Theor. Comput. Fluid Dyn. (2016) 30:41–66
DOI 10.1007/s00162-015-0358-5

ORIGINAL ARTICLE

Seth I. Dillard · James H. J. Buchholz · H. S. Udaykumar

From video to computation of biological fluid–structure interaction problems

Received: 9 January 2015 / Accepted: 27 June 2015 / Published online: 14 July 2015

Abstract This work deals with the techniques necessary to obtain a purely Eulerian procedure to conduct CFD simulations of biological systems with moving boundary flow phenomena. Eulerian approaches obviate difficulties associated with mesh generation to describe or fit flow meshes to body surfaces. The challenges associated with constructing embedded boundary information, body motions and applying boundary conditions on the moving bodies for flow computation are addressed in the work. The overall approach is applied to the study of a fluid–structure interaction problem, i.e., the hydrodynamics of swimming of an American eel, where the motion of the eel is derived from video imaging. It is shown that some first-blush approaches do not work, and therefore, careful consideration of appropriate techniques to connect moving images to flow simulations is necessary and forms the main contribution of the paper. A combination of level set-based active contour segmentation with optical flow and image morphing is shown to enable the image-to-computation process.

Keywords Moving boundaries · Level sets · Optical flow · Morphing · Image-based modeling · Cartesian grid methods

1 Introduction

This work is directed toward developing methods for modeling complex time-dependent geometries and fluid transport mechanisms in biological systems. Because biological systems change shape continually during movement and generally have complex shapes, the tasks of describing their surfaces and interaction with fluids and computationally meshing them for computational fluid dynamics (CFD) analyses have proven challenging. Examples of moving boundary problems of the type that would benefit from image-based modeling include aquatic animal locomotion studies with motions specified by video imaging, peristaltic motion of the intestine imaged ex vivo [1], heart valve dynamics imaged with 4D modalities, such as MRI, and a host of others. Current simulation techniques reported in the literature concerning animal locomotion appear to primarily involve defining simplified model geometries based upon measurements taken of animals directly, followed by prescribing some sort of motion in order to replicate observed behaviors, e.g., modeling a fish as an ellipsoid propelled by an oscillating flat plate [2,3]. Such models certainly lend a great deal of useful physical insight,

Communicated by Jeff D. Eldredge.

Seth I. Dillard (✉)
Department of Biomedical Engineering, Seamans Center for the Engineering Arts and Sciences, The University of Iowa, Iowa City, IA 52242-1527, USA
E-mail: seth-dillard@uiowa.edu
Tel.: 319-384-3573

James H. J. Buchholz · H. S. Udaykumar
Department of Mechanical and Industrial Engineering, Seamans Center for the Engineering Arts and Sciences, The University of Iowa, Iowa City, IA 52242-1527, USA

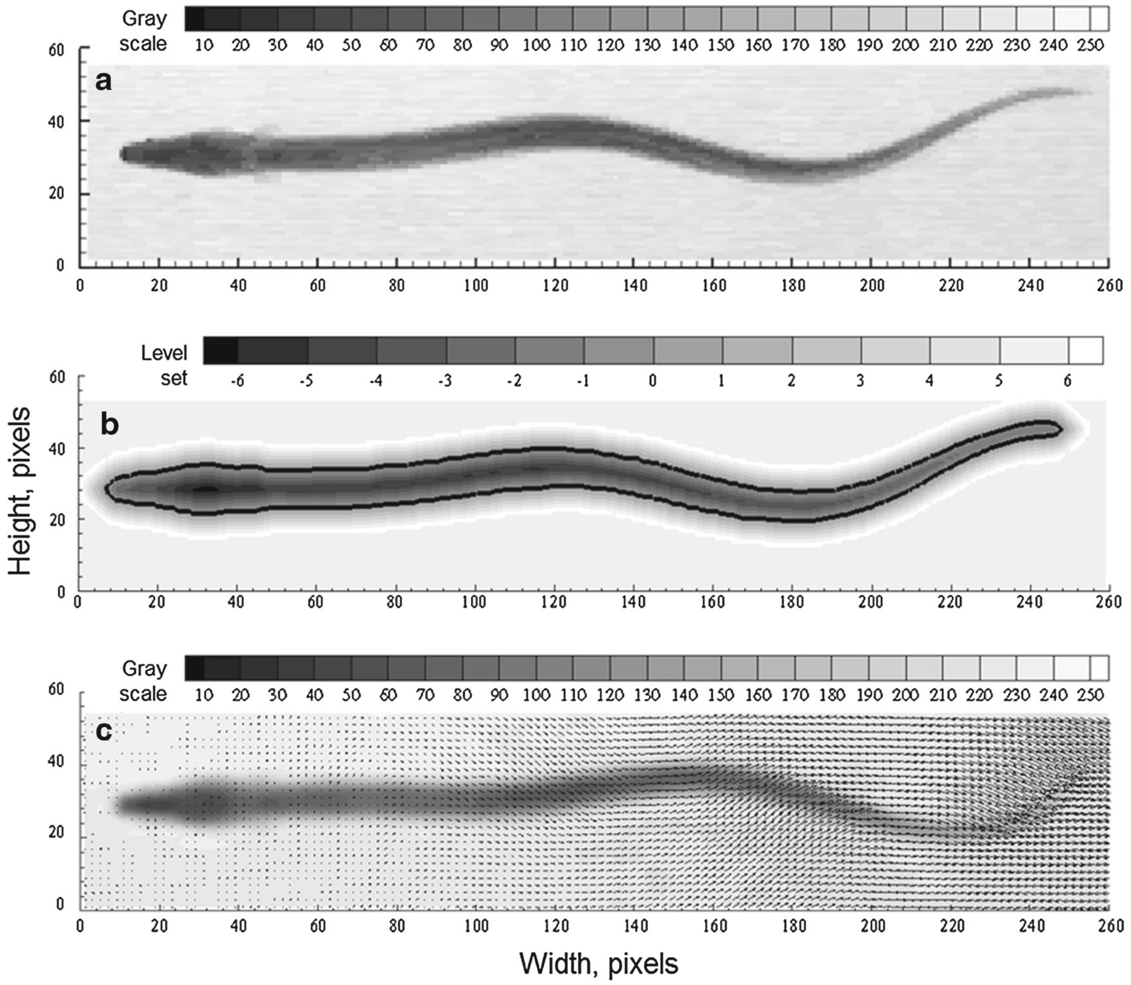

Fig. 1 **a** A single frame of the swimming American eel video sequence used in developing this work; **b** active contour segmentation converts the imaged object into a level set field; **c** optical flow vectors produced by applying the method of Brox et al. to a pair of frames in the swimming American eel video sequence

though limitations exist when attempting to model complex biological behaviors in a geometrically simplified fashion.

An alternative approach that may prove attractive, such as the one taken in this work, employs image segmentation to define directly the shapes of complex objects in a computational flow domain based on their visual appearance. This allows for accurate representation of geometry and motion and is an ongoing area of active research, particularly in the fields of medical imagery and computer vision [4–13]. However, conventional CFD approaches still require surface mesh generation (and volume mesh generation in 3D) in order to model moving boundaries, making for tedious implementation. It is therefore desirable to develop a modeling framework that bypasses mesh generation altogether, both to define the surface of the moving entities and to solve for the flow. This work proposes an approach to establishing such a framework, demonstrating employed methods on a sequence of video images illustrating the swimming of an American eel (Fig. 1a).

2 From moving images to flow computation

2.1 The flow solver with embedded moving boundaries

Techniques for solving transport phenomena in complex domains (with stationary or moving boundaries) have been developed in a fixed Cartesian grid Eulerian framework with local mesh refinement [14–16]. The challenge of developing techniques for treating the embedded boundaries as sharp entities has been addressed in several previous publications [14,15,17,18]. The current work is concerned with computing fluid motion in geometries defined by level sets extracted from images where the flow field is obtained by solving the incompressible Navier–Stokes equations:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}. \tag{2}$$

Here, $\boldsymbol{u}$ and p are the fluid velocity vector and pressure, and $\mathrm{Re} = UL/\nu$ is the Reynolds number. U and L are characteristic flow velocity and length scales, and ν is the kinematic fluid viscosity. The governing equations are advanced in time using a second-order four-step fractional step algorithm to segregate the pressure from the velocity [19,20]. The nonlinear and viscous terms are integrated temporally using the second-order explicit Adams–Bashforth and semi-implicit Crank–Nicolson schemes, and all spatial derivatives are approximated using second-order central differencing. The variables are stored at mesh cell centers with the exception of a set of cell face velocities (satisfying the divergence-free condition over each mesh cell) that are used to construct the advection terms.

Boundary conditions on embedded boundaries (defined by the zero-contours of the embedded narrow-band level set field) are applied using a least-squares formulation of the sharp interface method. In this approach, a quadratic least-squares method is used to extrapolate flow variable values to a set of ghost nodes immediately outside the flow domain [19]. The least-squares formulation allows for locally second-order spatial convergence rates on arbitrarily shaped domains. The implicit filtering inherent to least-squares methods also enhances the robustness of the approach when geometries with noisy surfaces are simulated. A detailed analysis of the current interface treatment is described in [19].

2.2 Moving boundaries

In the case of moving interfaces, boundary motion is tracked by advecting a level set field [21] using

$$(\phi_l)_t + \boldsymbol{V}_l \cdot \nabla \phi_l = 0. \tag{3}$$

In Eq. 3, $\boldsymbol{V}_l$ is the lth level set velocity field, which is derived directly from the physics of the problem. A fourth-order ENO scheme in space and fourth-order Runge–Kutta integration in time are used for the evolution of the level set field, and the phase of Cartesian grid cells (i.e., solid or fluid) is continuously stored and updated as necessary as objects move through the domain [15,22,23]. Since $\boldsymbol{V}_l$ is prescribed only by the physics on the interface (i.e., on the zero-level set), the velocity values at grid points that lie in the narrow band around the zero-level set need to be obtained. This is done by extension of the interfacial velocity [24] away from the front using

$$\psi_t + \boldsymbol{V}_{\mathrm{ext}} \cdot \nabla \psi = 0, \tag{4}$$

where ψ is the quantity (i.e., the interface velocity component $V_{l,x}$ or $V_{l,y}$) that needs to be extended away from the interface. A natural choice for the extension velocity is $\boldsymbol{V}_{\mathrm{ext}} = \mathrm{sign}\,(\phi_l)\,\frac{\nabla \phi_l}{|\nabla \phi_l|}$, which extends the velocity in a normal direction outward from an interface. A reinitialization procedure [25,26] is carried out after level set advection to return the ϕ-field to a signed distance function, i.e., to satisfy $|\nabla \phi_l| = 1$. Defining $(\phi_l)_o$ as the level set field prior to re-initialization, the following equation is solved to steady state, in order to re-initialize the level set field for the next advection step:

$$(\phi_l)_t + \boldsymbol{w} \cdot \nabla \phi_l = \mathrm{sign}\,(\phi_l) \tag{5}$$

$$\boldsymbol{w} = \mathrm{sign}\left((\phi_l)_o\right) \frac{\nabla (\phi_l)_o}{\left|\nabla (\phi_l)_o\right|}. \tag{6}$$

Here, $\mathrm{sign}\left((\phi_l)_o\right) = \frac{(\phi_l)_o}{\sqrt{(\phi_l)_o^2 + (\Delta x)^2}}$ and has the initial condition $\phi_l(\boldsymbol{x}, 0) = (\phi_l)_o(\boldsymbol{x})$.

An alternative to reinitialization proposed by Li et al., in which distance regularization is built directly into the level set formulation, promises to alleviate possible sources of error inherent to the reinitialization process by doing away with it altogether; for further details about this approach, the reader is referred to [27].

2.3 Image-based modeling: an Eulerian approach

The primary intent of this work is to devise an image-based approach suited to a purely Eulerian (Cartesian grid) sharp interface moving boundary flow solver, in which the need to generate meshes to conform to moving geometries is removed. In fact, in a purely Eulerian setting the need to employ any surface meshing to describe the embedded geometries is also obviated. Through image segmentation, a level set field can be generated on an image mesh and then later mapped directly onto a flow solver mesh by interpolation. The surface velocity of the imaged object is then computed from the image intensity field using *optical flow* [5,8,11]. However, image sequence frame rates are much too slow to match the small time steps required for high-resolution CFD simulations, so intermediate representations of object locations are required to fill in the missing information between image frames. It was initially thought that optical flow would act as the primary mechanism by which this missing level set information could be obtained, but advecting surfaces using optical flow field information alone did not work as anticipated. Thus, another computer vision technique, *image morphing* [28,29], designed to fulfill precisely the purpose we seek in completing the Eulerian modeling framework we set out to build, was employed to supply the missing information between image frames so that the relatively low temporal resolution of image sequences no longer precludes modeling behavior on much finer timescales. This suite of techniques that proceed from images to Eulerian flow computations is described in the following.

2.3.1 Processing video images to define embedded boundaries

Embedded surfaces are represented implicitly on the Cartesian mesh using level sets [17,21,24,30] as mentioned above. Images are processed using the active contour approach [9,31], which delivers level set fields that delineate the embedded object boundaries (Fig. 1b). Active contour evolution methods are based on early applications of elastic deformation theory to computer vision [32]. Contours are modeled as having elastic properties that act to minimize the energy of mismatch between an evolving field and some underlying topology, which in this context is defined by the arrangement of image brightness patterns. The method was initiated by Mumford and Shah [33] who proposed an energy functional that decomposes the image domain Ω into piecewise smooth regions delineated by a segmentation contour C,

$$E\left(f, C\right) = \mu L\left(C\right) + \lambda \int_{\Omega} \left(I_o - I\right)^2 \mathrm{d}\Omega + \int_{\Omega \setminus C} |\nabla I|^2 \mathrm{d}\Omega, \tag{7}$$

in which $L(C)$ is the total arc length of the segmentation contour and $\lambda > 0$, $\mu \geq 0$ are weighting parameters. The arc length penalty (regularization) term helps ensure that segmentation contours follow object boundaries smoothly by way of arc length minimization, while the second term quantifies the local mismatch between a representative function I and the original underlying image intensity I_o throughout the image domain, and the third term directs the evolving contour(s) toward boundaries defined by large gradients. The Mumford–Shah method was later modified by Chan and Vese [31] to delineate images into piecewise constant regions of average intensity rather than piecewise smooth regions of slowly varying intensity, eliminating the need for a smoothness constraint. Chan and Vese [31] also recast the energy minimization problem in terms of level sets, so that segmentation surfaces could be represented implicitly as zero-level isocontours of a signed distance level set field ϕ embedded in the Cartesian image space. This segmentation approach has found success in a variety of contexts, further details about which can be found in [34,35].

2.3.2 Optical flow

Optical flow has been an active area of research in the field of computer vision since it was first introduced by Horn and Schunck [11]. In their formulation, Horn and Schunck assumed that the brightness of any moving structure (measured by the intensity field I) in the image is constant, so that as the object moves

$$\frac{\mathrm{d}I}{\mathrm{d}t} = 0, \tag{8}$$

or, in the current Eulerian perspective,

$$\frac{\partial I}{\partial t} + \boldsymbol{u} \cdot \nabla I = 0 \tag{9}$$

Because intensity is a scalar value measured locally at each pixel comprising an image, another constraint is needed in order to provide an optical flow velocity, which has two components in a 2D image sequence. To this end, Horn and Schunck [11] further proposed that objects in an image will always undergo continuous deformation or rigid body motion and therefore that adjacent object regions must have similar velocities; the image intensity velocity field must vary smoothly everywhere except at object boundaries, where there are discontinuities. This introduces a smoothness constraint, the second constraint needed to construct a 2-component velocity field, which minimizes the magnitude of the gradient of the optical flow field velocity: $\sqrt{\nabla u \cdot \nabla u} = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2}$ [11].

Thus, the overall aim is to approximately satisfy the constraints by minimizing simultaneously [11]:

(a) the sum of the errors in image brightness

$$\varepsilon_b = I_x u + I_y v + I_t \tag{10}$$

(b) the departure from flow field smoothness

$$\varepsilon_s^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \tag{11}$$

where subscripts of image intensity I denote partial derivatives with respect to the subscripted variable. Minimizing these two error equations together is necessary and sufficient to obtain a complete flow field [11].

Since it is desired to determine the optical flow field over the entire image domain, the problem is posed as an integral functional, which represents a total error or energy to be minimized:

$$E^2 = \iint \mathcal{F} \mathrm{d}x\mathrm{d}y, \tag{12}$$

where

$$\mathcal{F} = \alpha^2 \varepsilon_s^2 + \varepsilon_b^2, \tag{13}$$

with a weighting factor α^2 introduced for the purpose of determining the relative importance of velocity and smoothness errors. Thus, the goal here is to find a mapping $\mathcal{F} : \mathbb{R}^2 \rightarrow \mathbb{R}$ that minimizes the energy E^2 characterizing image space $\Omega \subset \mathbb{R}^2$, and the mapping F is given by the solution to a pair of Euler–Lagrange equations [11,36,37]:

$$I_x \left(I_x u + I_y v + I_t\right) - \alpha^2 \nabla^2 u = 0 \tag{14}$$

$$I_y \left(I_x u + I_y v + I_t\right) - \alpha^2 \nabla^2 v = 0. \tag{15}$$

In practice, the relative weighting α^2 between brightness constancy and smoothness constraints is largely image dependent. Empirically, Jhunjhunwala and Rajagopalan [12] found that the best results were achieved on their test images by setting α equal to the gradient magnitude calculated at a given pixel, for every pixel in the domain, i.e.:

$$\alpha_i = \|\nabla I_i\| \tag{16}$$

Overall, the variational formulation proposed by Horn and Schunck [11] was found by them to work well for smooth sample images, but suffered large errors near discontinuities in some of the test images analyzed. A notable improvement was made by Brox et al. [8], in which gray value constancy is assumed as in previous works but left in the Lagrangian form

$$I\left(x, y, t\right) = I\left(x + u, y + v, t + 1\right) \tag{17}$$

for $I : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$. If $\boldsymbol{x} := [x, y, t]^{\mathrm{T}}$ and $\boldsymbol{w} := [u, v, 1]^{\mathrm{T}}$, then Eq. 17 can be written more compactly as [8]

$$I\left(\boldsymbol{x}\right) = I\left(\boldsymbol{x} + \boldsymbol{w}\right). \tag{18}$$

In addition to assuming brightness constancy, Brox et al. introduced an additional gradient constancy assumption which states that the relative brightness of an object and its surroundings stays the same regardless of an object's location or the time at which it is being viewed, effectively reducing sensitivity to overall changes in image brightness [8]:

$$\nabla I\left(\boldsymbol{x}\right) = \nabla I\left(\boldsymbol{x} + \boldsymbol{w}\right). \tag{19}$$

With the new constraint imposed on the optical flow field, the brightness error is changed from the standard Horn and Schunck formulation [11] to:

$$\varepsilon_b^2 = |I(\boldsymbol{x}+\boldsymbol{w}) - I(\boldsymbol{x})|^2 + \gamma\,|\nabla I(\boldsymbol{x}+\boldsymbol{w}) - \nabla I(\boldsymbol{x})|^2\,. \tag{20}$$

In Eq. 20, γ is a relative weighting between brightness constancy and brightness gradient constancy.

To further reduce sensitivity to violations of the brightness constancy assumption, Eq. 20 is tempered with a soft penalty function in order to lessen the impact of outliers in the image domain. Based on work by Black and Anandan [6], Brox et al. chose to achieve this sensitivity reduction using a modified L^1 norm function $\Psi\left(s^2\right) = \sqrt{s^2+\epsilon^2}$, where ϵ is a small value that prevents problems associated with a machine's attempt to numerically approximate the square root of zero [8]:

$$\varepsilon_b^2 = \Psi\left(|I(\boldsymbol{x}+\boldsymbol{w}) - I(\boldsymbol{x})|^2 + \gamma\,|\nabla I(\boldsymbol{x}+\boldsymbol{w}) - \nabla I(\boldsymbol{x})|^2\right). \tag{21}$$

In this way, ε_b^2 cannot vary linearly about noisy outlier pixels, but varies less abruptly as the square root of its original value.

The smoothness constraint is also used as in previous works, but is enforced over the entire spatio-temporal domain and is modified in the same way as the brightness constraint to lessen the unwanted effects of noise:

$$\varepsilon_s^2 = \Psi\left(|\nabla u|^2 + |\nabla v|^2\right), \tag{22}$$

where $\nabla := \left[\partial_x, \partial_y, \partial_t\right]$. Introducing the usual regularization weight α^2, the energy functional to be minimized takes on the familiar form

$$E^2(x,y) = \iiint \left(\varepsilon_b^2 + \alpha^2\varepsilon_s^2\right)\mathrm{d}\boldsymbol{x}. \tag{23}$$

As with the original optical flow approach, the goal is to find the functions u and v that minimize $E^2(x,y)$. The variational formulation again leads to a set of Euler–Lagrange equations, which now have the form [8]

$$\Psi'\left(I_t^2 + \gamma I_{xt}^2 + \gamma I_{yt}^2\right)\cdot\left[I_x I_t + \gamma I_{xx} I_{xt} + \gamma I_{xy} I_{yt}\right] - \alpha^2\nabla\cdot\left[\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right)\nabla u\right] = 0 \tag{24}$$

$$\Psi'\left(I_t^2 + \gamma I_{xt}^2 + \gamma I_{yt}^2\right)\cdot\left[I_y I_t + \gamma I_{yy} I_{yt} + \gamma I_{xy} I_{xt}\right] - \alpha^2\nabla\cdot\left[\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right)\nabla v\right] = 0, \tag{25}$$

with $\Psi'\left(s^2\right) = \frac{1}{2\sqrt{s^2+\epsilon^2}}$

The Euler–Lagrange equations defined in this way are nonlinear in $\boldsymbol{w}$, so fixed-point iterations are performed using a multigrid approach. If $\boldsymbol{w}^k := \left[u^k, v^k, 1\right]^{\mathrm{T}}$ with initialization $\boldsymbol{w}^o = [0,0,1]^{\mathrm{T}}$ at the coarsest grid level, then $\boldsymbol{w}^{k+1}$ is the solution of [8]

$$\begin{aligned}&\Psi'\left(\left[I_t^{k+1}\right]^2 + \gamma\left[I_{xt}^{k+1}\right]^2 + \gamma\left[I_{yt}^{k+1}\right]^2\right)\cdot\left[I_x^k I_t^{k+1} + \gamma I_{xx}^k I_{xt}^{k+1} + \gamma I_{xy}^k I_{yt}^{k+1}\right]\\ &-\alpha^2\nabla\cdot\left[\Psi'\left(\left|\nabla u^{k+1}\right|^2 + \left|\nabla v^{k+1}\right|^2\right)\nabla u^{k+1}\right] = 0\end{aligned} \tag{26}$$

$$\begin{aligned}&\Psi'\left(\left[I_t^{k+1}\right]^2 + \gamma\left[I_{xt}^{k+1}\right]^2 + \gamma\left[I_{yt}^{k+1}\right]^2\right)\cdot\left[I_y^k I_t^{k+1} + \gamma I_{yy}^k I_{yt}^{k+1} + \gamma I_{xy}^k I_{xt}^{k+1}\right]\\ &-\alpha^2\nabla\cdot\left[\Psi'\left(\left|\nabla u^{k+1}\right|^2 + \left|\nabla v^{k+1}\right|^2\right)\nabla v^{k+1}\right] = 0.\end{aligned} \tag{27}$$

Once a fixed-point solution in $\boldsymbol{w}^k$ is reached, the solution grid is refined and the coarse grid solution is used as an initialization at the new scale. However, Ψ' and I_*^{k+1} remain nonlinear and thus must be linearized before a solution for the optical flow field may be obtained. First-order Taylor expansions linearize I_*^{k+1} [8]:

$$I_t^{k+1} \approx I_t^k + I_x^k\,\mathrm{d}u^k + I_y^k\,\mathrm{d}v^k \tag{28}$$

$$I_{xt}^{k+1} \approx I_{xt}^k + I_{xx}^k\,\mathrm{d}u^k + I_{xy}^k\,\mathrm{d}v^k \tag{29}$$

$$I_{yt}^{k+1} \approx I_{yt}^k + I_{xy}^k \mathrm{d}u^k + I_{yy}^k \mathrm{d}v^k. \tag{30}$$

where

$$u^{k+1} = u^k + \mathrm{d}u^k \tag{31}$$

$$v^{k+1} = v^k + \mathrm{d}v^k. \tag{32}$$

For shorthand, Brox et al. define

$$(\Psi')_b^k := \Psi' \left(\left[I_t^k + I_x^k \mathrm{d}u^k + I_y^k \mathrm{d}v^k \right]^2 + \gamma \left[I_{xt}^k + I_{xx}^k \mathrm{d}u^k + I_{xy}^k \mathrm{d}v^k \right]^2 + \gamma \left[I_{yt}^k + I_{xy}^k \mathrm{d}u^k + I_{yy}^k \mathrm{d}v^k \right]^2 \right) \tag{33}$$

and

$$(\Psi')_s^k := \Psi' \left(\left| \nabla \left(u^k + \mathrm{d}u^k \right) \right|^2 + \left| \nabla \left(v^k + \mathrm{d}v^k \right) \right|^2 \right). \tag{34}$$

They term Eq. 33 “data term robustness” and Eq. 34 “smoothness diffusivity,” and with these new shorthand definitions write the Euler–Lagrange equations once again as

$$\begin{aligned} &(\Psi')_b^k \cdot \left\{ I_x^k \left[I_t^k + I_x^k \mathrm{d}u^k + I_y^k \mathrm{d}v^k \right] \right\} + \gamma \left(\Psi' \right)_b^k \left\{ I_{xx}^k \left[I_{xt}^k + I_{xx}^k \mathrm{d}u^k + I_{xy}^k \mathrm{d}v^k \right] \right. \\ &\left. + I_{xy}^k \left[I_{yt}^k + I_{xy}^k \mathrm{d}u^k + I_{yy}^k \mathrm{d}v^k \right] \right\} - \alpha^2 \nabla \cdot \left\{ \left(\Psi' \right)_s^k \nabla \left(u^k + \mathrm{d}u^k \right) \right\} = 0 \end{aligned} \tag{35}$$

$$\begin{aligned} &(\Psi')_b^k \cdot \left\{ I_y^k \left[I_t^k + I_x^k \mathrm{d}u^k + I_y^k \mathrm{d}v^k \right] \right\} + \gamma \left(\Psi' \right)_b^k \left\{ I_{yy}^k \left[I_{yt}^k + I_{xy}^k \mathrm{d}u^k + I_{yy}^k \mathrm{d}u^k \right] \right. \\ &\left. + I_{xy}^k \left[I_{xt}^k + I_{xx}^k du^k + I_{xy}^k dv^k \right] \right\} - \alpha^2 \nabla \cdot \left\{ \left(\Psi' \right)_s^k \nabla \left(v^k + dv^k \right) \right\} = 0. \end{aligned} \tag{36}$$

It should be noted that this is still a nonlinear system of equations for fixed point k, but now in increments of $\mathrm{d}u^k$ and $\mathrm{d}v^k$ [8]. The remaining nonlinearity in Ψ' is removed using a second inner fixed-point iteration loop over a new step l, where $\mathrm{d}u^{k,l}$ and $\mathrm{d}v^{k,l}$ denote the variables iterated on, with $\mathrm{d}u^{k,0} := 0$ and $\mathrm{d}v^{k,0} := 0$ [8]. Now the Euler–Lagrange equations take on the final form

$$\begin{aligned} &(\Psi')_b^{k,l} \cdot \left\{ I_x^k \left[I_t^k + I_x^k \mathrm{d}u^{k,l+1} + I_y^k \mathrm{d}v^{k,l+1} \right] \right\} + \gamma \left(\Psi' \right)_b^{k,l} \left\{ I_{xx}^k [I_{xt}^k + I_{xx}^k \mathrm{d}u^{k,l+1} + I_{xy}^k \mathrm{d}v^{k,l+1}] \right. \\ &\left. + I_{xy}^k \left[I_{yt}^k + I_{xy}^k \mathrm{d}u^{k,l+1} + I_{yy}^k \mathrm{d}v^{k,l+1} \right] \right\} - \alpha^2 \nabla \cdot \left\{ \left(\Psi' \right)_s^{k,l} \nabla \left(u^k + \mathrm{d}u^{k,l+1} \right) \right\} = 0 \end{aligned} \tag{37}$$

$$\begin{aligned} &(\Psi')_b^{k,l} \cdot \left\{ I_y^k \left[I_t^k + I_x^k \mathrm{d}u^{k,l+1} + I_y^k \mathrm{d}v^{k,l+1} \right] \right\} + \gamma \left(\Psi' \right)_b^{k,l} \left\{ I_{yy}^k [I_{yt}^k + I_{xy}^k \mathrm{d}u^{k,l+1} + I_{yy}^k \mathrm{d}v^{k,l+1}] \right. \\ &\left. + I_{xy}^k \left[I_{xt}^k + I_{xx}^k \mathrm{d}u^{k,l+1} + I_{xy}^k \mathrm{d}v^{k,l+1} \right] \right\} - \alpha^2 \nabla \cdot \left\{ \left(\Psi' \right)_s^{k,l} \nabla \left(v^k + \mathrm{d}v^{k,l+1} \right) \right\} = 0. \end{aligned} \tag{38}$$

Applying their method to a set of test images with known displacements, Brox et al. [8] found that resultant errors in the optical flow field were diminished significantly when compared to the Horn and Schunck [11] approach. Figure 1c illustrates the optical flow vector field resulting from applying the method of Brox et al. [8] to a pair of frames in the American eel video sequence used to develop this work.

2.3.3 Translating information from image domain to computational domain

A key issue in developing a purely Eulerian level set-based approach to using images/video in flow computations is the need to translate information processed in the image (i.e., pixelized) domain onto the computational (adaptively refined flow mesh) domain. This issue assumes two forms: temporal interpolation by way of image morphing and spatial interpolation onto the flow mesh.

2.3.3.1 Temporal interpolation

Temporal interpolation between image frames addresses the need to obtain the intermediate configurations of an object that are not captured directly in the imaging sequence, so that its variations in position may be rendered with the temporal density required to couple it with a simulated fluid flow environment. For instance, if a fluid flow of maximum velocity $U = 1$ is simulated on a uniform mesh with a grid spacing of $\Delta x = 0.001$, the Courant–Friedrichs–Lewy (CFL) condition requires that

$$CFL = \frac{U\Delta t}{\Delta x} \leq 1. \tag{39}$$

Thus, using the same example, two image frames separated by a time step of $\Delta T_{\text{Image}} = 1$ would need to be divided into 1000 smaller steps in order for the object motion to match smoothly with the fluid motion.

In practice, the maximum flow velocity is not constant throughout a CFD simulation (nor is the minimum grid spacing, if adaptive mesh refinement is employed), so time step sizes are continuously being updated by checking whether the CFL condition is being satisfied everywhere on the flow mesh. For this reason, it is necessary to keep track of the *running time* in the CFD solution process and correlate it to the frame rate in the image sequence being used to construct the moving boundaries embedded in the fluid mesh. To elucidate this point, let ΔT_{Image} represent a local image timescale, which has an initial time of $T_{\text{Image},o} = 0$ for the source frame and a final time of $T_{\text{Image},f} = 1$ for the target frame of an ordered pair in a sequence. Let ΔT_{flow} denote a corresponding flow timescale, having a start time t_{f1} during which an object's location is given by the source frame, and an end time t_{f2} during which an object's location matches that of the target frame. Any intermediate time t_f^* between t_{f1} and t_{f2} can then be correlated to a fraction of ΔT_{Image}, and this fraction can be used to determine the deformation required to correctly update the object's position (Fig. 2).

2.3.3.2 Image advection using optical flow

It was demonstrated in the previous section that the nonlinear optical flow method of Brox et al. offers an Eulerian description of objects moving through image frame sequences that promises a relatively high degree of accuracy. In our own tests, detailed in the next section, linear motion was captured quite well, even when the geometry was complex and the displacements between image frames were large. Errors for rotational motion were found to be higher, but those errors were often limited to regions located away from object boundaries. As such, it was hoped that an accurate description of motion could still be achieved in real image sequences, provided that they have adequate spatio-temporal resolutions to justifiably approximate their motions as linear within local pixel neighborhoods near object boundaries. It was thus also hoped that supplying an image or a segmented level set boundary with its corresponding optical flow vector field would be sufficient to move an object to its correct target location matching the next frame of the sequence. However, in practice this did not turn out to be the case.

The iterative process followed in the method of Brox et al. [8] is nonlinear, and pixels undergoing displacement do not, in general, follow the straight path that is represented by the final result in reaching their target destination. Thus, when the image brightness is advected in the usual sense by projecting the optical flow vectors onto the intensity gradient at each pixel location with the assumption of brightness constancy, that is by solving the advection problem,

$$\frac{\partial I}{\partial t} + \boldsymbol{u} \cdot \nabla I = 0 \tag{40}$$

or

$$\frac{\partial I}{\partial t} = -u\frac{\partial I}{\partial x} - v\frac{\partial I}{\partial y}, \tag{41}$$

the same end result is not achieved. It can easily be seen that this way of moving brightness patterns becomes particularly inaccurate in regions where optical flow vectors are oriented normal to the brightness gradient

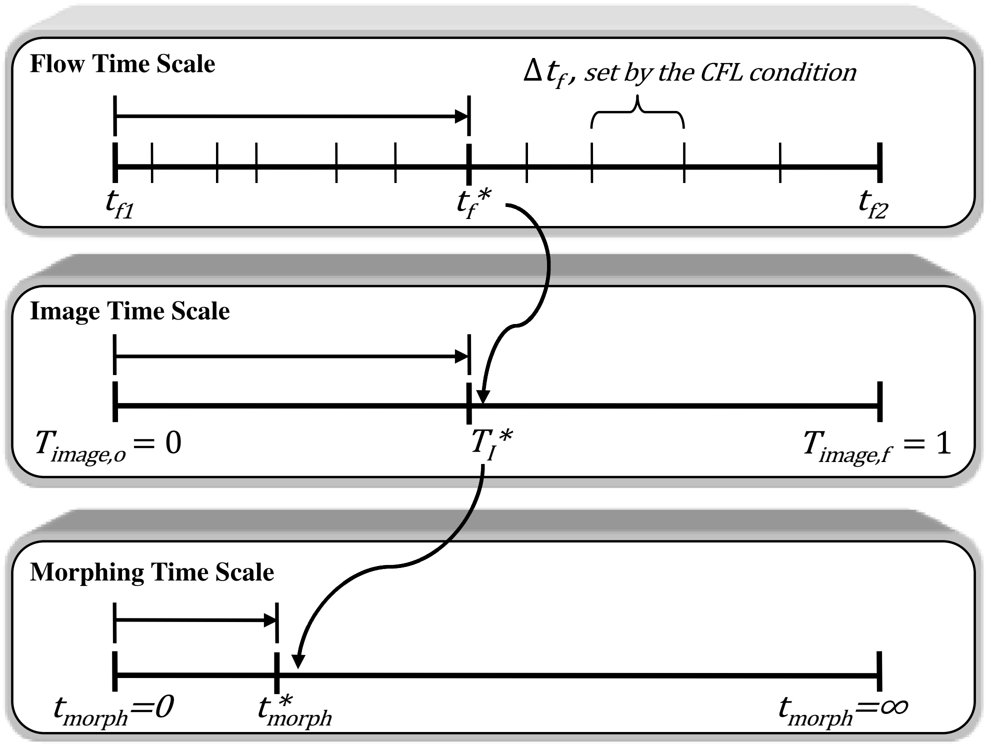

Fig. 2 Image-based modeling process involves three timescales. The flow timescale (*top*) is defined by the physics of the problem, e.g., Strouhal number, and is divided into flow time step sizes that are determined by satisfying the CFL condition. The image timescale (*middle*) is defined by the interval between image frames, within which the flow timescale determines the fraction of progress between source and target frames. This in turn determines a morphing timescale (*bottom*), during which elastic deformation moves the object of interest the desired fraction between source and target locations

(tangent to the object's boundary): $\boldsymbol{u} \cdot \nabla I \to 0$ in this situation, resulting in little or no motion of the imaged objects there regardless of the magnitude of the optical flow vector $|\boldsymbol{u}|$.

Attempts to move the level set representation of imaged objects, rather than the brightness patterns directly, under the influence of optical flow vectors failed for precisely the same reason. According to the level set equation, the motion of a level set occurs in a direction normal to itself driven by some speed function F:

$$\frac{\partial \phi}{\partial t} + F\,|\nabla \phi| = 0. \tag{42}$$

In this case, F is comprised of the optical flow vectors projected onto the normal vectors of the level set field surrounding an object's boundary or $F = \boldsymbol{u} \cdot \boldsymbol{n}$, giving

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{n}\,|\nabla \phi| = 0. \tag{43}$$

Rewriting the normal vector in terms of the gradient of the level set field,

$$\boldsymbol{n} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{44}$$

the level set equation becomes

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \frac{\nabla \phi}{|\nabla \phi|}\,|\nabla \phi| = 0, \tag{45}$$

or

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0. \tag{46}$$

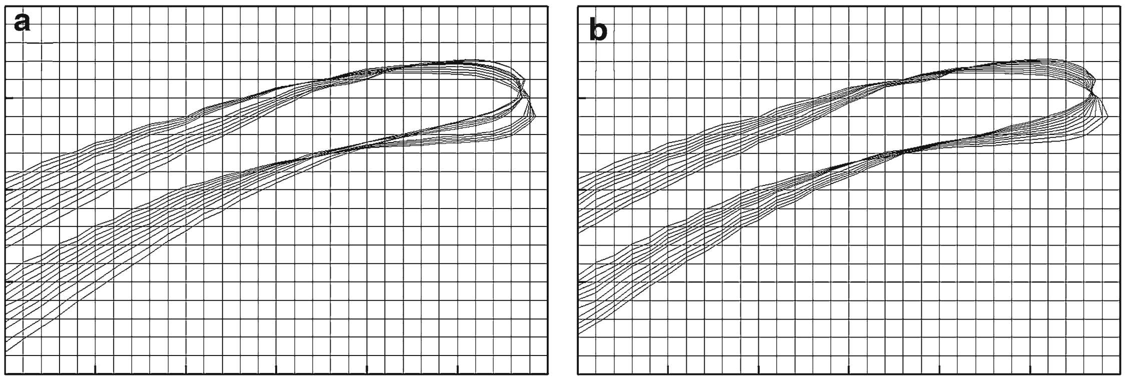

Fig. 3 Filling the gaps between image frames: **a** initial attempts to advect the level set using optical flow information resulted in jumps between frames in the tail region; **b** elastic morphing provided smooth, consistent motion throughout the gaps between image frames

This leaves us in exactly the same predicament we were in using the brightness advection equation (Eq. 40), because the gradient of a level set field about an object boundary is oriented the same way as the pixel intensity gradients marking the edge of the object. Thus, attempts to advect a level set field using optical flow vectors produced the same unfavorable results as similar attempts to advect the intensity field.

Figure 3 helps to elucidate the problems encountered during optical flow-based advection. Figure 3a shows five intermediate zero-level set positions describing the contour of the swimming American eel, taken at regular intervals for each of two optical flow field solutions that are temporally adjacent (9 positions total). It can be seen that along the part of the eel's body away from the tail near the left side of the figure, the level set contours are spaced regularly, indicating near constant motion there—the desired result. In this part of the image, the optical flow vectors are predominantly normal to the interface, pointing in the direction of the level set gradient and thus in the direction of the image brightness gradient. However, at the tip of the eel's tail, velocity vectors are oriented tangentially to the level set normal vectors, resulting in an under-prediction of motion there. The result is that the tail experiences a jump in position between the final advection step of one image frame pair and the first advection step of the subsequent image pair.

Despite the fact that the optical flow vectors produced by the method of Brox et al. possess a high degree of accuracy, these problems limit their usefulness for temporal interpolation; another method of moving interfaces through intermediate steps between image frames is needed. This is the subject of the next section.

2.3.3.3 Image morphing

Image morphing is defined as the process of constructing a smooth, natural-looking sequence of images between an ordered image pair [29], and consists of two steps. The first is warping, which typically involves a nonlinear coordinate transformation that aligns user-defined landmarks while imposing regularity or smoothness constraints on the coordinates that lie between landmarks [29]. The second step is blending, which fills in the differences in detail and color not captured during the warping process, and is typically based on pixel color/intensity interpolation. Applying the two steps incrementally results in a smooth transition from one image to the other that includes both the shape (warping) and the intensity/color (blending) [29].

In [29], Whitaker proposes an approach to image blending that is based on the shapes of level sets in the input images, making it well suited for the level set-based formulations recurring throughout this work. In this approach, a distance metric is constructed that penalizes images based on differences in the shapes of their level sets and thus acts in similar fashion to an elastic force that deforms the shapes of image objects as they are evolved between frames. This results in a set of transition images that represent intermediate shapes naturally through the blending process. Treating an image as a continuous function $F : \Omega \mapsto I$, where $\Omega \subset \mathbb{R}^2$ is the image domain and $I \subset \mathbb{R}$ is the set of intensity values contained in the image, $F(x, y)$ and $G(x, y)$ represent a source and a target image, respectively, with $(x, y) \in \Omega$. The image blending strategy involves constructing a family of images, indexed by α and starting with the source image $F(x, y)$ at $\alpha = 0$, which progressively appear more like the target image $G(x, y)$ as α increases [29]. Regarding the functions F and G as representations of points in some higher-dimensional function space, a linear interpolation is simply a straight-line path through the function space between F and G, parameterized by α. The path from F to G may be constructed by gradient descent over parameter t on a distance function that approaches zero as $F \to G$.

Alternatively, a path may be constructed on which F and G are both allowed to progress toward each other simultaneously with respect to t. The paths of these functions through the function space can be defined as $f(x, y, t)$ and $g(x, y, t)$, where $f(x, y, 0) = F(x, y)$ and $g(x, y, 0) = G(x, y)$, and the point where they meet in function space is the transition image between F and G. In this way, a blend, $b(x, y, \alpha)$, is obtained by re-parameterizing the solutions of f forward in time and the solutions of g backward in time to produce a sequence from F to G with respect to the parameter α.

Clearly, different metrics will produce different paths through the function space occupied by F to G and will thus lead to different blends between image frames. A metric typically employed in image analysis is the L_2 norm:

$$\varepsilon(t) = \frac{1}{2}\int_{\Omega} (f(x, y, t) - g(x, y, t))^2 \,\mathrm{d}x\mathrm{d}y. \tag{47}$$

Setting the temporal derivatives of f and g to the negative of the first variation of Eq. 47 gives a system of gradient descent equations over time t,

$$\frac{\partial f(x, y, t)}{\partial t} = g(x, y, t) - f(x, y, t) \tag{48}$$

$$\frac{\partial g(x, y, t)}{\partial t} = f(x, y, t) - g(x, y, t), \tag{49}$$

where $f(x, y, 0) = F(x, y)$ and $g(x, y, 0) = G(x, y)$. Since Eqs. 48 and 49 only contain derivatives in t, x and y may be regarded as parameters and the equations may be treated as a coupled pair of ordinary differential equations at each point in the domain, with the solution

$$f(x, y, t) = \frac{1}{2}(G(x, y) + F(x, y)) + \frac{1}{2}(F(x, y) - G(x, y))\,\mathrm{e}^{-2t} \tag{50}$$

$$g(x, y, t) = \frac{1}{2}(G(x, y) + F(x, y)) + \frac{1}{2}(G(x, y) - F(x, y))\,\mathrm{e}^{-2t}. \tag{51}$$

The steady-state solution to this pair of equations, $f(x, y, t) = g(x, y, t) = \frac{1}{2}(G(x, y) + F(x, y))$, is the transition image between F and G.

Because Eqs. 50 and 51 represent an exponential decay toward the transition image, the final blend is obtained by

$$b(x, y, \alpha) = \begin{cases} f(x, y, -\ln(1 - 2\alpha)) & \text{for} \quad 0 \le \alpha \le \frac{1}{2} \\ g(x, y, -\ln(2\alpha - 1)) & \text{for} \quad \frac{1}{2} \le \alpha \le 1 \end{cases}. \tag{52}$$

Here, it is noted that such a distance metric formulation on the image domain is limited in that it only considers single-pixel comparisons between images and that it is sensitive to nonlinear transformations acting on the intensity functions f and g. Therefore, it is proposed in [29] to treat the image domain as a set of isocontours (level sets) rather than a collection of individual pixels. This allows for inter-pixel relationships to be considered in the difference metric when blending an image from one frame to the next. Defining the kth level set of path f through function space from F to G as

$$\mathcal{L}_k = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \middle| \, f(x, y) = k \right\}, \tag{53}$$

a point's location can be determined to be either “inside” or “outside” of the level set k by applying the Heaviside function to the values of $f : \mathcal{H}(f(x, y) - k)$ (the definition of what is considered “inside” versus “outside” of a level set is arbitrary, as long as the definition is applied consistently).

Using this new level set definition of the image domain, a distance metric may be constructed as before, but now based upon the differences between regions contained within level sets on the two images (f and g), so that the metric's potential is proportional to the area of level set shapes in one image that are not in the other. Defining functions of f and g that are negative inside the kth level set and positive outside as

$$\mathcal{D}_k[f] = \frac{1}{2}(\mathcal{H}(k - f) - \mathcal{H}(f - k)) \tag{54}$$

$$\mathcal{D}_k[g] = \frac{1}{2}(\mathcal{H}(k-g) - \mathcal{H}(g-k)), \tag{55}$$

the overall distance metric for level set k is given by

$$\begin{aligned}\varepsilon(t) &= \frac{1}{2}\int_\Omega (\mathcal{D}_k[f(t)] - \mathcal{D}_k[g(t)])^2 \,\mathrm{d}x\,\mathrm{d}y \\ &= \frac{1}{4}\int_\Omega ([\mathcal{H}(k-f(t)) - \mathcal{H}(f(t)-k)] - [\mathcal{H}(k-g(t)) - \mathcal{H}(g(t)-k)])^2 \,\mathrm{d}x\,\mathrm{d}y.\end{aligned} \tag{56}$$

Minimizing the metric with respect to a single level set k leads to a pair of Euler–Lagrange equations

$$-d\varepsilon_{k,f} = \frac{1}{2}[\mathcal{H}(k-g)\delta(k-f) + \mathcal{H}(k-g)\delta(f-k) - \mathcal{H}(g-k)\delta(k-f) - \mathcal{H}(g-k)\delta(f-k)] \tag{57}$$

$$-d\varepsilon_{k,g} = \frac{1}{2}[\mathcal{H}(k-f)\delta(k-g) + \mathcal{H}(k-f)\delta(g-k) - \mathcal{H}(f-k)\delta(k-g) - \mathcal{H}(f-k)\delta(g-k)], \tag{58}$$

in which δ is the Dirac delta function (the derivative of $\mathcal{H}$), and $(\mathcal{H}(x) - \mathcal{H}(-x))\delta(x) = 0$. Because $\delta(-x) = \delta(x)$, Eqs. 57 and 58 reduce to

$$-d\varepsilon_{k,f} = \frac{g-k}{|g-k|}\delta(f-k) \tag{59}$$

$$-d\varepsilon_{k,g} = \frac{f-k}{|f-k|}\delta(g-k). \tag{60}$$

Since the δ functional performs a wave front propagation of the kth level set of f, while the gradient magnitude of f gives the same result on all level sets of f simultaneously, the Dirac delta term $\delta(f-k)$ in Eq. 59 can be remapped to $|\nabla f|$ whenever $f = k$ (the same holds true for g in Eq. 60). Thus, performing gradient descent on f and g and parameterizing the sequence of images by t give a pair of differential equations that are the level set equivalent of Eqs. 48 and 49:

$$\frac{\partial f}{\partial t} = \frac{g-f}{\left(\epsilon^2 + (g-f)^2\right)^{1/2}}|\nabla f| \tag{61}$$

$$\frac{\partial g}{\partial t} = \frac{f-g}{\left(\epsilon^2 + (f-g)^2\right)^{1/2}}|\nabla g|, \tag{62}$$

where ϵ is a small constant (set to 10^{-4} in [29]) provided to safeguard against division by zero in the limiting case where $f = g$.

In Whitaker's approach to morphing [29], the resulting distance metric equation (Eq. 47) is solved using gradient descent to give a pair of coupled ordinary differential equations (Eqs. 48, 49) and then resampled from the time parameter t to the transition parameter α. This resampling equation (Eq. 52) is constructed to ensure that the sum of absolute values of image differences will decrease at a constant rate with respect to α, giving a blend that appears to change consistently as α is varied from 0 to 1. This type of behavior may be replicated in the level set formulation of image blending by adjusting the sampling rate to produce a constant change in some image metric $\mathcal{D}(t)$, thereby giving the intervals in t at which to take "snapshots" in order to produce the desired sequence of blended images. Due to its straightforward implementation and the qualitatively good results it is able to consistently give, Whitaker proposed a root-mean-squared metric [29],

$$\mathcal{D}(t) = \left[\int_\Omega (f-g)^2 \,\mathrm{d}x\mathrm{d}y\right]^{1/2}, \tag{63}$$

with the discrete form

$$\mathcal{D}(t) = \sum_{i=1}^{n}\sum_{j=1}^{m} ABS\left(F_{i,j} - G_{i,j}\right) \tag{64}$$

to be used during implementation.

The algorithm given to produce n transition images between frame F and frame G is thus summarized as follows:

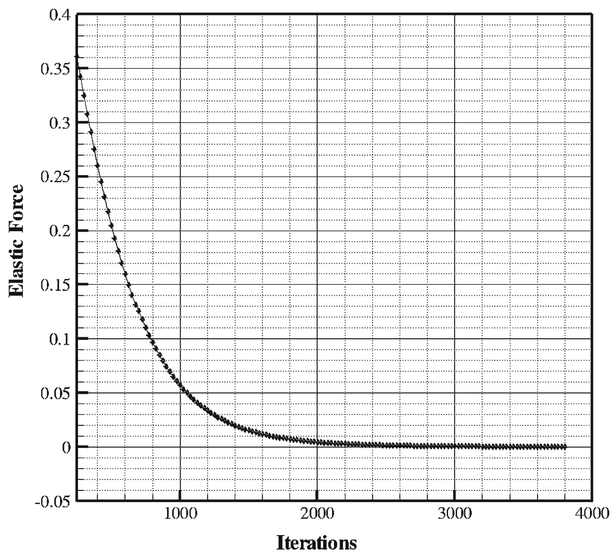

Fig. 4 Exponential decay of the elastic force through the morphing process requires sampling at exponentially increasing numbers of time intervals in order to produce a consistent motion

(1) Compute the difference metric on the source and target images to give

$$\mathcal{D}(0) = \left[\int_{\Omega} (F - G)^2 \, dx dy\right]^{1/2}. \tag{65}$$

(2) For each of the k transition images out of n total transition images between image frames, solve the level set blending equations (Eqs. 61, 62) with a forward differencing or Adams–Bashforth scheme until a time t is reached that satisfies

$$\mathcal{D}(t) = \frac{n-k}{n}\mathcal{D}(0). \tag{66}$$

In this way, image morphing proceeds consistently through the intermediate steps between image frames.

2.3.3.4 The image morphing algorithm

As shown in Sect. 2.3.3.3, the elastic force chosen to produce morphing is not linear, but rather decays exponentially as $\mathcal{D}(x, y) \to 0$ (Fig. 4). Morphing must therefore take place on a third timescale ΔT_{morph}, ranging between $t_{\text{morph}} = 0$, when $\alpha = 0$ and $k_{i,j}(0) = f_{i,j}$, and $t_{\text{morph}} = \infty$, when $\alpha = 1$ and $k_{i,j}(1) = g_{i,j}$. In order to deal with the disparity between the morphing timescale and the image timescale, the root-mean-squared metric proposed by Whitaker was used to obtain a sampling rate in which morphed image objects would match up with their correct locations throughout progression through the fluid flow timescale (Fig. 2). The algorithm is summarized as follows:

(1) Calculate the initial image metric value $\mathcal{D}(0) = [\int_{\Omega} (F - G)^2 \mathrm{d}x\mathrm{d}y]^{1/2}$.
(2) For any instant t_f^* falling between flow time intervals t_{f1} and t_{f2}, find the corresponding time fraction T_I^* between image frame times $T_{\text{Image},o} = 0$ and $T_{Image,f} = 1$ using an appropriate scaling relationship between t_f and T_I, i.e.,

$$T_I^* = \frac{t_f^* - t_{f1}}{\Delta T_{\text{flow}}}. \tag{67}$$

(3) Evolve the morphing process until the appropriate morphing pseudotime t_{morph}^* is reached by checking the metric value

$$\mathcal{D}\left(t_{\text{morph}}^*\right) = D(0) + \frac{\mathcal{D}(0) - \mathcal{D}(\infty)}{T_{\text{Image},o} - T_{\text{Image},f}}\left(T_I^* - T_{\text{Image},o}\right). \tag{68}$$

Because $\mathcal{D}(\infty) = 0$ when morphing is complete, and $T_{\text{Image},o} = 0$ and $T_{\text{Image},f} = 1$, Eq. 68 reduces to

$$\mathcal{D}\left(t_{\text{morph}}^*\right) = \mathcal{D}(0)\left(1 - T_I^*\right). \tag{69}$$

The morphing process is evolved until Eq. 69 is satisfied, thus giving morphed images at regular time intervals corresponding to the intervals at which flow field solutions are computed.

Figure 3b shows five zero-level set configurations obtained by morphing for each of two image frame pairs (9 total configurations). Comparing to Fig. 3a, it can be seen that the jumps in image position between the final morph of one frame and the initial state of the next have been eliminated; a consistent motion of the boundary has been achieved through use of the elastic force model and an appropriate sampling rate. However, like any method, the elastic force model chosen for image morphing is not perfect. The primary difficulty found with respect to our purposes involves the morphing path rather than the final result. Careful examination reveals that the eel's tail does not take a linear trajectory when morphing from one frame to the next, as one would expect based on the image information, but rather follows a trajectory in which the eel's overall body length appears to slightly shorten and then lengthen again. This happens because the distance metric has a zero value at nodes where the initial and final zero-level set contours cross; the contours are not separated there, and so there is no driving potential for motion. The effect is that contour motion can occur around this zero-potential point, but not through it, so a curved path around it is found during the iterative morphing process. This problem will always exist where zero-level contours of initial and target fields overlap, but can be minimized with greater temporal resolution to ensure that object boundaries are not largely displaced between frames.

2.3.3.5 Spatial Interpolation onto the Flow Mesh

The morphing algorithm just outlined provides a temporal interpolation from a coarsely sampled image timescale to a finely sampled fluid flow timescale. Similarly, modeled objects must be spatially interpolated from a coarse image domain consisting of pixels to a fine flow domain consisting of grid cells that may be locally refined to capture flow phenomena of interest. Thus, a key process in generating a purely Eulerian CFD model involves mapping the level set fields obtained by image analysis onto the flow domain.

This mapping is performed in two steps:

(1) First, the image pixel locations are converted to real (x, y) pairs (mapping the image domain $\Omega \longmapsto \mathbb{R}^2$), so that imaged objects can be easily scaled and placed wherever desired on the corresponding flow mesh. For example, flow domains in this work are typically constructed with dimensions of $\mathcal{O}(1)$ or $\mathcal{O}(10)$, having grid spacing $\Delta x \ll 1$ and containing objects that are of $\mathcal{O}(1)$ in size, in order to nondimensionalize the fluid flow problems and to facilitate the variation of important parameters like the Reynolds number. Thus, an imaged object with dimensions of, for example, 1000 pixels in length and 200 pixels in height could be converted to a modeled object of 1.0 unit length and 0.2 unit height by multiplying each (x, y) by a scaling parameter s. In addition, x- and y-shifting constants d_x and d_y can be added to position the zero-level set interface within the flow domain as desired. The result is a scaled, shifted version of each pixel location $\hat{x}_{i,j} = sx_{i,j} + d_{i,j}$.

(2) With each of the image pixels defined as a point with a scaled and shifted spatial address, the *flow mesh* points are swept over to examine whether they are within the bounds of $\hat{x}_{i,j}$, i.e., between $\hat{x}_{min}$ and $\hat{x}_{max}$ in the horizontal direction, and between $\hat{y}_{min}$ and $\hat{y}_{max}$ in the vertical direction. If they are, then their 4 nearest neighbors on the corresponding scaled and shifted image domain are used to interpolate the image level set value onto the flow mesh via bilinear interpolation. It is important to note here that when an image domain is rescaled and shifted for mapping to the fluid mesh, Δx_{image} is still not the same as Δx_{flow} in general. Thus, when mapping level set values from an image domain onto a fluid domain, the level set values must also be scaled appropriately. In this work, the pixel size Δx_{image} is set to 1.0 for convenience, allowing us to simply multiply level set values interpolated from the image domain by Δx_{flow} to get the correct scaling. A narrow-band approach [21] is chosen for efficiency, such that any flow mesh cell containing a level set magnitude $|\phi| > 6\Delta x_{\text{flow}}$ is simply set to $\phi = 6\Delta x_{\text{flow}} \text{sign}(\phi)$.

Although the elastic force model was chosen for image morphing due to the limitations of optical flow-based advection, optical flow vectors are still used to set boundary conditions on the zero-level set contour in moving boundary flow simulation problems; indeed, optical flow may be the only way to do this in a Eulerian setting without relying on the use of surface points. So, the optical flow vector field is interpolated onto the flow mesh from the image domain just as the level set field values are (Fig. 5). Since optical flow vectors are computed on image sequences under the assumption that pixels are unit size and frames are separated by some arbitrary unit time, the resultant vector field is given in terms of $pixels/frame$. The optical flow velocity field is mapped from the image domain to the flow mesh in just the same manner as the level set field, scaling by Δx_{flow} to get the correct velocity magnitude. Furthermore, varying the image frame rate ΔT_{image} allows

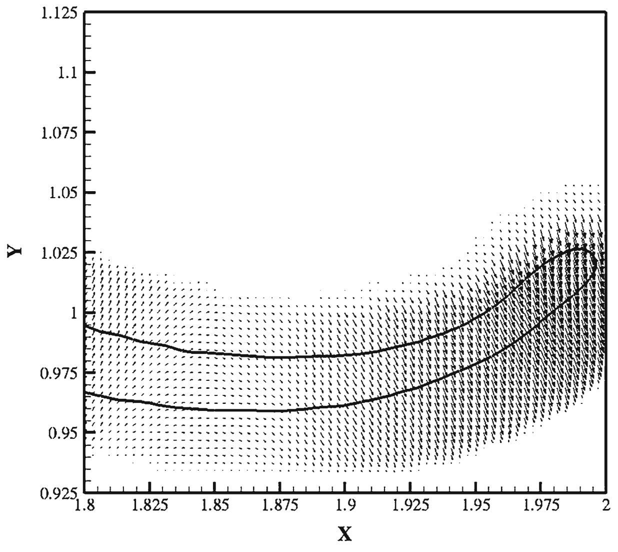

Fig. 5 Optical flow vectors computed at the object boundary are extended normally into the level set narrow band [21] to supply boundary velocity conditions during flow computations

for the evaluation of different physical conditions. For example, the swimming eel's Strouhal number can be doubled by simply halving the defined frame rate ΔT_{image}. Since the optical flow field is piecewise constant between image frames, a 3-frame strategy is used to create a piecewise linear optical flow field and thereby avert the discontinuous jumps in velocity otherwise produced by a piecewise constant profile:

$$\boldsymbol{u} = \boldsymbol{u}_{1\to 2} + \boldsymbol{u}_{2\to 3} T^{*}_{\text{image}}, \tag{70}$$

where $\boldsymbol{u}_{1\to 2}$ and $\boldsymbol{u}_{2\to 3}$ represent optical flow velocity vectors solved between one ordered frame pair and the next, respectively, and T^{*}_{image} is the (dimensionless) fraction of time elapsed between frames 1 and 2. Over-relaxation may also be used to smooth the transition between piecewise linear segments.

It is noted that an image must be morphed and re-mapped to the corresponding flow domain after each fluid time step during CFD simulations. While this accounts for little of the total run time compared with solving for the optical flow field, or certainly compared to solving for the fluid flow field, it would still be computationally wasteful to sweep over the entire flow domain every single time step, find corresponding image domain locations and then update the values everywhere (even if they're "updated" to the same value). To eliminate this inefficiency, the flow solver code has a built-in memory structure containing only points that lie within the narrow-band level set, storing their spatial locations within the flow domain along with their level set values. Mapping the image level set and interface velocity from the scaled and shifted image domain to the flow domain need only takes place within the narrow band where such information is necessary for setting boundary conditions. By sweeping over these level set "tube" points rather than the entire flow domain, the vast majority of flow grid points can often be eliminated from the search and update process.

To summarize the entire image-to-computation process, an outline of the algorithm is provided in Fig. 6.

3 Results

3.1 Testing optical flow

The optical flow method of Brox et al. was tested on a test image prescribed with three types of motion before applying it to the real image sequence (Fig. 7). The image was of dimension 128 × 128 pixels and consisted of a 7-pointed star shape defined in [38], which was given a numerical Heaviside function contour to provide a smooth but steep edge gradient. This shape was chosen for several reasons: It features regions of high curvature near its edges and center, it possesses nonuniform gradients, and its radial nature allows for combinations of linear and angular motion to be imposed. As suggested in [8], calculating optical flow fields in a multi-resolution manner on grids of increasing levels of refinement can help reduce the chances of an optical flow solution becoming trapped in a local minimum that does not satisfy global minimization of the

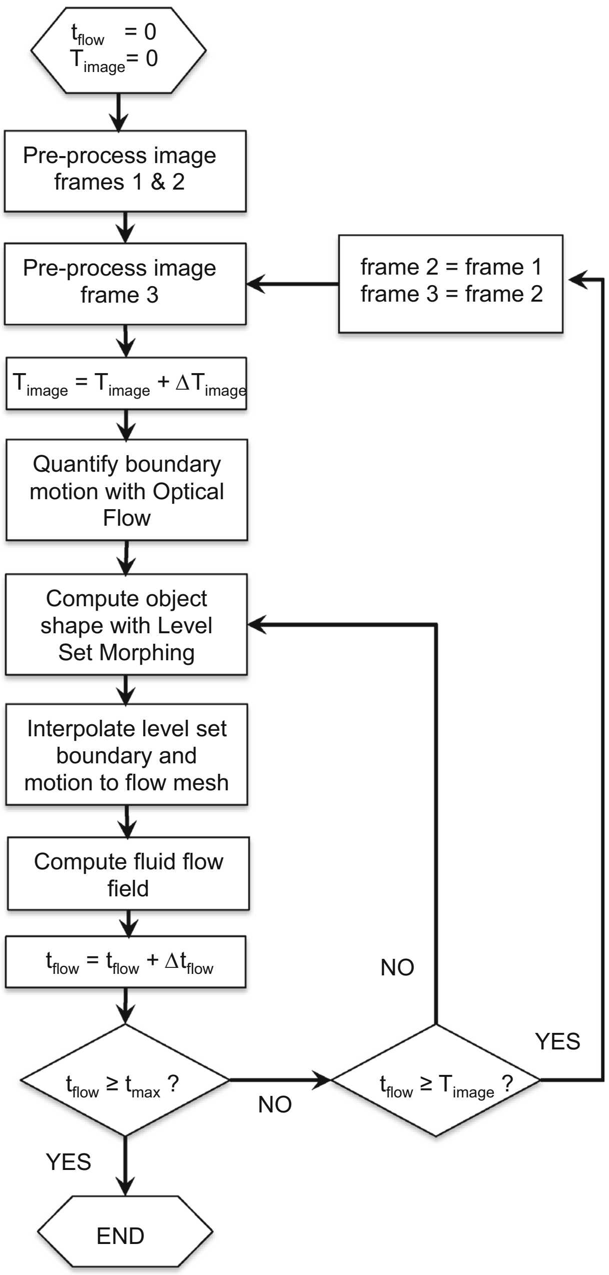
t_{flow} = 0
T_{image} = 0
Pre-process image frames 1 & 2
Pre-process image frame 3
frame 2 = frame 1
frame 3 = frame 2
$T_{image} = T_{image} + \Delta T_{image}$
Quantify boundary motion with Optical Flow
Compute object shape with Level Set Morphing
Interpolate level set boundary and motion to flow mesh
Compute fluid flow field
$t_{flow} = t_{flow} + \Delta t_{flow}$
$t_{flow} \geq t_{max}$?
NO
$t_{flow} \geq T_{image}$?
NO
YES
YES
END

Fig. 6 Summary of the image-to-computation algorithm

constraint equations. Thus, optical flow was solved on three different grid levels of side length 32, 64 and 128 pixels, respectively, with each coarse solution providing the initial conditions interpolated onto the grid at the next level of refinement. In all of the test image cases, the smoothness weight α^2 was set to 10.0, with the gradient weight (the weighting of the nonlinear terms) set to $\gamma = 1.0$.

The first tests were run by imposing a simple translational motion on the shape by setting $U = V = 5.0$ pixels/frame. Figure 8 and Table 1 illustrate the small error given by the method of Brox et al. [8] when solving for translational motion, even with large displacements. Maximum magnitude error was under 2 % and maximum angular error was under 0.5° (these maxima were also restricted to small regions within the domain, with the overall error being under 1 % and 0.02°, respectively). Imposing pure rotational motion of $\dot{\theta} = \frac{\pi}{24}$ rad/frame (Fig. 9) gave substantially larger maximum magnitude errors approaching 50 % and angular errors approaching 6°. However, the error was predominantly located away from the shape's boundaries

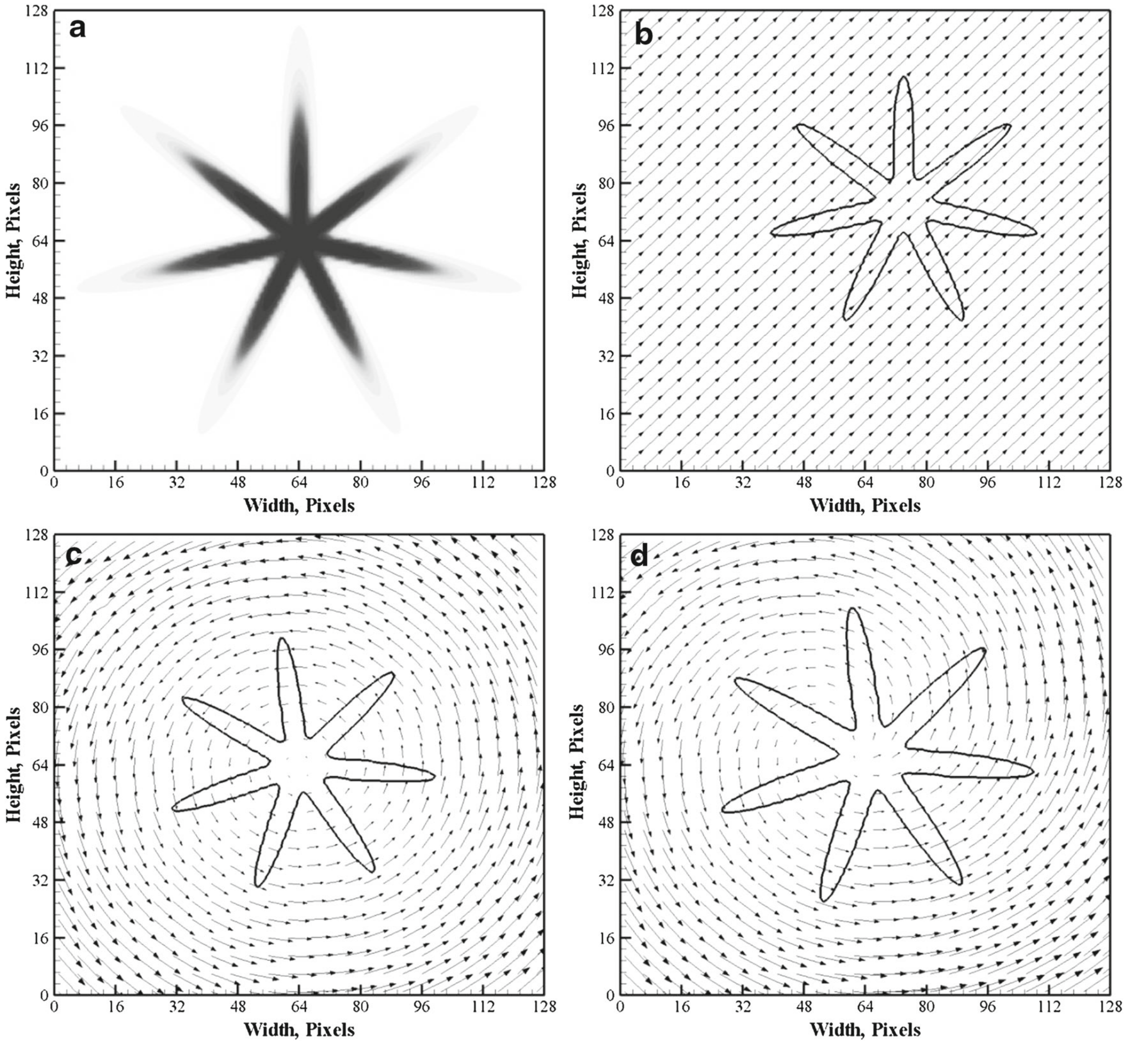

Fig. 7 A 128×128 *star-shaped* Heaviside image was used for optical flow testing (**a**). The shape was prescribed with three different types of motion: **b** translation with $= V = 5.0$ pixels/frame; **c** rotation at a rate of $\dot{\theta} = \pi/24$ rad/frame; and **d** a combination of translation with $U = V = 1.0$ pixel/frame, rotation of $\dot{\theta} = \pi/24$ rad/frame, and radial expansion of $\dot{r} = 1.0$ pixel/frame

(Fig. 9c, d) where the error was considerably smaller on average (4.2 % and 1.03°). Finally, three types of motion were imposed on the Heaviside star shape simultaneously: translation with $U = V = 1.0$ pixel/frame, rotation with $\dot{\theta} = \frac{\pi}{24}$ rad/frame and radial expansion of the shape at a rate of $\dot{r} = 1.0$ pixel/frame. In this case, the Brox method [8] began to give larger errors (Fig. 10) even at the boundaries (11.9 % and 4.2° average). However, it is notable that the majority of this error was concentrated along the left arms of the shape near the shifted center of rotation where velocity magnitudes were low and thus sensitive to error. This is particularly true where the arms converge at the center and thus the radius of curvature of the angular velocity field is small, giving a strongly rotational sense to the flow. These results were considered encouraging given the translational nature of local motions observed in the swimming eel video sequence toward which this work is directed. However, the results do indicate that caution must be exercised if optical flow is to be applied to image sequences exhibiting combinations of significant translational and rotational motion.

3.2 Image-to-computation algorithms applied to simulate swimming of an eel

The prototype image case used for this development project was that of a swimming American eel, modeled from video footage supplied by Dr. Eric Tytell. The eel was imaged while swimming in a water tunnel apparatus

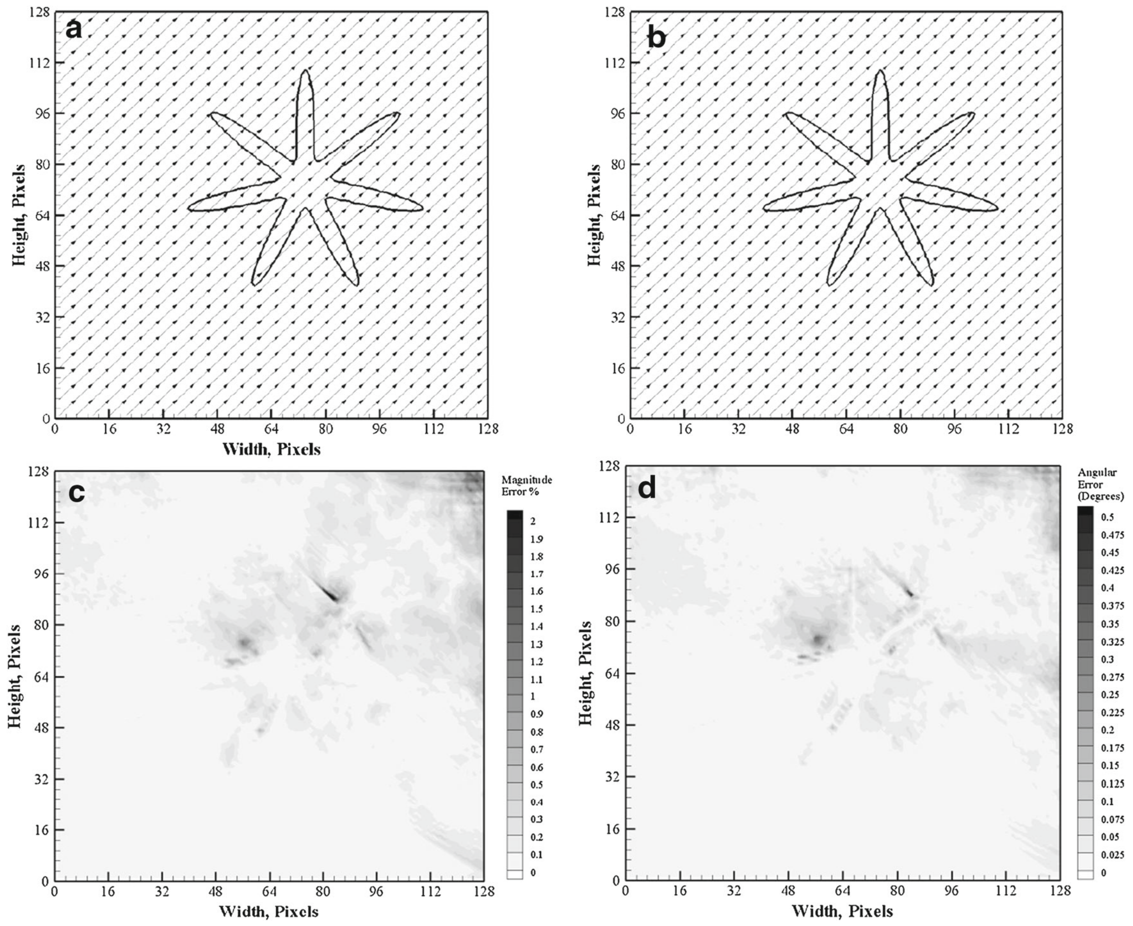

Fig. 8 Optical flow solution of the translating Heaviside star, $= V = 5.0$ pixels/frame: **a** exact vector field; **b** Brox et al. vector field; **c** velocity magnitude error %; **d** velocity angular error in degrees

Table 1 Average velocity magnitude and angular errors for three test image cases: translating motion with $U = V = 5.0$ pixels/frame; rotating motion with $\dot{\theta} = \frac{\pi}{24}$ rad/frame; and a combination of translation with $U = V = 1.0$ pixel/frame, rotation with $\dot{\theta} = \frac{\pi}{24}$ rad/frame, and radial expansion of $\dot{r} = 1.0$ pixel/frame

Imposed motion	Average velocity magnitude error (%)		Average angular error (°)	
	Image domain	Object boundary	Image domain	Object boundary
Translation	0.095	0.118	0.019	0.027
Rotation	24.96	4.165	2.310	1.031
Translation, rotation, expansion	32.45	11.91	5.397	4.202

Errors are averaged both over the entire image and over pixels adjacent to the object boundary

during hydrodynamic studies of its anguilliform swimming motion. The eel's dimensions and boundary conditions for the experimental setup are supplied in [39]. For the current investigation, 36 video frames containing one complete tail beat cycle of the eel's motion were selected for segmentation and then copied nine more times in sequence to produce a total of 360 video frames containing ten identical tail beats. Maximum displacement occurred at the tip of the eel's tail at a rate of ~2 pixels/frame in the y-direction. Image segmentation was performed with active contours [9,31] as described in [34].

In Tytell's work, the eel being studied was measured to be $L = 20$ cm in length and was placed in a water tunnel in which it swam at a steady-state rate of 28 $\text{cm}\,\text{s}^{-1}$, or 1.4 $\text{L}\,\text{s}^{-1}$. Thus, the Reynolds number of the swimming eel based on its body length is roughly 56,000. The Strouhal number of the swimming eel (St $= \frac{fA}{U}$, where f is vortex shedding frequency, A is the characteristic length of the problem—the peak-

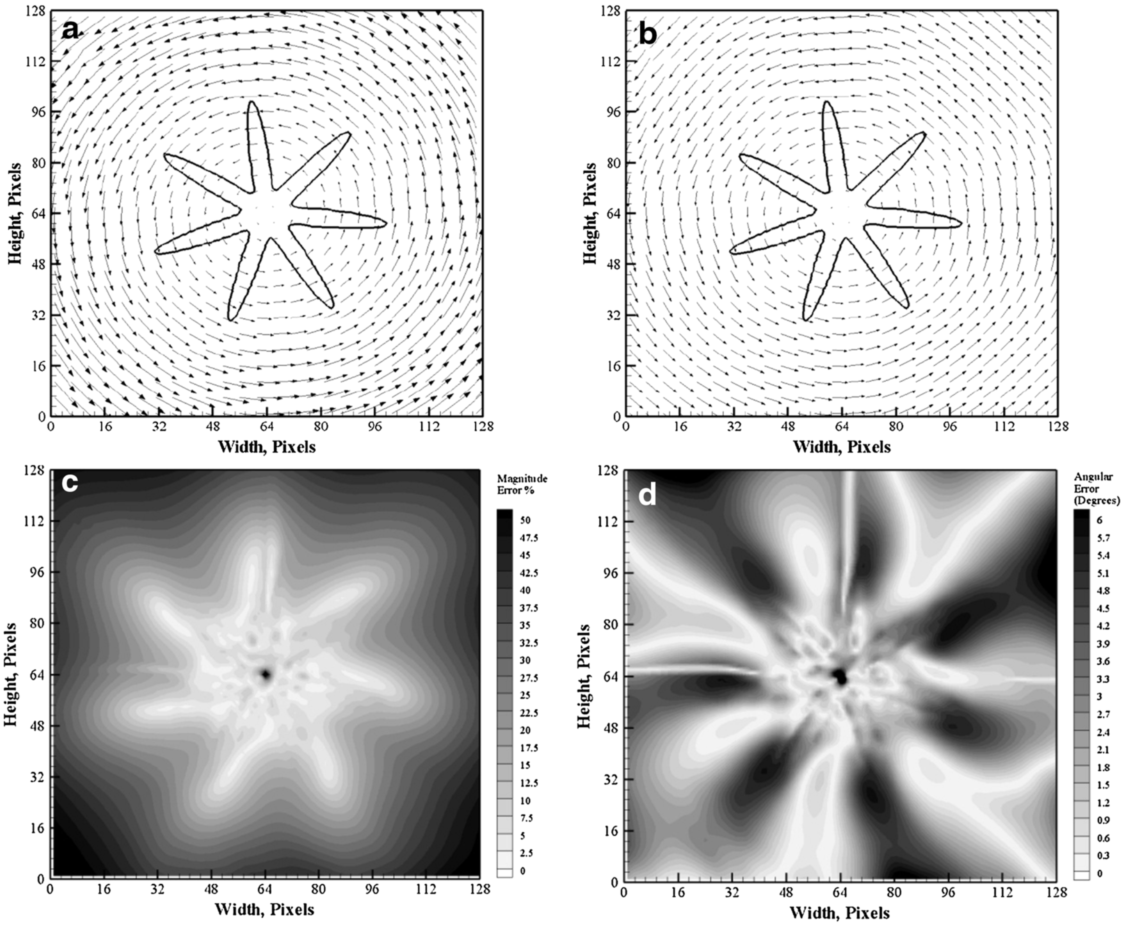

Fig. 9 Optical flow solution of the rotating Heaviside star, $\dot{\theta} = \pi/24$ rad/frame: **a** exact vector field; **b** Brox et al. vector field; **c** velocity magnitude error %; **d** velocity angular error in degrees

to-peak amplitude, or distance traversed by the eel's tail in this case—and U is the free-stream velocity of the fluid environment) was reported to be ~0.3, based upon tail beat amplitude of approximately 7 % of the eel's body length (~0.07 L) and a frequency f of 3.1 Hz. The direct numerical simulation (DNS) nature of the flow solver used for these simulations limited cases to Reynolds numbers that were more than an order of magnitude lower than that produced by the swimming eel in the experimental setup; a Reynolds number of 5000 was studied here. Three different Strouhal numbers, $St = 0.3$, 0.5 and 0.7, were simulated and examined for variations in wake structure and thrust magnitude, for reasons of hydrodynamic interest.

For each case, the eel geometry was scaled to one unit body length and initialized in a domain of dimension 5 units length by 2 units height, with a coarse grid spacing of $\Delta x_1^{\text{flow}} = 0.0125$ and four levels of mesh refinement, giving a minimum grid spacing of $\Delta x_4^{\text{flow}} = 0.0025$. Open flow conditions were approximated by initializing the velocity field with a horizontal (u-velocity) magnitude of $1.0\,\text{L}\,\text{s}^{-1}$, imposing inlet conditions of $u = 1.0\,\text{L}\,\text{s}^{-1}$ on the west side of the domain, specifying the east as an outlet and setting Dirichlet conditions with u-velocity of $1.0\,\text{L}\,\text{s}^{-1}$ on the north and south boundaries. In each case, the eel's position was fixed for 1000 time steps in order to allow for flow field maturation before imposing swimming motion. Fluid density was specified as $\rho = 1.0$, so that the Reynolds number could be assigned simply by setting the dynamic viscosity value μ. Strouhal number variation was accomplished by specifying the time step size ΔT_{image} between each of the 36 video frames.

Results of the three cases simulated (Figs. 11, 12, 13) are encouraging in that they appear to share some of the physical trends revealed on review of Tytell's experimental observations (keeping in mind that we are still limited to a 2-D representation here). In their analysis of wake structures created by the steadily swimming eel using particle image velocimetry (PIV), Tytell et al. reported finding a *2p-type* wake structure dominated

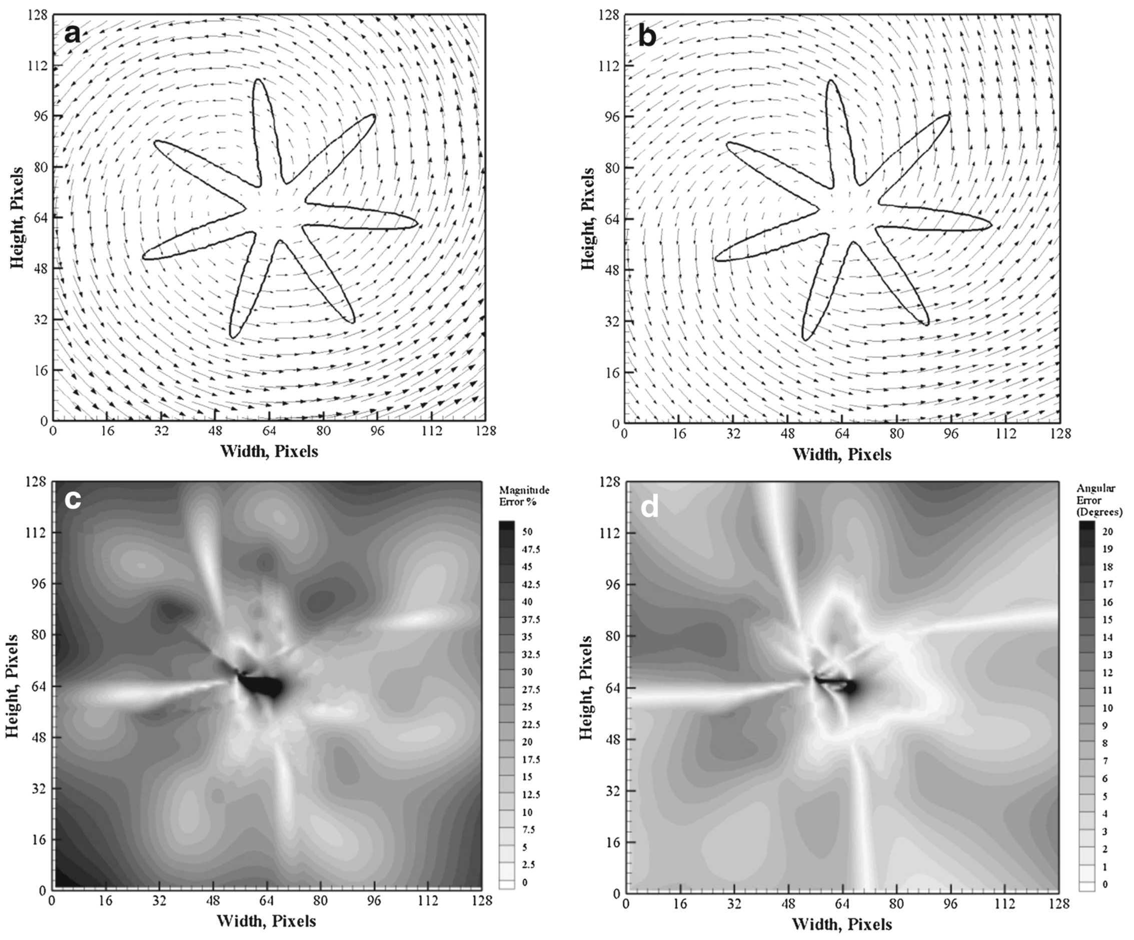

Fig. 10 Optical flow solution of the translating, rotating and radially expanding Heaviside star; $U = V = 1.0$ pixel/frame, $\dot{\theta} = \pi/24$ rad/frame, $\dot{r} = 1.0$ pixel/frame: **a** exact vector field; **b** Brox vector field; **c** velocity magnitude error %; **d** velocity angular error in degrees

by lateral jets, formed by vortex pairs of opposite sign aligned parallel with the direction of flow [39]. They observed that each reversal of the eel's tail direction during swimming began a process of shedding boluses of fluid that were "sucked" into the troughs of waves traveling along the eel's body and accumulated in the boundary layer. The continuous lateral sweeping of the tail resulted in vorticity being advected downstream from the tail's tip in the form of an elongated vortical shear layer, which, once shed from the rear of the body, was unstable in its geometric configuration and rolled up into two vortices—a primary and a secondary vortex—of the same rotational sense. As the tail changed direction, the same thing happened on the opposite side of the body to give another primary and secondary vortex pair, each possessing a sense opposite to the previous vortex pair. In this way, two sets of vortex pairs were formed during each complete tail beat, resulting in two lateral jets of opposite direction alternately staggered in the eel's wake.

Examination of Figs. 11, 12 and 13 reveals similar behaviors exhibited by our CFD results, though perhaps with more subtlety due to the reduced Reynolds number and the fact that this is a 2D simulation of a highly three-dimensional flow. Vorticity contours in Fig. 14 show an elongated shear layer affected by the eel's traversing tail, though the weaker of the two vortices generated by the shear layer's unstably high aspect ratio is diffused shortly after formation in each instance, again likely due in part to the relatively low Reynolds number prescribed in these simulations. In each case, the primary vortices shed from the eel's tail remain close to the center of the wake, in single file *2s* formation rather than alternately staggered to each side. Thus, the wake signature features strong lateral components (Fig. 11) and lacks the clearly defined axial flow acceleration often found in the wakes of carangiform swimmers (Fig. 12). This intuitively follows from kinematic differences between the two types of swimmers; carangiform swimming is marked by a rigid body propelled with an isolated tail, whereas anguilliform swimming involves a large percentage of the body in the

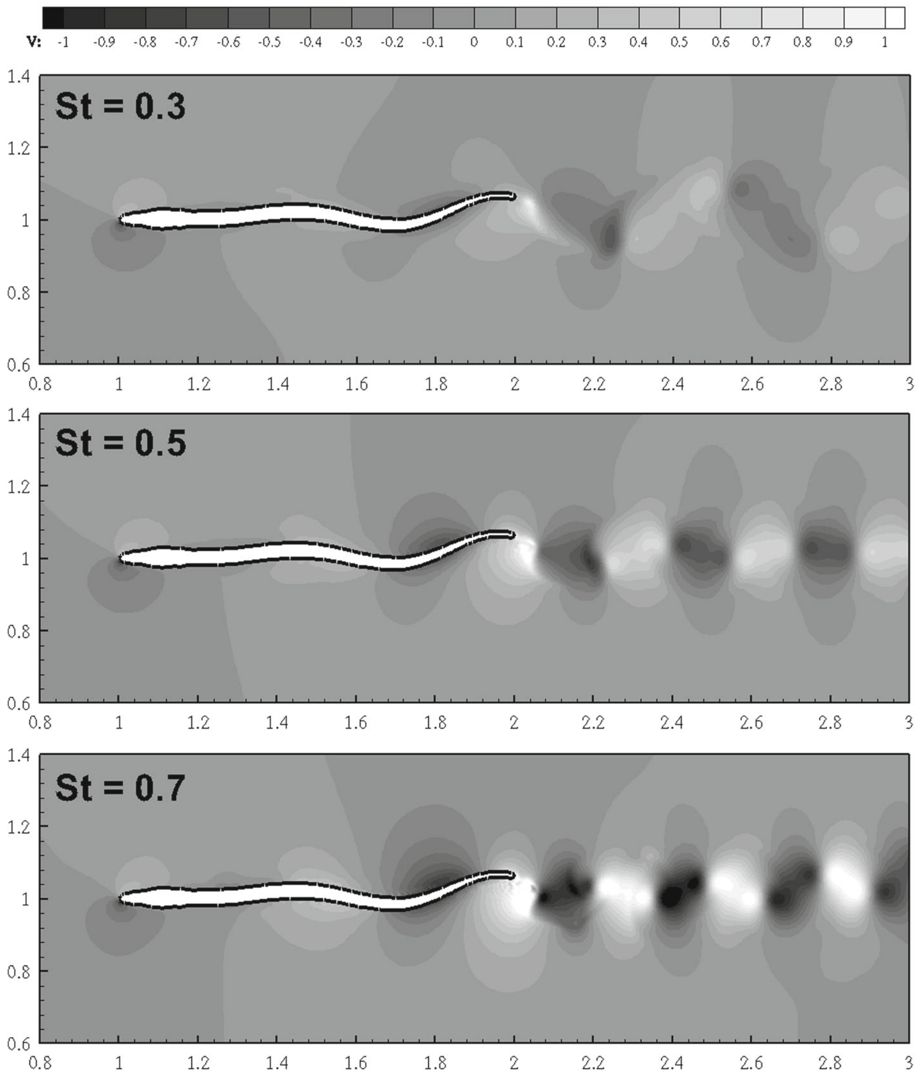

Fig. 11 Lateral component of velocity computed at the three Strouhal numbers evaluated

swimming motion. Thus, it is not unexpected that a significant amount of forward motion should be provided by momentum generated along the length of an eel's body as proposed by Tytell et al. in [39], an idea supported by the apparent axial acceleration experienced by fluid in the troughs of the swimming eel's waveform as it proceeds distally as illustrated in Fig. 13.

3.3 Limitations of the present image-to-computation algorithm

This novel Eulerian method completed for this work dispenses with many of the limitations that are inherent in Lagrangian or semi-Lagrangian approaches by obviating the process of generating meshes to conform to or describe embedded moving objects. However, the current approach comes with limitations of its own. One such limitation lies with the assumptions made regarding motion; in the optical flow calculations, objects are assumed to move along the direction of their intensity gradients. The constraint of smoothness in combination with the nonlinear iterative path taken in the method of Brox et al. [8] gives optical flow vector components that are not constrained to follow image gradients everywhere per se, but there are still problems with assessing tangential boundary motion (i.e., parallel to the imaged boundary and normal to its intensity gradient) correctly. This is particularly true in regions that have little or no curvature. Another limitation of the present Eulerian method lies with the morphing process. The elastic force model used to move a level set contour between

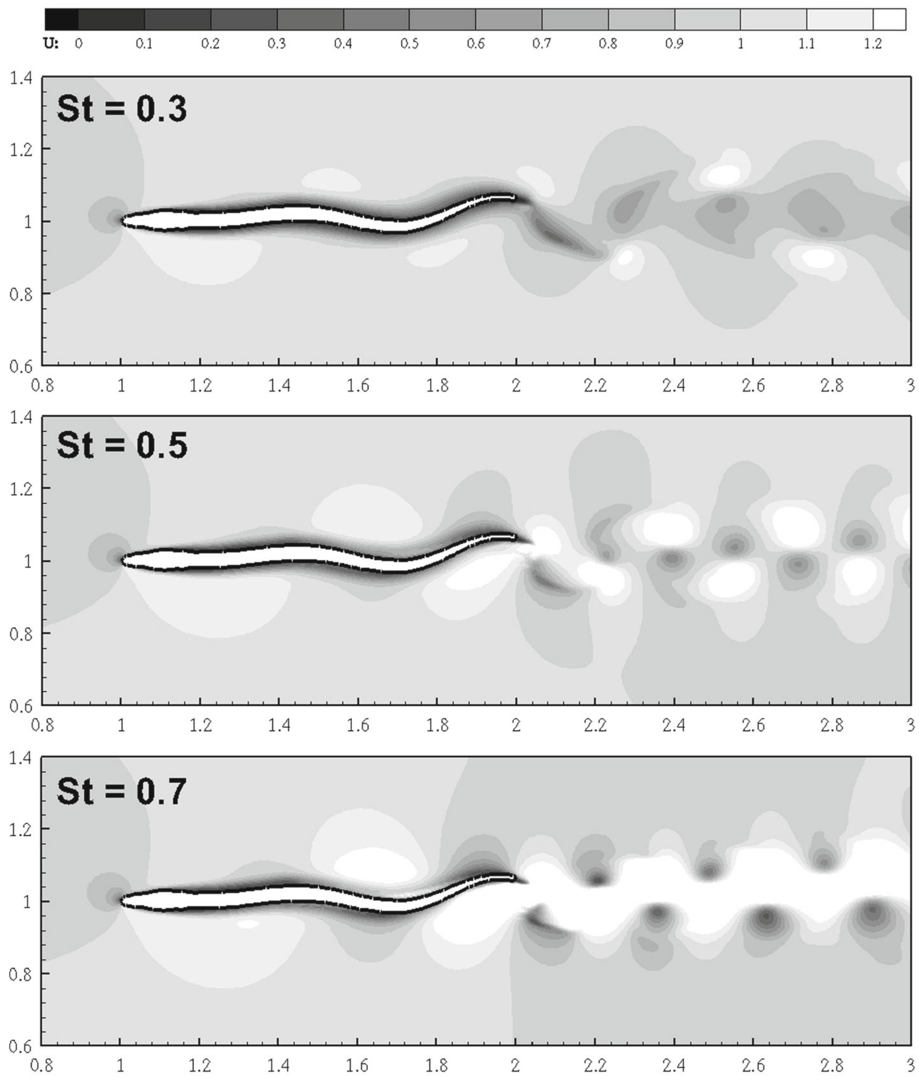

Fig. 12 Axial component of velocity at the three Strouhal numbers evaluated. The eel's anguilliform swimming motion produces a set of vortical structures from which momentum is ejected with a strong lateral component in the wake; thus, a clear thrust generation signature such as that found in the wakes of carangiform swimmers is absent here

frames does not conserve contour length and produces errors where contours cross each other at steep angles. In the case of the American eel, this created the effect of shortening the tail between image frames, leading to errors between the level set's actual motion and that described by the optical flow field used to set boundary conditions in that region. It should also be pointed out that the present methods suffer the same limitation inherent to all image-based modeling algorithms, namely that of image quality. Low-quality images plagued by noise, poor contrast, low resolution and inconsistent features between image frames will all conspire to produce unacceptable results when used for CFD simulations. Much effort has been made in the field of image processing to alleviate these issues, including specialized denoising techniques (e.g., [40,41]), contrast enhancement methods (e.g., [42,43]) and image segmentation tools designed to work in the presence of noise and poor contrast (e.g., [44,45]). However, it still remains that any algorithm—particularly one designed to perform over many frames of a moving image sequence such as that described in this work—is dependent upon relatively high-quality images in order to function reliably. Finally, it should be pointed out that the present formulation is limited to moving 2D images; however, in our experience, contour-based segmentation and morphing have proven equally effective when applied to 3D image sets, leaving optical flow as the primary limiting factor. This limitation may be addressed in a couple of different ways: First, if the motion being studied is mostly planar in nature, like in the present case of the swimming American eel, 2D optical flow techniques

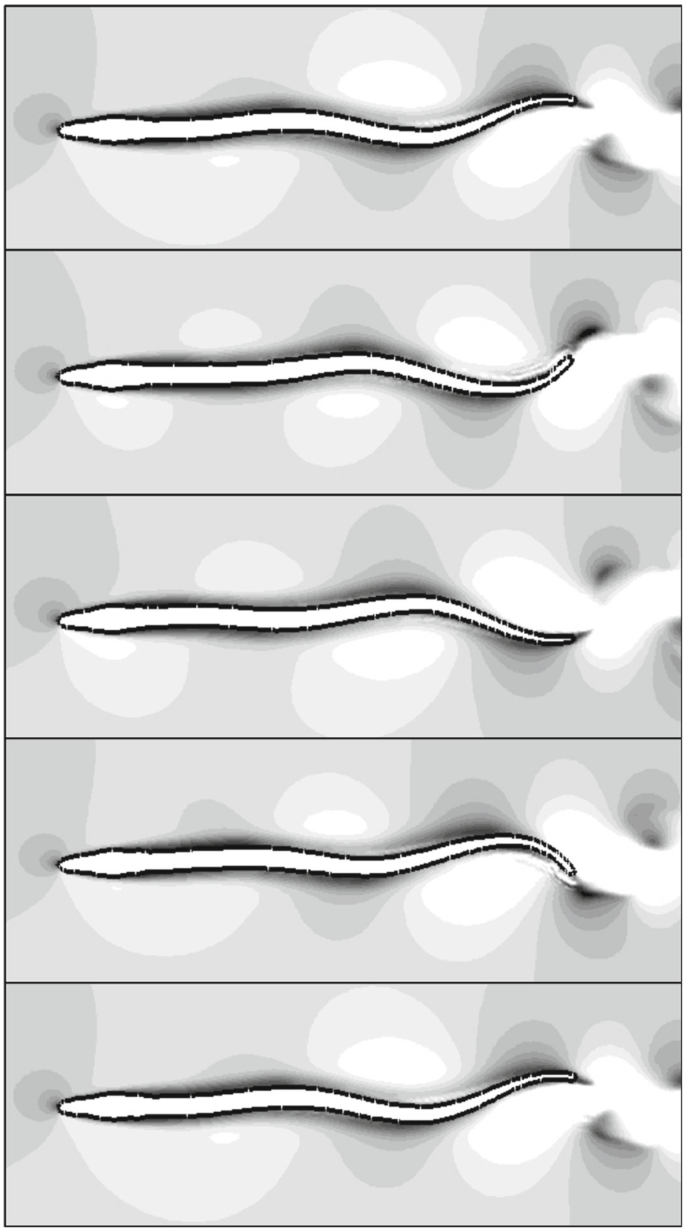
Fig. 13 Axial velocity plotted over a complete tail beat helps illustrate fluid transport along the troughs of the eel's axial waveform

may be applied to individual slices of a 3D data set just as they are to individual image frames currently. Otherwise, it is possible to extend the entire framework to 4D (3D + time) with the addition of a 3D optical flow constraint as suggested in, e.g., [46,47], thus giving a true representation of 3D motion.

4 Conclusions

The Eulerian method developed here provides a route to image-based modeling of moving boundary biological flows, as the implementation is straightforward and entirely (i.e., all the way from the image domain to flow computations) follows the level set framework employed in the CFD code. There is no tedious point generation algorithm, no surface smoothing, no assumptions regarding point correspondence and no need to convert from an Eulerian image field to a Lagrangian surface mesh and then back to an Eulerian field again. Through image segmentation, a level set field is generated on an image mesh and then mapped directly onto a flow solver

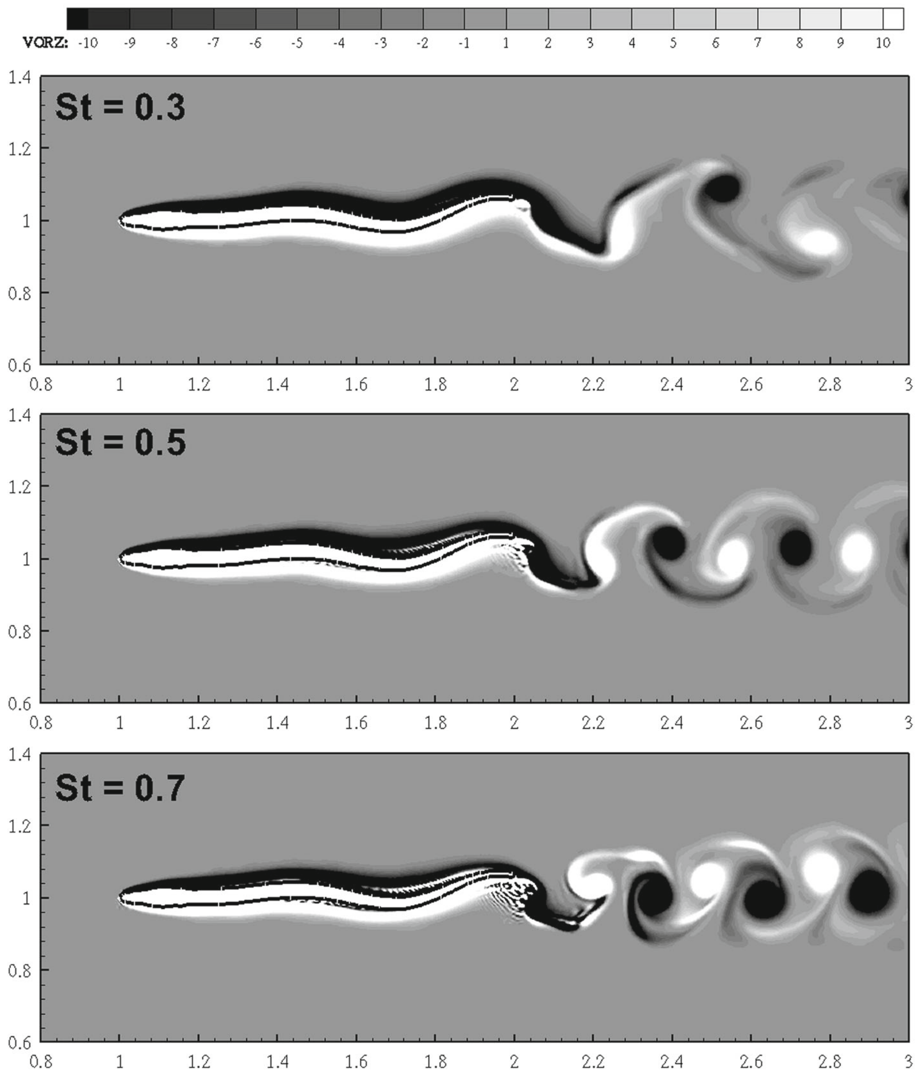

Fig. 14 Contours of vorticity computed at three different Strouhal numbers

mesh by interpolation. The surface velocity of the imaged object is computed from the intensity field using nonlinear optical flow, which has been tested in this work on a synthetic 7-pointed star shape with prescribed kinematics. Gaps between image frames acquired at rates too slow for high-resolution CFD simulations are filled in with another computer vision technique, image morphing, to supply the missing information between image frames so that the relatively low temporal resolution of image sequences would no longer preclude modeling behavior on much finer timescales. By linking together active contour segmentation, optical flow motion tracking and image morphing techniques, a purely Eulerian sharp interface route to modeling moving boundaries has been successfully obtained.

References

1. Schulze-Delrieu, K.: Visual parameters define the phase and the load of contractions in isolated guinea pig ileum. Am. J. Physiol. Gastrointest. Liver Physiol. **276**(6), G1417–G1424 (1999)
2. Adkins, D., Yan, Y.Y.: CFD simulation of fish-like body moving in viscous liquid. J. Bionic Eng. **3**, 147–153 (2006)
3. Liu, H.: Simulation-based biological fluid dynamics in animal locomotion. Appl. Mech. Rev. **58**, 269–282 (2005)
4. Alvino, C., et al.: Multigrid computation of rotationally invariant non-linear optical flow. In: IEEE (2005)
5. Amiaz, T., Kiryati, N.: Piecewise smooth dense optical flow via level sets. Int. J. Comput. Vis. **68**(2), 111–124 (2006)

6. Black, M., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. Comput. Vis. Image Underst. **63**(1), 75–104 (1996)
7. Borshukov, G. et al.: Motion segmentation by multistage affine classification. IEEE Trans. Image Process. **6**(11), 1591–1594 (1997)
8. Brox, T., et al.: High accuracy optical flow estimation based on a theory for warping. In: 8th European Conference on Computer Vision (2004)
9. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
10. Ha, J., et al.: Active contours and optical flow for automatic tracking of flying vehicles. In: 2004 American Control Conference (2004)
11. Horn, B., Schunck, B.: Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
12. Jhunjhunwala, P., Rajagopalan, S.: Optical flow based volumetric spatio-temporal interpolation. In: 30th Annual International IEEE EMBS Conference (2008)
13. Kichenassamy, S. et al.: Conformal curvature flows: from phase transitions to active vision. Arch. Ration. Mech. Anal. **134**, 275–301 (1996)
14. Udaykumar, H., Krishnan, S., Marella, S.V.: Adaptively refined, parallelised sharp interface Cartesian grid method for three-dimensional moving boundary problems. Int. J. Comput. Fluid Dyn. **23**(1), 1–24 (2009)
15. Marella, S. et al.: Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. J. Comput. Phys. **210**, 1–31 (2005)
16. Vigmostad, S.C. et al.: Fluid–structure interaction methods in biological flows with special emphasis on heart valve dynamics. Int. J. Numer. Methods Biomed. Eng. **26**(34), 435–470 (2010)
17. Liu, H. et al.: Sharp interface Cartesian grid method II: a technique for simulating droplet interactions with surfaces of arbitrary shape. J. Comput. Phys. **210**, 32–54 (2005)
18. Sambasivan, S.K., Udaykumar, H.: A sharp interface method for high-speed multi-material flows: strong shocks and arbitrary materialpairs. Int. J. Comput. Fluid Dyn. **25**(3), 139–162 (2011)
19. Mousel, J.A.: A massively parallel adaptive sharp interface solver with application to mechanical heart valve simulations. In: Mechanical and Industrial Engineering, University of Iowa, p. 207 (2012)
20. Zang, Y., Street, R., Koseff, J.: A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. J. Comput. Phys. **114**(1), 18–33 (1994)
21. Sethian, J.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, New York (1999)
22. Krishnan, S., Udaykumar, H.S., Marshall, J.S., Chandran, K.B.: Two-dimensional dynamic simulation of platelet activation during mechanical heart valve closure. Ann. Biomed. Eng. **34**(10), 1519–1534 (2006)
23. Ye, T. et al.: An Accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. J. Comput. Phys. **156**(2), 209–240 (1999)
24. Sethian, J.: Evolution, implementation, and application of level set and fast marching methods for advancing fronts. J. Comput. Phys. **169**(2), 503–555 (2001)
25. Sussman, M., Fatemi, E.: An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. SIAM J. Sci. Comput. **20**(4), 1165–1191 (1999)
26. Sussman, M. et al.: An improved level set method for incompressible two-phase flows. Comput. Fluids **27**(5–6), 663–680 (1998)
27. Li, C. et al.: Distance regularized level set evolution and its application to image segmentation. IEEE Trans. Image Process. **19**(12), 3243–54 (2010)
28. Breen, D.E., Whitaker, R.T.: A level-set approach for the metamorphosis of solid models. IEEE Trans. Vis. Comput. Graph. **7**(2), 173–192 (2001)
29. Whitaker, R.T.: A level-set approach to image blending. IEEE Trans. Image Process. **9**(11), 1849–1861 (2000)
30. Sethian, J., Smareka, P.: Level set methods for fluid interfaces. Annu. Rev. Fluid Mech. **35**, 341–372 (2003)
31. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. Int. J. Comput. Vis. **50**(3), 271–293 (2002)
32. Terzopoulos, D. et al.: Elastically deformable models. Comput. Graph. **21**(4), 205–214 (1987)
33. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational-problems. Commun. Pure Appl. Math. **42**(5), 577–685 (1989)
34. Dillard, S. et al.: Techniques to derive geometries for image-based Eulerian computations. Eng. Comput. **31**(3), 530–566 (2014)
35. Dillard, S.I., Mousel, J.A., Shrestha, L., Raghavan, M.L., Vigmostad, S.C.: From medical images to flow computations without user-generated meshes. Int. J. Numer. Methods Biomed. Eng. **30**(10), 1057–1083 (2014)
36. Greenwood, D.: Classical Dynamics. Dover Publications, New York (1977)
37. Lanczos, C.: The Variational Principles of Mechanics, 4th edn. Dover Publications, New York (1970)
38. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed—algorithms based on Hamilton–Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
39. Tytell, E., Lauder, G.: The hydrodynamics of eel swimming I: wake structure. J. Exp. Biol. **207**, 1825–1841 (2004)
40. Yu, Y.J., Acton, S.T.: Speckle reducing anisotropic diffusion. IEEE Trans. Image Process. **11**(11), 1260–1270 (2002)
41. Zong, X.L., Laine, A.F., Geiser, E.A.: Speckle reduction and contrast enhancement of echocardiograms via multiscale nonlinear processing. IEEE Trans. Med. Imaging **17**(4), 532–540 (1998)
42. Sakellaropoulos, P., Costaridou, L., Panayiotakis, G.: A wavelet-based spatially adaptive method for mammographic contrast enhancement. Phys. Med. Biol. **48**(6), 787–803 (2003)
43. Xiao, D., Ohya, J.: Contrast enhancement of color images based on wavelet transform and human visual system. In: IASTED International Conference on Graphics and Visualization in Engineering. Clearwater, FL (2007)
44. Ashton, E.A., Parker, K.J.: Multiple resolution Bayesian segmentation of ultrasound images. Ultrason. Imaging **17**(4), 291–304 (1995)

45. Boukerroui, D. et al.: Segmentation of ultrasound images—multiresolution 2D and 3D algorithm based on global and local statistics. Pattern Recognit. Lett. **24**(4–5), 779–790 (2003)
46. Kalmoun, E.M., Köstler, H., Rüde, U.: 3D optical flow computation using a parallel variational multigrid scheme with application to cardiac C-arm CT motion. Image Vis. Comput. **25**(9), 1482–1494 (2007)
47. Barron, J., Thacker, N.: Tutorial: Computing 2D and 3D Optical Flow. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester (2005)