J Math Imaging Vis (2010) 37: 112–131
DOI 10.1007/s10851-010-0196-4

Vesicles and Amoebae: On Globally Constrained Shape Deformation

Ishay Goldin · Jean-Marc Delosme · Alfred M. Bruckstein

Published online: 19 March 2010

Abstract Modeling the deformation of shapes under constraints on both perimeter and area is a challenging task due to the highly nontrivial interaction between the need for flexible local rules for manipulating the boundary and the global constraints. We propose several methods to address this problem and generate "random walks" in the space of shapes obeying quite general possibly time varying constraints on their perimeter and area. Design of perimeter and area preserving deformations are an interesting and useful special case of this problem. The resulting deformation models are employed in annealing processes that evolve original shapes toward shapes that are optimal in terms of boundary bending-energy or other functionals. Furthermore, such models may find applications in the analysis of sequences of real images of deforming objects obeying global constraints as building blocks for registration and tracking algorithms.

Keywords Constrained shape evolution · Constant shape-factor deformations · Amoeba motion · Vesicles

This research was partially supported by The Center for Complexity Science (CCS) No. 2006-70.

I. Goldin (✉) · A.M. Bruckstein
Center for Intelligent Systems, Computer Science Department, Technion, I.I.T., Haifa, Israel
e-mail: goldin@cs.technion.ac.il

A.M. Bruckstein
e-mail: freddy@cs.technion.ac.il

J.-M. Delosme
IBISC Laboratory (Computer science, Integrative Biology and Complex Systems), Evry University, 91025 Evry, France
e-mail: delosme@ibisc.univ-evry.fr

1 Introduction

Reinhard Lipowsky, in his overview on *Vesicles and Biomembranes*, [16], states that "vesicles exhibit a great variety of nonspherical shapes. This polymorphism arises, to a large extent, from two global constraints":

– "The surface area of a bilayer membrane is constant (at constant temperature) since the exchange in molecules between the membrane and the solution is negligible" ... "the stretching of these lipid bilayers is limited to rather small deformations. They start to rupture as soon as their area is changed by about one percent".
– The volume of a vesicle is constant (at constant temperature and environment) since, [16], "it is determined by the osmotic pressure arising from those solutes that cannot permeate the bilayer membrane".

Therefore, the mathematical description of vesicle dynamics requires three dimensional models for deforming shapes while preserving both their surface area and volume.

In this work we explore the world of two-dimensional shapes deforming under perimeter and area constraints. A 2D shape is described by a simple closed planar curve. Its perimeter, L, and its area, A, must obey the isoparametric inequality [5], $4\pi A \leq L^2$. At equality, the entire world of shapes consists of circular disks. However, if the circle is deflated, i.e., for a given perimeter, the area is required to be smaller than the maximum permitted by the isoparametric inequality, the world of possible shapes becomes infinitely rich. A planar shape or a simple and closed contour may be described either explicitly, as a parametric continuous curve (or a discrete polygon), or implicitly, as the zero level set of a bivariate function. The various representation types have advantages and drawbacks, and their richness and power of expression depends on the number of parameters they employ.

It would be nice to have a way of selecting "a random shape" of a given perimeter and area. However, the problem with selecting such "random points" in the world of shapes, while imposing restrictions like perimeter and area, is that nonlinear, complex and dissimilar constraints are enforced over the many variables of the particular representation in use. Furthermore, it is very difficult and often impossible to specify or enforce a probability distribution on the "space of shapes".

Indeed, suppose a planar shape is described via an arbitrary parametric representation of a C^{D_0}-continuous closed curve $\mathbf{C}(p) = (x(p), y(p)),\ p \in [0, 1]$. In this representation, the constraints for the perimeter, area and smooth closure of the contour (D_0 continuous derivatives) may be written as follows:

$$\forall i \in \{0, \ldots, D_0\}: \quad \mathbf{C}^{(i)}(0) = \mathbf{C}^{(i)}(1) \tag{1}$$

$$L(\mathbf{C}) = \oint_C \|\mathbf{C}_p\|\, dp = \mathcal{L} \tag{2}$$

$$A(\mathbf{C}) = \frac{1}{2} \oint_C \mathbf{C} \times \mathbf{C}_p\, dp = \mathcal{A} \tag{3}$$

where $\times$ marks the (signed magnitude of the) cross product. The expression for the area is a result of Green's theorem, and it assigns a positive area to a counter-clockwise contour. Using this representation, to select a shape with a given area and perimeter is equivalent to finding a continuous, smooth and non-self-intersecting (a property that is quite hard to explicitly express) vector function, $\mathbf{C}(p)$, obeying the above constraints.

Alternatively, the boundary of a planar shape could be defined as a non-self-intersecting polygon, X, with vertices $\{X_i = (x_i, y_i)\}_{i=0}^{m-1}$ and $X_m = X_0$. The perimeter and area constraints are, in this case:

$$L(X) = \sum_{i=0}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} = \mathcal{L} \tag{4}$$

$$A(X) = \frac{1}{2} \sum_{i=0}^{m-1} (x_i y_{i+1} - x_{i+1} y_i) = \mathcal{A} \tag{5}$$

and here again, the non-self-intersecting conditions are quite complex. Since we are only interested in the contour's shape up to an Euclidean transformation, we can (for example) take X_0 to be at the origin, $(0, 0)$, and X_1 on the positive x-axis, $(x_1, 0)$, i.e., $x_0, y_0, y_1 = 0$. In all, to select a random shape is equivalent to the selection of a random "non-self-intersecting" solution of a highly underdetermined system of two nonlinear equations having $2m - 3$ unknowns, not a trivial task. A way to simplify matters would be by enforcing the curve to obey some additional properties. For example, we could force the boundary polygon to have equal length edges. Such "arc-length-like" representations can provide relatively good polygonal approximations of any arbitrary curve. In this context, we may replace (4) with a set of m simpler equations constraining all polygon edges to have l_e length:

$$\forall i = 0, \ldots, m-1: \quad (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = l_e^2 \tag{6}$$

Altogether, since X_0 and X_1 are now fixed, we have a system of m nonlinear equations with $2m - 4$ unknowns, and we seek a random non-self-intersecting solution for it.

The same problem can also be posed in terms of the angles, $\{\theta_i\}_{i=0}^{m-1}$, that each unit polygon edge makes with an arbitrary "horizon". In that case, the perimeter constraint is implicitly imposed by the number of edges, m, and their length, l_0, and the closure and area constraints are given by:

$$\sum_{i=0}^{m-1} \cos\theta_i = \sum_{i=0}^{m-1} \sin\theta_i = 0 \tag{7}$$

$$\sum_{i=0}^{m-1} \sum_{j=1}^{i-1} \sin(\theta_i - \theta_j) = 2\mathcal{A}/l_0^2 \tag{8}$$

Here we have three equations for "only" $m - 1$ unknowns, but still no simple way to obtain a "random" feasible non-self-intersecting solution.

The examples given above show that modeling random shapes in general and under given perimeter and area constraints in particular is not a well posed and well defined problem. Hence, we shall deal here with a simpler problem: given an initial shape, C_0, with perimeter $\mathcal{L}_0$ and area $\mathcal{A}_0$, we shall seek to express infinitesimal (or small) deformations of this shape that will preserve its perimeter and area or modify them at a predefined rate. In terms of each selected representation method, we shall deal with the problem of "obtaining new shapes from old ones" while obeying the mathematical constraints imposed on the shape representation parameters. Formally speaking, for a given initial shape, $C(0)$, with perimeter $\mathcal{L}(0)$ and area $\mathcal{A}(0)$, and given a desired time evolving perimeter, $\mathcal{L}(t)$, and area, $\mathcal{A}(t)$, we seek to express evolution rules that satisfy $L(C(t)) = \mathcal{L}(t)$ and $A(C(t)) = \mathcal{A}(t)$.

2 Related Work

Several previous research efforts dealt with the problem of deforming contours while preserving either their perimeter or their area. In [9], Gage studied an area-preserving curvature flow and its properties. Other curve evolution rules, the Euclidean, affine, and similarity invariant geometric diffusion flows were adjusted in [20] so that the shape is continuously scaled to preserve either its perimeter or its area. Note however, that these works dealt only with a set of predefined

flows rather than general deformations, and only either the perimeter or the area were fixed.

In the field of graphics, Sauvage et al. suggested methods for multiresolution editing of curves such that their perimeter [21, 23] *or* their area [22] are preserved. These methods, however, require iterative constrained optimization processes. A deformation is applied to the curve by modifying any control point at any resolution level. Then, a "target curve" is iteratively optimized for smoothness and minimum deviation from its original shape under length or area constraints. In contrast, we would like to have deformation methods that are analytic and non-iterative. In [6], Elber presents methods for multiresolution editing of B-spline curves subject to various linear constraints. In addition to the positional, tangential and orthogonal constraints that are linear, Elber suggests that the bilinear form of the area constraint can be dealt with as a pair of interchangeable linear constraints, one for each of its dimensions. Sauvage et al. and Elber then joined forces to present a fast and detail preserving generalization of this work in [24] where they deform 3D B-spline surfaces subject to volume constraints. Still, none of these methods enable the representation or the deformation of shapes subject to both the perimeter and the area constraints.

Recently, a very interesting study of shapes having minimal contour bending energy under perimeter and area constraints was undertaken by Arreaga et al. [1], Capovilla et al. [2]. We shall show here how our shape evolution ideas can be used to flow toward optimal shapes in the minimal bending energy sense. In this context we note that Okabe presented in [18] an interesting set of partial differential equations governing a perimeter and area preserving contour evolution in the direction of the negative gradient of the bending energy functional. In the limit, these equations implicitly describe "stationary" (i.e. optimal) contours with respect to the energy functional.

3 Curve Evolution

Epstein and Gage proved in [7], that a general curve deformation can always be expressed by the motion of its points in the direction of the curve's local normal, as follows:

$$\frac{\partial \mathbf{C}(s,t)}{\partial t} = \varepsilon(s,t)\hat{n}(s,t) \tag{9}$$

where s is the curve's arc-length parametrization. Therefore, a differential deformation can be bijectively expressed by the choice of a geometrically defined function, $\varepsilon(s)$. The constraints we want to impose over the rate of change of the contour's enclosed area and perimeter are:

$$\frac{d}{dt}\big[L(\mathbf{C})\big] = \frac{\partial}{\partial t}\left[\oint_C \|\mathbf{C}_p\|\,dp\right] = \mathcal{L}_t(t) \tag{10}$$

$$\frac{d}{dt}\big[A(\mathbf{C})\big] = \frac{\partial}{\partial t}\left[\frac{1}{2}\oint_C \mathbf{C}\times\mathbf{C}_p\,dp\right] = \mathcal{A}_t(t) \tag{11}$$

Using differential geometry arguments [12], these constraints can be expressed as constraints on $\varepsilon(s)$, or $\varepsilon(p)$. If $\hat{n}$ is directed outward, and K is the curvature, then

$$\frac{d}{dt}\big[L(\mathbf{C})\big] = \oint_C \varepsilon K\,ds = \oint_C \varepsilon K\|\mathbf{C}_p\|\,dp = \mathcal{L}_t(t) \tag{12}$$

$$\frac{d}{dt}\big[A(\mathbf{C})\big] = \oint_C \varepsilon\,ds = \oint_C \varepsilon\|\mathbf{C}_p\|\,dp = \mathcal{A}_t(t) \tag{13}$$

However, how can we find such $\varepsilon(s)$ functions? One of the possibilities is to define $\varepsilon(s)$ as a finite sum of elementary (basis) functions:

$$\varepsilon(s) \triangleq \sum_{r=0}^{m} \alpha_r b_r(s) \tag{14}$$

and seek constraints on the finite set of coefficients, $\{\alpha_r\}$. Notice that, since m here is a finite number, not every evolution function, $\varepsilon(s)$, is expressible. The richness of the world of possible deformations modeled in this way depends on the type and number of basis functions involved in defining, or rather designing, $\varepsilon(s)$, and for most applications, handling many parameters is costly and complicated, hence we must select and use basis functions with care.

Substituting the representation (14) into (12) and (13) yields

$$\begin{aligned}\oint_C \varepsilon K\,ds &= \oint_C \left(\sum_{r=0}^{m}\alpha_r b_r\right) K\,ds \\ &= \sum_{r=0}^{m}\alpha_r\left(\oint_C K b_r\,ds\right) = \mathcal{L}_t(t)\end{aligned} \tag{15}$$

$$\begin{aligned}\oint_C \varepsilon\,ds &= \oint_C \left(\sum_{r=0}^{m}\alpha_r b_r\right) ds \\ &= \sum_{r=0}^{m}\alpha_r\left(\oint_C b_r\,ds\right) = \mathcal{A}_t(t)\end{aligned} \tag{16}$$

and by defining $B_r^L \triangleq \oint_C K b_r\,ds$ and $B_r^A \triangleq \oint_C b_r\,ds$, (15), and (16) become linear in the coefficients of ε's representation in terms of the selected "basis" functions:

$$\sum_{r=0}^{m}\alpha_r(t)B_r^L(t) = \mathcal{L}_t(t) \tag{17}$$

$$\sum_{r=0}^{m}\alpha_r(t)B_r^A(t) = \mathcal{A}_t(t) \tag{18}$$

Notice that the $\{B_r^L\}$ and the $\{B_r^A\}$ terms are scalar descriptors of the complete contour at a given point in time.

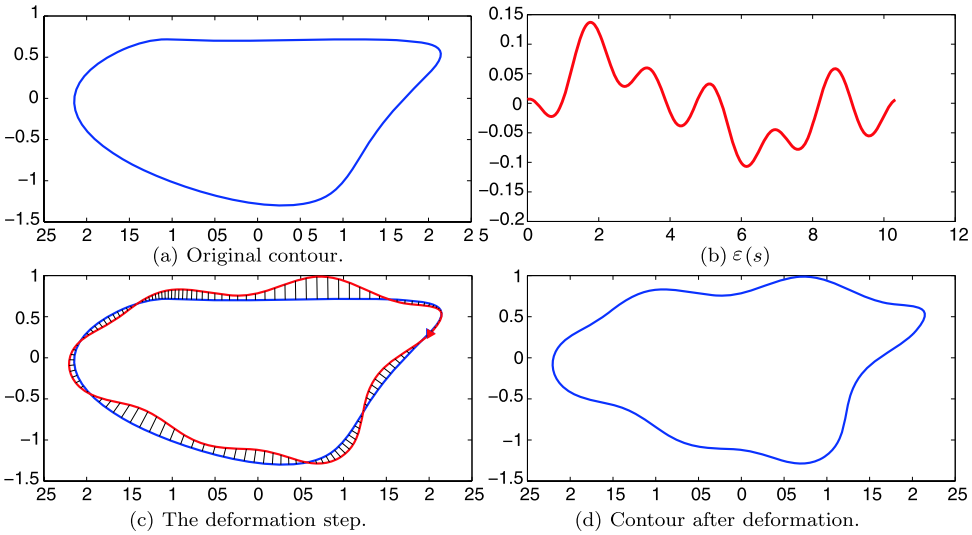

Fig. 1 A curve evolution example. The *contour* in (**a**) is deformed into the one in (**d**) via the $\varepsilon(s)$ evolution function shown in (**b**). The *triangle* in (**c**) marks the contour's arbitrary starting point. Notice that the evolution shown is of finite rather than infinitesimal time interval. The basis functions in use are $\{\cos(2\pi rs/L)\}_{r=1,2,3,6}$, $\{\sin(2\pi rs/L)\}_{r=1,2}$

They do not depend on $\varepsilon(s)$, but only on the contour's trace, $\mathbf{C}(s)$, on the chosen representation "basis functions" in use, $\{b_r(s)\}_{r=0}^{m}$, and on the arbitrary choice of a starting point, $s=0$, on the contour.

Another constraint on the deformation function, $\varepsilon(s)$, is that it has to keep the curve continuous and smooth to a certain degree, D_0, depending on the application. For that to happen, $\varepsilon(s)$ has to be smooth enough everywhere, and even as continuous basis functions are in use, the endpoints constraints, $\forall i \in \{0,\ldots,D_0\}$: $\varepsilon^{(i)}(0) = \varepsilon^{(i)}(\mathcal{L}(t))$, lead to more linear constraints on the set of coefficients, $\{\alpha_m\}$:

$$\forall i \in \{0,\ldots,D_0\}: \quad \sum_{r=0}^{m} \alpha_r(t) b_r^{(i)}(0) = \sum_{r=0}^{m} \alpha_r(t) b_r^{(i)}\big(\mathcal{L}(t)\big) \tag{19}$$

In all, the $\{\alpha_r\}$ coefficients are required to satisfy the linear system consisting of (17), (18) and (19). As expected, choosing the right basis representation can considerably simplify the problem. Taking basis functions that are cyclic over the domain of the contour's perimeter, $[0,\mathcal{L}(t)]$, like the Fourier basis functions for example, readily satisfies (19). Furthermore, (18) depends, via B_r^A's, only on the mean values of the basis functions. In some bases, all but one of the basis functions have zero mean. Again, the Fourier basis is a good example. In those cases, assuming the nonzero mean function is normalized to have unit mean, the right area rate of change can readily be guaranteed by setting its coefficient, α_0, to $\mathcal{A}_t(t)$. Hence, (17) remains the only constraint on ε's coefficients. How is the contour deformed then? Assuming that we want the deformation to depend upon exactly $m+1$ parameters, a number that is completely application dependent, the deformation is described by $\partial\mathbf{C}(s,t)/\partial t = \hat{n}(s,t)\sum_{r=0}^{m}[\alpha_r(t)b_r(s)]$. Notice that except for the nonzero mean function, the choice of the other m basis functions is also arbitrary and application dependent. In the Fourier basis for example, it might make sense to take the lower frequency terms, $\{\cos(2\pi rs/L),\sin(2\pi rs/L)\}_{r=1}^{m/2}$. For the Fourier and similar bases, any subset of $m-1$ coefficients out of $\{\alpha_r\}_{r=1}^{m}$ can be set with no restrictions applied. Then, the remaining coefficient is determined by solving the linear (17). Next, the contour is deformed via $\mathbf{C}(s,t+dt) = \mathbf{C}(s,t) + \hat{n}(s,t)\,dt\sum_{r=0}^{m}[\alpha_r(t)b_r(s)]$. Figure 1 shows an example of shape evolution according to randomly generated parameters. In this example, Fourier basis functions were in use, five parameters were randomly set, and one was determined accordingly via (17).

In some applications, further information might be available about the rules governing the deformation of the shape. Prior information may either be incorporated into the probability distribution functions of the randomly generated coefficients, or used to impose further constraints on $\{\alpha_r\}$. To significantly simplify applying those constraints, we should express them, if possible, via linear constraints on $\{\alpha_r\}$.

4 The "Local Evolve–Global Rescale" Model

Amoebae (see e.g. Fig. 2) are unicellular organisms famous for their ability to "morph". Amoebae continually form and retrace so-called pseudopodia, i.e., "false arms and legs", in various directions. The pseudopodia test the environment for likely meals or possible dangers. Each time such arm is formed, the main body of the amoeba contracts, in both surface area and volume, to allow the arm's growth. The arm may later be retracted or the whole body may move toward the arm, reincorporating it, etc.

We would like our mathematical shape deformation model to be able to describe such behaviors, i.e., to allow

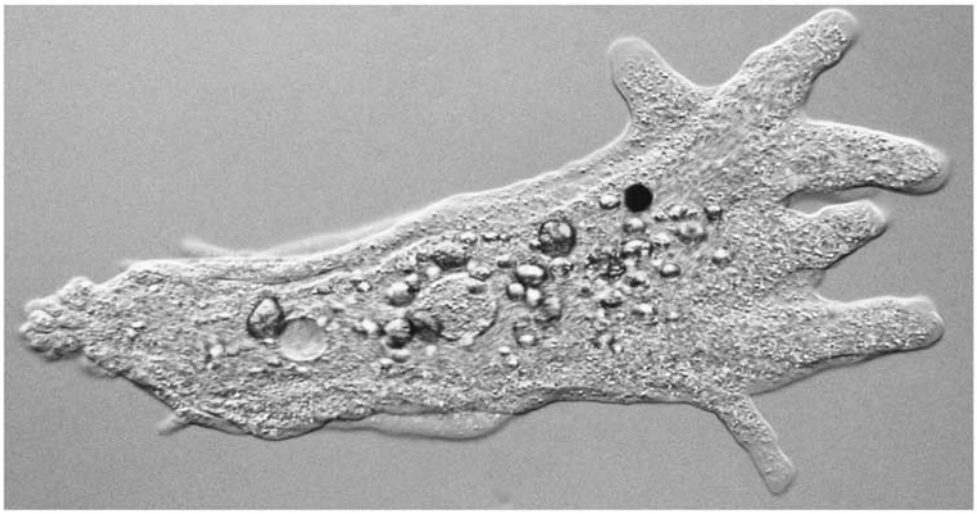

Fig. 2 An amoeba (picture taken from the Carnegie Institute for Science web page, www.carnegieinstitution.org/first_light_case/horn/lessons/cellimages.html)

a local segment of the contour to intensively move outward while the rest of the contour mildly adjusts itself to the change in order to obey the global constraints on the perimeter and the area. The global adjustments should preserve the contour's shape except for the "active region". The "*local evolve global rescale*" principle refers to a class of deformation techniques describing such deformations for various shape representation methods. This section describes the "local evolve global rescale" evolution of continuous contours while polygonal and splinegonal deformation methods following this principle are presented in Sects. 5.2 and 6.

Assume, for example, that we want to grow a certain segment of the contour into a protrusion or an arm. In other words, we would like the evolution rule to rapidly push outward a certain area of the contour while the rest is slowly deforming in a way that (a) preserves its general shape, and (b) makes the contour obey its global constraints. We shall start with a possible description of the local rapid deformation. A *bump function*, $\varepsilon_{bump}(s)$, is a function over the $[0, L(t)]$ domain describing an arm growing curve evolution rule. Good examples of bump functions could be the well known B-spline basis functions [4, 8, 25]:

$$\frac{\partial \mathbf{C}}{\partial t} = \varepsilon_{bump}(s)\hat{n} = \alpha N_{i,j}(s)\hat{n}, \quad s \in \left[0, L(t)\right] \tag{20}$$

where $N_{i,j}(s)$ is defined by:

$$N_{i,j}(s) = \left(\frac{s - s_i}{s_{i+j-1} - s_i}\right) N_{i,j-1}(s) + \left(\frac{s_{i+j} - s}{s_{i+j} - s_{i+1}}\right) N_{i+1,j-1}(s) \tag{21}$$

$$N_{i,1}(s) = \{1 \text{ if } s_i \le s < s_{i+1},\ 0 \text{ otherwise}\} \tag{22}$$

These piecewise polynomials of degree $j-1$ are recursively defined over an easy to manipulate knot vector, $(s_0, s_1, \ldots, s_{k+j})$. They have useful properties like having a local nonzero domain, nonnegativity everywhere, zero endpoint values and derivatives, C^{j-2} continuity over the knots, etc. The general shape of these functions is shown in Fig. 3b. The knot vector, the order, j, the index, i, and the scaling parameter, α, are chosen to best fit the shape evolution characteristics of the specific application. The nonzero domain of the bump function will be centered "on" the desired arm growing region of the contour as can be seen in Fig. 3.

In fact, depending on the desired behavior of the deformable shape, the $\varepsilon(s)$ function can take other forms. For example, a shape might grow several arms simultaneously by taking an evolution function that is the sum of a variety of bump functions. Any evolution function is allowed as long as it is cyclic having a period of $L(t)$ (or D_0 equal endpoints derivatives), this, in order to keep the contour closure smooth.

However, since these evolution functions do not, in general, satisfy the perimeter and area constraints, global correction terms should be added to the flow. These terms must deform the contour in an application dependent "natural way". We shall first select to base our correction terms on the Euclidean geometric curvature flow, $\partial \mathbf{C}/\partial t = -K\hat{n}$. Indeed, in [9], Gage proposes a variation on this flow that preserves the contour's area through time. This idea was further developed in [20], where the authors introduced a perimeter preserving version of this flow. We shall here use both of these flows to adjust the overall curve deformation to obey the desired time varying constraints on the shape's perimeter and area.

The perimeter preserving Euclidean geometric curvature flow is expressed by [20]:

$$\frac{\partial \mathbf{C}}{\partial t} = \left(-K + \frac{\oint K^2\, ds}{L(\mathbf{C})}\mathbf{C} \cdot \hat{n}\right)\hat{n} \tag{23}$$

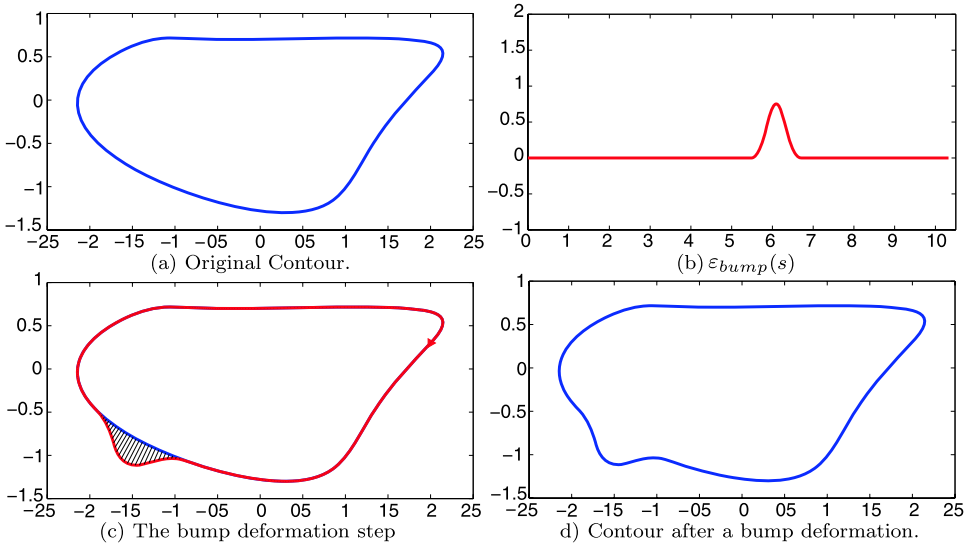

Fig. 3 A curve evolution example. The *contour* in (**a**) is deformed into the one in (**d**) via the $\varepsilon_{bump}(s)$ evolution bump function shown in (**b**). The *triangle* in (**c**) marks the starting point of placing the bump function over the contour. Notice that the evolution shown is of finite rather than infinitesimal time interval

and the Euclidean geometric curvature flow without area shrinkage is given by [9]:

$$\frac{\partial \mathbf{C}}{\partial t} = \left(-K + \frac{\pi \mathbf{C}\cdot\hat{n}}{A(\mathbf{C})}\right)\hat{n} \tag{24}$$

The proof of perimeter and area preservation of these flows proceeds as follows:
Using integration by parts, we have that

$$\oint (\mathbf{C}\cdot K\hat{n})\,ds = \oint \left(\mathbf{C}\cdot(-\hat{t}_s)\right)ds = -\oint (\mathbf{C}\cdot\mathbf{C}_{ss})\,ds \tag{25}$$

$$= -[\mathbf{C}\cdot\mathbf{C}_s]_0^{\mathcal{L}(t)} + \oint (\mathbf{C}_s\cdot\mathbf{C}_s)\,ds \tag{26}$$

$$= 0 + \oint \mathbf{C}_s^2\,ds = \oint 1\,ds = L(\mathbf{C}) \tag{27}$$

Therefore, the flow defined by (23) satisfies

$$\frac{d}{dt}\big[L(\mathbf{C})\big]$$

$$= \oint \varepsilon K\,ds = \oint \left(-K + \frac{\oint K^2\,ds}{L}\mathbf{C}\cdot\hat{n}\right)K\,ds \tag{28}$$

$$= -\oint K^2\,ds + \frac{\oint K^2\,ds}{L}\oint (\mathbf{C}\cdot K\hat{n})\,ds \tag{29}$$

$$= -\oint K^2\,ds + \frac{\oint K^2\,ds}{L}L = 0 \tag{30}$$

Concerning the area variation, since $\oint K\,ds = 2\pi$ and $A(\mathbf{C}) = \frac{1}{2}\oint (\mathbf{C}\cdot\hat{n})\,ds$, we have that

$$\frac{d}{dt}\big[A(\mathbf{C})\big]$$

$$= \oint \varepsilon\,ds = \oint \left(-K + \frac{\pi\mathbf{C}\cdot\hat{n}}{A}\right)ds \tag{31}$$

$$= \frac{\pi}{A}\oint (\mathbf{C}\cdot\hat{n})\,ds - \oint K\,ds = \frac{\pi}{A}2A - 2\pi = 0 \tag{32}$$

The idea behind both of these global dimension-preserving flows is that the shape is continuously scaled while it deforms. The infinitesimal scaling term, $\partial\mathbf{C}/\partial t = \lambda\mathbf{C}$, can be expressed via its projection on the direction of the normal to the curve, $\partial\mathbf{C}/\partial t = (\lambda\mathbf{C}\cdot\hat{n})\hat{n}$. The rate in which the contour is being scaled is designed to precisely overcome the rate of change of the perimeter or the area due the curvature flow.

The overall evolution equation is therefore

$$\frac{\partial \mathbf{C}}{\partial t} = \varepsilon_{net}(s)\hat{n}$$

$$= \varepsilon_{bump}(s)\hat{n} + \alpha_2\left(-K + \frac{\oint K^2\,ds}{L(t)}\mathbf{C}\cdot\hat{n}\right)\hat{n}$$

$$+ \alpha_3\left(-K + \frac{\pi\mathbf{C}\cdot\hat{n}}{A(t)}\right)\hat{n} \tag{33}$$

And the perimeter and area rates of change are

$$\frac{d}{dt}\big[L(\mathbf{C})\big]$$

$$= \oint K\varepsilon_{net}(s)\,ds$$

$$= \oint K\left[\varepsilon_{bump}(s) + \alpha_3\left(-K + \frac{\pi\mathbf{C}\cdot\hat{n}}{A}\right)\right]ds \tag{34}$$

$$= \oint K\varepsilon_{bump}(s)\,ds$$

$$+ \alpha_3\left(\frac{\pi}{A}\oint (\mathbf{C}\cdot K\hat{n})\,ds - \oint K^2\,ds\right) \tag{35}$$

$$= \oint K\varepsilon_{bump}(s)\,ds + \alpha_3\left(\frac{\pi L}{A} - \oint K^2\,ds\right) \tag{36}$$

$$\begin{aligned}
&\frac{d}{dt}\big[A(\mathbf{C})\big] \\
&\quad = \oint \varepsilon_{net}(s)\,ds \\
&\quad = \oint \left[\varepsilon_{bump}(s) + \alpha_2\left(-K + \frac{\oint K^2\,ds}{L}\mathbf{C}\cdot\hat{n}\right)\right] ds \qquad (37)\\
&\quad = \oint \varepsilon_{bump}(s)\,ds \\
&\qquad + \alpha_2\left(\oint\left(\frac{\oint K^2\,ds}{L}\mathbf{C}\cdot\hat{n}\right)ds - \oint K\,ds\right) \qquad (38)\\
&\quad = \oint \varepsilon_{bump}(s)\,ds \\
&\qquad + \alpha_2\left(\frac{\oint K^2\,ds}{L}\oint \mathbf{C}\cdot\hat{n}\,ds - 2\pi\right) \qquad (39)
\end{aligned}$$

However, since we require the overall evolution to satisfy $dL/dt = \mathcal{L}_t(t)$ and $dA/dt = \mathcal{A}_t(t)$, we can solve for coefficients of the correction terms:

$$\begin{aligned}
\alpha_3 &= \frac{(L_t(t) - \oint K\varepsilon_{bump}(s)\,ds)A}{\pi L - A\oint K^2\,ds} \\
\alpha_2 &= \frac{(A_t(t) - \oint \varepsilon_{bump}(s)\,ds)L}{\oint K^2\,ds \oint \mathbf{C}\cdot\hat{n}\,ds - 2\pi L}
\end{aligned} \qquad (40)$$

where $L_t(t) = (\mathcal{L}(t+dt) - L(t))/dt$ and $A_t(t) = (\mathcal{A}(t+dt) - A(t))/dt$. In all, the generation of such perimeter and area constrained evolution requires as input only the bump function parameters and its positioning on the contour. All other terms in $\varepsilon_{net}(s)$ then follow from (33) and (40).

We can get the same mathematical expression for the overall flow from another viewpoint that might be more straightforward. We consider a deformation rule that has three components as follows, $\varepsilon_{net} = \varepsilon_{bump} + \beta_2\varepsilon_{natural} + \beta_3\varepsilon_{scale}$. The ε_{bump} term is, as explained above, an unconstrained deformation of the contour. The other two terms then compensate for the deviation ε_{bump} caused in the global constraints. The $\varepsilon_{natural}$ term is some easy to model flow that is "natural" in some way to the object being modeled. Here we define it to be the curvature flow, for its many useful properties like producing no shocks, topology and convexity preservation, and a tendency to drive shapes toward disks [10, 11]. Setting the right β_2 value, the deformation rule defined by both the natural flow term and the bump function term obeys the right time varying "shape factor" given by $4\pi\mathcal{A}(t)/\mathcal{L}^2(t)$. The last term, $\varepsilon_{scale} = \mathbf{C}\cdot\hat{n}$, with the proper selection of β_3, infinitesimally rescales the contour to the desired dimensions. Hence, setting the right coefficients, β_2 and β_3, the overall flow, $\varepsilon_{net} = \varepsilon_{bump} - \beta_2 K + \beta_3(\mathbf{C}\cdot\hat{n})$, is made to obey the time varying perimeter and area constraints. The right coefficients are determined by expressions similar to (40) above.

The curvature flow correction term we proposed above for $\varepsilon_{natural}(s)$ is certainly not the only one possible. Other "natural" flows might be chosen, as long as they produce no shocks and suitably evolve the contour's shape factor.

This rather general "local evolve global rescale" deformation technique is an alternative to the one presented in Sect. 3, and it can be useful in cases where we have prior knowledge about the types of global deformations of the object we are modeling. When modeling an amoeba, for instance, we know what type of activity the object is going through, for example, growing an arm, and we further assume something about the natural tendencies of its shape deformation, for example, a tendency to regain a circular shape or to flow toward a shape that optimizes a global geometric functional such as minimizing of the elastic bending energy as will be further explored in Sect. 7. This prior information allows us, at the expense of defining shape dependent "basis functions", to express typical deformations via significantly fewer parameters.

4.1 Numerical Implementation

Osher and Sethian advocated in [19] the use of implicit representation of curves (or surfaces) as zero level sets of three (or higher) dimensional surfaces for the temporal evolution of such shapes. Indeed, a curve $\mathbf{C}(t)$ is expressed by $\{\mathbf{C}(t) \in \mathbb{R}^2 : \Phi(\mathbf{C}, t) = 0\}$, the zero set of a deforming bivariate function, $\Phi : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$. By differentiating $\Phi(\mathbf{C}, t) = 0$ with respect to t we have that

$$\nabla\Phi \cdot C_t + \Phi_t = 0 \qquad (41)$$

and since $\hat{n} = -\nabla\Phi/\|\nabla\Phi\|$ (the sign being determined by the arbitrary choice of $\Phi > 0$ within the contour), any $\partial C/\partial t = \varepsilon\hat{n}$ deformation of a level set contour can be expressed via

$$\Phi_t = \varepsilon\|\nabla\Phi\| \qquad (42)$$

The numerical implementation is carried out on a fixed grid and consists of a discretization and representation of $\Phi(x, y)$ at sample points, (i, j), $\Phi(i, j)$. First, the bump function, $\varepsilon_{bump}(s)$, can be replaced with a bivariate bump function, $\varepsilon_{bump}(i, j)$. For example, we can use a 2D Gaussian function centered around (x_0, y_0), a point in the center of the active region of the contour,

$$\varepsilon_{bump}(i, j) = \alpha_1 e^{-\frac{(i-x_0)^2+(j-y_0)^2}{\sigma^2}} \qquad (43)$$

The bivariate bump can take any other shape as long as it is continuous; no other constraints are enforced. After $\varepsilon_{bump}(i, j, t)$ is set, it is added to the embedding surface, $\Phi_{bump}(i, j, t) = \Phi(i, j, t) + \varepsilon_{bump}(i, j, t)\|\nabla\Phi\|\Delta t$. Then, the coefficients governing the rates of perimeter preserving

and area preserving curvature flows are calculated (alternatively, we can calculate β_2 and β_3, the rates of enforcing curvature flow and scaling):

$$\begin{aligned}\alpha_3 &= \frac{(\mathcal{L}(t+\Delta t)-L)A}{\Delta t(\pi L - A\oint K^2\,ds)}\\ \alpha_2 &= \frac{(\mathcal{A}(t+\Delta t)-A)L}{\Delta t(\oint K^2\,ds \oint \mathbf{C}\cdot\hat{n}\,ds - 2\pi L)}\end{aligned} \tag{44}$$

and the embedding function obeys its zero level sets global constraints by following

$$\begin{aligned}&\Phi(i,j,t+\Delta t)\\ &\quad=\Phi_{bump}(i,j,t)\\ &\qquad+\left[\alpha_2\left(-K+\frac{\oint K^2\,ds}{\mathcal{L}(t)}\mathbf{C}\cdot\hat{n}\right)\right.\\ &\qquad\left.+\alpha_3\left(-K+\frac{\pi\mathbf{C}\cdot\hat{n}}{\mathcal{A}(t)}\right)\right]\|\nabla\Phi\|\Delta t\end{aligned} \tag{45}$$

where all terms are measured over $\Phi_{bump}(i,j,t)$. The curvature and the unit normal of level sets passing through each of the grid points can be approximated since we have explicit formulae for them. For instance, the curvature of a level set of $\Phi(x,y)$ at (x,y) is given by

$$K=\frac{\Phi_{xx}\Phi_y^2-2\Phi_x\Phi_y\Phi_{xy}+\Phi_{yy}\Phi_x^2}{(\Phi_x^2+\Phi_y^2)^{3/2}} \tag{46}$$

The estimation of instantaneous global measures of the curve is done by sampling the zero level set. The perimeter and the area are calculated from the sample points positions, $\{\mathbf{C}_i\}_{i=1}^{Q}$. Sample points features like the curvature, K_i, or the unit normal, $\hat{n}_i$, are approximated by suitable interpolation from neighboring grid points, and their integrals are estimated by their sum, weighted by their associated arc-length. For example, the bending elastic energy is estimated by:

$$\oint K^2\,ds=\sum_{i=1}^{Q}K_i\frac{\|\mathbf{C}_{i+1}-\mathbf{C}_i\|+\|\mathbf{C}_i-\mathbf{C}_{i-1}\|}{2} \tag{47}$$

In all, the deformation step of (45) can be applied to all grid points since it is made out of only local terms, like K and $\mathbf{C}\cdot\hat{n}$, that are evaluated at any grid point, and global terms, like the contour's perimeter, its area, and its bending elastic energy, which can be regarded as constants over all grid points. An example of perimeter and area preserving shape deformation is shown in Fig. 4.

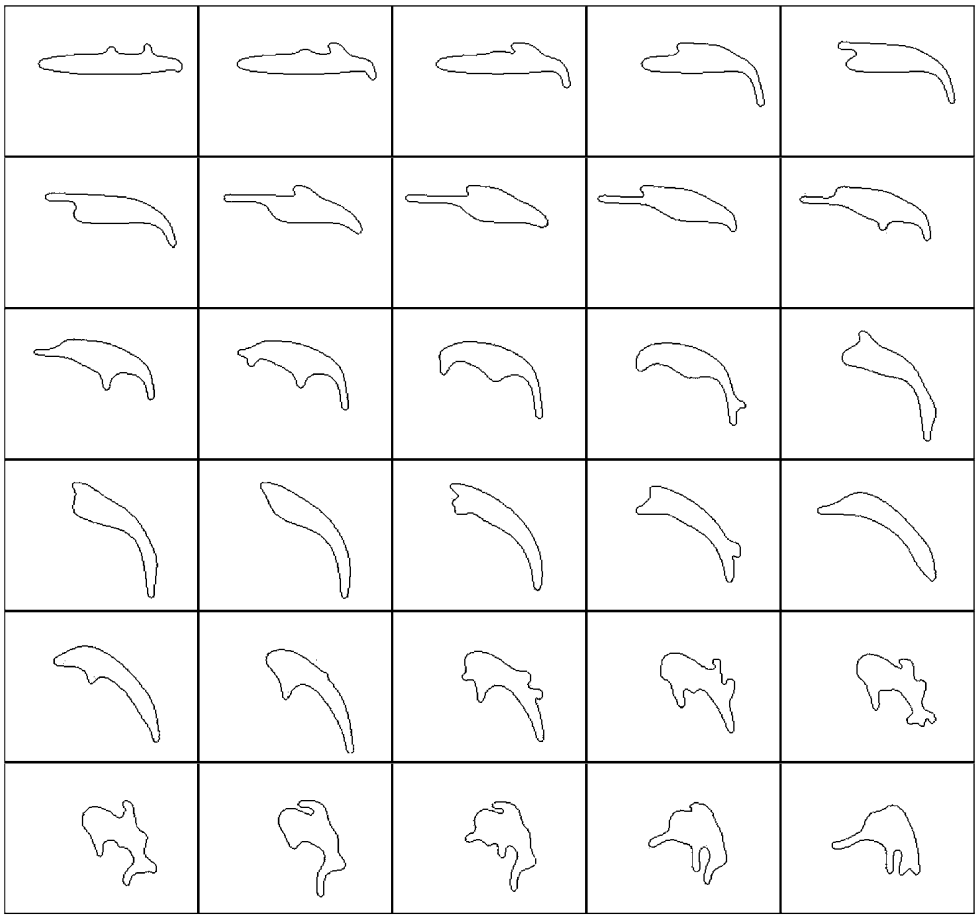

Fig. 4 An example of perimeter and area preserving shape deformation produced via a level sets implementation of the "local evolve global rescale" model. At any time, the above amoeba is growing two pseudopodia at locations that are determined by random walks along the perimeter of the shape

A great advantage of using the level set method is that it inherently avoids contour self intersections, while enabling shapes to change their topology as they deform. This characteristic may be helpful or harmful depending on the application. Topology-preserving implementations of the level set method are presented in [13], and these can be called upon if such shape deformations are necessary.

5 Deforming Globally Constrained Polygons

Suppose we want to deform a contour that is given as a discrete vector of points in the plane, $\{X_i = (x_i, y_i)\}_{i=0}^{m-1}$ with $X_m = X_0$, constrained to obey

$$\sum_{i=0}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} = \mathcal{L}(t) \tag{48}$$

$$\frac{1}{2} \sum_{i=0}^{m-1} (x_i y_{i+1} - x_{i+1} y_i) = \mathcal{A}(t) \tag{49}$$

Again, we would like to obtain new solutions from known ones. We would like to express an infinitesimal deformation, $\{\delta X_i = (\delta x_i, \delta y_i)\}_{i=0}^{m-1}$, that would satisfy $L(X + \delta X) = \mathcal{L}(t + \delta t)$ and $A(X + \delta X) = \mathcal{A}(t + \delta t)$, that is

$$\sum_{i=0}^{m-1} \sqrt{\big((x_{i+1} + \delta x_{i+1}) - (x_i + \delta x_i)\big)^2 + \big((y_{i+1} + \delta y_{i+1}) - (y_i + \delta y_i)\big)^2} = \mathcal{L}(t + \delta t) \tag{50}$$

$$\frac{1}{2} \sum_{i=0}^{m-1} \big((x_i + \delta x_i)(y_{i+1} + \delta y_{i+1}) - (x_{i+1} + \delta x_{i+1})(y_i + \delta y_i)\big) = \mathcal{A}(t + \delta t) \tag{51}$$

In order to simplify the algebra and assure reasonable parametrization, we can replace the perimeter constraint with the requirement for m equal length segments, each of length l_e:

$\forall i = 0, \ldots, m-1$:

$$(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = l_e^2(t) \tag{52}$$

$$\big((x_{i+1} + \delta x_{i+1}) - (x_i + \delta x_i)\big)^2 + \big((y_{i+1} + \delta y_{i+1}) - (y_i + \delta y_i)\big)^2 = l_e^2(t + \delta t) \tag{53}$$

Then, substituting (52) into (53) and (49) into (51), marking $\Delta x_i \triangleq x_{i+1} - x_i$ and manipulating, yields

$\forall i = 0, \ldots, m-1$:

$$2\Delta x_i(\delta x_{i+1} - \delta x_i) + (\delta x_{i+1} - \delta x_i)^2 + 2\Delta y_i(\delta y_{i+1} - \delta y_i) + (\delta y_{i+1} - \delta y_i)^2 = l_e(t + \delta t) - l_e(t) \tag{54}$$

$$\sum_{i=0}^{m-1} x_i \delta y_{i+1} + \delta x_i y_{i+1} + \delta x_i \delta y_{i+1} - x_{i+1} \delta y_i - \delta x_{i+1} y_i - \delta x_{i+1} \delta y_i = \mathcal{A}(t + \delta t) - \mathcal{A}(t) \tag{55}$$

Still, we have an under-determined and nonlinear system of equations with many variables and dissimilar imposed constraints. This system is hard to solve algebraically, and the use of some geometric intuition can simplify matters considerably. First, notice that instead of requiring the deformation method to obey constraints on both perimeter and area, we can restrict it to only obey the correct shape factor, $4\pi A/L^2$. In this case, the shape can be scaled to regain the desired dimensions. Sections 5.1 and 5.2 introduce two new solutions to the above problem. Section 5.1, like Sect. 3 above, presents a method that is "global" in the sense that its effect is not restricted to a local polygonal region. Section 5.2 on the other hand follows the "local evolve global rescale" principle presented for the continuous case in Sect. 4.

5.1 Pentagonal Deformation Model for Polygonal Boundaries

The pentagon model is a perimeter preserving and area constrained contour deformation method that works by restricting the global polygonal boundary deformation to be a piece-wise Euclidean transformation. We shall hence geometrically transform the problem of solving (54) and (55) into a problem that can be completely algebraically addressed. In order to obey preservation of shape factor, let us designate by $\tilde{X}(t)$ a scaled version of $X(t)$, that is, $\tilde{X}(t) \triangleq X(t)/\lambda(t)$. The method proposed below will therefore be allowed to produce instantaneous evolution equations for $\tilde{X}(t)$ that do not obey the given $\mathcal{L}(t)$ and $\mathcal{A}(t)$ constraints, but instead, preserve $\tilde{X}$'s perimeter while its area deforms such that it obeys $A(\tilde{X}(t))/L^2(\tilde{X}(t)) = \mathcal{A}(t)/\mathcal{L}^2(t)$. The first new constraint, perimeter preservation, is given by $L(\tilde{X}(t)) = \mathcal{L}(0)$, the new desired area evolution rule is therefore:

$$\tilde{\mathcal{A}}(t) = \mathcal{L}^2(0)\mathcal{A}(t)/\mathcal{L}^2(t) \tag{56}$$

and the time varying scaling parameter is then given either by $\lambda(t) = \mathcal{L}(t)/\mathcal{L}(0)$ or by $\lambda(t) = \sqrt{\mathcal{A}(t)/\tilde{\mathcal{A}}(t)}$.

To start with, consider three consecutive contour points. If we require the two corresponding edges to stay rigid with respect to each other, the displacement of the middle point is determined by the motion of the edge connecting the first and the third points. That way, the number of variables in the problem is reduced by the two displacement variables of the middle point. We can similarly handle longer discrete curve segments by defining their motion via the motion of

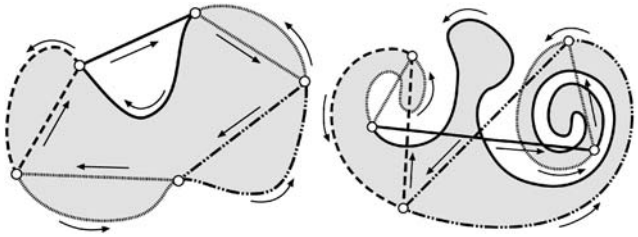

Fig. 5 Examples of arbitrary pentagons selections. Notice that the pentagon need not be convex and it can self intersect as long as it obeys increasing parametrization. The *arrows* point in the directions of the integrals shown in the first and second rows of (58), a different line type for each colum

the edge connecting their endpoints. Using this principle we reduce the original problem into the deformation of a planar polygon with five vertices only. At this point, as shown below, the problem becomes fully tractable algebraically.

A general deformation step of this method is defined by a set of five of the contour's points. The pentagon defined by edges connecting these five points (in the order of traversal of the boundary, i.e., increasing order of parametrization, the starting point being chosen at the first special point) need not be convex, and it can even self-intersect as shown in Fig. 5. The deformation of the original curve segments between every pair of consecutive pentagon points will be rigid and hence will be the Euclidean transformation defined by the motion of the corresponding pentagon edge (see e.g. Fig. 6). This, by no means restricts the domain of shapes for which this deformation can be applied or the range of shapes that can be eventually obtained. An example of perimeter and area preserving deformation is shown in Fig. 7. The Euclidean motion assures that the length of the curve segment remains constant and so is the (possibly negative) area between the curve segment and the pentagon edge. The contour is deformed by exploiting the parametric freedom in changing the pentagon's angles while the pentagon edges' lengths remain fixed. The overall perimeter is preserved because it is preserved by each of the five segments separately (and it is in all obeying the required perimeter constraint, $\mathcal{L}(t)$, since the shape is suitably scaled). The area is changing at the right pace because the only change in area arises from the change in the pentagon's area, and this can be designed to obey an instantaneous rate of $\tilde{\mathcal{A}}_t(t)$. Formally speaking, by denoting $a(p) \triangleq \mathbf{C}(p) \times \mathbf{C}_p(p)$, the area of the contour may be written as follows:

$$\begin{aligned} A(\mathbf{C}) &= \frac{1}{2}\oint_C a(p)\,dp \\ &= \frac{1}{2}\int_{p_1}^{p_2} a(p)\,dp + \frac{1}{2}\int_{p_2}^{p_3} a(p)\,dp + \cdots \\ &\quad + \frac{1}{2}\int_{p_5}^{p_1} a(p)\,dp \end{aligned} \tag{57}$$

To $A(\mathbf{C})$ we can add and subtract the similar integral over the edges of the pentagon (in the opposite direction, with any parametrization, $\tilde{p}$, and with $\tilde{a}(\tilde{p})$ denoting an infinitesimal area element, defined similarly to $a(p)$ on the edges of the pentagon):

$$\begin{aligned} &A(\mathbf{C}) \\ &= \int_{p_1}^{p_2} \frac{a(p)}{2}\,dp + \int_{p_2}^{p_3} \frac{a(p)}{2}\,dp + \cdots + \int_{p_5}^{p_1} \frac{a(p)}{2}\,dp \\ &\quad + \int_{\tilde{p}_2}^{\tilde{p}_1} \frac{\tilde{a}(\tilde{p})}{2}\,d\tilde{p} + \int_{\tilde{p}_3}^{\tilde{p}_2} \frac{\tilde{a}(\tilde{p})}{2}\,d\tilde{p} + \cdots + \int_{\tilde{p}_1}^{\tilde{p}_5} \frac{\tilde{a}(\tilde{p})}{2}\,d\tilde{p} \\ &\quad + \int_{\tilde{p}_1}^{\tilde{p}_2} \frac{\tilde{a}(\tilde{p})}{2}\,d\tilde{p} + \int_{\tilde{p}_2}^{\tilde{p}_3} \frac{\tilde{a}(\tilde{p})}{2}\,d\tilde{p} + \cdots + \int_{\tilde{p}_5}^{\tilde{p}_1} \frac{\tilde{a}(\tilde{p})}{2}\,d\tilde{p} \end{aligned} \tag{58}$$

Notice that the first terms of the first and second rows express together the area of the region bounded between the first pentagon edge and the corresponding curve segment. In the same way, the second terms of these rows express the area between the second pentagon edge and its corresponding segment, and so forth for all five term-pairs. Notice also that the third row is a piecewise integral expressing the area of the pentagon itself. Having said that, and denoting by $A_i,\ i = 1, \ldots, 5$ the signed areas bounded by the pentagon edges and the contour, we can write:

$$A(\mathbf{C}) = \sum_{i=1}^{5} A_i + A_{pentagon} \tag{59}$$

Since the five regions with areas A_i discussed above are "attached" to the corresponding pentagon edges and undergo only Euclidean, hence area preserving, transformations, $\sum_{i=1}^{5} A_i$ is constant during the deformation. It is clear therefore that all the change in the area of the contour results from the change in the area of the pentagon.

The next issue that we shall address is the freedom we have in deforming the pentagon. A pentagon is defined by ten parameters; two for each of its vertices. since we only deal with a shape's deformation, we can disregard its position and orientation by choosing a coordinate system whose origin is fixed to the first pentagon point and the x-axis points in the direction of the edge to the second pentagon point. The seven parameters left are subject to five edge length constraints and another constraint for the pentagon's area, hence a single degree of freedom. How is the pentagon deformed? As laid out in [12], changing one of its angles explicitly determines all others, that is, completely determines the pentagon's shape up to Euclidean transformation. Deforming the pentagon and setting the Euclidean motion it undergoes, determines $\{(\delta x_i, \delta y_i)\}$ for the five pentagon points. All other $\{(\delta x_i, \delta y_i)\}_{i=0}^{m-1}$ displacements are calculated by applying to each of these points the Euclidean

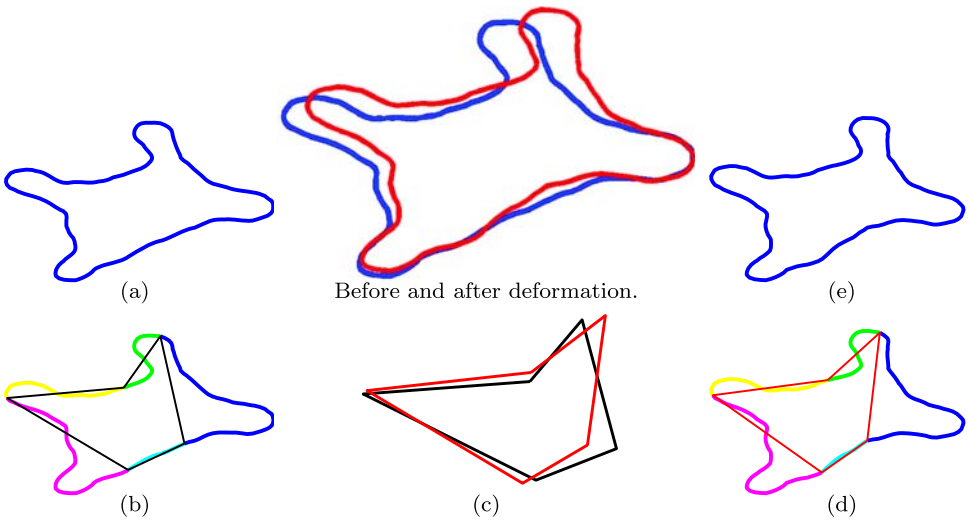

Fig. 6 Pentagon shape deformation: (**a**) Contour before deformation. (**b**) Choice of pentagon defines curve segments. (**c**) Pentagon deformation. (**d**) Curve segments repositioned on pentagon. (**e**) Contour after deformation

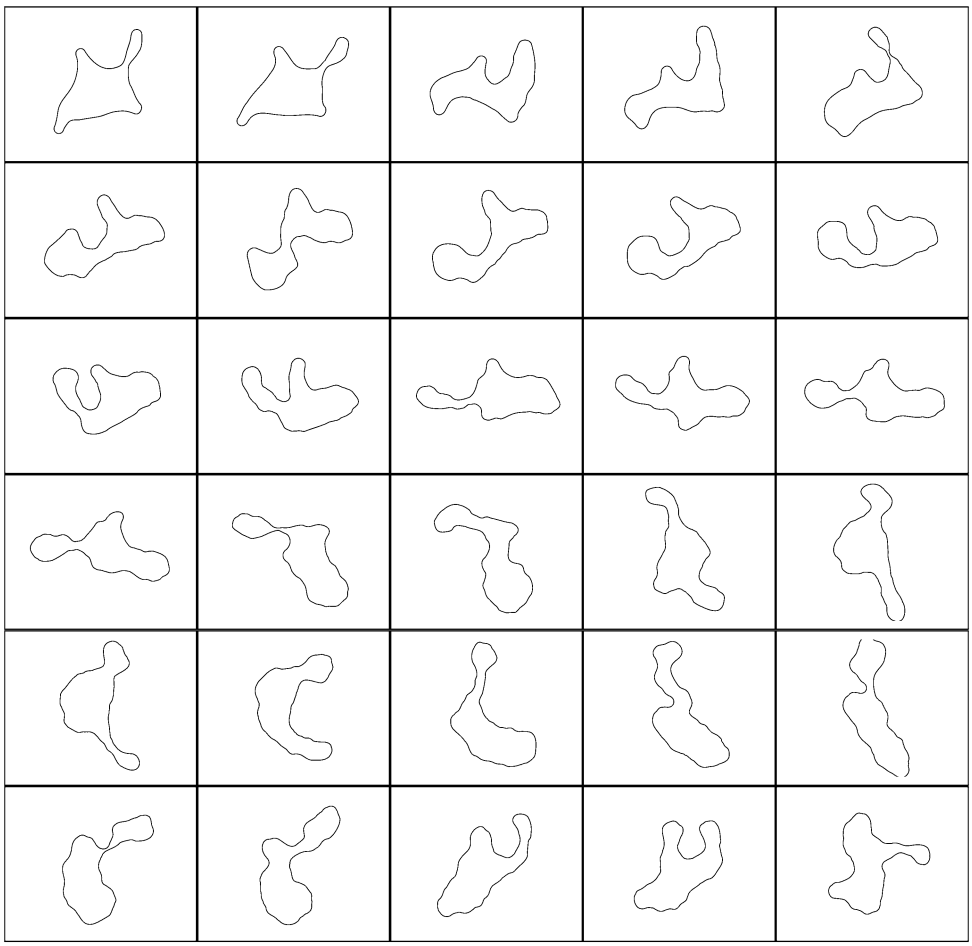

Fig. 7 An example of a random perimeter and area preserving shape deformation produced via the pentagonal deformation method

transformation that produced the displacement of the corresponding pentagon edge.

However, why use a pentagon? Why not rely on polygons with fewer or with more edges? It turns out that a pentagon is the simplest polygon allowing area and edge length preserving deformations. Clearly, higher order polygons could also be used. They however require one more parameter for each additional edge above five, and will enable more complex deformations at the expense of much higher complexity.

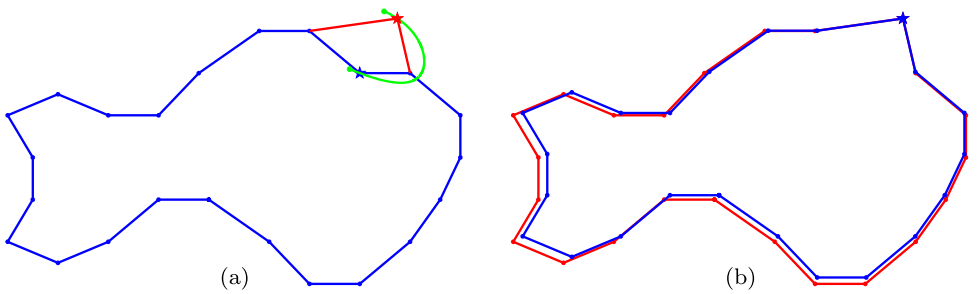

Fig. 8 (**a**) The *green curve* marks the locus of points onto which the *blue* stared point can be moved satisfying the desired shape factor constraint for the special case of perimeter and area preserving deformation. (**b**) The *blue contour* is the result of scaling the *red contour* about the new position of the stared point

The pentagonal deformation model can be applied to both discrete and continuous curves. It has the advantage of preserving the curve's parametrization, however, it does not preserve the curve's smoothness since directional discontinuities are introduced at the vertices of the selected pentagon. The smoothness breaks do not raise any problem in the context of polygonal curves, but in general, the local changes should be minimized at each step. There are two ways to ensure "smoothness". First, the deformation may be applied via (infinitesimally) small steps ensuring that the selected points are barely moving and none of them undergoes a significant distortion or bending. Another option is to follow a significant, global, deformation step by a local process of curvature minimization via pentagons defined over five consecutive curve points (see Sect. 7 for relevant optimization techniques).

5.2 Moving One Point at a Time

For a polygonal contour, another way to express an infinitesimal deformation is via the "local evolve global rescale" principle, that is, to move a single point at a time, followed by a second, rescaling, stage that affects all points, see e.g. Fig. 8. For every point, X_i, there is a one dimensional locus of points, a planar curve, along which it can be moved such that after rescaling, the contour obeys its desired perimeter and area constraints. Conveniently, this curve can be explicitly expressed (details in [12]).

The rescaling stage itself also contains some degree of freedom that can be employed to our advantage. The choice of an origin point about which the shape is scaled is equivalent to the choice of a translation vector. The origin may be designed to keep the shape's center of mass fixed and thereby eliminate the global motion associated with the local evolution. Alternatively, rescaling around the locally evolved area creates a more intuitive and natural shape flow effectively modelling random locomotion.

6 Constrained Deformation of Splinegons

A shape's boundary may also be expressed as a *splinegon*, i.e., a spline guided by a polygon of control points. Several studies propose methods that enable enforcing various global constraints to the intuitive approach of contour deformation via control points editing. A B-spline representation is appropriate because of its locality properties (moving a single control point has a limited domain effect). Elber [6] presents a method for multiresolution editing of B-spline curves subject to various linear constraints. Then, he expands this method to the area constraint. The area of a B-spline contour may be written via $A = \sum_i \sum_j X_i Y_j B_{ij}$ where $\{(X_i, Y_i)\}$ are the control points and $\{B_{ij}\}$ are constant coefficients (pre-computed integrations over the B-spline basis functions in use). Elber suggests that this bilinear equation can be dealt with as a pair of interchangeable linear constraints; one for $\{X_i\}$, one for $\{Y_i\}$. In [24], Sauvage et al. present a fast and detail preserving generalization of the above work for deforming 3D B-spline surfaces subject to volume constraints.

Another useful curve description method is via multiresolution representation. Sauvage et al. suggested methods for multiresolution editing of curves such that either their perimeter [21, 23] *or* their area [22] is preserved. In all those methods, a deformation is applied to the curve by modifying any control point at any resolution level. Then, the curve is iteratively optimized for smoothness and minimal deviation from its (modified) shape subject to perimeter or area constraints. However, these authors never dealt with contour deformations subject to both perimeter and area constraints.

Following the "local evolve global rescale" principle once again, we suggest to deform a contour by moving one or more control points in a way that obeys the desired shape factor, $4\pi \mathcal{A}(t)/\mathcal{L}^2(t)$. As before, this shape can later be scaled (by scaling the control points) to the required perimeter, $\mathcal{L}(t)$, and area, $\mathcal{A}(t)$. A control point position has two degrees of freedom, X_k and Y_k, while the area vs. length ratio imposes only a single constraint. As in Sect. 5.2, it turns out that for every control point there is a planar curve along which it can be moved while obeying the desired shape factor constraint (and of course, the curve passes through the point for shape factor preserving deformations). Again, there are several possible parameterizations describing points along this curve. The scale factor, λ, is a natural

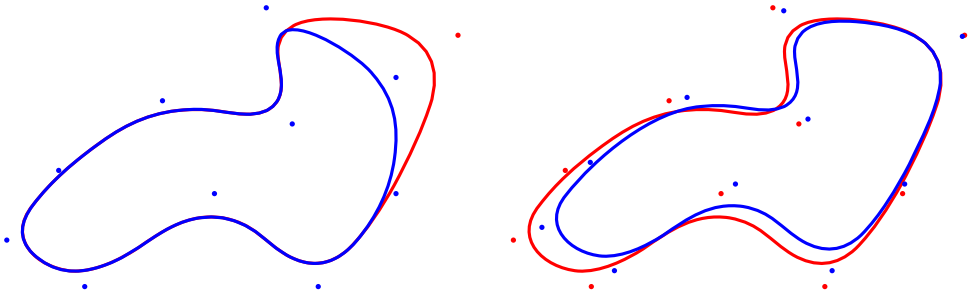

Fig. 9 One step of perimeter and area preserving shape deformation via B-spline control point editing. An *up-right* motion of the *top right* control point is followed by a rescaling of all control points centered around the "active region"

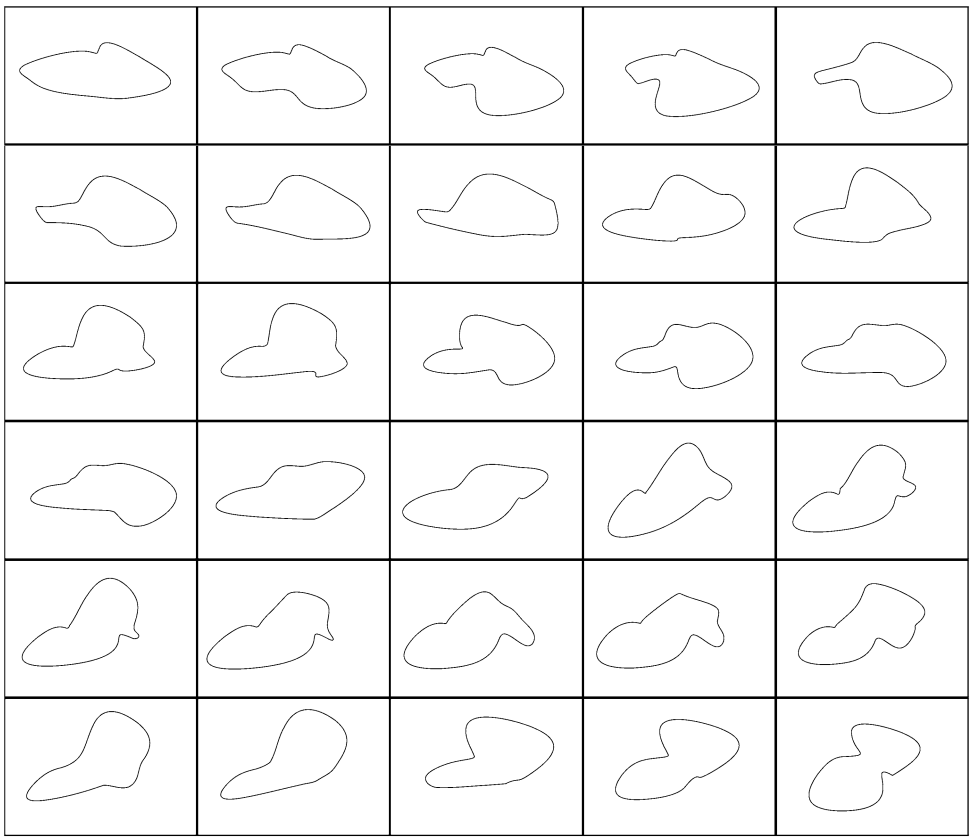

Fig. 10 Perimeter and area preserving random shape deformation via B-spline control point editing. The *above cubic* B-spline contour is defined using twelve control points

choice for that single degree of freedom, but the step size or direction can also be used. Unfortunately, solving for the locus of control point positions, $(X_k(\lambda), Y_k(\lambda))$, is by no means explicit or trivial.

As seen in [6, 27], the area constraint leads to linear equations in X_k and Y_k. However, the perimeter constraint imposes a need for a numerical solution for the control point's displacement. An example of shape deformation via B-spline control points editing is shown in Figs. 9 and 10.

7 Optimal Shapes

In many applications we are interested in shapes that optimize a certain quality measure under various constraints. It is sometimes assumed that cell membranes, under surface area and volume constraints, seek to minimize their total bending energy, a surface integral over the square of the mean curvature, $E = \iint_S M^2\,dA$, see e.g. [16]. In aeronautics, wing designers try to maximize a wing's lift force under drag, weight, size and strength constraints, etc. Our general approach to constrained shape deformations is useful in obtaining some of these optimal shapes. The deformation methods enable us to evolve arbitrary shapes toward optimal ones while preserving their perimeter and area.

An interesting recent work done by Arreaga et al. [1], Capovilla et al. [2] studied the shapes of contours having minimal boundary bending "elastic energy" subject to constraints of constant perimeter and area. The functional for minimization is defined by $\mathcal{F}(\mathbf{C}) \triangleq \oint_C K^2(s)\,ds$. The constraints are added to the functional via Lagrange multipliers,

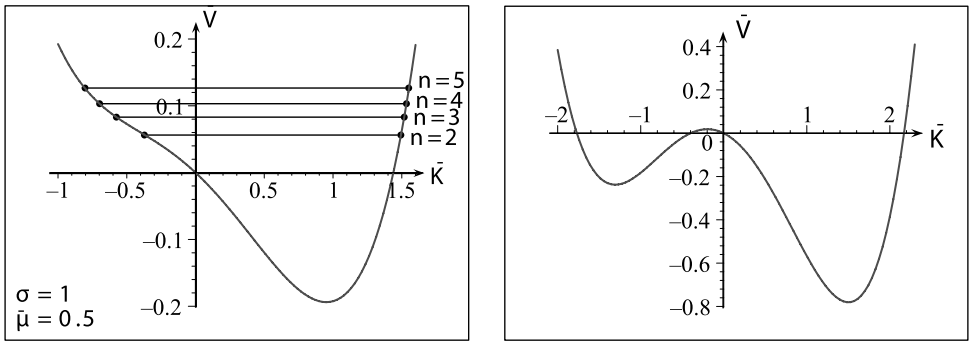

Fig. 11 The *curve* of potential, $V(K)$, takes one of the two possible types of curves shown above. The *curve on the left* shows some of the discrete set of energy levels corresponding to *closed curves* (curves taken from Arreaga et al. [1])

as follows:

$$\tilde{\mathcal{F}}(\mathbf{C}) = \mathcal{F}(\mathbf{C}) + \mu\big(L(\mathbf{C}) - \mathcal{L}\big) + \sigma\big(A(\mathbf{C}) - \mathcal{A}\big) \tag{60}$$

The starting point of the analysis is the *Euler-Lagrange* equation satisfied by these optimal shapes (see e.g. [17]):

$$2K'' + K^3 - \mu K - \sigma = 0 \tag{61}$$

Then, an interesting way of solving this complicated differential equation of the fourth degree in stages is proposed. Since a plane curve is determined, up to rigid motions, by its curvature, the first stage of the solution can be to solve for the curvature in terms of the arc-length. This step is done by rewriting (61) in the following form:

$$K'' = -\frac{d}{dK}V(K), \quad V(K) \equiv \frac{1}{8}K^4 - \frac{\mu}{4}K^2 - \frac{\sigma}{2}K \tag{62}$$

This is a very clever mapping of the optimal contours problem into a problem of determining the motion of a fictitious particle in a quartic potential, K being the displacement of the particle and s playing the role of time (notice that the same "trick" can be used in solving other problems as long as their equation of motion is of the form $K'' = f(K)$). As in the physical world, the "total energy" is defined by the sum of the kinetic and potential energy (taking a unit mass),

$$E = \frac{1}{2}K'^2 + V(K) \tag{63}$$

and we can see that it is conserved along the motion of the fictitious particle, as follows:

$$\begin{aligned}\frac{dE}{ds} &= \frac{d}{ds}\left(\frac{1}{2}K'^2 + V(K)\right) \\ &= K'K'' + \frac{dV}{dK}\frac{dK}{ds} = K'K'' + \big(-K''\big)K' = 0\end{aligned} \tag{64}$$

The optimal, stationary, shapes correspond to the motion of the fictitious particle on the curve of potential, $V(K)$. Every selection of the μ and σ parameters produces a different potential curve. Two such curves reproduced from [1] are shown in Fig. 11. Different potential curves result in different optimal contours. In addition, for any given potential curve, there is an infinite but discrete set of possible "total energy" levels, each corresponding to a different stationary contour. (This set is discrete because it consists only of the energy levels that correspond to closed curves. The feasible energy levels are found via a numerical procedure.) Each such (μ, σ, E_i) triplet corresponds to a different equilibrium shape. For each triplet, $K(s)$ is obtained via a numerical integration process over the equation of motion of the particle in the "potential-well". Then, a second numerical integration stage produces $\mathbf{C}(s)$ from $K(s)$ up to Euclidean transformations. Figure 12 shows a few examples of optimal contours so obtained, again reproduced from [1]. When the fictitious particle is given a higher total energy (a feasible energy level other than the lowest possible, not to be confused with the contour's bending energy), the corresponding optimal contour is more complex and has a higher type of symmetry as can be seen in Figs. 12b–d, f–h.

Note that the technique proposed above does not enable setting the perimeter and the area of the minimal shape we look for. The only parameters it allows to adjust freely are μ and σ and these affect the perimeter and the area via an implicit function that is, in fact, not even continuous. We here propose a direct method of numerically obtaining optimal shapes of a priori set perimeter and area. Given the set of global constraints, we propose to start with an arbitrary selected initial shape having the exact perimeter and area requirements. Notice that such shapes are easily found since, for example, we can take as an initial shape a rectangle or an ellipse satisfying these conditions; a task that is equivalent to finding the right horizontal and vertical scaling factors of the unit square or circle. As can be seen in Fig. 13, given an initial shape, it is deformed toward optimality under constraints of constant perimeter and area. The deformation methods that are more suitable for this process are the "global" ones, like the basis-functions curve evolution (Sect. 3) and the pentagonal deformation (Sect. 5.1). In each of these methods, at each iteration of the flow, a small set of parameters is randomly selected to describe the deformation. For the basis-functions evolution, a set of basis

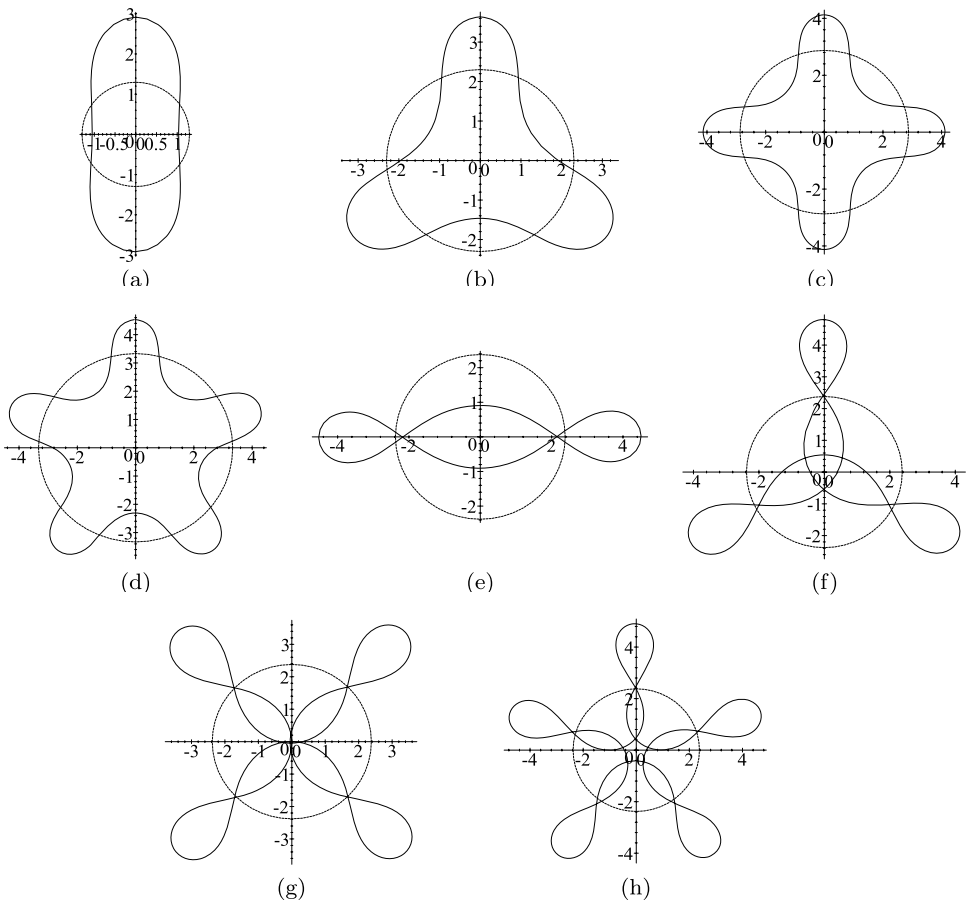

Fig. 12 Optimal shapes in terms of bending elastic energy under perimeter and area constraints (taken from Arreaga et al. [1]). Figures (**b–d**) and (**f–h**) have a high order symmetry and they correspond to a fictitious particle given a higher energy. As seen in Figs. (**e–h**), certain values of μ and σ cause the contour to self intersect

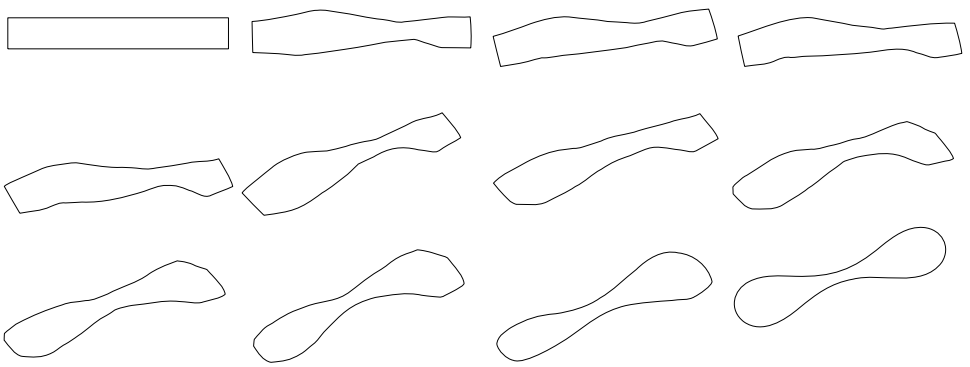

Fig. 13 An optimization process initialized with a rectangle. The rectangle's edges are easily set to obey desired perimeter and area constraints

functions is determined and while some of the coefficients are randomly set, other coefficients may be varied to optimize the bending energy functional (a third group of coefficients is determined by satisfying the global constraints of (17), (18) and (19)). For the pentagonal method, five random boundary points are selected, leaving a single degree of freedom in the deformation of the pentagon. For all deformation methods, in each step of the optimization process, setting the values of the parameters defining the instantaneous deformation may take several forms. It can be a unit step in the negative energy gradient direction or a full gradient descent process determining the values that minimize the energy the most. It can also be an annealing like process in which values are randomly drawn and are then accepted at probabilities that depend on their contribution to the bending energy measure. All these processes are repeated with different selections of parameters resulting in the evolution (or convergence) of shapes toward stationary optimal shapes. Figure 14 shows some "optimal" shapes we obtained via the pentagonal deformation method.

For a given and fixed perimeter, as the constrained area decreases, the optimal shapes become the ones shown in

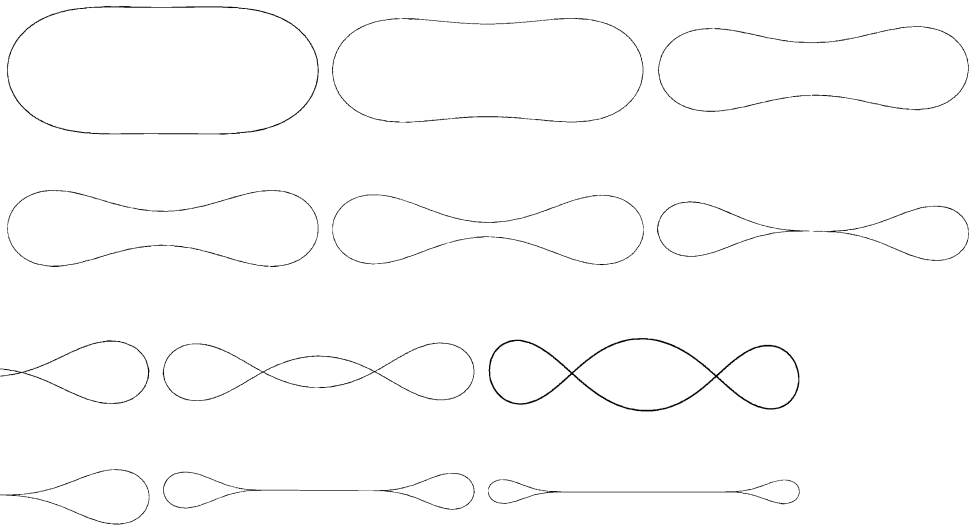

Fig. 14 Curves of minimal elastic energy under area and length constraints. Shapes were obtained using the pentagonal deformation method

Fig. 15 Curves of minimal elastic energy under area and length constraints as obtained using the pentagonal deformation method. Relatively to Fig. 14, the above shapes have a lower shape factor ($4\pi A/L^2$). Shapes in the *top row* were allowed to form self intersections and so they all produce negative area regions (*middle pieces*). Shapes in the *bottom row* were not allowed to self intersect and so they preserved their initial shape topology

Fig. 15. In the top row, we show the curves that were obtained when we allowed the curves to cross themselves. In that case, the shapes resemble the ones obtained by Arreaga et al. in [1], that is, they approximately obey the Euler-Lagrange expression of (61). The shapes shown in the bottom row were not allowed to self-intersect during the optimization flow, and so, they converged to "optimal" non-self-intersecting shapes that could not be obtained via the method presented in [1], and cannot obey the Euler-Lagrange expression anymore.

Figure 16 presents more complex shapes. These optimal shapes are results of optimization flows in cases where the initial shapes had numerous self intersections. Notice that if a contour's *turning number* is defined by the (integer) number of cycles around the origin completed by its unit tangent, these shapes have a turning number that is not restricted to one (actually, the turning number is always one less than the number of "leaves" growing out of the central region of the contour). Similar shapes were not presented in [1], since (11) therein restricts the analysis to unit turning number contours. It restricts the angle by which the tangent gets rotated in one full oscillation of the particle to be $2\pi/n,\ n = 2, 3, \ldots$. By allowing this angle to be $2\pi m/n$, $n = 2, 3, \ldots, m = 0, 1, \ldots$, more complex optimal shapes obeying the Euler-Lagrange conditions may be obtained, resulting in shapes like those presented in Fig. 16.

The shapes having complex symmetries as presented by Arreaga et al. (and reproduced here in Fig. 12b–d, f–h) are optimal in the sense that they obey the Euler-Lagrange equation. However, they correspond to local rather than global minima. These shapes are known to have higher bending

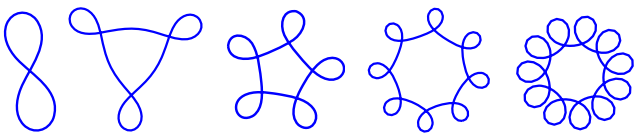

Fig. 16 Minimal elastic energy under area and length constraints. Initial contours had numerous self intersections and resulting optimal shapes have a winding number that is not one

energies than the peanut-like shapes of equal perimeter and area, [1], shapes similar to those seen in Fig. 14. The higher order symmetry shapes correspond to stationary points that are so sensitive (in the sense of being local minima with a very small attraction basin) that they practically never result in an optimization process. Even when we initialized the optimization processes with "locally optimal" shapes of a higher type of symmetry (sampled from Fig. 12) and used very slow and smooth deformations, the optimization processes readily deformed these shapes toward peanut-like shapes of the same dimensions, as can be seen in Fig. 17.

So far, we have discussed optimal shapes as a function of given perimeter and area. Alternatively, these shapes can be considered via their shape factor and a scaling term. Since scaling a contour by λ scales it perimeter by λ and its instantaneous curvature by λ^{-1}, its bending energy, $\oint K^2\,ds$, is scaled by $\lambda(\lambda^{-1})^2 = \lambda^{-1}$. Therefore, we can introduce a normalized measure of the bending energy by multiplying $\mathcal{F}$ by the curve's perimeter. This scale invariant energy measure was introduced by Weiss [28] and by Bruckstein and Netravali [3]. For optimal shapes, this measure clearly de-

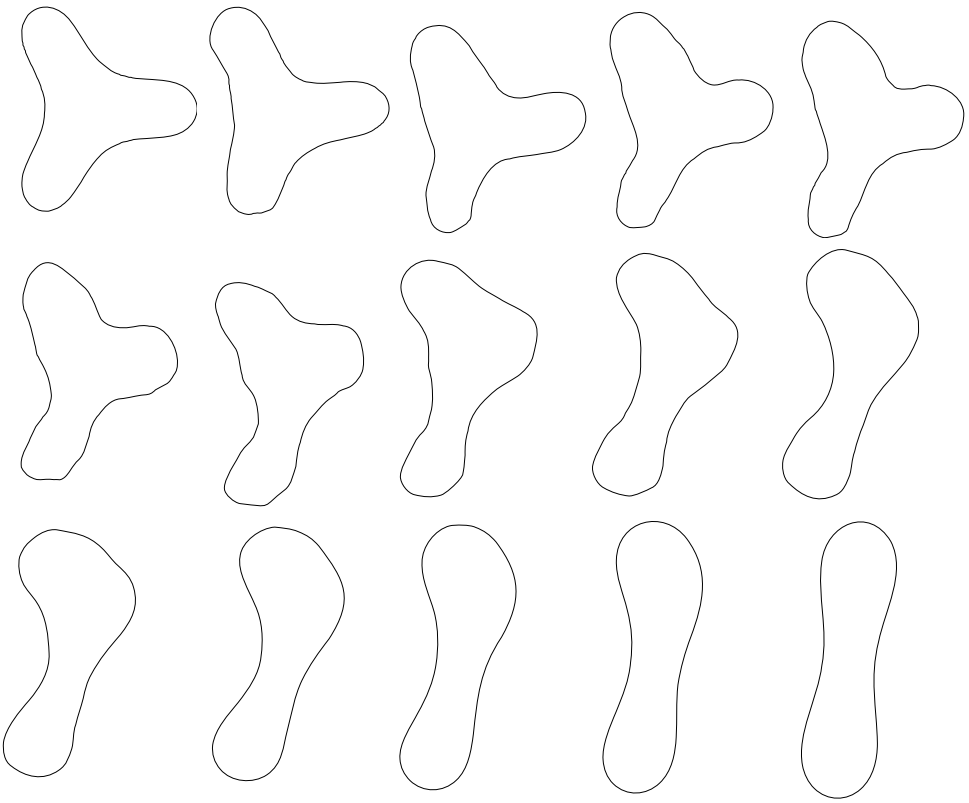

Fig. 17 An optimization process initialized with a higher order "locally optimal shape" as sampled from Fig. 12. Notice that the shape flows toward the global minimizer peanut

pends only on the shape factor, $4\pi A/L^2$, that is,

$$\mathcal{F}_{norm} \triangleq \big(\mathcal{F}(L,A)\big)L = f\left(\frac{4\pi A}{L^2}\right) \tag{65}$$

The methods for obtaining optimal shapes discussed above enabled us to experimentally obtain a function describing this normalized bending energy as a function of the shape factor as seen in Fig. 18. A similar function was discussed by Arreaga et al. [1] (see Fig. 14 therein). However, only the horizontal and vertical asymptotes of this function could be analytically derived, and the function itself was drawn by asymptote-directed interpolation.

Our experimentally derived curve of Fig. 18 might turn out to be important in understanding some real life phenomena. It turns out that vesicles are seen in cells in quite a variety of sizes and morphologies [14, 15]. Under certain conditions, a large vesicle might split and produce smaller vesicles which might then split at their turn. In fact, a process of vesicle splitting due to volume reduction can be analyzed in conjunction with the tendency of membranes to minimize their boundary bending energy. For simplicity, consider a two dimensional vesicle of perimeter L and area A. Its minimal boundary bending energy is given by $\mathcal{F}(L,A) = f(4\pi A/L^2)/L$. If the vesicle's area decreases while its perimeter remains the same, say, due to a change in the environment's osmotic pressure, its elastic energy increases considerably, and the vesicle becomes less stable, more likely to break and reshape. Indeed, as can be seen in Figs. 15 and 18, the optimal configurations for low shape factor contours (i.e. small area) are shapes in which membranes from opposite sides come very close together. The membranes in these shapes divide the vesicle into two separate sides where transport of matter between them is practically impossible. Suppose the closest opposite points of the membrane attract each other and stick together inducing a vesicle split, this process will generate two new vesicles with a twice higher shape factor ($A \to \frac{A}{2}, L \to \frac{L}{2} \Rightarrow \frac{4\pi A}{L^2} \to \frac{4\pi(A/2)}{(L/2)^2} = 2\frac{4\pi A}{L^2}$). This process can of course reoccur upon further area decrease; each of the halves can split, and so on.

Using deformation methods similar to those described above we can optimize shapes with respect to other criteria. Examples of possible objective functionals can be integrals over higher order curve derivatives ($\oint k'(s)\,ds, \ldots$) and/or associated powers ($\|k\|, k^3, \ldots$), moment of inertia ($\oint r^2\,da$) and other moments, potential functions due to various external forces, and more.

We note that such methods may also allow the study of the forces and laws governing the formation of certain real world shapes. Given an object of some (possibly temporary) "optimal shape", a "reverse engineering" process may be able to retrieve its energy functional. Take the shape of a vesicle for example, besides a desire to minimize bending energy, vesicle shape is also influenced by osmotic pressures, adhesion forces, and the electric properties of the

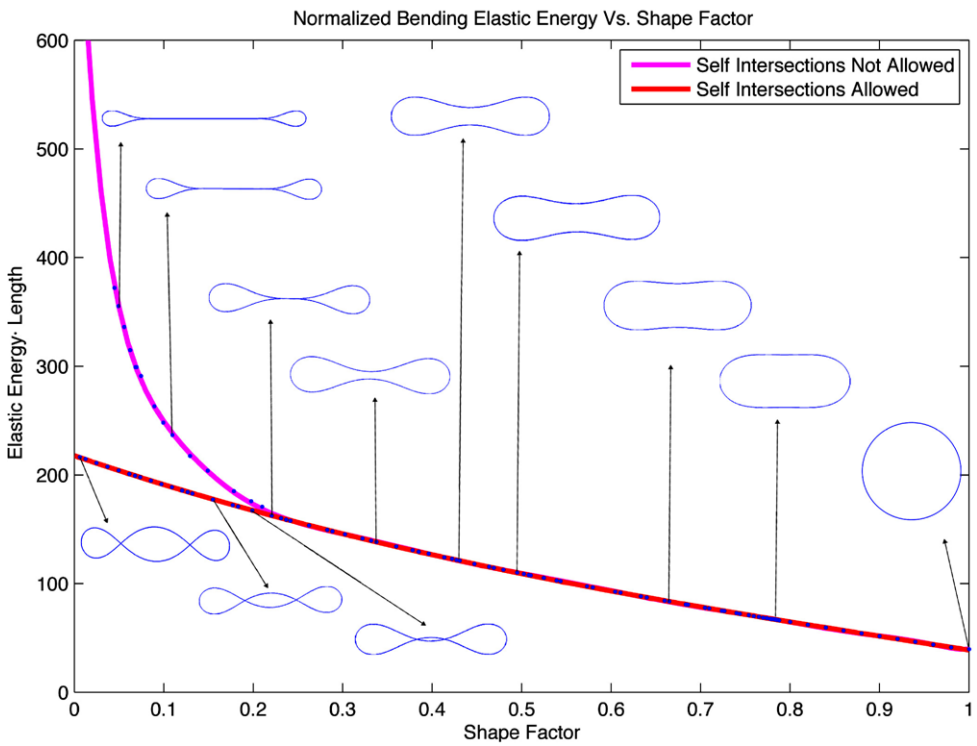

Fig. 18 A *curve* describing the normalized bending energy of optimal shapes as a function of their shape factor (optimal shapes are ones which minimize their bending energy under perimeter and length constraints). *Each point on the curve* corresponds to a certain optimal shape and any arbitrary scaled version of it

membrane's molecules [16]. Given the shape of a vesicle, it may be possible to figure out the forces acting on it. Expressing the forces parametrically defines a function that measures, for any set of parameters, the deviation of an optimal shape with respect to the functional induced by these forces from the target shape. This function may be minimized yielding the true forces present.

8 Running Times

The implementation of all deformation methods was done in Matlab^{TM} over an intel® core^{TM}2 Duo CPU with 2 G of RAM. The "local evolve global rescale" level sets amoebae evolved over a 200 by 200 pixels grid at about eight frames (deformation steps) per second. It is safe to assume that an optimized C/C++ code using a narrow band [26] could easily generate "real time" smooth deformations. The same statement holds for the pentagonal deformation model that evolved 300 edges polygonal amoebae at over 80 steps per second.

Shape optimization is a computationally heavier process. A rectangular polygon of 50 by 100 edges sides converged into an approximately optimal shape using the pentagonal deformation method in about six seconds. This process takes longer than a random walk with the same number of steps because each pentagonal step involves a one dimensional numeric optimization. A smoother initial shape, e.g. an ellipse, converges much slower because of the limited advancement made by every pentagonal step.

9 Generalizations

The most obvious generalization of perimeter and area constrained planar shapes is that of volume and surface area constrained 3D shapes. Some of the deformation methods proposed above may be adjusted to enable the deformation of such 3D shapes while others may not. The same principles that were used for continuous curve evolution methods in Sects. 3 and 4 may be used to state constraints on $\varepsilon(s1, s2)$, a bivariate function describing the evolution rule of a closed 3D surface in the direction of its normals. These constraints may be written as a series of basis functions yielding linear constraints over the coefficients (as in Sect. 3), or they may be satisfied by a two dimensional bump function, some natural flow (mean curvature flow for instance), and a scaling term (as in Sect. 4). The pentagonal deformation method on the other hand has no obvious 3D generalization.

The perimeter and area constraints are certainly not the only global constraints that may be enforced over a deforming planar shape. Other constraints may include a shape's moment of inertia, $(\iint r^2\,da)$, or higher order moments, average curvature, boundary bending elastic energy, or other accumulative curve derivatives and/or associated powers, curve or area integrals of external force fields, fixed curve points and/or tangents, etc. Some of the deformation methods proposed above may be easily adjusted to follow other constraints. The boundary bending energy constraint for example yields the following constraint on the $\varepsilon(s)$ deforma-

tion function (as in (17) and (18)):

$$\oint_C 2K^3\varepsilon(s) - \varepsilon_{ss}(s)\,ds = 0 \tag{66}$$

This equation can be treated in the same manner as (12) and (13) for the perimeter and the area. It can either be written as a linear constraint over basis functions coefficients as in (17) and (18), or it can be used to set the coefficients of a "natural" and a scaling flow terms that are used in order to overcome the effect of a local bump function evolution as in Sect. 4.

The number of global constraints in use may also be altered. Such changes will of course require some adjustments, for example, replacing the pentagon with a higher order polygon (one additional edge for each new constraint) or adding linear constraints over the basis functions coefficients in the continuous curve evolution method of Sect. 3.

10 Concluding Remarks

This work presented a series of methods for carrying out perimeter and area constrained planar shape deformations. The methods studied involved both continuous and discrete (polygonal and splinegonal) shape representations and highlighted the difficulties in dealing with the complex interactions between local rules of deformation and global constraints. One application we studied is that of globally constrained shape optimization processes that can be implemented using goal-directed gradient-descent annealing-type shape deformation processes. We hope that this study will lead to further work in this area.

References

1. Arreaga, G., Capovilla, R., Chryssomalakos, C., Guven, J.: Area-constrained planar elastica. Phys. Rev. E **65**(3), 031801 (2002)
2. Capovilla, R., Chryssomalakos, C., Guven, J.: Elastica hypoarealis. Eur. Phys. J. B **29**, 163–166 (2002)
3. Bruckstein, A.M., Netravali, A.N.: On minimal energy trajectories. Comput. Vis. Graph. Image Process. **49**(3), 283–296 (1990)
4. De Boor, C.: A Practical Guide to Splines. Springer, Berlin (1978)
5. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces, pp. 33–35. Prentice Hall, New York (1976), Chap. 1
6. Elber, G.: Multiresolution curve editing with linear constraints. In: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA '01, pp. 109–119. ACM, New York (2001)
7. Epstein, C.L., Gage, M.: The curve shortening flow. In: Chorin, A., Majda, A. (eds.) Wave Motion: Theory, Modelling, and Computation, pp. 17–59. Springer, Berlin (1987)
8. Farin, G.E.: NURBS: From Projective Geometry to Practical Use. AK Peters, Natick (1999)
9. Gage, M.: On an area preserving evolution equation for plane curves. Contemp. Math. **51**, 51–62 (1986)
10. Gage, M., Hamilton, R.S.: The heat equation shrinking convex plane curves. Differ. Geom. **23**, 69–96 (1986)
11. Grayson, M.A.: The heat equation shrinks embedded plane curves to round points. Differ. Geom. **26**, 285–314 (1987)
12. Goldin, I., Delosme, J.-M., Bruckstein, A.M.: Vesicles and amoebae: on globally constrained shape deformation. Tech. Report, Center for Intelligent Systems, Computer Science Dep. Technion, Haifa, Israel (2008)
13. Han, X., Xu, C., Prince, J.L.: A topology preserving level set method for geometric deformable models. IEEE Trans. Pattern Anal. Mach. Intell. **25**(6), 755–768 (2003)
14. Képès, F.: Secretory compartments as instances of dynamic self-evolving structures. Acta Biotheor. **50**(4), 209–221 (2002)
15. Képès, F., Rambourg, A., Satiat-Jeunemaître, B.: Morphodynamics of the secretory pathway. Int. Rev. Cytol. **242**, 55–120 (2005)
16. Lipowsky, R.: Vesicles and biomembranes. Encycl. Appl. Phys. **23**, 199–222 (1998)
17. Malcolm, M.A.: On the computation of nonlinear spline functions. SIAM J. Numer. Anal. **14**(2), 254–282 (1977)
18. Okabe, S.: The motion of an elastic closed curve with constant enclosed area. Var. Probl. Relat. Top. **1405**(15), 197–213 (2006)
19. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
20. Sapiro, G., Tannenbaum, A.: Area and length preserving geometric invariant scale-spaces. IEEE Trans. Pattern Anal. Mach. Intell. **17**(1), 67–72 (1995)
21. Sauvage, B., Hahmann, S., Bonneau, G.-P.: Length preserving multiresolution editing of curves. Computing **72**(1–2), 161–170 (2004)
22. Sauvage, B., Hahmann, S., Bonneau, G.-P.: Area preserving deformation of multiresolution curves. Comput. Aided Geom. Des. **22**(4), 349–367 (2005)
23. Sauvage, B., Hahmann, S., Bonneau, G.-P.: Length constrained multiresolution deformation for surface wrinkling. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06). IEEE Comput. Soc., Washington (2006)
24. Sauvage, B., Hahmann, S., Bonneau, G.-P., Elber, G.: Detail preserving deformation of B-spline surfaces with volume constraint. Comput. Aided Geom. Des. **22**, 678–696 (2008)
25. Schoenberg, I.J.: Cardinal Spline Interpolation. Society for Industrial and Applied Mathematics, Philadelphia (1973)
26. Sethian, J.A.: Level, Set Methods and Fast Marching Methods—Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge, UK (1999)
27. Ueda, K.: Signed area of sectors between spline curves and the origin. In: Proceedings of the 1999 International Conference on Information Visualisation, pp. 309–314. IEEE Comput. Soc., Washington (1999)
28. Weiss, I.: 3D shape representation by contours. Comput. Vis. Graph. Image Process. **41**(1), 80–100 (1988)

Ishay Goldin received B.Sc. and M.Sc. degrees in Computer Science from the Technion, Israel Institute of Technology, in 2005 and 2008, respectively. His M.Sc. research was on modeling area and perimeter constrained shape evolutions and optimization, and he earned his degree with "Summa Cum Laude" for his work. Since 2008 he is an algorithm designer at Surgix Ltd., a start-up company specializing in the field of medical image processing. His research interests are in image and signal processing and analysis, computer vision, and applied geometry.

Jean-Marc Delosme received the engineer's degree and the specialization in automatic control degree from the École Nationale Supérieure de l'Aéronautique et de l'Espace, France, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University.
Dr. Delosme joined the faculty of the Department of Electrical Engineering at Yale University in 1983 after a year as a research associate at the Information Systems Laboratory, Stanford University. Since 1996 he is Professor of Computer Science at the University of Evry, France. His current research interests are in computational issues in image processing, 3D flow simulations, optimizations for the design of streaming applications, and the robust control of flows within embedded systems or biological networks.

Alfred M. Bruckstein received the B.Sc. and M.Sc. degrees in Electrical Engineering, from the Technion, Israel Institute of Technology, in 1977 and 1980, respectively. His M.Sc. thesis dealt with models of coding in the nervous system. Then he joined the Information Systems Laboratory in the Electrical Engineering Department at Stanford University, in California, working on topics in direct and inverse scattering and signal processing and earned a Ph.D. degree in Electrical Engineering with a thesis on Scattering Models in Signal Processing.
From 1984 he has been on faculty at the Technion, Haifa, and presently he is holder of the Ollendorff Chair in Science. He served from 2002 to 2005 as the Dean of the Technion Graduate School, and is now the Head of the university's Excellence Program.
In the autumn of 1981 he visited the Physics Department of Groningen University, with the Biophysics Group, analyzing the responses of movement-sensing giant neurons in the visual system of the fly. In the summers of 1980 and 1982 he visited the Massachusetts Institute of Technology, at the Man-Vehicle Laboratory and the Information and Decision Systems Laboratory, respectively. From 1987, until 2001, Professor Bruckstein was a permanent visiting Member of Technical Staff at Bell Laboratories at Murray Hill, NJ, and he spent the years 1995–1997 there on a sabbatical leave from the Technion. At Bell Labs he was involved in image processing and analysis research, doing work on omnidirectional cameras, watermarking methods, PCB and VLSI inspection processes and holographic image representations. In 2002–2003 he was a Visiting Chaired Professor at Tsinghua University in Beijing, and is, during 2009–2010, a Visiting Professor at the Nanyang Technological University in Singapore, in the School of Physical and Mathematical Sciences, and at the Institute for Media Innovations. He also spent several summers in Paris, visiting Evry University and CEREMADE at Paris-Dauphine, working on various topics in Image Analysis for biological applications.
His research interests are in distributed, ant-robotics, and multi-agent systems, image and signal processing and analysis,pattern recognition, and various aspects of applied geometry. He has also done work, and is still interested in algorithmic aspects of inverse scattering problems, and in point processes as mathematical models in neurophysiology.