewness will be negative if θ_m corresponds to the head direction and positive if θ_m corresponds to the tail direction. In our approach, we compute a factor $\bar{\gamma}$ that is proportional to the skewness

$$\bar{\gamma} = \mu_{30}\cos^3\theta_m + 3\mu_{21}\cos^2\theta_m\sin\theta_m + 3\mu_{12}\cos\theta_m\sin^2\theta_m + \mu_{03}\sin^3\theta_m$$

where

$$\begin{aligned}\mu_{30} &= M_{30} - 3x_{cm}M_{20} + 2x_{cm}^2M_{10}\\ \mu_{21} &= M_{21} - 2x_{cm}M_{11} - y_{cm}M_{20} + 2x_{cm}^2M_{01}\\ \mu_{12} &= M_{12} - 2y_{cm}M_{11} - x_{cm}M_{02} + 2y_{cm}^2M_{10}\\ \mu_{03} &= M_{03} - 3y_{cm}M_{02} + 2y_{cm}^2M_{01}\end{aligned}$$

are the centralized moments. The orientation estimation can then be adjusted following the rule

$$\theta_m \longleftarrow \begin{cases}\theta_m + \pi, & \bar{\gamma} > 0\\ \theta_m, & \bar{\gamma} \le 0.\end{cases}$$

That is, if $\bar{\gamma} > 0$, then the distribution is incorrectly skewed and therefore the angle should be flipped.

Fig. 7 shows a snapshot of the successful segmentation of ten live fish where the fish are very close to one another. The measured heading θ_m is shown for each fish as a short arrow. The tracked position of each fish is represented by a colored dot and the estimated heading angle is represented by the longer colored arrow; these are the output of the state estimation described in the next section.

B. State Estimation

We estimate the state of the tracked objects using UKFs. We assume the dynamics of each tracked object are independent, and we use a separate UKF for each object. For each tracked object, the input to the UKF is the measurement vector $\mathbf{z} = (x_{cm}, y_{cm}, \theta_m)$ as computed in the segmentation scheme described in Section III-A. For each tracked object, the state is modeled as

$$\mathbf{x}(t) = [x_o \quad y_o \quad \theta_o \quad s_o]^T$$

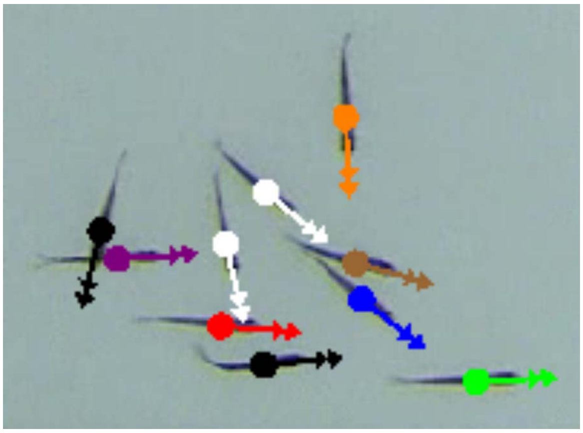


Fig. 7. ***Snapshot of ten live golden shiners with tracking output overlaid. Each colored dot represents a fish position. Each arrow represents the corresponding heading angle: the shorter arrow is the measured heading θ_m and the longer arrow is the estimated heading $\hat{\theta}$.***

where (x_o, y_o) is the modeled centroid location of the object, θ_o is the modeled heading angle of the object, and s_o is the modeled speed of the object.

The measurement model for each UKF is given by

$$\mathbf{z}(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{x}(t) + \boldsymbol{\eta}(t) \tag{1}$$

where $\boldsymbol{\eta}$ is a measurement noise vector. The noise vector is assumed to be drawn at each time step from a zero-mean normal distribution with diagonal covariance matrix $R = \mathrm{diag}\{\sigma_{pm}^2, \sigma_{pm}^2, \sigma_{\theta m}^2\}$, where σ_{pm} and $\sigma_{\theta m}$ are the variance in position and heading measurements, respectively. These variances can be modified online in our implementation.

The dynamic model for each object state is given by

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \begin{pmatrix} s_o(t)\Delta t \cos\theta_o(t) \\ s_o(t)\Delta t \sin\theta_o(t) \\ 0 \\ 0 \end{pmatrix} + \boldsymbol{\zeta}(t) \tag{2}$$

where Δt is the time step and $\boldsymbol{\zeta}$ is a disturbance vector. The disturbance vector is assumed to be drawn from a normal distribution with zero mean and a diagonal covariance matrix $Q = \mathrm{diag}\{\sigma_d^2, \sigma_d^2, \sigma_\theta^2, \sigma_s^2\}$, where σ_d, σ_θ, and σ_s are disturbance variances for position, heading, and speed, respectively. The disturbance variances can also be modified online.

Each UKF iteration has two steps: prediction and update. During the prediction step, the state estimate $\hat{\mathbf{x}}$ is projected forward in time according to the dynamic model (2) with $\boldsymbol{\zeta} = 0$ to produce the predicted state $\bar{\mathbf{x}}$. The measurement model (1) with $\boldsymbol{\eta} = 0$ is used to compute the predicted measurement $\bar{\mathbf{z}}$ from the predicted state $\bar{\mathbf{x}}$. During the update step, the state estimate $\hat{\mathbf{x}}$ is updated by comparing the predicted measurement $\bar{\mathbf{z}}$ with the actual measurement $\mathbf{z}$. The estimation of covariance matrices is used in both prediction and update steps. The output of each UKF at time t is the updated estimate $\hat{\mathbf{x}}(t)$, which gives the new estimate of the position, heading, and speed of the object at time t.

There are two additional processing steps inside the state estimation scheme that improve the quality of the estimation. The first addresses the data association problem, which refers to the problem of determining which measurement vector should be associated to which state vector at each time step. This ensures that we track the same fish in consecutive time steps. Because the number of real and robotic fish in a given experiment remains fixed and the EMMG algorithm extracts every individual object of interest, the association is always one to one. We apply the efficient Hungarian matching algorithm developed by Kuhn [29] and expanded by Munkres [30], and we leverage the implementation in C [49]. At time t, the algorithm makes associations between measured objects and tracked objects so that the sum of distances between the tracked objects' estimated positions at time $t-1$ and the associated measured objects' centroids at time t is minimized. This method works well in practice and is computationally inexpensive for moderate numbers of objects.

We also perform additional processing on the heading measurement at each time step of the UKF for two important reasons. The first is that occlusions and other noise in the image can cause incorrectly oriented measurements despite application of the algorithm described above for determining head and tail directions. The second is that, for continuity, the heading state must evolve over $\mathbb{R}$, whereas the orientation measurement is always in the range $[-\pi, \pi]$.

To address the first concern, a position history of each tracked object is stored for five time steps and the measurement is aligned with the direction of displacement between the current and the oldest position as long as the displacement is sufficiently large (comparable to one half of a body length). If the displacement is not large enough or there is insufficient history, the previous state is used for alignment. The alignment decision is made according to the following rule:

$$\theta_m \longleftarrow \begin{cases} \theta_m + \pi, & \left|\cos\dfrac{\theta_m - \alpha}{2}\right| \le \rho \\ \theta_m & \left|\cos\dfrac{\theta_m - \alpha}{2}\right| > \rho \end{cases}$$

where α is the alignment reference angle (e.g., the displacement direction or the previous heading) and ρ parameterizes the degree of alignment required. We use $\rho = \sqrt{2}/2$, since it corresponds to flipping the measurement angle if it is separated from the alignment reference by more than $\pi/2$.

To preserve continuity of the heading angle estimate, the measurement is again modified following the rule

$$\theta_m \longleftarrow \hat{\theta} + \tan^{-1}\frac{\sin(\theta_m - \hat{\theta})}{\cos(\theta_m - \hat{\theta})}$$

where $\hat{\theta}$ is the current heading estimate from the UKF. For example, if the current heading estimate is $\hat{\theta} = \pi$ and the measurement is $\theta_m = -3.13$ (i.e., the actual heading has rotated slightly counterclockwise since the last time step), this algorithm adjusts the measurement to $\theta_m = -3.13 + 2\pi \approx 3.15$. It is important here to calculate the arctangent using a four-quadrant method; the implementation uses the C function `atan2`.

C. Estimating Properties of the School

To compute properties of the fish school, we use the estimated states of individual fish that are being tracked. For example, the centroid location of the school is computed by averaging the estimated positions of the tracked fish. Linear momentum is computed by summing the estimated velocities of the tracked fish.

In the case that we are not otherwise tracking individual fish, we opt for a more efficient approach in which we approximate school properties using only a subset of fish and their measured positions and orientations as provided by the segmentation algorithm. For example, the centroid location of the school is computed in this case by averaging the centroid locations of the subset of fish that have been segmented.

We use thresholds in the segmentation algorithm to adjust how much of the school to include in the subset—the goal is to strike a good balance between computational speed and accuracy of estimates of school properties. When fish are not included in the subset, it is typically because they are too small or because a cluster of fish appear as one large connected component. Therefore, the error properties of estimating centroid location or any of the quantities discussed below are related to the distribution properties of small fish and clusters within the school. This is difficult to predict and can vary even within the same group of fish over the course of an experiment. Optimistically, we may assume that segmentation errors are distributed uniformly across the group and therefore estimation errors are small.

To estimate the polarity of the school, we compute a measure of synchrony of direction of motion that derives from the study of coupled oscillator dynamics [50]. Suppose there are N fish and fish i has measured heading $\theta_i(t)$ at time t. The synchrony measure $p(t) \in [0,1]$ is computed as

$$p(t) = \frac{1}{N}\left(\sum_{i=1}^{N}\sum_{k=1}^{N}\cos(\theta_i(t) - \theta_k(t))\right)^{1/2}. \tag{3}$$

When $p = 1$, all the fish are heading in the same direction and the school is maximally polarized. When $p = 0$ the fish are heading in very different directions and the school is minimally polarized. For example, when fish in a school are uniformly distributed and moving around a circle, then p will be close to zero. Similarly, when fish are milling around and moving in random directions that are uniformly distributed, then p will also be close to zero. More sophisticated calculations can distinguish among these motion patterns and other features of directionality within a school [8].

One method we use to estimate the boundary of the fish school is to compute a bounding ellipse from the spatial covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix}$$

where

$$\sigma_{xx} = \sum_{i=1}^{N}(x_i - x_C)^2 \quad \sigma_{yy} = \sum_{i=1}^{N}(y_i - y_C)^2$$

$$\sigma_{xy} = \sum_{i=1}^{N}(x_i - x_C)(y_i - y_C)$$

(x_i, y_i) is the estimated position of fish i, and (x_C, y_C) is the estimated centroid of the school. Then, the bounding ellipse can be approximated as a level set of the bivariate Gaussian distribution in x and y with covariance Σ.

Level sets of this distribution are equivalent to level sets of the function

$$f(x,y) = \frac{\sigma_{xx}(y - y_C)^2 - 2\sigma_{xy}(x - x_C)(y - y_C) + \sigma_{yy}(x - x_C)^2}{\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2}$$

which is proportional to the argument of the exponential in the distribution. The level sets $f(x,y) = c$ are ellipses centered at (x_C, y_C) with major axis rotated from the x-axis by an angle

$$\beta = \frac{1}{2}\tan^{-1}\frac{2\sigma_{xy}}{\sigma_{xx} - \sigma_{yy}}.$$

The expression $\sqrt{\lambda_i c}$ gives the semimajor axis for $i = 2$ and the semiminor axis for $i = 1$, where $\lambda_2 > \lambda_1$ are the eigenvalues of Σ. We make the choice

$$c = \frac{1}{2\left(\pi^2\left(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2\right)\right)^{\frac{1}{4}}}.$$

This yields an ellipse that visually matches well the fish schools in our tests and demonstrations. For example, Fig. 1 shows four frames from the overhead video with the centroid locations and bounding ellipses as calculated by this method. The ellipse algorithm is computationally inexpensive, but it has the potential to poorly represent the boundary of a school with a more complex shape as is the case in Fig. 3. Viable alternatives include convex hull computation (for which there are many algorithms available) and alpha shape algorithms (see [51]).

It is possible to imagine classes of experiments in which it is important to determine the location of the fish that is closest to the robotic fish and yet still within the group, the position on the boundary of the group that is closest to the robotic fish, the position of the fish in the front or rear of the group, and so on. These are all computations that can be made with the estimates available from our tracking routine.

IV. DEMONSTRATIONS

In this section, we describe two demonstrations of our testbed with feedback-controlled robotic fish that are designed to interact in real time with a live fish school. The demonstrations provide just two examples of the many ways in which the robotic fish can be manipulated to expand the possibilities for behavioral experiments with fish schools.

A. School Centroid Chasing

In the first demonstration, a robotic predator fish uses real-time feedback in order to chase the centroid of a live fish school. By segmenting out a fraction of the individual fish and using their positions to estimate the location of the centroid, the robotic fish can persistently chase the moving centroid of the school. The feedback control law is as described in the example of Section II-B2 where the robotic fish steers to head towards a target and the target is the time-varying centroid of the school.

This demonstration provides a manipulation of a robotic heterospecific that cannot be achieved without real-time tracking and control. The automated real-time feedback is needed because the live fish move in response to the robotic predator, which moves continuously in response to the live fish. Thus, preplanning the motion of the robotic fish is not possible because the centroid of the fish school cannot be known in advance. Similarly, manual control of the robotic fish would likely fail because the human operator would have difficulty computing the continuously and possibly fast changing centroid location.

Fig. 3 shows one frame from the overhead video stream during a centroid chasing demonstration. The robotic predator fish is modeled after a koi, as shown in Fig. 5(b), and it chases the centroid of a school of golden shiners. In Fig. 3, the tracked position of the robotic predator fish is shown with a white dot and the tracked heading by a white arrow. To track the robotic fish efficiently, an algorithm was used whereby segmentation for the robotic fish was first restricted to a small square region of the image about the robotic fish's last known position. If no blob was found in this square region, then segmentation for the robotic fish was done on the full image. The restricted segmentation area is shown with a red box in Fig. 3. The threshold τ for segmenting live individual fish was set high to speed up the computation; the resulting subset of segmented fish contained fewer than one-fourth of the fish school. Those fish that were measured are identified in Fig. 3; the measured position of each fish is shown as a blue dot and the corresponding measured orientation is shown with a small blue arrow. The green circle shows the centroid of the measured subset of fish. In Fig. 3, the robotic fish can be seen to be heading straight for the centroid of the school.

B. Triggered Dart Towards School

In the second demonstration, a robotic predator fish is triggered to dart towards a live fish school when the real-time estimate of the fish school polarity $p(t)$, defined in (3), is close to zero. For small values of $p(t)$, the school is likely to be randomly milling around or moving in a circle. The robotic fish is again the black model koi of Fig. 5(b) and a school of live golden shiners is under pursuit. The robotic fish begins in a random position in the tank, and when $p(t)$ is below a threshold, a dart is initiated. To dart, the robotic fish accelerates quickly towards the current centroid of the school.

In this demonstration, a relatively low value was used for the segmentation threshold τ. This allowed approximately 50% of the fish school to be included in the polarity computation. The value of the attack threshold for $p(t)$ was 0.5. Variations of the demonstration are possible where the trigger depends not only on polarity but also on position of the school, shape of the school, etc.

V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we describe a new, cyber–physical implementation of interacting live and robotic fish that extends the current scope of behavioral experimentation with fish schools. Central to the implementation, the robotic fish can be manipulated to behave as conspecifics or heterospecifics using real-time feedback control. The real-time feedback is provided by means of real-time tracking of individual fish and real-time computation of fish school properties. Control laws are designed that allow the robotic fish to adopt behaviors that are responsive to the other fish (live and robotic) and the rest of the environment.

We describe two demonstrations in which a robotic fish adopts a predatory responsive behavior, and the dynamic response of the live fish can be measured. In the first demonstration, the robotic predator continuously chases the centroid of the live fish school, and in the second demonstration, the robotic predator darts towards the centroid of the live fish school when the school's polarity is below a threshold.

There are many other possibilities for new kinds of investigations with our cyber–physical implementation, including those in which the robotic fish behave as conspecifics. Indeed, it has recently been possible to infer the mechanism of interactions among golden shiners [52]. It was demonstrated that the position and speed of neighbors play an important role, but fish do not explicitly match body orientation. These new insights introduce the possibility of implementing highly realistic interactions with one, or multiple, replica conspecifics.

Real-time feedback-controlled robotic fish provide an excellent opportunity for testing specific hypotheses that are very difficult, or impossible, with conventional experimentation. For example, with robotic fish it is possible to investigate higher order components of interactions, such as the influence of acceleration. While in [52], it was demonstrated that position, speed, and acceleration are likely to be important, it was not possible to fully elucidate their influence due to intrinsic correlations between individuals.

Another important avenue of research that can benefit from our new technology concerns the relationship between the spatial position adopted by individuals and the influence they have on the motion characteristics of the school. A replica fish could be moved to specific positions within the group and then made to suddenly rotate, or accelerate, to mimic detection of a predator. The response of real fish to such perturbations would be quantified using the HD tracking data output. Similarly, the robotic predator could be used to create controlled perturbations to investigate how individuals in groups, both individually and collectively, assess and respond to threats. ■

Acknowledgment

The authors would like to thank A. Stewart for building the tank, G. Young for coding the EMMG algorithm, and Y. Katz for discussions on tracking. They would also like to thank L. Lee, Q. Wan, M. Abromowitz, and Y. Katz for help with creation of the model fish.

REFERENCES

[1] A. J. W. Ward, D. J. T. Sumpter, I. D. Couzin, P. J. B. Hart, and J. Krause, "Quorum decision-making facilitates information transfer in fish shoals," *Proc. Nat. Acad. Sci.*, vol. 105, no. 19, pp. 6948–6953, 2008.

[2] J. J. Faria, J. R. G. Dyer, R. O. Clèment, I. D. Couzin, N. Holt, A. J. W. Ward, D. Waters, and J. Krause, "A novel method for investigating the collective behaviour of fish: Introducing 'Robofish'," *Behav. Ecol. Sociobiol.*, vol. 64, pp. 1211–1218, 2010.

[3] J. K. Parrish and L. Edelstein-Keshet, "Complexity, pattern and evolutionary trade-offs in animal aggregation," *Science*, vol. 284, pp. 99–101, 1999.

[4] W. D. Hamilton, "Geometry for the selfish herd," *J. Theor. Biol.*, vol. 31, pp. 295–311, 1971.

[5] R. Vabo and L. Nottestad, "An individual-based model of fish school reactions: Predicting antipredator behaviour as observed in nature," *Fisheries Oceanogr.*, vol. 6, pp. 155–171, 1997.

[6] J. Krause and G. D. Ruxton, *Living in Groups*. Oxford, U.K.: Oxford Univ. Press, 2002.

[7] I. D. Couzin, "Collective cognition in animal groups," *Trends Cogn. Sci.*, vol. 13, no. 1, pp. 36–43, 2009.

[8] D. A. Paley, N. E. Leonard, R. Sepulchre, D. Grunbaum, and J. K. Parrish, "Oscillator models and collective motion: Spatial patterns in the dynamics of engineered and biological networks," *IEEE Control Syst. Mag.*, vol. 27, no. 4, pp. 89–105, Aug. 2007.

[9] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis, "Collective motion, sensor networks, and ocean sampling," *Proc. IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007.

[10] C. M. Breder, "Equations descriptive of fish schools and other animal aggregations," *Ecology*, vol. 35, no. 3, pp. 361–370, 1954.

[11] I. D. Couzin and J. Krause, "Self-organization and collective behavior in vertebrates," *Adv. Study Behavior*, vol. 32, pp. 1–75, 2003.

[12] B. Partridge and T. Pitcher, "The sensory basis of fish schools: Relative roles of lateral line and vision," *J. Comparat. Physiol. A, Neuroethol. Sensory Neural Behav. Physiol.*, vol. 135, pp. 315–325, 1980.

[13] R. Tegeder and J. Krause, "Density dependence and numerosity in fright stimulated aggregation behaviour of shoaling fish," *Phil. Trans. R. Soc. B*, vol. 350, pp. 381–390, 1995.

[14] S. Viscido, J. Parrish, and D. Grunbaum, "Individual behavior and emergent properties of fish schools: A comparison of observation and theory," *Mar. Ecol. Progr. Ser.*, vol. 273, pp. 239–249, 2004.

[15] D. J. Hoare, I. D. Couzin, J.-G. J. Godin, and J. Krause, "Context-dependent group size choice in fish," *Animal Behavior*, vol. 67, pp. 155–164, 2004.

[16] E. Hensor, I. D. Couzin, R. James, and J. Krause, "Modelling density-dependent fish shoal distributions in the laboratory and field," *Oikos*, vol. 110, pp. 344–352, 2005.

[17] J. Gautrais, C. Jost, M. Soria, A. Campo, S. Motsch, R. Fournier, S. Blanco, and G. Theraulaz, "Analyzing fish movement as a persistent turning walker," *Math. Biol.*, vol. 58, pp. 429–445, 2009.

[18] J. Krause, A. F. T. Winfield, and J.-L. Deneubourg, "Interactive robotics in experimental biology," *Trends Ecol. Evol.*, vol. 26, no. 7, pp. 369–375, 2011.

[19] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tache, I. Said, V. Durier, S. Canonge, J. M. Ame, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, and J. L. Deneubourg, "Social integration of robots into groups of cockroaches to control self-organized choices," *Science*, vol. 318, pp. 1155–1158, 2007.

[20] M. Milinski, J. H. Luth, R. Eggler, and G. A. Parker, "Cooperation under predation risk: Experiments on costs and benefits," *Proc. R. Soc. Lond. B*, vol. 264, pp. 831–837, 1997.

[21] M. Aureli, V. Kopman, and M. Porfiri, "Free-locomotion of underwater vehicles actuated by ionic polymer metal composites," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 4, pp. 603–614, Aug. 2010.

[22] E. Morais, M. F. Campos, F. Padua, and R. Carceroni, "Particle filter-based predictive tracking for robust fish counting," in *Proc.*

XVIII Brazilian Symp. Comput. Graph. Image Process., 2005, pp. 367–374.

[23] S. Butail and D. A. Paley, "3D reconstruction of fish schooling kinematics from underwater video," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2438–2443.

[24] A. Feldman, M. Hybinette, T. Balch, and R. Cavallaro, "The multi-ICP tracker: An online algorithm for tracking multiple interacting targets," *J. Field Robot.*, 2012, to be published.

[25] A. D. Straw, K. Branson, T. R. Neumann, and M. H. Dickinson, "Multi-camera real-time three-dimensional tracking of multiple flying animals," *J. R. Soc. Interface*, vol. 8, no. 56, pp. 395–409, Jul. 2010.

[26] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. Dickinson, "High-throughput ethomics in large groups of *drosophila*," *Nature Methods*, vol. 6, pp. 451–457, 2009.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc., B*, vol. 39, no. 1, pp. 1–38, 1977.

[28] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon Press, 1995.

[29] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, pp. 83–97, 1955.

[30] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Indust. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

[31] M. Isard and J. MacCormick, "BraMBLe: A Bayesian multi-blob tracker," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, pp. 34–41.

[32] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 279–290.

[33] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. 11th Int. Symp. Aerosp./Defense Sens. Simul. Controls*, pp. 182–193, 1997.

[34] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. Adaptive Syst. Signal Process. Commun. Control Symp.*, 2000, pp. 153–158.

[35] Merlin Systems Corp., *Merlin Robotics,* 2011. [Online]. Available: http://www.merlinsystemscorp.co.uk/index.php/e-shop/merlin-robotics.html

[36] SmoothOn, Inc., 2011. [Online]. Available: http://www.smooth-on.com

[37] Allied Vision Technologies, *Guppy F-080: FireWire Camera,* 2011. [Online]. Available: http://www.alliedvisiontec.com/us/products/cameras/firewire/guppy/f-080bc.html

[38] D. T. Swain, *MADTraC Framework Source Documentation*, 2011. [Online]. Available: http://poincare.princeton.edu/MADTraC/docs/

[39] G. Bradski, *OpenCV (Open Computer Vision) Library*, Mar. 2011. [Online]. Available: http://opencv.willowgarage.com/wiki/Welcome

[40] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[41] F. Chang, C.-J. Chen, and C.-J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Comput. Vis. Image Understand.*, vol. 93, no. 2, pp. 206–220, 2004.

[42] D. Weihs, "The mechanism of rapid starting of slender fish," *Biorheology*, vol. 10, pp. 343–350, 1973.

[43] P. W. Webb and J. M. Skadsen, "Strike attacks of Esox," *Can. J. Zoology*, vol. 58, pp. 1462–1469, 1980.

[44] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Dept. Electr. Eng. Comput. Sci., Univ. California Berkeley, Berkeley, CA, Tech. Rep. TR-97-021, Apr. 1998.

[45] I. D. Dinov, *Expectation Maximization and Mixture Modeling Tutorial. UC Los Angeles: Statistics Online Computation Resource*, 2008. [Online]. Available: http://www.escholarship.org/uc/item/1rb70972

[46] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[47] J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," in *Proc. Graph. Models Image Process.*, 1992, pp. 438–460.

[48] A. de Froment, Personal communication, 2009.

[49] B. P. Gerkey, *C Implementation of the Hungarian Method*, 2008. [Online]. Available: http://robotics.stanford.edu/ gerkey/tools/hungarian.html

[50] S. H. Strogatz, "From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators," *Physica D*, vol. 143, pp. 1–20, 2000.

[51] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 4, pp. 551–559, Jul. 1983.

[52] Y. Katz, C. Ioannou, K. Tunstrom, C. Huepe, and I. D. Couzin, "Inferring the structure and dynamics of interactions in schooling fish," *Proc. Nat. Acad. Sci.*, 2011.

ABOUT THE AUTHORS

Daniel T. Swain (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from The Pennsylvania State University, State College, in 2003 and 2005, respectively, where his research projects included vision-based feedback control of laser welding and detection of contraband using nuclear quadrupole resonance spectroscopy with high-temperature superconducting probes. Currently, he is working towards the Ph.D. degree in the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ.

His current research interests include the role of individual dynamics in information flow in natural and engineered collective motion, applied dynamical systems and control theory, robotics, and computer vision.

Iain D. Couzin received the B.Sc. degree from the University of Saint Andrews, Saint Andrews, Scotland, in 1996, the Ph.D. degree from The University of Bath, Bath, U.K., in 1999, and the M.A. degree from the University of Oxford, Oxford, U.K., in 2004.

He held postdoctoral positions at the University of Leeds, Leeds, U.K. (2000–2002), Princeton University, Princeton, NJ (2002–2003), and the University of Oxford, where he was Junior Fellow in the Sciences at Balliol College (2003–2005) and Royal Society University Research Fellow (2005–2007). He is now an Assistant Professor in the Department of Ecology and Evolutionary Biology and Associated Faculty in the Princeton Environmental Institute, the Program in Applied and Computational Mathematics and the Princeton Institute for Computational Science and Engineering at Princeton University. His research aims to reveal the fundamental principles that underlie evolved collective behavior, and consequently he works on a wide range of biological systems from insect swarms to fish schools and human crowds.

Naomi Ehrich Leonard (Fellow, IEEE) received the B.S.E. degree in mechanical engineering from Princeton University, Princeton, NJ, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1991 and 1994, respectively.

From 1985 to 1989, she worked as an Engineer in the electric power industry for MPR Associates, Inc. She is now the Edwin S. Wilsey Professor of Mechanical and Aerospace Engineering and an associated faculty member of the Program in Applied and Computational Mathematics at Princeton University. She works in nonlinear control and dynamics with current interests in cooperative control, collective motion, and collective decision making. Applications include mobile sensor networks, collective animal behavior, and decision dynamics in mixed teams of humans and robots.