2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Biologically Motivated Shape Optimization of Foraging Fronts

Musad Haque, Amir Rahmani, Magnus Egerstedt, and Anthony Yezzi

Abstract— **Social animals often form a predator front to charge through an aggregation of prey. It is observed that the nature of the feeding strategy dictates the geometric shape of these charging fronts. Inspired by this observation, we model foraging multi-robot fronts as a curve moving through a prey density. We optimize the shape of the curve using variational arguments and simulate the results to illustrate the operation of the proposed curve optimization algorithm.**

I. INTRODUCTION

Charging through an aggregation of prey is a common foraging strategy among social animals. Using this technique, predators form a specific formation to create a front that moves together, in unison, towards the collection of prey. The U-shaped African lion front known as the "catcher's mitt" and the "wall method" exhibited by Bottlenose dolphins are examples of these geometric strategies [1], [2]. Our goal is to formalize this, i.e., to draw inspiration from nature and optimize the shape of the predator fronts for foraging multi-agent systems.

Foraging has received significant attention in the multi-robot community (for a representative sample, see [3], [4], [5], [6], [7]); yet previous work primarily focuses on the search and retrieval aspects of foraging stationary objects or cooperative agents. In [3], the effects of physical interference is presented for different foraging strategies and the effects of behavioral diversity of the foraging group is studied in [4], where the behaviors range from "homogeneous" to "specialized." Bio-inspired foraging strategies for static environments, based on ants and bees, are presented in [5] and [6], respectively. Here, we instead focus on the geometric shape of the foraging front in a dynamic scenario involving prey that are not explicitly cooperative.

We develop a curve flow algorithm that modifies the shape of the predator front to maximize the total energy intake, i.e. the total amount of prey swept by the front. A potential application for this work is the clean up of oil spills. Until now, unmanned vehicles have been deployed in the spill site to collect data on ocean properties and survey the extent of damage. However, we propose to utilize a multi-robot system for an efficient clean up task of spilled oil. Fig. 1 shows how a group of robots coordinate to drive a flexible suction boom towards a spill site. Using the proposed curve flow algorithm, we can optimize the shape of this boom for an efficient cleanup of the oil spill as a function of the oil dynamics.

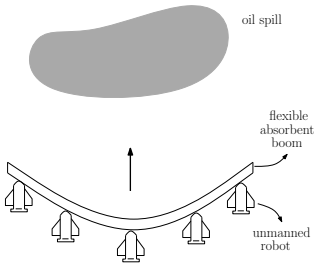


Fig. 1. A group of unmanned vehicles are driving a flexible, absorbent boom towards an oil spill. Optimizing the shape of the boom, to remove the largest amount of oil, is a possible application of the proposed curve flow algorithm.

The problem of finding the optimal charging front was initially addressed in [8], but the effort was restricted to the simulation of quadratic curves under various predator-prey interactions. The curve flow algorithm developed in this paper is based on curve evolution techniques, which are widely used in the field of image processing, e.g. see [9], [10], [11]. Active contours for image segmentation evolve an initial curve under a cost function to detect objects. One common approach is to model the initial curve as a level set and define the optimality condition based on the speed of the curve, e.g. [9], [10]. Here, we follow a similar strategy to that of [11], where an arc length parameterized curve is evolved according to a gradient ascent based deformation algorithm.

The remainder of the paper is organized as follows: Section II introduces the curve-based model of the charging front and presents the curve flow algorithm. Examples that illustrate the operation of the algorithm are provided in Section III. Conclusions are presented in Section IV.

II. CURVE EVOLUTION MODEL

In this section, we present our curve-based model of the predator front, define the energy over a curve, and produce an algorithm to increase the energy intake by deforming the shape of the curve. We model the prey aggregation as a $2D$ time-varying density function denoted by $u(x,y,t)$, where $u : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ describes prey density at position (x,y) at time t. This density function changes according to partial differential equations (PDEs) representing the movement of the prey. In biology, this approach is known as the "population framework" [13] and unlike the agent-based models of the prey (e.g. [14]), this approach does not require tracking the movement of individual prey-like agents.

This work is partially supported by the Office of Naval Research through the MURI, Heterogeneous Unmanned Networked Teams (HUNT).

M. Haque, A. Rahmani, M. Egerstedt, and A. Yezzi are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA musad.haque@gatech.edu, {amir.rahmani, magnus.egerstedt, ayezzi}@ece.gatech.edu

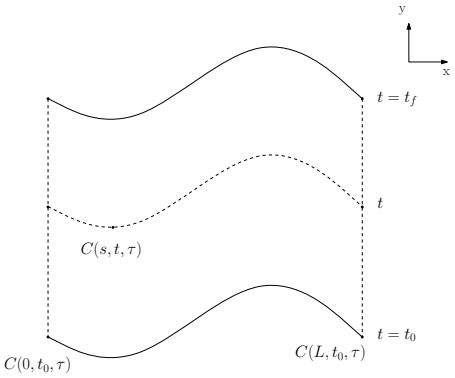


Fig. 2. A curve front sweeps in the positive y direction with unit speed while maintaining its shape.

.

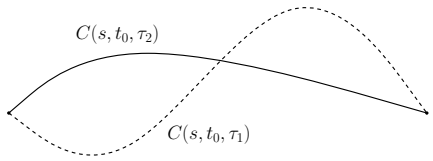


Fig. 3. Curves evolving under the proposed algorithm share the same endpoints at $t = t_0$.

We use PDEs to denote prey movement for the purpose of simulations only. In general, the curve flow algorithm developed in this paper is independent of the choice of prey movement representation, and is only a function of prey density at each time.

A. Predator Fronts as Curves

We model the predator front as a curve of fixed shape sweeping through the aggregation of the prey, as shown in Fig. 2. Without loss of generality, we assume that the front sweeps the area with unit speed and in the positive y direction while maintaining its shape. We define the energy intake of the front as the sum of the prey it sweeps over time and our goal is to find the best front shape that maximizes this energy intake.

Let the predator front be given by the curve, $C(s,t,\tau)$, where s is the arclength parameter, t denotes time, and $\tau \in \mathbb{R}$ parameterizes a family of time-varying curves. If we denote the total length of the curve by $L(\tau)$, then $s \in [0, L(\tau)]$. We will find the optimum shape of the front by evolving the curve C, using gradient ascent and moving in a direction that increases the energy intake. The main idea of the algorithm is to start with a curve shape $C(s,t,0)$ and let it sweep the prey density from $t = 0$ to $t = t_f$ and compute the energy intake. Then, we deform the shape of the curve (with respect to τ) such that the energy intake is increased during the next sweep. We repeat these steps until the best curve shape is found.

It should be noted that with this curve-based model of the predator front, we are assuming a continuum of predators instead of the common agent-based model of foragers seen in [3]. Moreover, we assume that all curve shapes have identical endpoints, i.e. the endpoints of the curve stay the same regardless of the deformation in the shape of the curve (Fig. 3).

We represent the energy-intake during a sweep of the curve, i.e. the amount of prey being "eaten" during a charge, as

$$E(\tau) = \int_0^{t_f} \int_0^{L(\tau)} u(C(s,t,\tau),t)\ ds\ dt, \tag{1}$$

where $u(C(s,t,\tau))$ represents the prey density at position $C(s,t,\tau)$ at time t. Our goal is to find the curve shape that maximizes this energy and we choose to use gradient ascent to update the curve shape, i.e. in such a way that the gradient of $E(\tau)$ with respect to τ is increased. The derivative of $E(\tau)$ with respect to τ is given by

$$\frac{dE(\tau)}{d\tau} = \int_0^{t_f} \frac{d}{d\tau} \int_0^{L(\tau)} u(C(s,t,\tau),t)\ ds\ dt, \tag{2}$$

and to compute this derivative, we introduce a parameter $p \in [0,1]$ to replace the s parameterization of the curve with a parameterization that is not τ dependent. (For this substitution, we follow the method outlined in [11].) From the definition of arc length[1],

$$L(\tau) \quad = \quad \int_0^{L(\tau)} ds = \int_0^1 \|C_p(p,\tau)\|\ dp, \tag{3}$$

from which it follows that

$$ds = \|C_p(p,\tau)\|\ dp, \tag{4}$$

and

$$\frac{dE(\tau)}{d\tau} = \int_0^{t_f} \int_0^1 \left(u(C(s,t,\tau),t)\|C_p(p,\tau)\|\right)_\tau\ dp\ dt. \tag{5}$$

Using the chain rule, we get

$$\frac{dE(\tau)}{d\tau} = \int_0^{t_f} \int_0^1 (\ u_\tau \|C_p\| + u\|C_p\|_\tau\ )\ dp\ dt. \tag{6}$$

Notice that,

$$u_\tau = \nabla u \cdot C_\tau, \tag{7}$$

where $\nabla u(C(p,t,\tau),t)$ is the $2D$ spatial gradient of u. Also, we have that

$$\|C_p\|_\tau^2 = 2\|C_p\|\|C_p\|_\tau, \tag{8}$$

and

$$(C_p^T C_p)_\tau = 2C_{p\tau}^T C_p, \tag{9}$$

where the superscript T denotes transpose. As a result,

$$\|C_p\|_\tau = C_{p\tau}^T \frac{C_p}{\|C_p\|} = C_{\tau p} \cdot \overrightarrow{T}, \tag{10}$$

where we note that the partial derivatives of C can be exchanged and we have introduced the unit tangent of the curve,

$$\overrightarrow{T}(p,\tau) = \frac{C_p}{\|C_p\|}. \tag{11}$$

[1]For conciseness, we let f_x represent the partial derivative $\frac{\partial f}{\partial x}$ of a function $f(x,y)$ and denote the second-order partial derivative $\frac{\partial f}{\partial x \partial y}$ by f_{xy}.

Next, we substitute (7) and (10) into (6) and revert back to the s parameterization to obtain

$$\frac{dE(\tau)}{d\tau} = \int_0^{t_f} \int_0^{L(\tau)} \left(\nabla u \cdot C_\tau + u C_{\tau s} \cdot \overrightarrow{T}(s,\tau) \right) \, ds \, dt. \quad (12)$$

Using integration by parts, $\int_0^{L(\tau)} u C_{\tau s} \cdot \overrightarrow{T} \, ds =$

$$\left[u C_\tau \cdot \overrightarrow{T} \right]_{s=0}^{s=L(\tau)} - \int_0^{L(\tau)} C_\tau \cdot \left(u \overrightarrow{T} \right)_s \, ds \quad (13)$$

Since we assume that the endpoints of the curve are fixed for all values of τ, $C_\tau(0,t,\tau) = 0$ and $C_\tau(L(\tau),t,\tau) = 0, \forall \tau, t \in [0,t_f]$. Equation (12) can now be written as

$$\frac{dE(\tau)}{d\tau} = \int_0^{t_f} \int_0^{L(\tau)} \left(\nabla u \cdot C_\tau \; - \; C_\tau \cdot \left(u \overrightarrow{T} \right)_s \right) \, ds \, dt. \quad (14)$$

Using the chain rule,

$$\begin{aligned} \left(u \overrightarrow{T} \right)_s &= u_s \overrightarrow{T} + u \overrightarrow{T}_s \\ &= \nabla u \cdot C_s + u \kappa \overrightarrow{N} \\ &= \nabla u \cdot \overrightarrow{T} + u \kappa \overrightarrow{N}, \end{aligned}$$

where we introduce two more intrinsic geometric properties of the curve: the unit normal $\overrightarrow{N}(s,\tau)$ and curvature $\kappa(s,\tau)$, through the relation $\overrightarrow{T}_s = \kappa \overrightarrow{N}$. Thus, $dE(\tau)/d\tau =$

$$\begin{aligned} & \int_0^{t_f} \int_0^{L(\tau)} C_\tau \cdot \left[-u\kappa \overrightarrow{N} + \nabla u - \left(\nabla u \cdot \overrightarrow{T} \right) \overrightarrow{T} \right] \, ds \, dt \\ = & \int_0^{t_f} \int_0^{L(\tau)} C_\tau \cdot \left[-u\kappa \overrightarrow{N} + \left(\nabla u \cdot \overrightarrow{N} \right) \overrightarrow{N} \right] \, ds \, dt \\ = & \int_0^{t_f} \int_0^{L(\tau)} C_\tau \cdot \left[\nabla u \cdot \overrightarrow{N} - u\kappa \right] \overrightarrow{N} \, ds \, dt. \end{aligned} \quad (15)$$

Utilizing the fact that the only time-dependent functions are u and ∇u, we have

$$\frac{dE(\tau)}{d\tau} = \int_0^{L(\tau)} C_\tau \cdot \left[\overrightarrow{N}^T \int_0^{t_f} \nabla u \, dt - \kappa \int_0^{t_f} u \, dt \right] \overrightarrow{N} \, ds. \quad (16)$$

With this expression of $dE(\tau)/d\tau$, we will next present the curve flow algorithm used to characterize the most efficient predator front.

B. Curve Flow Algorithm

The main idea of the algorithm is to deform the shape of the curve so that the energy consumed by the predator front is increasing. To this end, our curve flow algorithm is inherently a gradient ascent algorithm, and for the following curve evolution:

$$C_\tau := \left[\overrightarrow{N}^T \int_0^{t_f} \nabla u \, dt - \kappa \int_0^{t_f} u \, dt \right] \overrightarrow{N}, \quad (17)$$

we have

$$\frac{dE(\tau)}{d\tau} = \int_0^{L(\tau)} \| C_\tau(s,\tau) \|^2 \, ds, \quad (18)$$

i.e., $dE(\tau)/d\tau$ is non-negative. Moreover, with such a choice for curve evolution, we are not required to explicitly define τ; instead, it is driving the curve evolution by our choice of C_τ. However, note that the cost function given by (1) places no restriction on the length of the curve. More specifically, the $\kappa \int_0^{t_f} u \, dt \overrightarrow{N}$ term, which represents a backward diffusion term in (17), can potentially generate infinitely long curves to increase the energy.

One way to address this would be to introduce a cost function that penalized the length of the curve. Consider the cost function, J, given by

$$J(\tau) = E(\tau) - \rho L(\tau), \quad (19)$$

where E is the energy function given by (1) and ρ is some positive constant. We take the derivative with respect to τ and obtain,

$$\frac{dJ(\tau)}{d\tau} = \frac{dE(\tau)}{d\tau} - \rho \frac{dL(\tau)}{d\tau}. \quad (20)$$

From (16), we have an expression for $dE(\tau)/d\tau$ and what remains is to find an expression for $dL(\tau)/d\tau$. The total arc length of the curve can be written as

$$L(\tau) = \int_0^1 \| C_p \| \, dp, \quad (21)$$

and by taking the derivative with respect to τ, we have

$$\frac{dL(\tau)}{d\tau} = \int_0^1 C_{p\tau}^T \overrightarrow{T} \, dp \quad (22)$$

Noting the fact that the end points do not change with respect to τ and using integration by parts, we get

$$\frac{dL(\tau)}{d\tau} = -\int_0^{L(\tau)} C_\tau^T \overrightarrow{N} \kappa \, ds. \quad (23)$$

With (16) and (23), we can rewrite (20) as $dJ(\tau)/d\tau =$

$$\int_0^{L(\tau)} C_\tau \cdot \left[\overrightarrow{N}^T \int_0^{t_f} \nabla u \, dt - \kappa \left(\int_0^{t_f} u \, dt - \rho \right) \right] \overrightarrow{N} \, ds.$$

Thus, we propose the following evolution:

$$C_\tau := \left[\overrightarrow{N}^T \int_0^{t_f} \nabla u \, dt - \kappa \left(\int_0^{t_f} u \, dt - \rho \right) \right] \overrightarrow{N}, \quad (24)$$

with the result that $dJ(\tau)/d\tau$ is non-negative. The update rule for the curve becomes

$$C(s,0,\tau_{next}) = C(s,0,\tau) + (\tau_{next} - \tau) C_\tau(s,\tau), \quad (25)$$

except at the endpoints, where the curve shape does not change.

III. EXAMPLES

For the curve flow algorithm developed in the previous section, we are only required to specify the distribution of prey at each time; it is independent of movement laws used to describe the motion of the prey aggregation and the predator-prey interaction model. In this section, we provide two examples where we apply the curve flow algorithm: we begin with the simple case of no predator-prey interactions, and next present a case where more sophisticated prey are scared of the dolphins. Simulation results are provided to illustrate the operation of the algorithm.

A. Prey Model

In the case without any predator-prey interaction, the aggregation of prey is modeled using a diffusion equation; a reaction-diffusion equation is used in the case with more sophisticated prey.

1) Diffusion: The movement of prey is described by the following equation:

$$\frac{\partial u(x,y,t)}{\partial t} = v_0 \left(\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} \right), \qquad (26)$$

where $v_0 \in \mathbb{R}_+$ is the thermal diffusivity. The prey diffuses from its initial density, $u(x,y,0)$, at a "speed" of v_0, regardless of the location of the predator front. The diffusion of the prey is shown as contours levels in Fig. 4.

This model can adequately describe diffusion of chemicals such as oil on the surface of the water, an example of the potential applications of the proposed algorithm presented earlier.

2) Reaction-Diffusion: A reaction-diffusion process is a more natural representation of the prey movement than a simple diffusion process (as the one used in the previous subsection) since it incorporates the prey response to a predator charge. In general, a reaction-diffusion process models the changes in a substance under: 1) reaction - the influence of another substance and 2) diffusion - the spatial distribution. There are numerous mathematical models of a reaction–diffusion process and the one we use is known as the FitzHugh-Nagumo model. Because of its simplicity, this model is widely used in the field of mathematical biology to describe the firing of neurons and the propagation of nerve action potentials under the excitation of ion movement across a membrane [16]. We tailor the system of partial differential equations used to describe the FitzHugh-Nagumo model in [16] to model the propagation of prey under the excitement of the predator front as follows:

$$\frac{\partial u(x,y,t)}{\partial t} = v(q) \left(\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} \right) - \sigma q, \quad (27)$$

where $\sigma \in \mathbb{R}_+$ and the diffusion coefficient, v, now depends on the predator location, q. For a curve $C(s,t,\tau)$, we define the predator location as follows:

$$q(x,y,t,\tau) = \begin{cases} 1 & \text{if } (x,y) \text{ on } C(s,t,\tau) \\ 0 & \text{otherwise} \end{cases}, \qquad (28)$$

where $s \in \{0,\ldots,L(\tau)\}$. The diffusion coefficient is modeled as

$$v(q) = \begin{cases} v_0 + \lambda & \text{if } q = 1 \\ v_0 & \text{otherwise} \end{cases}, \qquad (29)$$

where $\lambda \in \mathbb{R}_+$. Such a formulation for the thermal diffusivity captures the idea of the prey being "scared" in the presence of predators. For a location (x,y), when $q = 0$ (i.e., there are no predators present in that location), the prey-flock diffuses according to the nominal speed of v_0; but when $q = 1$, they diffuse faster at a speed of $v_0 + \lambda$. We also capture the idea of prey being "removed" with the $-\sigma q$ term.

This model can be used to represent aggregation of smart agents in a swarm or material that react to the movement of the clearing front.

Our mathematical models for predator fronts and prey aggregations are based on creating simple, yet rich biological models. Recall that the goal of the work is not biomimicry, but to draw inspiration from biology for engineering applications.

B. The Curve Flow Algorithm

The implementation of curve flow algorithm for a diffusion-based prey density is presented in Algorithm (1). By using (26), where the prey movement and predator positions are completely decoupled, the algorithm only requires us to calculate the prey density terms, $u(x,y,t)$ and $\nabla u(x,y,t)$, once. These values are stored and subsequently accessed each time the curve position is updated during a sweep. Also, notice that in Routine (2), the unit normal and the curvature is calculated only for the interior points on the curve. As a result, the two endpoints of the cure do not vary with τ, a requirement that was analytically useful in the derivation of $dE(\tau)/d\tau$.

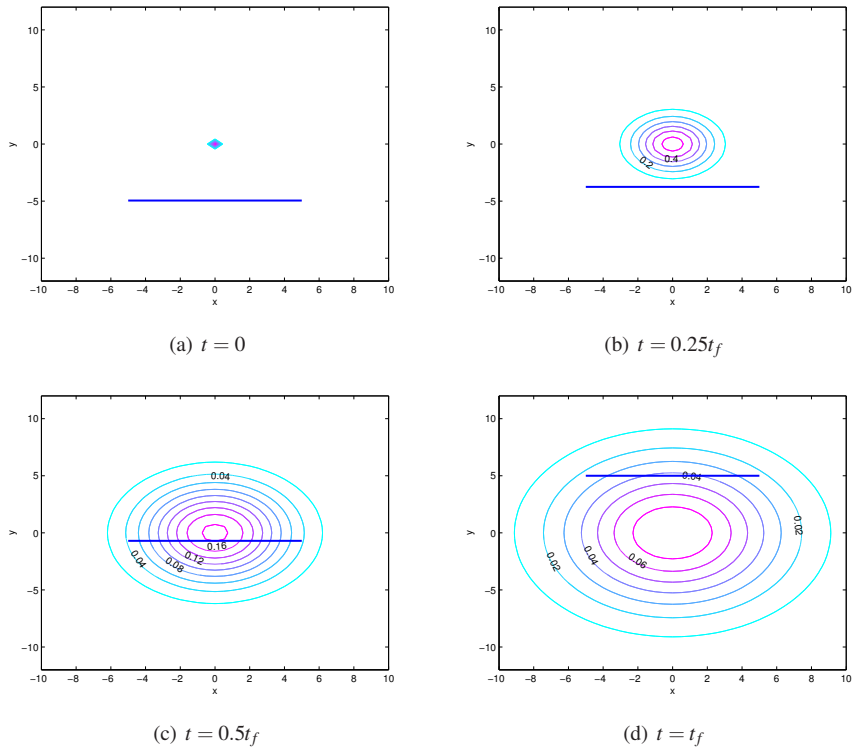
The algorithm for the reaction-diffusion case is similar to Algorithm (1), with the difference that density terms should be calculated inside Routine (3) at each time.

Algorithm 1 Curve flow algorithm

```
Specify initial prey distribution u(x,y,0)
Calculate ∇u(x,y,0)
for t = 0 : dt : t_f do
    Calculate u(x,y,t+dt)
    Calculate ∇u(x,y,t+dt)
end for
τ ⇐ 0
Generate N points to specify initial curve C(0,0,0)
L(0) ⇐ length of C(0,0,0)
Call Routine (2)
while ‖C_τ(s,τ)‖ > ε do
    τ ⇐ τ + 1
    L(τ) ⇐ length of C(0,0,τ)
    Call Routine (2)
end while
```

Routine 2 Calculate C_τ

```
for s = L(τ)/N to ((N−1)/N)L(τ) do
    Calculate N⃗(s,τ)
    Calculate κ(s,τ)
    Call Routine (3)
    Calculate C_τ(s,τ) according to (24)
    C(s,0,τ+1) ⇐ C(s,0,τ) + γC_τ(s,τ)
    return C(s,0,τ+1)
    return C_τ(s,0,τ)
end for
```



(a) $t = 0$ (b) $t = 0.25t_f$

(c) $t = 0.5t_f$ (d) $t = t_f$

Fig. 4. The curve at $\tau = 0$, a line, is swept through a prey density. The density, centered at $(0,0)$ and diffusing according to (26), is represented with a contour map.

Routine 3 Calculate $\int_0^{t_f} u(C)$ and $\int_0^{t_f} \nabla u(C)$

```
int_u(C) ⇐ 0
int_∇u(C) ⇐ 0
for t = 0 : dt : t_f do
    Calculate u(C(s, t + dt, τ))
    Calculate ∇u(C(s, t + dt, τ))
end for
Calculate int_u(C)
Calculate int_∇u(C)
return int_u(C)
return int_∇u(C)
```

C. Simulations

The foraging area is represented by a $2D$ mesh, where $x_{min} = -10$, $x_{max} = 10$, $y_{min} = -10$, $y_{max} = 10$, and the mesh spacing is $\Delta x = \Delta y = 0.5$. The initial prey density is a rectangle (1x13 units), where the center of the bottom edge is located at $(0,-5)$, as shown in Fig. 5(a). The diffusion process is numerically solved with thermal diffusivity set to 0.5.

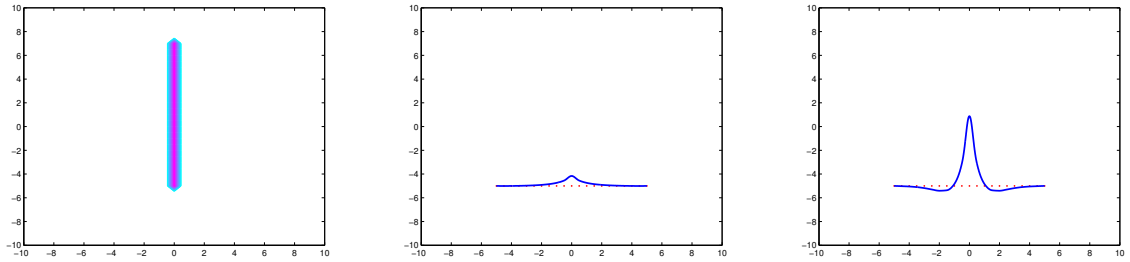
For each τ, the resulting curve is swept through the prey density from $t_i = 0$ to $t_f = 10$, with a time step of $\Delta t = 0.05$. We use 21 data points to characterize the curve and the initial curve (at $\tau = 0$) is a straight line (shown as a dotted line in Figs. 5 and 6).

Fig. 5 depicts the two evolutions of the curve: one under the pure diffusion of prey and the other is for the reaction-diffusion case. The curves illustrated at Figs. 5(b) and 5(c) respectively represent the optimal predator front for the diffusion and the reaction-diffusion case.

Although the shape of the optimal curve depends on the nature of its interaction with prey, for the prey model used in this paper, the overall tendency of the curve is to add more length to the locations where prey is highly concentrated at the beginning of the sweep. As seen in Fig. 5, the curve that maximizes energy intake for the reactively diffusing prey adds more length to these locations than the curve under the purely diffusive case. More specifically, since prey diffuse faster once they sense the predator front, we notice that the agents in the off-center positions tend to arrange themselves behind one another to collect the prey that diffuse due to the presence of the agent located in the center of the curve. Further, the shape of the optimal curve for the reaction-diffusion case depends on the thermal diffusivity of prey - a longer curve is generated when prey diffuse faster in the presence of predators (Fig. 6).

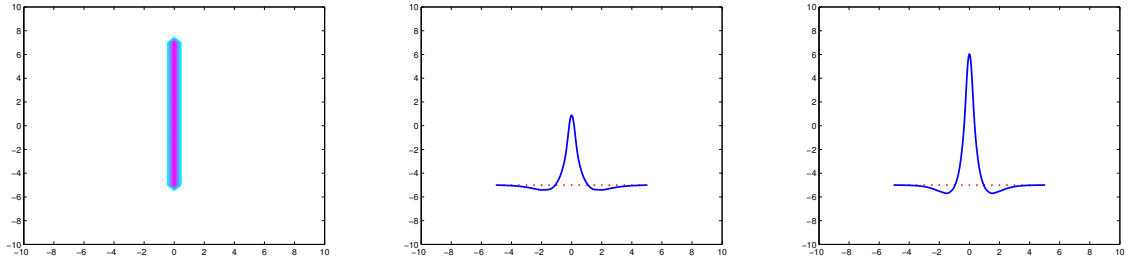
IV. CONCLUSION

In this bio-inspired work, we characterize the most efficient shape of the predator front to maximize the amount of prey swept. Prey aggregation is modeled using the population framework and the predator front is modeled as a $2D$ curve. Using curve evolution techniques, an algorithm is proposed to deform the predator front to maximize the total energy



(a) Initial distribution of prey. (b) Diffusion case. (c) Reaction-diffusion case.

Fig. 5. Evolution of the predator front under the curve flow algorithm given by (24) for two types of prey movement processes. (a) The prey density is represented by contour levels. (b)-(c) The dotted line depicts the initial estimate of the shape of the curve, and the solid line represents the optimal shape of the curve.



(a) Initial distribution of prey. (b) $\lambda = 5v_0$ (c) $\lambda = 10v_0$

Fig. 6. The optimal curve is shown for two different values of λ. Note that a larger value of λ represents the case where prey diffuses faster in the presence of predators. More specifically, a larger λ increases the thermal diffusivity of prey in (29).

intake of the predators. This algorithm is independent of the method for which the prey density is calculated. Diffusion and reaction-diffusion prey aggregation models are presented as examples for which simulations show the application of the proposed algorithm. Potential applications of this work include design of the suction boom for surface oil skimming and design of the net front in capturing smart marine mines.

REFERENCES

[1] K. Pryor and K. Norris, *Dolphin Societies*, Berkeley, CA: University of California Press, 1998.

[2] R.D. Estes, *Behavior Guide to African Mammals*, Berkeley, CA: University of California Press, 1991.

[3] D.A. Shell and M.J. Matarić, "On foraging strategies for large-scale multi-robot systems", *Intl. Conf. on Intelligent Robots and Systems*, 2006, pp. 2717–2723.

[4] T. Balch, "The impact of diversity on performance in multi-robot foraging", *Proc. Third Conf. on Autonomous Agents*, 1999, pp. 92–99.

[5] T.H. Labella, M. Dorigo, and J. Deneubourg, Self-organised task allocation in a group of robots, *Distributed Autonomous Robotic Systems*, 2004.

[6] N. Lemmens, S. Jong, K. Tuyls, and A. Nowé, "Bee Behaviour in Multi-agent Systems", *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, 2008, pp. 145–156.

[7] G. Ferrari-Trecate, M. Egerstedt, A. Buffa and M. Ji, Laplacian Sheep: A Hybrid, Stop-Go Policy for Leader-Based Containment Control, *Hybrid Systems: Computation and Control*, Springer-Verlag, 2006, pp. 212–226.

[8] M. Haque, A. Rahmani, and M. Egerstedt, "Geometric Foraging Strategies in Multi-Agent Systems Based on Biological Models," *Conference on Decision and Control*, Atlanta, USA, Dec 2010.

[9] Y. Shi and W. Karl, A fast level set method without solving PDEs, *Int. Conf. on Acoustics, Speech, and Signal Processing*, 2005, pp. 97–100.

[10] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, 2001, pp. 266–277.

[11] S. Lankton, D. Nain, A. Yezzi, and A. Tannenbaum, "Hybrid Geodesic Region-Based Curve Evolutions for Image Segmentation", *SPIE Medical Imaging*, 2007.

[12] C. Packer and A.E. Pusey, Divided We Fall: Cooperation among Lions, *Scientific American Magazine*, May 1997.

[13] S.-H. Lee, H.K. Pak, and T.-S. Chon, Dynamics of prey-flock escaping behavior in response to predator's attack, *Journal of Theoretical Biology*, vol. 240, 2006, pp 250–259.

[14] M.A. Haque, A.R. Rahmani, and M. Egerstedt, "A Hybrid, Multi-Agent Model of Foraging Bottlenose Dolphins", *Third IFAC Conf. on Analysis and Design of Hybrid Systems*, Zaragoza, Spain, 2009.

[15] J.E. Rhodes and G.S. Holder, *Concept for Future Naval Mine Countermeasures in Littoral Power Projection*, 1998.

[16] J.D. Murray, *Mathematical Biology I: An Introduction*, New York, NY: Springer, 2002.

[17] R. Heinsohn and C. Packer, Complex Cooperative Strategies in Group-Territorial African Lions, *Science*, vol. 269, September 1995, pp. 1260–1262.

[18] J.G.F. Francis, The QR Transformation I, *Comput. J.*, vol. 4, 1961, pp 265-271.

[19] H. Kwakernaak and R. Sivan, *Modern Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ; 1991.

[20] D. Boley and R. Maier, "A Parallel QR Algorithm for the Non-Symmetric Eigenvalue Algorithm", *in Third SIAM Conference on Applied Linear Algebra*, Madison, WI, 1988, pp. A20.