2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

3D reconstruction of fish schooling kinematics from underwater video

Sachit Butail and Derek A. Paley

Abstract— **This paper describes a probabilistic framework to estimate the shape and position of multiple fish in a school. We model the fish shape as an ellipsoid with a curvature coefficient that allows us to incorporate bending. An expression for the extremal contour in terms of state parameters is used to derive a likelihood function for shape. We present a motion model that uses curvature as an input to the turning rate. Tracking is performed using a particle filter with joint probabilistic data association. We evaluate our algorithm using simulated data and further characterize its performance using real data from a laboratory experiment with six giant danios.**

I. INTRODUCTION

Animal aggregations have fascinated and inspired researchers studying collective behavior in many species [14]. Where biologists stand to gain from tools in engineering that help advance the understanding of animal groups, engineers use this improved understanding to design bio-inspired robotic systems. Among animals that demonstrate collective behavior, fish are particularly attractive because a wide variety of schooling fish are easy to procure and maintain in a laboratory environment.

While there are many control strategies that appear to replicate collective behavior [24], [12], there are far fewer experiments that have managed to quantify this kind of behavior in nature. A major reason for this is that automatic tracking of multiple targets is inherently hard. Advances in computer vision techniques have helped: e.g., tracking positions of up to eight fish in three dimensions [26]. Still, an underwater environment presents challenges such as changing light conditions, clutter and reflections. We would not expect fish to exhibit natural behavior were we to put markers on them. A typical schooling behavior as seen from an underwater camera consists of numerous occlusions, speed bursts and sharp turns.

Fast starts and quick turns are common swimming behavior, often as a precursor to an escape or attack [25]. In a fish school the collective response to an external cue can take place within a fraction of a second. In such a scenario the pose of each fish may not give a complete picture. The sensory volume of a fish that determines nearest neighbor interactions is dependent on the instantaneous pose and shape of the fish body [10]. In this paper we build a framework to track position, orientation, and shape of individual fish in a school.

S. Butail is a graduate student in the Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA sbutail@umd.edu

D. A. Paley is an Assistant Professor in the Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA dpaley@umd.edu

Fish schools have been tracked in both their natural environment [11] and in laboratories [26], [17]. Positions of up to fourteen fish have been tracked in two dimensions [17] and groups of four and eight fish have been tracked in three dimensions [26]. In [11] an acoustic sensor is used to track individual fish in a school from a moving platform. In [26], [17] least squares fitting is used to join track segments already matched on the image plane.

We are not aware of any prior instance of automatic shape tracking of multiple fish in a school. In the context of computer vision, our objective is to track the structure of multiple non-rigid objects through time. We have approximate knowledge of our target geometry when it is not in motion. Structure of a rigid object can be estimated using feature extraction [21] or optical flow [1]. For a non-rigid object, however, shape estimation is relatively difficult. Feature-tracking algorithms extended to non-rigid objects involve distorting a regular shape along the low frequency modes. A detailed model is then created through training data [15], or determined probabilistically [22]. We found feature tracking to be unsuccessful in tracking multiple similar looking objects. Another method not tested here called *shape from silhouette* builds a 3D approximation of an object based on overlapping volumes from within the silhouettes of multiple views [8].

As our objective is to solve a multi-target tracking problem, we are concerned with data association and occlusions. There is an extensive amount of literature on both these aspects that are addressed in different applications [3], [18], [13]. For example, in [13], [19] occlusions are explicitly handled by using prior knowledge about target geometry.

In this paper we utilize prior information about our target shape in a probabilistic tracking framework. We model fish motion as having constant speed along its body direction (heading) with orientation driven by random turning rates. The yaw motion is proportional to curvature about the yaw axis. We model fish shape as an ellipsoid that can be bent along its center line as a function of a single parameter called curvature. This model relates the shape and its 2D projection. State estimation is performed using a particle filter with joint probabilistic data association for motion correspondence. We evaluate our algorithm using simulated data and further characterize its performance using a laboratory experiment with six giant danios (Danio aequipinnatus) in a 300 gal tank.

The contributions of this paper are as follows:

- We mathematically relate the 3D shape of a fish body to its 2D image under perspective projection, using a model of a bent ellipsoid.
- We implement a probabilistic estimation algorithm to

automatically track 3D position and shape of individual fish in a school.

The paper is outlined as follows: Section II provides a background on nonlinear estimation and data association methods. Section III presents the bent ellipsoid model and describes the mathematical relation of the 3D model to its 2D projection according to the tracking variables (position, heading and curvature) in terms of a likelihood function. The problem is then cast in a particle filtering framework. Section IV describes experimental results and Section V provides a summary of the paper and highlights of our ongoing work.

II. Multi-Fish Tracking and Data Association

In this section we give an overview of nonlinear estimation techniques followed by a brief discussion on data association techniques needed to resolve measurement target association.

A. Nonlinear estimation

Both the motion and measurement models discussed in this paper are nonlinear functions of state. In this case a linear estimator such as Kalman filter would diverge easily. In vision-based systems, nonlinear estimators such as the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter are often used to track targets [2]. While the extended Kalman filter (EKF) has been successfully used in target-tracking systems, particle filters perform better than the EKF in scenarios that involve high nonlinearities and non-Gaussian distributions [2]. In this paper we use a standard sampling importance resampling (SIR) particle filter to track multiple fish in a school. Within a particle filter we use a measurement likelihood function to encode our confidence in the information we receive. A likelihood function is a conditional probability $P(Z|X)$ of a measurement Z given state X [3].

Although we use only heading in our state vector, in order to model shape we need the body orientation represented by a rotation matrix $[\hat{\boldsymbol{x}}\ \hat{\boldsymbol{y}}\ \hat{\boldsymbol{z}}]$. We make the assumption that a fish does not roll about its center line. Using the mean $\hat{\boldsymbol{x}}$ we compute the cross product of the vertical axis in the inertial frame with $\hat{\boldsymbol{x}}$ to get the $\hat{\boldsymbol{y}}$ direction in the body frame. The rotation matrix is completed by setting $\hat{\boldsymbol{z}} = \hat{\boldsymbol{x}} \times \hat{\boldsymbol{y}}$.

B. Data association

Data association, common to all multi-target tracking systems, is the task of matching measurements to targets. In our context, for example, this implies maintaining the same measurement-target associations through consecutive frames. In an environment with clutter it is typical to get more measurements than the number of targets. A simple strategy is to assign a measurement to the nearest measurement estimate; this strategy is called the nearest neighbor filter (NNF) [3]. An optimal Bayesian filter in this scenario would take into account all of the past history of measurement-target associations and branch out a path from each such pairing to assign a probability to the latest set of measurements. The number of paths in such a scenario increases exponentially with the number of measurements [16]. Fortunately, there exist several heuristics to trim these paths [16] (For a review of data association techniques please refer to [9].)

In this paper we use a method called joint probabilistic data association (JPDA) [3]. The JPDA algorithm assigns probability values to measurement-target associations based on current measurements and state estimates. These values are then used to assign a weight to each association. The final update to a target estimate during a time-step is a weighted sum of all possible measurement updates. At any time-step k, the set of all valid target-measurement associations, $\boldsymbol{\theta}$, is generated based on a gating volume. A feasible event $\theta \in \boldsymbol{\theta}$ is created such that (i) each measurement has only one source and (ii) each target (excluding clutter) produces exactly one measurement or no measurements at all. The joint measurement-target association probability β_{ij} between measurement j and target i is [3]

$$\beta_{ij} = \sum_{\theta \in \boldsymbol{\theta}} P(\theta|\boldsymbol{Z}^k), \tag{1}$$

where $\boldsymbol{Z}^k$ is the set of all measurements up to time k. $P(\theta|\boldsymbol{Z}^k)$ as per Bayes theorem is the product of the measurement-association likelihood function $P(Z[k]|\theta, \boldsymbol{Z}^{k-1})$ and the prior $P(\theta)$. $Z[k]$ represents all measurements at k. All unassigned measurements are assumed to be uniformly distributed across the entire observation region. The probability of each association in a feasible event is computed using the measurement likelihood function. In a particle filtering framework, the measurement likelihood function is averaged over all samples [18].

III. Reconstructing fish kinematics from video

In this section we present a model of fish shape, followed by a model of fish motion. Measurement models for extracting position and shape (which includes orientation) from video are described and cast in a particle-filtering algorithm.

A. Fish shape as a bent ellipsoid

A simple yet robust method to track an articulated object is to model it as a series of connected quadrics [19], [8]. An even closer approximation of different shapes within an object can be achieved by modifying the general quadric equation to form a superquadric [20], or an extended quadric [28]. All of the above strategies, however, add many more variables to the state space. We use a similar approach of modeling the fish shape as a quadric—an ellipsoid—but in a way that allows us to use a single extra parameter, K, which we call the curvature, to represent bending of the fish body during swimming.

We begin with the equation of an ellipse centered at the origin and oriented along the horizontal axis. Given a point on the ellipse $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$, its equation can be written as $u_1^2/a^2 + u_2^2/b^2 = 1$, where a and b are the lengths of the semi-major and-minor axes respectively. To bend this ellipse we re-define the long axis as a curve, $u_2 = f(u_1)$. The new equation becomes $u_1^2/a^2 + (u_2 - f(u_1))^2/b^2 = 1$. For example, to bend the ellipse in a crescent we set $f(u_1) =$

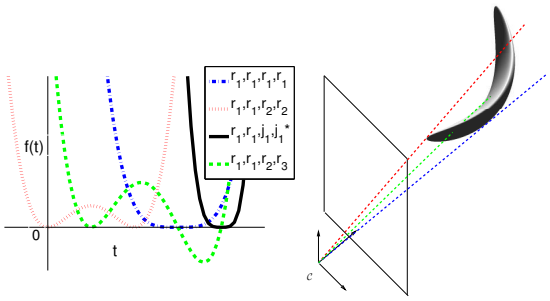

Fig. 1. Possible solutions to a quartic equation with at least one repeated root. Also shown is a bent ellipsoid with its projection on camera plane showing corresponding scenarios in the same color.

Ku_1^2, where K is the curvature[1]. When $K = 0$, the ellipse is straight; when $K \neq 0$, it bends in a parabola.

A similar strategy is employed to bend an ellipsoid in three dimensions [7]. Consider an ellipsoid with dimensions a, b, c such that $a > c > b$. The equation for an ellipsoid that is bent on its shortest dimension is

$$\frac{r_{\mathcal{B},1}^2}{a^2} + \frac{(r_{\mathcal{B},2} - Kr_{\mathcal{B},1}^2)^2}{b^2} + \frac{r_{\mathcal{B},3}^2}{c^2} = 1, \qquad (2)$$

where $\boldsymbol{r}_{\mathcal{B}} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^T$ is a point in a body frame $\mathcal{B}$. We re-write equation (2) in matrix form as

$$\tilde{\boldsymbol{r}}_{\mathcal{B}}^T Q_{\mathcal{B}} \tilde{\boldsymbol{r}}_{\mathcal{B}} + \frac{K^2 r_{\mathcal{B},1}^4}{b^2} - \frac{2K r_{\mathcal{B},1}^2 r_{\mathcal{B},2}}{b^2} = 0, \qquad (3)$$

where $Q_{\mathcal{B}} = \text{diag}\{1/\text{a}^2, 1/\text{b}^2, 1/\text{c}^2, -1\}$, and $\tilde{\boldsymbol{r}}_{\mathcal{B}} \triangleq \begin{bmatrix} \boldsymbol{r}_{\mathcal{B}}^T & 1 \end{bmatrix}^T$ is the homogeneous representation of $\boldsymbol{r}_{\mathcal{B}}$. To express (2) in inertial coordinates, we use the 4×4 transformation matrix T from the inertial frame $\mathcal{I}$ to frame $\mathcal{B}$ such that $\tilde{\boldsymbol{r}_{\mathcal{B}}} = T\tilde{\boldsymbol{r}}$. We can write (3) as

$$\tilde{\boldsymbol{r}}^T Q_{\mathcal{C}} \tilde{\boldsymbol{r}} + \frac{K^2 (T_1 \tilde{\boldsymbol{r}})^4}{b^2} - \frac{2K (T_1 \tilde{\boldsymbol{r}})^2 (T_2 \tilde{\boldsymbol{r}})}{b^2} = 0, \qquad (4)$$

where $Q_{\mathcal{C}} = T^T Q_{\mathcal{B}} T$ and T_i denotes the ith row of T.

Having written the equation of a bent ellipsoid in inertial coordinates, we now proceed to project it onto an image plane under perspective projection. Similar analysis has been employed for quadrics [6], [19]. Without loss of generality we assume the camera frame $\mathcal{C}$ is coincident with the inertial frame $\mathcal{I}$. We represent a line from the origin by $L(t) = \boldsymbol{l}t$ with $\boldsymbol{l} = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T$ and t a free parameter. Replacing $\tilde{\boldsymbol{r}}$ in (4) with $\begin{bmatrix} L(t)^T & 1 \end{bmatrix}^T$ yields a fourth-order polynomial in t. For a point to lie on the bounding contour of the 3D shape, the polynomial should have a single root. The projection of each such point will form the extremal contour in the image plane. We seek to find an expression for such a contour in order to extract information about the 3D shape.

If the discriminant of a polynomial with real coefficients is zero, it will have at least two roots that are same [4].

[1]Note that this bending causes a proportional increase in the total length of the ellipse that can be adjusted by setting the length of semi-major axis as a function of K. In the example above, where $f(u_1) = Ku_1^2$, we can replace a in the second equation by a_b such that $a_b^2 + K^2 a_b^4 = a^2$, which implies $a_b = 1/K\sqrt{-1 + \sqrt{1 + 4K^2a^2}/2}$, for $K \neq 0$

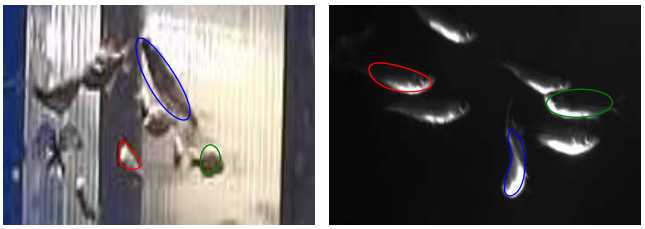

Fig. 2. Modeling fish as bendable ellipsoids. Shown are image frames from two cameras with image planes normal to each other. The extremal contours for each fish use the ellipsoid model.

(This condition is equivalent to having a point on the surface tangent to its position vector.) Demanding that the discriminant be zero raises four possible scenarios (Fig. 1):

i. All four roots are the same
ii. Two roots are the same and two are distinct
iii. Two roots are the same same and two are imaginary
iv. Two pairs of repeated roots

Requirement (i) above for a quartic (fourth-order polynomial) is too strong as we can have cases where two roots are same and the other two do not exist. The only case that is undesirable is case (ii), i.e., when we have 3 distinct real roots. Looking at Fig. 1 we see that the projection of points corresponding to such cases will lie inside the silhouette.

The discriminant for a quartic of the form $p_4t^4 + 4p_3t^3 + 6p_2t^2 + 4p_1t + p_0 = 0$ is [4]

$$I^3 - 27J^2 = 0, \qquad (5)$$

where $I = p_4p_0 - 4p_3p_1 + 3p_2^2$ and $J = p_4p_2p_0 + 2p_3p_2p_1 - p_4p_1^2 - p_3^2p_0 - p_2^2$. In our case the coefficients $p_i, i = 0, ..., 4$ contain values from the $Q_{\mathcal{C}}$ and T matrices along with curvature K. For example $p_0 = Q_{\mathcal{C},4,4} + K^2/b^2 T_{1,4}^4 - 2K/b^2 T_{2,4} T_{1,4}^2$ where the subscript $_{i,j}$ on matrices $Q_{\mathcal{C}}$ and T denote element (i, j). (See the appendix for a list of all coefficients.) We normalize the vector $\boldsymbol{l}$ with respect to l_3 such that $L(t) = \begin{bmatrix} u & v & 1 \end{bmatrix}^T t$ with $\begin{bmatrix} u & v \end{bmatrix}^T$ denoting a point in the image plane for a camera with unit focal length. The points on the image plane that satisfy (5) lie inside or on the silhouette of the projection.

B. Particle-filtering framework

We begin with a model that approximates fish motion. The following assumptions are made:

- A fish shape is approximated by a bent ellipsoid.
- The ratios between semi-major, medium and minor axes, $a/b, a/c$, are the same for all fish.
- A fish does not roll about its centerline; fish maneuver by yaw and pitch motions only.

The target state vector $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{r}^T, \boldsymbol{x}^T, K, s \end{bmatrix}^T$ at any time comprises position $\boldsymbol{r} \in \mathbb{R}^3$, heading $\boldsymbol{x} \in \mathbb{R}^3$, curvature $K \in \mathbb{R}$, and speed $s \in \mathbb{R}$. The full orientation of a fish is found by completing the orthonormal frame as discussed in Section II-A. K is assumed to decay exponentially while being perturbed by a Gaussian disturbance $d_K = \mathbb{N}(0, \sigma_K^2)$. The motion model is

$$\begin{aligned} \dot{\boldsymbol{r}} &= s\boldsymbol{x}; \;\; \dot{\boldsymbol{x}} = \boldsymbol{u} \times \boldsymbol{x} \\ \dot{K} &= -\lambda K; \;\; \dot{s} = 0, \end{aligned} \qquad (6)$$

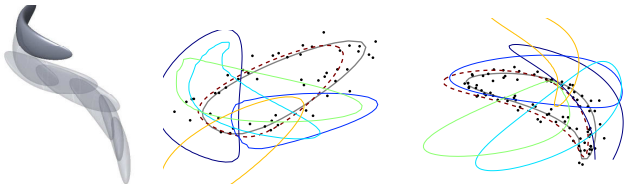

Fig. 3. Motion and measurement models. The left most figure shows how the ellipsoid turns in the direction of curvature. The grey contour in the two figures on the right is generated by projecting a bent ellipsoid onto two camera frames. The black dots are noisy measurements. Random contours are generated and weighted using the likelihood function for the shape. The contours that were weighted highest are shown in red dashed lines.

where $\boldsymbol{u} = \begin{bmatrix} w, h, \psi q \end{bmatrix}^T$ is the control input denoting the turning rate about each axes in the body frame. $\psi = \text{atan}(\text{Ka})$ is the angle of inclination from the center of ellipsoid to its tip.

For the stochastic version of (6) we model the unknown turning rates as Gaussian random variables such that $w_w = \mathbb{N}(0, \sigma_w^2)$, $w_h = \mathbb{N}(0, \sigma_h^2)$ and $w_q = \mathbb{N}(0, \sigma_q^2)$. We set $\sigma_w << \sigma_h$ and $\sigma_q = 0$ (the fish turns more than it pitches)

$$\begin{aligned} d\boldsymbol{r} &= s\boldsymbol{x}dt;\ d\boldsymbol{x} = dw_u \times \boldsymbol{x} \\ dK &= -\lambda K dt + dw_K;\ ds = dw_s, \end{aligned} \tag{7}$$

where $dw_u = \begin{bmatrix} dw_w, dw_h, \psi dw_q \end{bmatrix}^T$

The following likelihood functions represent the measurement model for position and shape. Our observations consist of the centroid position $\boldsymbol{u} = \begin{bmatrix} u & v \end{bmatrix}^T$ and silhouette $s = \{u_i, v_i; i = 1, ..., n_s\}$ of each blob, where n_s is the number of pixels in the silhouette. The centroid measurement on a camera image plane with focal length f (in pixels) as a function of target center position $\boldsymbol{r}$ is $\boldsymbol{u} = f \begin{bmatrix} r_1/r_3, r_2/r_3 \end{bmatrix}^T + I_2 \eta_p$, where η_p is a two dimensional Gaussian noise vector for u and v and I_2 is the 2×2 identity matrix. The likelihood function for location of a single measurement, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{r})$, and estimate, $\hat{\boldsymbol{u}}$, pair is

$$P_{pos}(\boldsymbol{u}|\boldsymbol{r}) = \mathbb{N}(\boldsymbol{u}; \hat{\boldsymbol{u}}, \Sigma). \tag{8}$$

$\mathbb{N}(\boldsymbol{u}; \hat{\boldsymbol{u}}, \Sigma)$ denotes a normal distribution function with mean $\hat{\boldsymbol{u}}$ and noise covariance matrix $\Sigma \in \mathbb{R}^{2\times 2} = \text{diag}\{\sigma_{\text{u}}^2, \sigma_{\text{v}}^2\}$.

We use silhouette points to estimate shape. Consider a point $\boldsymbol{u}_i = \begin{bmatrix} u_i, v_i \end{bmatrix}^T$ on the silhouette of an object in the image plane. We wish to find the probability $p(\boldsymbol{u}_i|\boldsymbol{r}, \boldsymbol{x}, K)$. Given that $\boldsymbol{r}, \boldsymbol{x}$ and K project an extremal contour $\{(u_j, v_j)|(u_j, v_j) \in C, j = 1, ..., n_c\}$ where n_c is the number of points in the contour, we can write the probability as $p(\boldsymbol{u}_i|C)$. The probability of a point on the silhouette $\boldsymbol{u}_i$ given contour C (assuming that measurements are normally distributed about the true state) can be written as the sum

$$p(\boldsymbol{u}_i|C) = \sum_{\boldsymbol{u}_j \in C} \mathbb{N}(\boldsymbol{u}_i; \boldsymbol{u}_j, \Sigma),$$

where $\boldsymbol{u}_j = \begin{bmatrix} u_j, v_j \end{bmatrix}^T$. The likelihood function for shape P_{sha}^l for camera l is the product of probabilities over all the silhouette points

$$P_{sha}^l = \prod_{\boldsymbol{u}_i \in s} p(\boldsymbol{u}_i|C). \tag{9}$$

TABLE I

PARAMETER VALUES USED FOR TRACKING

Parameter	Value	Parameter	Value
σ_u	3.0 pixels	σ_q	2.0 rad/s
σ_v	3.0 pixels	σ_h	0.20 rad/s
σ_K	0.01	σ_w	0.001 rad/s
σ_s	5 mm/s	λ	10

The contours that align with the silhouette will always have a point on the silhouette and are consequently weighted the highest. In a multi-camera setup, where multiple perspectives of a scene are captured at the same instant, the combined likelihood function is a product over the likelihood for all cameras, $P_{sha} = \prod_l P_{sha}^l$. Fig. 3 shows an example of weighting contours using the shape likelihood function. Note that position and shape likelihood functions are not independent since silhouette points depend on the position of a target in space. We use a position-only likelihood function in scenarios where it is difficult to use shape information such as during occlusions. Due to symmetry in the shape representation there is a forward-backward ambiguity in the likelihood function.

For the first time step, the particle filter is initialized manually and the length of each fish is assigned. Measurement-target association performed at each step is based on the position-only likelihood function. If the projected extremal contours of two targets overlap within a camera frame, an occlusion is assumed and samples are weighted on the basis of position only. When a target is again found in at least two cameras without occlusions, the heading $\boldsymbol{x}$ and curvature K are interpolated from the last time-step without occlusion. The particle filtering algorithm for tracking multiple fish is described below.

Initialize a set of samples N for each target j as a normal distribution about the initial value. For each time-step k:

1. For each target i and a set of validated measurements based on a suitable gating volume [23], compute β_{ij} using (1).
2. For each target i
   a) Compute particle weights and normalize for all measurements $Z[k]$ at time step k, according to

$$\tilde{w_i} = \sum_{j \in Z[k]} \beta_{ij} P_m;\ w_i = \tilde{w_i} [\sum_{p=1}^{N_t} \tilde{w}_{i,p}]^{-1},$$

   where $P_m = P_{pos}$ if the target is occluded and $P_m = P_{sha}$ otherwise.
   b) Resample using normalized weights w_i.
   c) Estimate the output by computing the mean of each parameter over all sample values.
   d) Propagate each particle using motion model (7).

IV. EXPERIMENTAL METHODS AND RESULTS

Obtaining ground truth is not possible for all the parameters that we wish to estimate. In order to characterize the performance of our tracking algorithm, particularly the likelihood function and motion model, we run it on a single

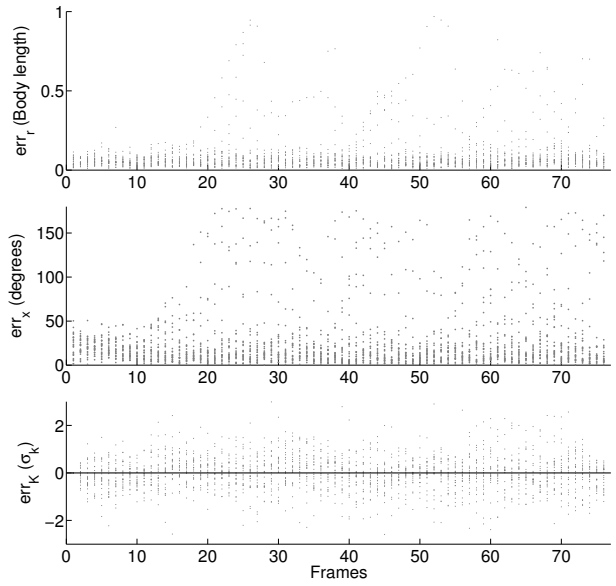

Fig. 4. Results from 25 Monte Carlo runs of a single fish simulation. Errors plotted as dots are shown for position, heading and curvature. For heading the angle between the estimated vector and true value is shown.

simulated fish. The simulated motion is generated by (6) and the shape is constructed using the method described in Section III-A. Fig. 3 shows a few steps in such a simulation. At each step, the extremal contours were projected onto two different camera planes and peppered with pixel noise having $\Sigma = \mathrm{diag}\{2, 2\}$. These contours were used as silhouette measurements for the tracker. Fig. 4 shows the results from 25 Monte Carlo runs of the algorithm using five hundred samples with a time-step of 1/15 seconds for 5 seconds each.

For our laboratory experiment we tracked six giant danios. These schooling fish are 4–5 cm long. The fish were moved from a 20 gal tank into a 300 gal (1.2 m wide and 0.91 m deep) tank and kept there for a few days to acclimatize. Schooling behavior was observed soon after the fish were moved into the large tank. Four CWC-620WP Speco cameras were mounted on the inside wall of the tank such that the maximum volume was covered. (Three cameras were mounted at the bottom, middle and top levels spaced at 120 degree intervals. The bottom and top cameras were at an inclination pointing into the tank. A fourth camera was mounted on top opposite to the other top camera.) Intrinsic calibration was performed using the MATLAB$^{\text{TM}}$ calibration toolbox [5]. Extrinsic calibration was performed by moving a checkerboard between the cameras and propagating the extrinsic parameters, $\begin{bmatrix} R & \boldsymbol{t} \end{bmatrix}$, between overlapping camera views until all camera positions and orientations were known with respect to a common inertial frame.

Noise in images was a major concern despite the laboratory conditions. Changing lighting effects in water gave rise to glare and clutter in images. The background was modelled as a running Gaussian average at each pixel [27]. The silhouettes were obtained using a Canny edge detector on the binary image after filling up any dark pixels.

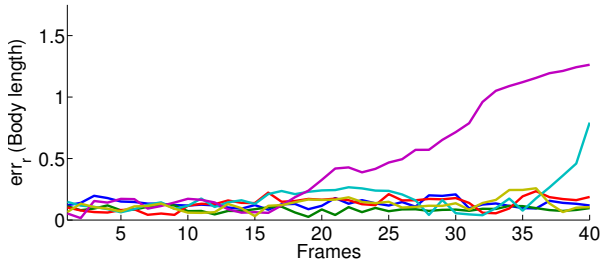

Fig. 5. Tracking results with real data for position only. The plot shows the absolute position error for each fish.

Verification of results: The results for position tracking were verified using a GUI created in MATLAB in which a user can click on fish and perform the association manually. User marked centroid positions were then used to output a least-squares estimate of position and projected back on to the images for verification. Shape was verified by projecting the occluding contour on the image plane and comparing with the silhouette measurement.

Results & discussion: We are able to automatically track position of all six fish despite occlusions for over forty frames. Fig. 5 shows the error norm of difference between the position vectors and user-verified (ground-truth) data for all six fish. The tracker's performance often degrades during extended occlusions or when the fish are not visible in at least two cameras, as can be seen in Fig. 5 for the purple fish after frame 20 and, for the green fish, after frame 35.

Shape estimation (Fig. 6) is highly dependent on measurements, which were often noisy. Not all fish shapes were tracked well. Errors can be attributed to occlusions, quick turns and inexact shape representation. We can address occlusions by using cameras with large field of view, which ensures that the fish stay within two cameras. Quick turns can be addressed using high-speed camera to track shape during fast maneuvers. Lastly, a better approximation of shape can be made by using bends that are not necessarily about the center of a fish body by introducing an additional parameter in the bent-ellipsoid equation, (2).

V. Conclusion

We describe a probabilistic framework to estimate position and shape of multiple fish in a school using underwater cameras. This framework will be used to quantitatively analyze fish-schooling kinematics during subsequent behavioral experiments. We mathematically model fish shape using a bent ellipsoid with time-varying curvature. We develop an expression for the extremal contour of such a shape under perspective projection. We use a motion model with constant speed and curvature-dependent turning rate in the body frame. The tracking algorithm uses a particle filter with joint probabilistic data association to estimate shape.

As part of ongoing work we are exploring methods to improve shape estimation and track more targets. A probabilistic framework allows us to include a high resolution camera to augment the shape estimation. We also intend to detect occlusions and compensate for them. In order to track

Fig. 6. Tracking results for shape. The bent ellipsoid shape is projected back onto two cameras at five different time-steps. Black arrows represent heading. A target track is lost during occlusion in the 30th frame

larger datasets these algorithms may be implemented on a parallel processing architecture such as CUDA™.

REFERENCES

[1] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 384–401, 1985.

[2] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, Feb 2002.

[3] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[4] A. B. Basset. *An Elementary Treatise on Cubic and Quartic Curves*. Cambridge University Press, 1901.

[5] J. Y. Bouguet. Camera calibration toolbox for Matlab http://www.vision.caltech.edu/bouguetj/calib_doc/, 2008.

[6] S. Butail and D. Paley. Vision-based estimation of three-dimensional position and pose of multiple underwater vehicles. *IEEE/RSJ International Conference, IROS*, pages 2477–2482, 2009.

[7] Chunguang "Ken" Cao and Timothy S. Newman. A new framework for recovery of shape of the right ventricle from GBP spect images. *Proc. of the 43rd ACM Southeast conf.*, 2005.

[8] Kong Man Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette across time part i: Theory and algorithms. *International J. of Computer Vision*, 62(3):221 – 247, May 2005.

[9] Ingemar J. Cox. A review of statistical data association for motion correspondence. *Int. J. Comput. Vision*, 10(1):53–66, 1993.

[10] K. Faucher, E. Parmentier, C. Becco, N. Vandewalle, and P. Vandewalle. Fish lateral system is required for accurate control of shoaling behaviour. *Animal Behaviour*, doi:10.1016/j.anbehav.2009.12.020, 2010.

[11] N.O. Handegard, R. Patel, and V. Hjellvik. Tracking individual fish from a moving platform using a split-beam transducer. *The Journal of the Acoustical Society of America*, 118:2210, 2005.

[12] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Conf. on Decision and Control*, pages 988–1001, Dec. 2002.

[13] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. *California Partners for Advanced Transit and Highways (PATH)*, 1994.

[14] Julia K. Parrish and William M. Hammer. *Animal groups in three dimensions*. Cambridge University Press, 1997.

[15] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 13(7):730–742, 1991.

[16] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 1979.

[17] Chad Schell, Stephen P. Linder, and James R. Zeidler. Tracking highly maneuverable targets with unknown behavior. *Proc. of the IEEE 92(3): 558*, 2004.

[18] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving objects with a mobile robot. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:371, 2001.

[19] B. Stenger, P.R.S. Mendonca, and R. Cipolla. Model-based 3d tracking of an articulated hand. *In Proc. IEEE Computer Vision and Pattern Recognition CVPR*, 2:II–310–II–315 vol.2, 2001.

[20] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: deformable superquadrics. *IEEE trans. on Pattern Analysis and Machine Intelligence, PAMI*, 1991.

[21] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[22] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 30, 2008.

[23] J. Vermaak, S.J. Godsill, and P. Perez. Monte Carlo filtering for multi-target tracking and data association. *IEEE Trans. on Aerospace and Electronic systems*, 41(1):309–332, 2005.

[24] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel Type of Phase Transition in a System of Self-Driven Particles. *Physical Review Letters*, 75:1226–1229, August 1995.

[25] J.J. Videler. *Fish swimming*. Springer, 1993.

[26] Steven V. Viscido, Julia K. Parrish, and Daniel Grünbaum. Individual behavior and emergent properties of fish schools: a comparison of observation and theory. *Marine Ecology Progress Series*, 10.3354/meps273239, 2004.

[27] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI*, 19(7):780–785, 1997.

[28] Lin Zhou and C. Kambhamettu. Extending superquadrics with exponent functions: modeling and reconstruction. *Computer Vision and Pattern Recognition, CVPR*, 1999.

APPENDIX

EQUATION FOR A BENT ELLIPSOID UNDER PERSPECTIVE PROJECTION

A bent ellipsoid as seen from the camera frame is a trivariate equation in r_1, r_2, r_3 where $\boldsymbol{r} = \begin{bmatrix} r_1, r_2, r_3 \end{bmatrix}^T$ is a point in the camera frame. Setting $\boldsymbol{r} = L(t)$ as a function of scaling parameter in t we get a fourth-order polynomial equation in t of the form $p_4 t^4 + 4p_3 t^3 + 6p_2 t^2 + 4p_1 t + p_0 = 0$. The coefficients $p_i, i = 0, ..., 4$, of the above polynomial are (subscript $\mathcal{C}$ is dropped from $Q_{\mathcal{C}}$ for clarity):

$$
\begin{aligned}
p_0 =& Q_{4,4} + \frac{1}{b^2}(K^2 T_{1,4}^4 - 2KT_{2,4}T_{1,4}^2) \\
4p_1 =& 2\boldsymbol{l}^T Q_{1:3,4} + \frac{1}{b^2}(4K^2(T_{1,1:3}\boldsymbol{l})T_{1,4}^3 - 2K(T_{2,1:3}\boldsymbol{l})T_{1,4}^2 \\
& - 4KT_{2,4}(T_{1,1:3}\boldsymbol{l})T_{1,4}) \\
6p_2 =& \boldsymbol{l}^T Q_{1,1:3}\boldsymbol{l} + \frac{1}{b^2}(6K^2(T_{1,1:3}\boldsymbol{l})^2 T_{1,4}^2 - \\
& 4K(T_{2,1:3}\boldsymbol{l})(T_{1,1:3}\boldsymbol{l})T_{1,4} - 2KT_{2,4}(T_{1,1:3}\boldsymbol{l})^2) \\
4p_3 =& \frac{1}{b^2}(6K^2(T_{1,1:3}\boldsymbol{l})^2 T_{1,4} - 2K(T_{2,1:3}\boldsymbol{l})(T_{1,1:3}\boldsymbol{l})^2) \\
p_4 =& \frac{K^2}{b^2}(T_{1,1:3}\boldsymbol{l})^4
\end{aligned}
$$