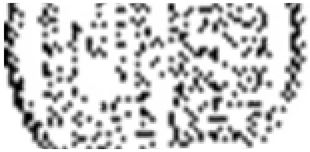
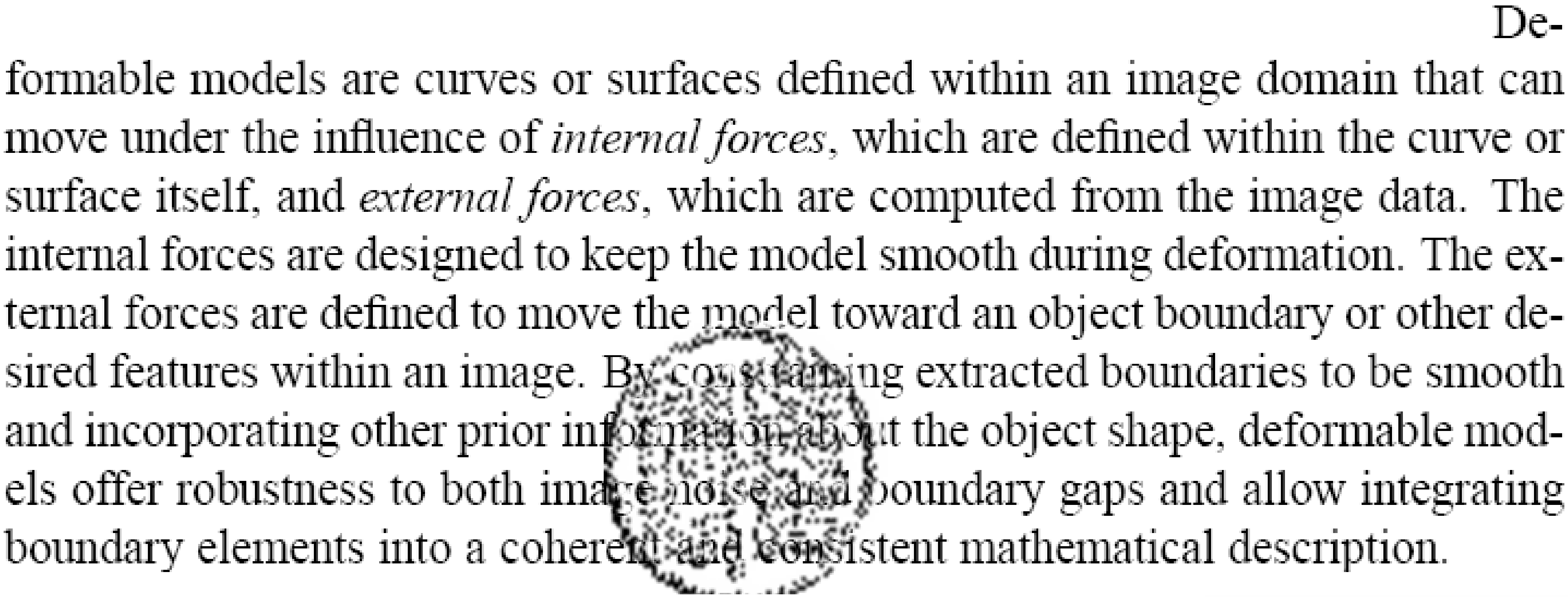
ACTIVE CONTOURS MODELS



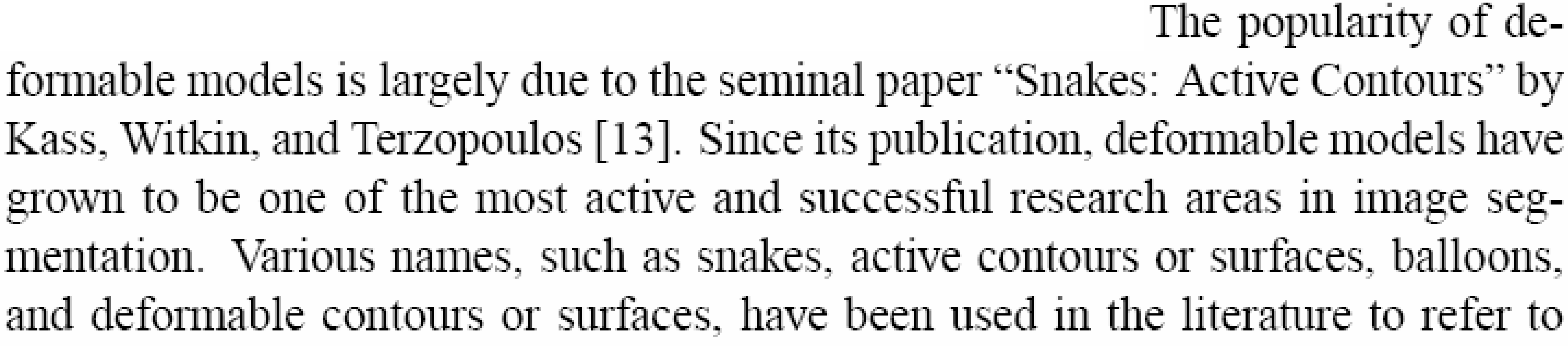
Il Choi, CTO, Ph.D.

UniInfo Co.Ltd.

Introduction



Deformable models are curves or surfaces defined within an image domain that can move under the influence of *internal forces*, which are defined within the curve or surface itself, and *external forces*, which are computed from the image data. The internal forces are designed to keep the model smooth during deformation. The external forces are defined to move the model toward an object boundary or other desired features within an image. By constraining extracted boundaries to be smooth and incorporating other prior information about the object shape, deformable models offer robustness to both image noise and boundary gaps and allow integrating boundary elements into a coherent and consistent mathematical description.



The popularity of deformable models is largely due to the seminal paper "Snakes: Active Contours" by Kass, Witkin, and Terzopoulos [13]. Since its publication, deformable models have grown to be one of the most active and successful research areas in image segmentation. Various names, such as snakes, active contours or surfaces, balloons, and deformable contours or surfaces, have been used in the literature to refer to deformable models.

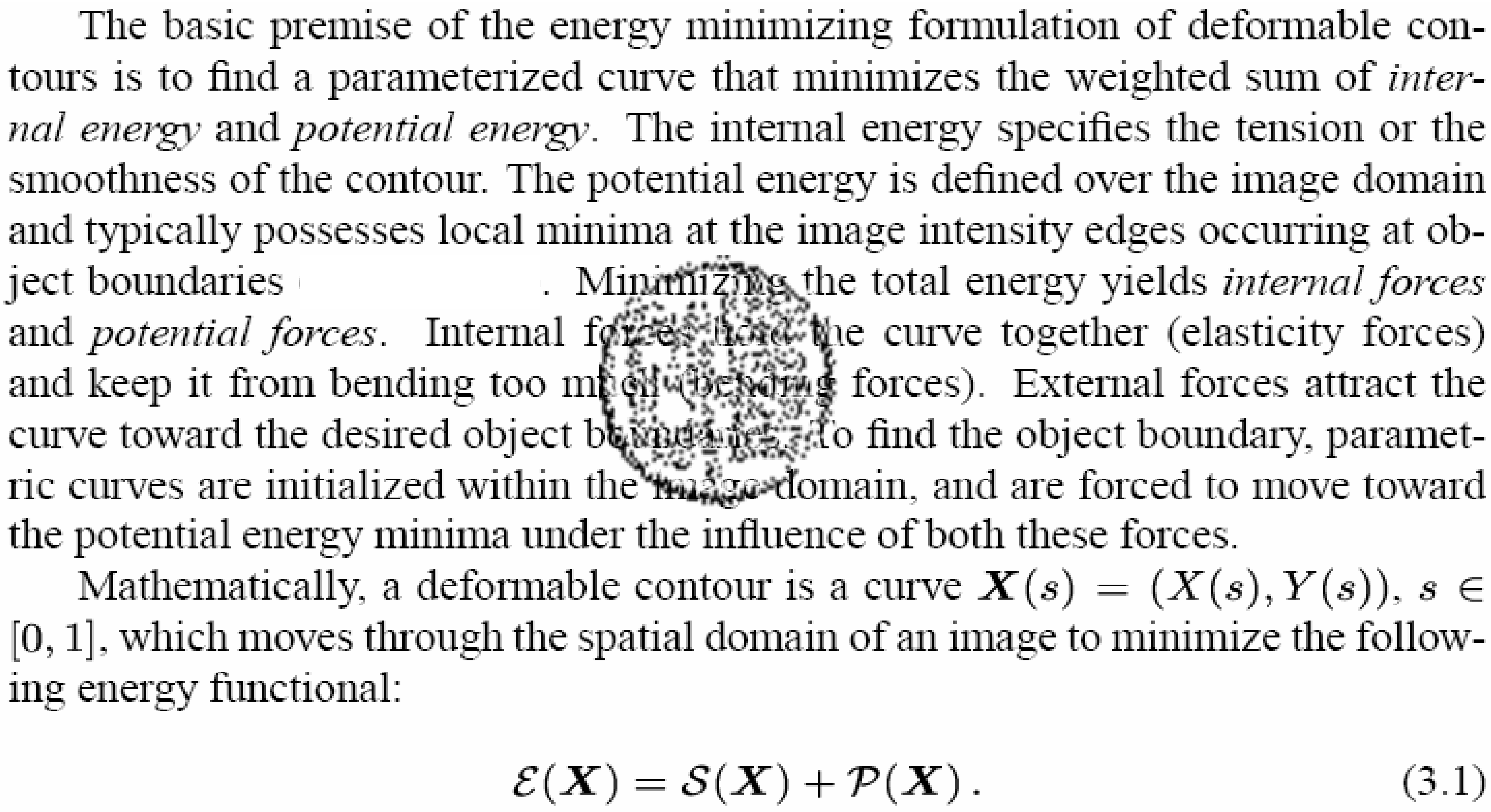
There are basically two types of deformable models: *parametric deformable models* (cf. [13, 21–23]) and *geometric deformable models* (cf. [24–27]). Parametric deformable models represent curves and surfaces explicitly in their parametric forms during deformation. This representation allows direct interaction with the model and can lead to a compact representation for fast real-time implementation. Adaptation of the model topology, however, such as splitting or merging parts during the deformation, can be difficult using parametric models. Geometric deformable models, on the other hand, can handle topological changes naturally. These models, based on the theory of curve evolution [28–31] and the level set method [32, 33], represent curves and surfaces implicitly as a level set of a higher-dimensional scalar function. Their parameterizations are computed only after complete deformation, thereby allowing topological adaptivity to be easily accommodated. Despite this fundamental difference, the underlying principles of both methods are very similar.

Parametric Active Contours

3 major problems: (1) Topology adaptation
(ref. T-snakes, level-set)
(2) Increasing Capture range
(3) Concavity adaptation

To overcome (2) and (3),
Balloon force, Gradient Vector Flow methods, and
Geometric Active Contours
had been proposed.

Energy Minimizing Formulation



The basic premise of the energy minimizing formulation of deformable contours is to find a parameterized curve that minimizes the weighted sum of *internal energy* and *potential energy*. The internal energy specifies the tension or the smoothness of the contour. The potential energy is defined over the image domain and typically possesses local minima at the image intensity edges occurring at object boundaries . Minimizing the total energy yields *internal forces* and *potential forces*. Internal forces hold the curve together (elasticity forces) and keep it from bending too much (bending forces). External forces attract the curve toward the desired object boundaries. To find the object boundary, parametric curves are initialized within the image domain, and are forced to move toward the potential energy minima under the influence of both these forces.

Mathematically, a deformable contour is a curve $\boldsymbol{X}(s) = (X(s), Y(s))$, $s \in [0, 1]$, which moves through the spatial domain of an image to minimize the following energy functional:

$$\mathcal{E}(\boldsymbol{X}) = \mathcal{S}(\boldsymbol{X}) + \mathcal{P}(\boldsymbol{X})\,. \tag{3.1}$$

The first term is the internal energy functional and is defined to be

$$\mathcal{S}(\boldsymbol{X}) = \frac{1}{2}\int_0^1 \alpha(s)\left|\frac{\partial \boldsymbol{X}}{\partial s}\right|^2 + \beta(s)\left|\frac{\partial^2 \boldsymbol{X}}{\partial s^2}\right|^2 ds\,. \tag{3.2}$$

The first-order derivative discourages stretching and makes the model behave like an elastic string. The second-order derivative discourages bending and makes the model behave like a rigid rod. The weighting parameters $\alpha(s)$ and $\beta(s)$ can be used to control the strength of the model's tension and rigidity, respectively. In practice, $\alpha(s)$ and $\beta(s)$ are often chosen to be constants.

The second term is the potential energy functional and is computed by integrating a potential energy function $P(x, y)$ along the contour $\boldsymbol{X}(s)$:

$$\mathcal{P}(\boldsymbol{X}) = \int_0^1 P(\boldsymbol{X}(s))ds\,. \tag{3.3}$$

The potential energy function $P(x, y)$ is derived from the image data and takes smaller values at object boundaries as well as other features of interest.

Regardless of the selection of the exact potential energy function, the procedure for minimizing the energy functional is the same. The problem of finding a curve $\boldsymbol{X}(s)$ that minimizes the energy functional $\mathcal{E}$ is known as a variational problem [35]. It has been shown that the curve that minimizes $\mathcal{E}$ must satisfy the following Euler-Lagrange equation [13, 22]:

$$\frac{\partial}{\partial s}\left(\alpha \frac{\partial \boldsymbol{X}}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 \boldsymbol{X}}{\partial s^2}\right) - \nabla P(\boldsymbol{X}) = 0\,. \tag{3.6}$$

To gain some insight about the physical behavior of deformable contours, we can view Eq. (3.6) as a force balance equation

$$\boldsymbol{F}_{\text{int}}(\boldsymbol{X}) + \boldsymbol{F}_{\text{pot}}(\boldsymbol{X}) = 0\,, \tag{3.7}$$

where the internal force is given by

$$\boldsymbol{F}_{\mathrm{int}}(\boldsymbol{X}) = \frac{\partial}{\partial s}\left(\alpha \frac{\partial \boldsymbol{X}}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 \boldsymbol{X}}{\partial s^2}\right) \tag{3.8}$$

and the potential force is given by

$$\boldsymbol{F}_{\mathrm{pot}}(\boldsymbol{X}) = -\nabla P(\boldsymbol{X})\,. \tag{3.9}$$

The internal force $\boldsymbol{F}_{\mathrm{int}}$ discourages stretching and bending while the potential force $\boldsymbol{F}_{\mathrm{pot}}$ pulls the contour toward the desired object boundaries.

To find a solution to Eq. (3.6), the deformable contour is made dynamic by treating $\boldsymbol{X}(s)$ as a function of time t as well as s — i.e., $\boldsymbol{X}(s,t)$. The partial derivative of $\boldsymbol{X}$ with respect to t is then set equal to the left-hand side of Eq. (3.6) as follows:

$$\gamma\frac{\partial \boldsymbol{X}}{\partial t} = \frac{\partial}{\partial s}\left(\alpha\frac{\partial \boldsymbol{X}}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 \boldsymbol{X}}{\partial s^2}\right) - \nabla P(\boldsymbol{X})\,. \tag{3.10}$$

The coefficient γ is introduced to make the units on the left side consistent with the right side. When the solution $\boldsymbol{X}(s,t)$ stabilizes, the left side vanishes and we achieve a solution of Eq. (3.6). We note that this approach of making the time derivative term vanish is equivalent to applying a gradient descent algorithm to find the local minimum of Eq. (3.1) [34]. Thus, the minimization is solved by placing an initial contour on the image domain and allowing it to deform according to Eq. (3.10).

External Forces

(1) Gaussian Potential Force
(2) Pressure (Balloon) Force
(3) Distance Potential Force
(4) Gradient Vector Flow

Gaussian Potential Forces

a gray-level image $I(x, y)$ viewed as a function of continuous position variables (x, y), a typical potential energy function designed to lead a deformable contour toward step edges is

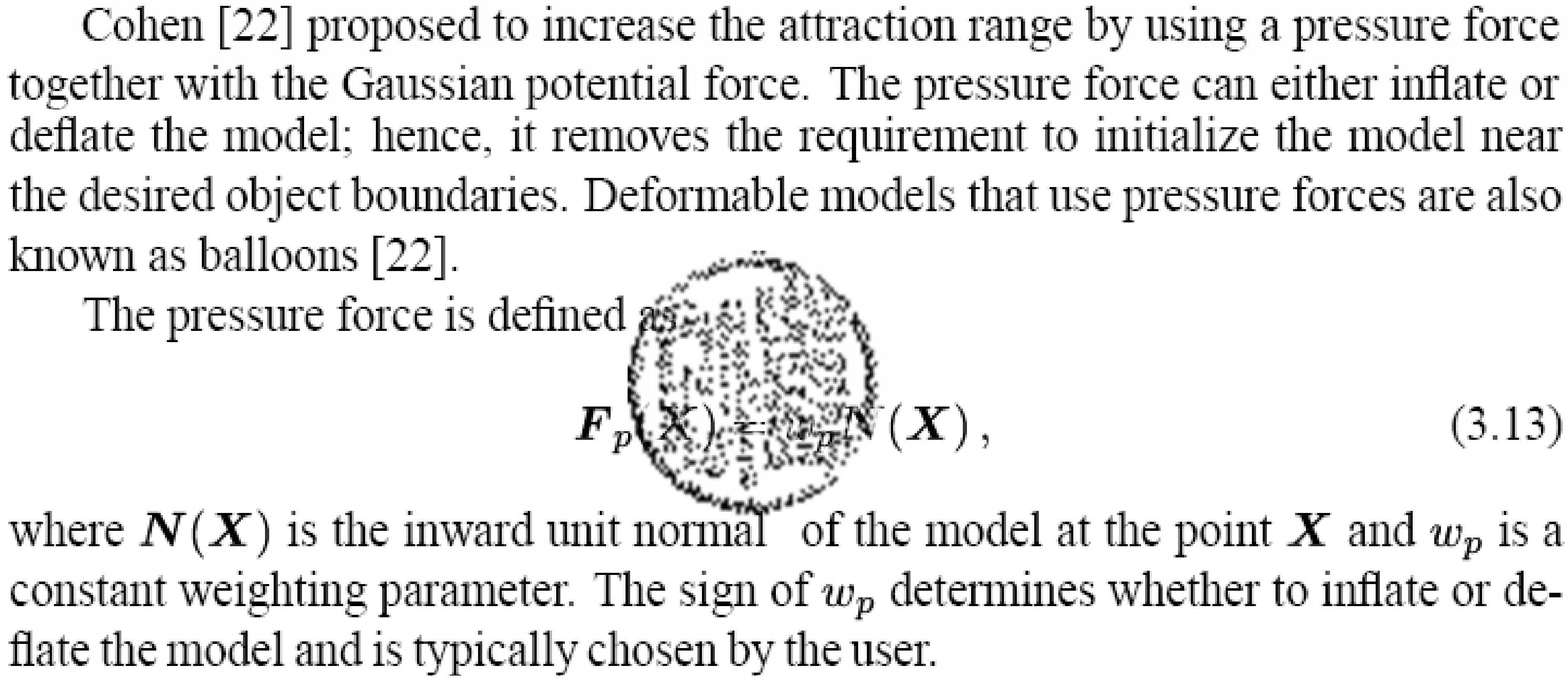
$$P(x,y) = -w_e \left|\nabla[G_\sigma(x,y) * I(x,y)]\right|^2 , \tag{3.4}$$

where w_e is a positive weighting parameter, $G_\sigma(x, y)$ is a two-dimensional Gaussian function with standard deviation σ, ∇ is the gradient operator, and $*$ is the 2D image convolution operator. If the desired image features are lines, then the appropriate potential energy function can be defined as follows:

$$P(x,y) = w_l[G_\sigma(x,y) * I(x,y)] , \tag{3.5}$$

where w_l is a weighting parameter. Positive w_l is used to find black lines on a white background, while negative w_l is used to find white lines on a black background. For both edge and line potential energies, increasing σ can broaden its attraction range. However, larger σ can also cause a shift in the boundary location, resulting in a less accurate result (this problem can be addressed by using potential energies calculated with different values of σ).

Pressure (Balloon) Force



Cohen [22] proposed to increase the attraction range by using a pressure force together with the Gaussian potential force. The pressure force can either inflate or deflate the model; hence, it removes the requirement to initialize the model near the desired object boundaries. Deformable models that use pressure forces are also known as balloons [22].

The pressure force is defined as

$$\boldsymbol{F}_p(\boldsymbol{X}) = w_p \boldsymbol{N}(\boldsymbol{X}), \tag{3.13}$$

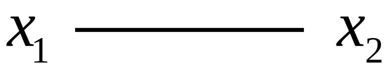
where $\boldsymbol{N}(\boldsymbol{X})$ is the inward unit normal of the model at the point $\boldsymbol{X}$ and w_p is a constant weighting parameter. The sign of w_p determines whether to inflate or deflate the model and is typically chosen by the user.

The value of w_p determines the strength of the pressure force. It must be carefully selected so that the pressure force is slightly smaller than the Gaussian potential force at significant edges, but large enough to pass through weak or spurious edges. When the model deforms, the pressure force keeps inflating or deflating the model until it is stopped by the Gaussian potential force.

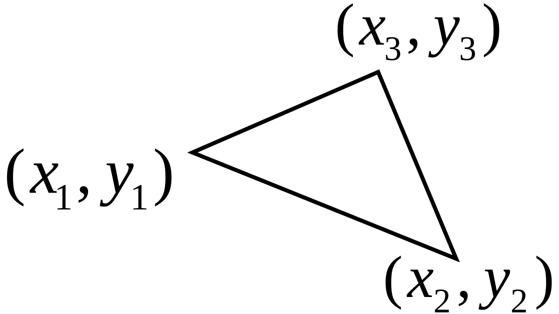
A disadvantage in using pressure forces is that they may cause the deformable model to cross itself and form loops (cf. [39]).

Area Revisit

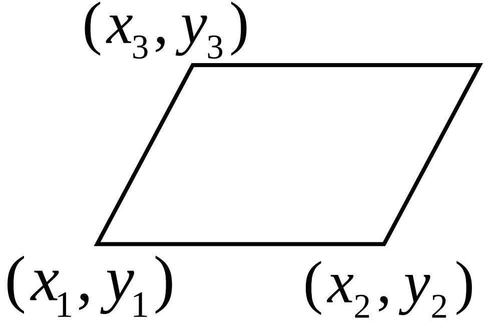
(1) The lengh of a line segment with its end points x_1, x_2 : $\frac{1}{1}\begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix}$.



(2) The area of triangle in the *xy* plane: $\frac{1}{1\cdot 2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$.



(3) The area of parallelogram in the *xy* plane: $\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$.



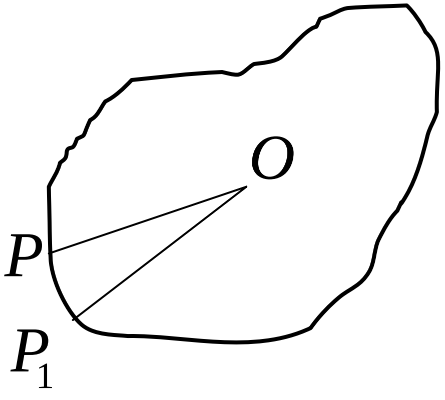
(4) The enclosed area of a simple curve in the *xy* plane

If we approximate the curve by poltgons having an increasing number of shorter and shorter sides and calculate the limit of the areas of these polygons. If P(x,y) and $\mathrm{P}_1(x+dx, y+dy)$ are two neighboring vertices of such an approximating polygon, then its area consists of a sum of elementary triangles(OPP$_1$), that is of summands:

$$\frac{1}{2}\begin{vmatrix} 0 & 0 & 1 \\ x & y & 1 \\ x+dx & y+dy & 1 \end{vmatrix} = \frac{1}{2}(xdy - ydx),$$

In the limit, this sum becomes the line integrals

$$\frac{1}{2}\int (xdy - ydx).$$



$$\frac{1}{2}\oint \mathbf{C}\times\mathbf{T}ds = \frac{1}{2}\oint \mathbf{C}\bullet\mathbf{N}ds$$

where $\mathbf{C}=\mathbf{C}(p)=\mathbf{C}(x(p),y(p)),\ \mathbf{T}=\dfrac{\partial\mathbf{C}(p)}{\partial p},\mathbf{T}\bullet\mathbf{N}=0$

Lagrange-Euler Equation of the above equation

Distance Potential Force

Another approach for extending attraction range is to define the potential energy function using a distance map as proposed by Cohen and Cohen [40]. The value of the distance map at each pixel is obtained by calculating the distance between the pixel and the closest boundary point, based either on Euclidean distance [41] or Chamfer distance [42]. By defining the potential energy function based on the distance map, one can obtain a potential force field that has a large attraction range.

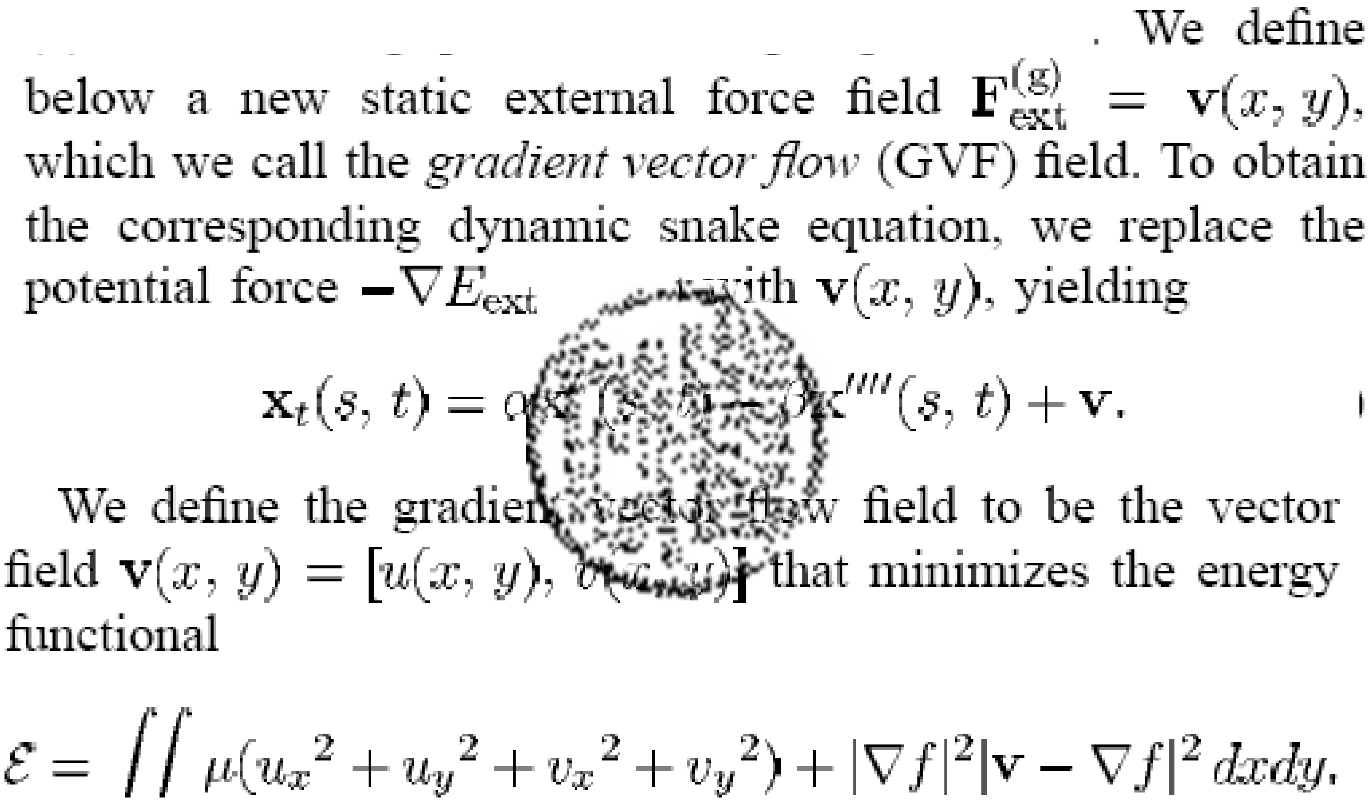
Given a computed distance map $d(x, y)$, one way of defining a corresponding potential energy, introduced in [40], is as follows:

$$P_d(x, y) = -w_d \exp[-d(x, y)^2]\,. \tag{3.14}$$

The corresponding potential force field is given by $-\nabla P_d(x, y)$.

The distance potential force is based on the principle that the model point should be attracted to the nearest edge points. This principle, however, can cause difficulties when deforming a contour or surface into boundary concavities [43].

Gradient Vector Flow [10][43]



. We define below a new static external force field $\mathbf{F}_{\text{ext}}^{(\text{g})} = \mathbf{v}(x, y)$, which we call the *gradient vector flow* (GVF) field. To obtain the corresponding dynamic snake equation, we replace the potential force $-\nabla E_{\text{ext}}$ with $\mathbf{v}(x, y)$, yielding

$$\mathbf{x}_t(s, t) = \alpha \mathbf{x}''(s, t) - \beta \mathbf{x}''''(s, t) + \mathbf{v}.$$

We define the gradient vector flow field to be the vector field $\mathbf{v}(x, y) = [u(x, y), v(x, y)]$ that minimizes the energy functional

$$\mathcal{E} = \iint \mu({u_x}^2 + {u_y}^2 + {v_x}^2 + {v_y}^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 \, dx dy.$$

This variational formulation follows a standard principle, that of making the result smooth when there is no data. In particular, we see that when $|\nabla f|$ is small, the energy is dominated by sum of the squares of the partial derivatives of the vector field, yielding a slowly varying field. On the other hand, when $|\nabla f|$ is large, the second term dominates the integrand, and is minimized by setting $\mathbf{v} = \nabla f$. This produces the desired effect of keeping $\mathbf{v}$ nearly equal to the gradient of the edge map when it is large, but forcing the field to be slowly-varying in homogeneous regions. The parameter μ is a regularization parameter governing the tradeoff between the first term and the second term in the integrand. This parameter should be set according to the amount of noise present in the image (more noise, increase μ).

We note that the smoothing term—the first term within the integrand —is the same term used by Horn and Schunck in their classical formulation of optical flow

Using the *calculus of variations* it can be shown that the GVF field can be found by solving the following Euler equations

$$\mu\nabla^2 u - (u - f_x)({f_x}^2 + {f_y}^2) = 0$$
$$\mu\nabla^2 v - (v - f_y)({f_x}^2 + {f_y}^2) = 0$$

where ∇^2 is the Laplacian operator. These equations provide further intuition behind the GVF formulation. We note that in a homogeneous region [where $f(x, y)$ is constant], the second term in each equation is zero because the gradient of $f(x, y)$ is zero. Therefore, within such a region, u and v are each determined by Laplace's equation, and the resulting GVF field is interpolated from the region's boundary, reflecting a kind of competition among the boundary vectors. This explains why GVF yields vectors that point into boundary concavities.

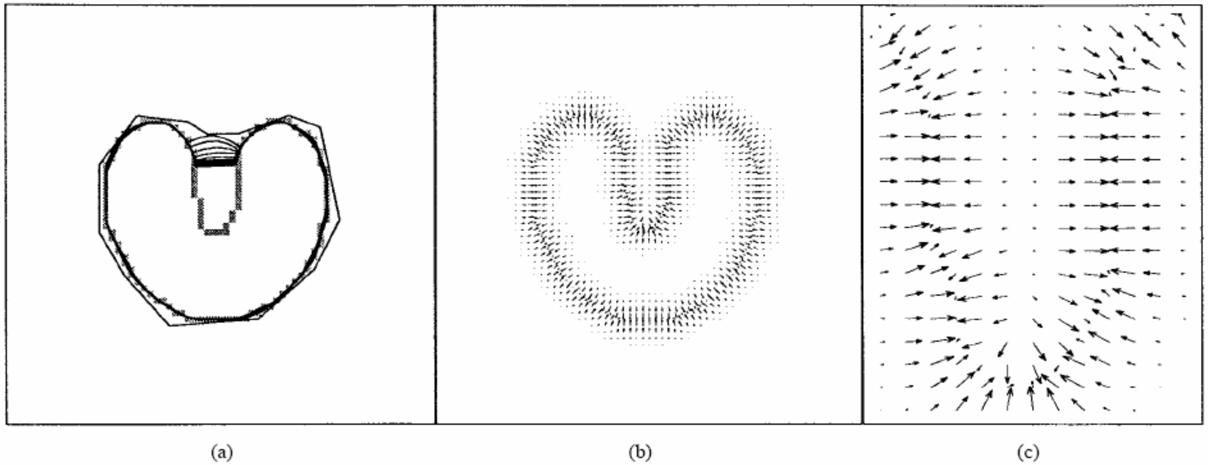


Fig. 1. (a) Convergence of a snake using (b) traditional potential forces, and (c) shown close-up within the boundary concavity.

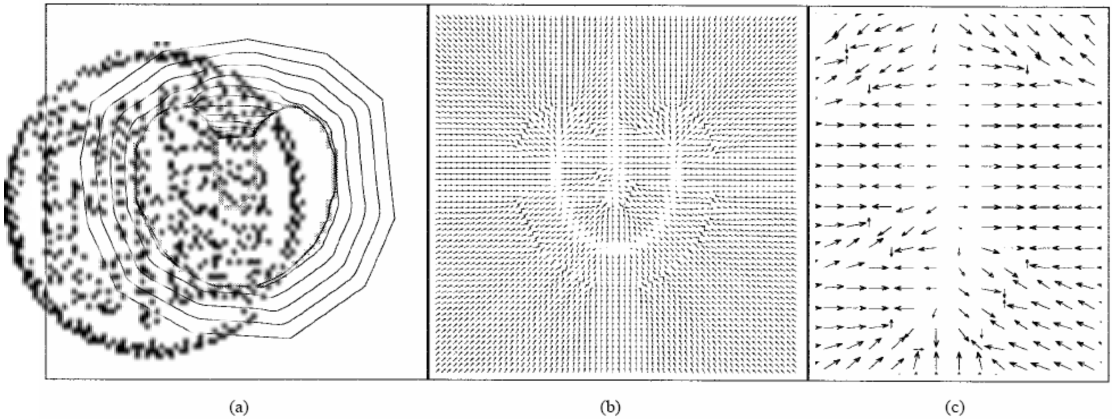


Fig. 2. (a) Convergence of a snake using (b) distance potential forces, and (c) shown close-up within the boundary concavity.

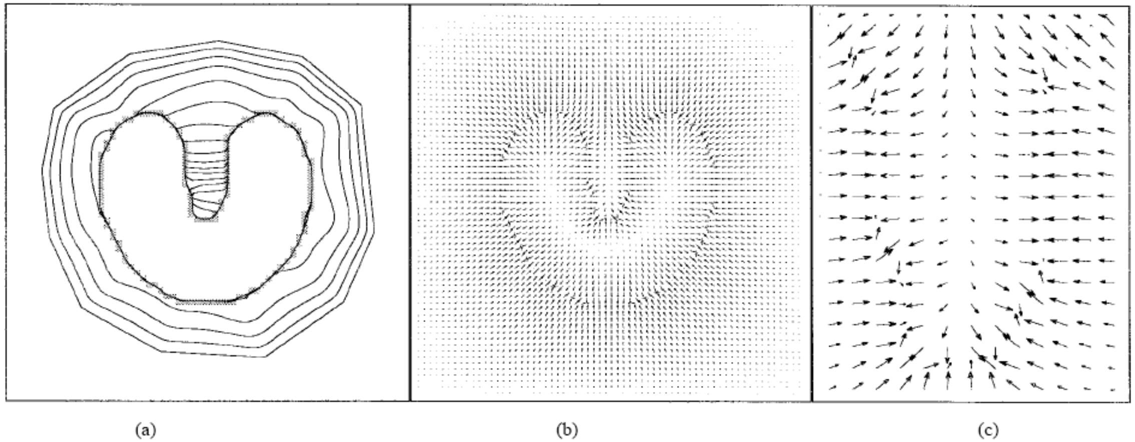
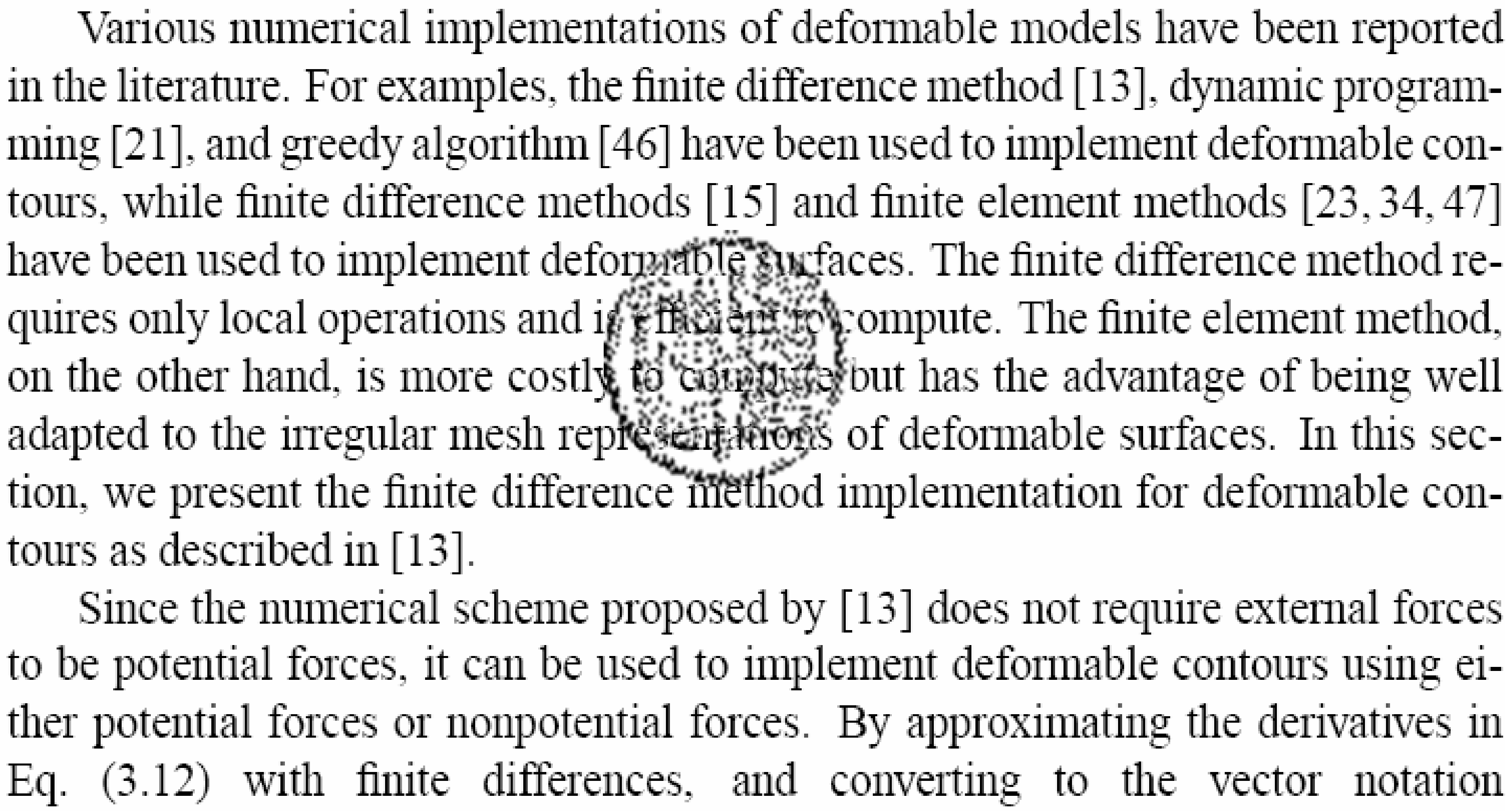


Fig. 3. (a) Convergence of a snake using (b) GVF external forces, and (c) shown close-up within the boundary concavity.

Numerical Implementation



Various numerical implementations of deformable models have been reported in the literature. For examples, the finite difference method [13], dynamic programming [21], and greedy algorithm [46] have been used to implement deformable contours, while finite difference methods [15] and finite element methods [23, 34, 47] have been used to implement deformable surfaces. The finite difference method requires only local operations and is efficient to compute. The finite element method, on the other hand, is more costly to compute but has the advantage of being well adapted to the irregular mesh representations of deformable surfaces. In this section, we present the finite difference method implementation for deformable contours as described in [13].

Since the numerical scheme proposed by [13] does not require external forces to be potential forces, it can be used to implement deformable contours using either potential forces or nonpotential forces. By approximating the derivatives in Eq. (3.12) with finite differences, and converting to the vector notation

$\boldsymbol{X}_i^n = (X_i^n, Y_i^n) = (X(ih, n\Delta t), Y(ih, n\Delta t))$, we can rewrite Eq. (3.12) as

$$\begin{aligned} \gamma \frac{\boldsymbol{X}_i^n - \boldsymbol{X}_i^{n-1}}{\Delta t} &= \frac{1}{h^2}[\alpha_{i+1}(\boldsymbol{X}_{i+1}^n - \boldsymbol{X}_i^n) - \alpha_i(\boldsymbol{X}_i^n - \boldsymbol{X}_{i-1}^n)] \\ &\quad - \frac{1}{h^4}[\beta_{i-1}(\boldsymbol{X}_{i-2}^n - 2\boldsymbol{X}_{i-1}^n + \boldsymbol{X}_i^n) \\ &\quad -2\beta_i(\boldsymbol{X}_{i-1}^n - 2\boldsymbol{X}_i^n + \boldsymbol{X}_{i+1}^n) \\ &\quad +\beta_{i+1}(\boldsymbol{X}_i^n - 2\boldsymbol{X}_{i+1}^n + \boldsymbol{X}_{i+2}^n)] + \boldsymbol{F}_{\text{ext}}(\boldsymbol{X}_i^{n-1}) \quad (3.20) \end{aligned}$$

where γ is the damping coefficient, $\alpha_i = \alpha(ih)$, $\beta_i = \beta(ih)$, h the step size in space, and Δt the step size in time. In general, the external force $\boldsymbol{F}_{\text{ext}}$ is stored as a discrete vector field, i.e., a finite set of vectors defined on an image grid. The value of $\boldsymbol{F}_{\text{ext}}$ at any location $\boldsymbol{X}_i$ can be obtained through a bilinear interpolation of the external force values at the grid points near $\boldsymbol{X}_i$.

Equation (3.20) can be written in a compact matrix form as

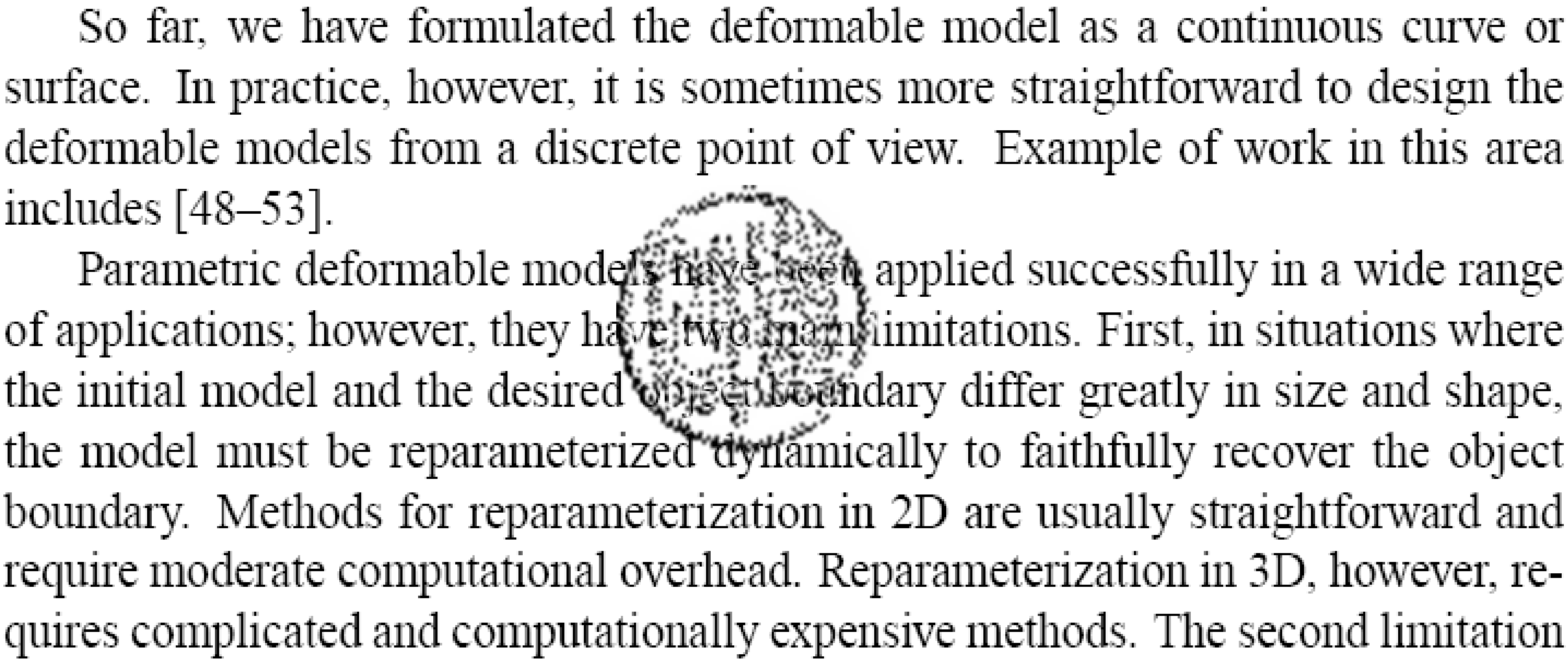
$$\frac{\boldsymbol{X}^n - \boldsymbol{X}^{n-1}}{\tau} = \boldsymbol{A}\boldsymbol{X}^n + \boldsymbol{F}_{\text{ext}}(\boldsymbol{X}^{n-1}), \tag{3.21}$$

where $\tau = \Delta t/\gamma$, $\boldsymbol{X}^n$, $\boldsymbol{X}^{n-1}$, and $\boldsymbol{F}_{\text{ext}}(\boldsymbol{X}^{n-1})$ are $m \times 2$ matrices, and $\boldsymbol{A}$ is an $m \times m$ pentadiagonal banded matrix with m being the number of sample points. Equation (3.21) can then be solved iteratively by matrix inversion using the following equation:

$$\boldsymbol{X}^n = (\boldsymbol{I} - \tau\boldsymbol{A})^{-1}[\boldsymbol{X}^{n-1} + \tau\boldsymbol{F}_{\text{ext}}(\boldsymbol{X}^{n-1})]. \tag{3.22}$$

The inverse of the matrix $\boldsymbol{I} - \tau\boldsymbol{A}$ can be calculated efficiently by LU decomposition[2]. The decomposition needs only to be performed once for deformation processes that do not alter the elasticity or rigidity parameters.

Discussion



So far, we have formulated the deformable model as a continuous curve or surface. In practice, however, it is sometimes more straightforward to design the deformable models from a discrete point of view. Example of work in this area includes [48–53].

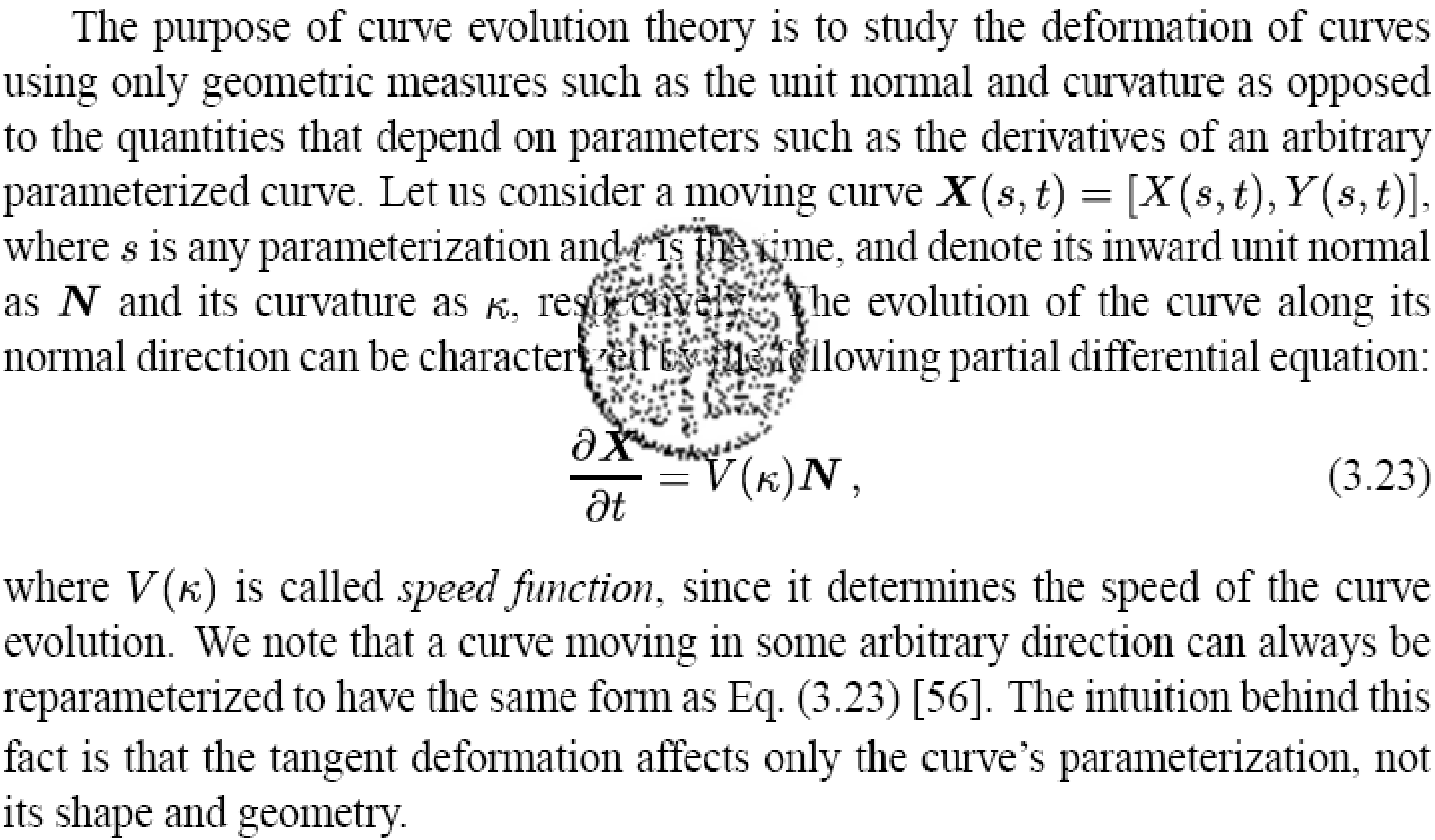
Parametric deformable models have been applied successfully in a wide range of applications; however, they have two main limitations. First, in situations where the initial model and the desired object boundary differ greatly in size and shape, the model must be reparameterized dynamically to faithfully recover the object boundary. Methods for reparameterization in 2D are usually straightforward and require moderate computational overhead. Reparameterization in 3D, however, requires complicated and computationally expensive methods. The second limitation with the parametric approach is that it has difficulty dealing with topological adaptation such as splitting or merging model parts, a useful property for recovering either multiple objects or an object with unknown topology. This difficulty is caused by the fact that a new parameterization must be constructed whenever the topology change occurs, which requires sophisticated schemes [54, 55].

DEMOS and Coffee Break

Geometric Active Contours

Geometric deformable models, proposed independently by Caselles et al. [24] and Malladi et al. [25], provide an elegant solution to address the primary limitations of parametric deformable models. These models are based on curve evolution theory [28–31] and the level set method [32, 33]. In particular, curves and surfaces are evolved using only geometric measures, resulting in an evolution that is independent of the parameterization. As in parametric deformable models, the evolution is coupled with the image data to recover object boundaries. Since the evolution is independent of the parameterization, the evolving curves and surfaces can be represented implicitly as a level set of a higher-dimensional function. As a result, topology changes can be handled automatically.

Curve Evolution Theory Revisit



The purpose of curve evolution theory is to study the deformation of curves using only geometric measures such as the unit normal and curvature as opposed to the quantities that depend on parameters such as the derivatives of an arbitrary parameterized curve. Let us consider a moving curve $\boldsymbol{X}(s,t) = [X(s,t), Y(s,t)]$, where s is any parameterization and t is the time, and denote its inward unit normal as $\boldsymbol{N}$ and its curvature as κ, respectively. The evolution of the curve along its normal direction can be characterized by the following partial differential equation:

$$\frac{\partial \boldsymbol{X}}{\partial t} = V(\kappa)\boldsymbol{N}\,, \tag{3.23}$$

where $V(\kappa)$ is called *speed function*, since it determines the speed of the curve evolution. We note that a curve moving in some arbitrary direction can always be reparameterized to have the same form as Eq. (3.23) [56]. The intuition behind this fact is that the tangent deformation affects only the curve's parameterization, not its shape and geometry.

The most extensively studied curve deformations in curve evolution theory are *curvature deformation* and *constant deformation*. Curvature deformation is given by the so-called *geometric heat equation*

$$\frac{\partial \boldsymbol{X}}{\partial t} = \alpha \kappa \boldsymbol{N} \,,$$

where α is a positive constant. This equation will smooth a curve, eventually shrinking it to a circular point [57]. The use of the curvature deformation has an effect similar to the use of the elastic internal force in parametric deformable models.

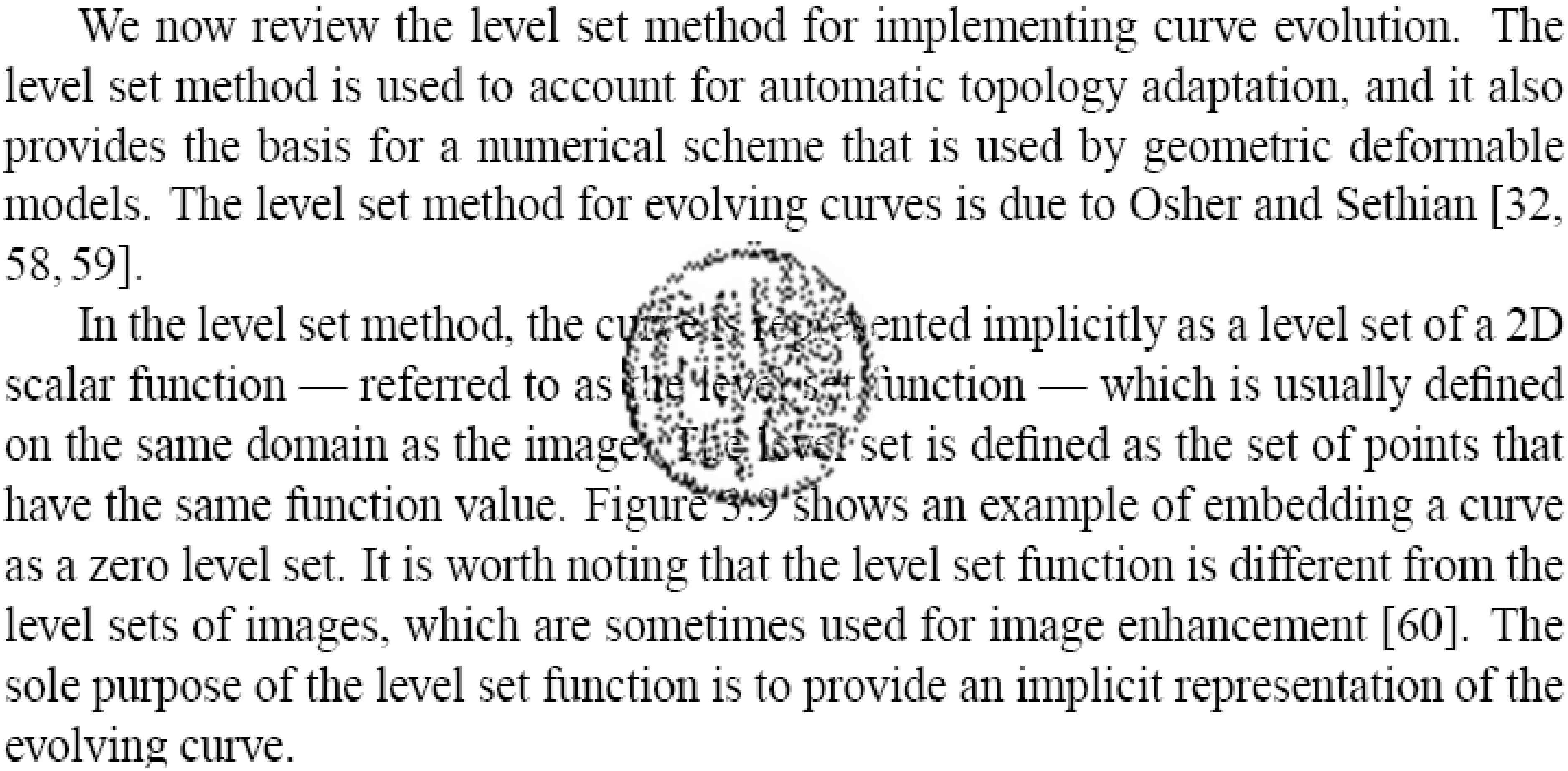
Constant deformation is given by

$$\frac{\partial \boldsymbol{X}}{\partial t} = V_0 \boldsymbol{N} \,,$$

where V_0 is a coefficient determining the speed and direction of deformation. Constant deformation plays the same role as the pressure force in parametric deformable models. The properties of curvature deformation and constant deformation are complementary to each other. Curvature deformation removes singularities by smoothing the curve, while constant deformation can create singularities from an initially smooth curve.

The basic idea of the geometric deformable model is to couple the speed of deformation (using curvature and/or constant deformation) with the image data, so that the evolution of the curve stops at object boundaries. The evolution is implemented using the level set method. Thus, most of the research in geometric deformable models has been focused in the design of speed functions.

Level Set Method



We now review the level set method for implementing curve evolution. The level set method is used to account for automatic topology adaptation, and it also provides the basis for a numerical scheme that is used by geometric deformable models. The level set method for evolving curves is due to Osher and Sethian [32, 58, 59].

In the level set method, the curve is represented implicitly as a level set of a 2D scalar function — referred to as the level set function — which is usually defined on the same domain as the image. The level set is defined as the set of points that have the same function value. Figure 3.9 shows an example of embedding a curve as a zero level set. It is worth noting that the level set function is different from the level sets of images, which are sometimes used for image enhancement [60]. The sole purpose of the level set function is to provide an implicit representation of the evolving curve.

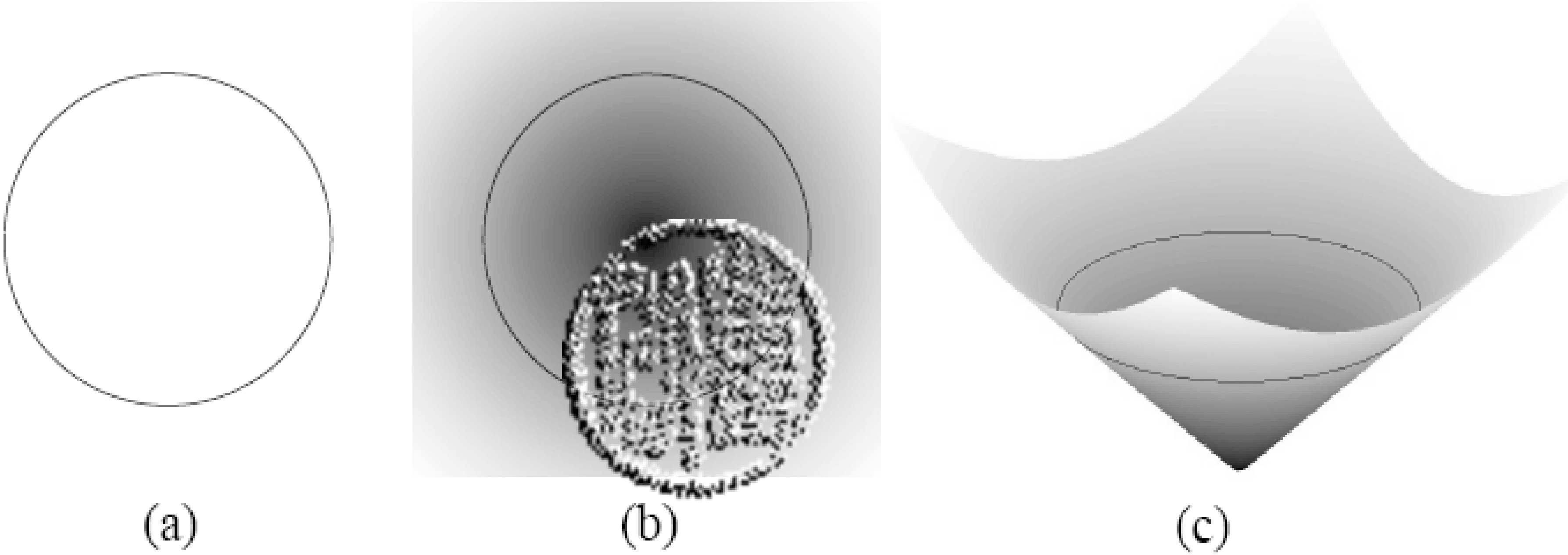


Figure 3.9: An example of embedding a curve as a level set. (a) A single curve. (b) The level set function where the curve is embedded as the zero level set (in black). (c) The height map of the level set function with its zero level set depicted in black.

Instead of tracking a curve through time, the level set method evolves a curve by updating the level set function at fixed coordinates through time. This perspective is similar to that of an Eulerian formulation of motion as opposed to a Lagrangian formulation, which is analogous to the parametric deformable model. A useful property of this approach is that the level set function remains a valid function while the embedded curve can change its topology. This situation is depicted in Fig 3.10.

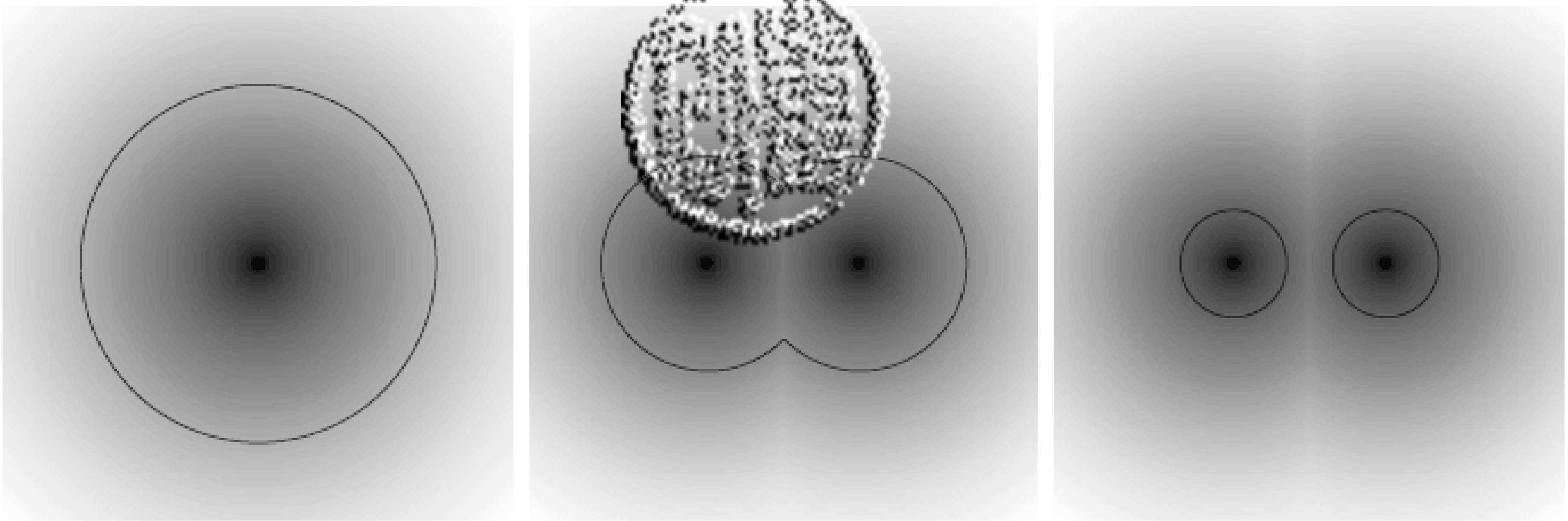


Figure 3.10: From left to right, the zero level set splits into two curves while the level set function still remains a valid function.

We now derive the level set embedding of the curve evolution equation (3.23). Given a level set function $\phi(x, y, t)$ with the contour $\boldsymbol{X}(s, t)$ as its zero level set, we have

$$\phi[\boldsymbol{X}(s,t), t] = 0\,.$$

Differentiating the above equation with respect to t and using the chain rule, we obtain

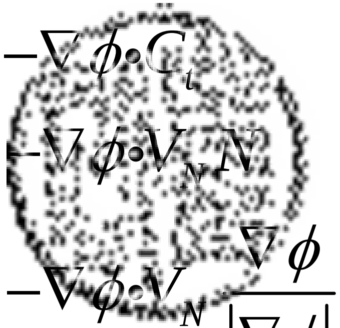
$$\frac{\partial \phi}{\partial t} + \nabla\phi \cdot \frac{\partial \boldsymbol{X}}{\partial t} = 0\,, \tag{3.24}$$

where $\nabla\phi$ denotes the gradient of ϕ.

$$\phi(x(t), y(t), t) = costant$$

$$\frac{d\phi}{dt} = 0 = \phi_t + \phi_x x_t + \phi_y y_t$$

Thus,



$$\begin{aligned} \phi_t &= -\nabla\phi \bullet C_t \\ &= -\nabla\phi \bullet V_N N \\ &= -\nabla\phi \bullet V_N \frac{\nabla\phi}{|\nabla\phi|} \\ &= -V_N \nabla\phi \bullet \frac{\nabla\phi}{|\nabla\phi|} \\ &= -V_N |\nabla\phi| \end{aligned}$$

We assume that ϕ is negative inside the zero level set and positive outside. Accordingly, the inward unit normal to the level set curve is given by

$$\boldsymbol{N} = -\frac{\nabla\phi}{|\nabla\phi|}\,. \tag{3.25}$$

Using this fact and Eq. (3.23), we can rewrite Eq. (3.24) as

$$\frac{\partial\phi}{\partial t} = V(\kappa)|\nabla\phi|\,, \tag{3.26}$$

where the curvature κ at the zero level set is given by

$$\kappa = \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}\,. \tag{3.27}$$

The relationship between Eq. (3.23) and Eq. (3.26) provides the basis for performing curve evolution using the level set method.

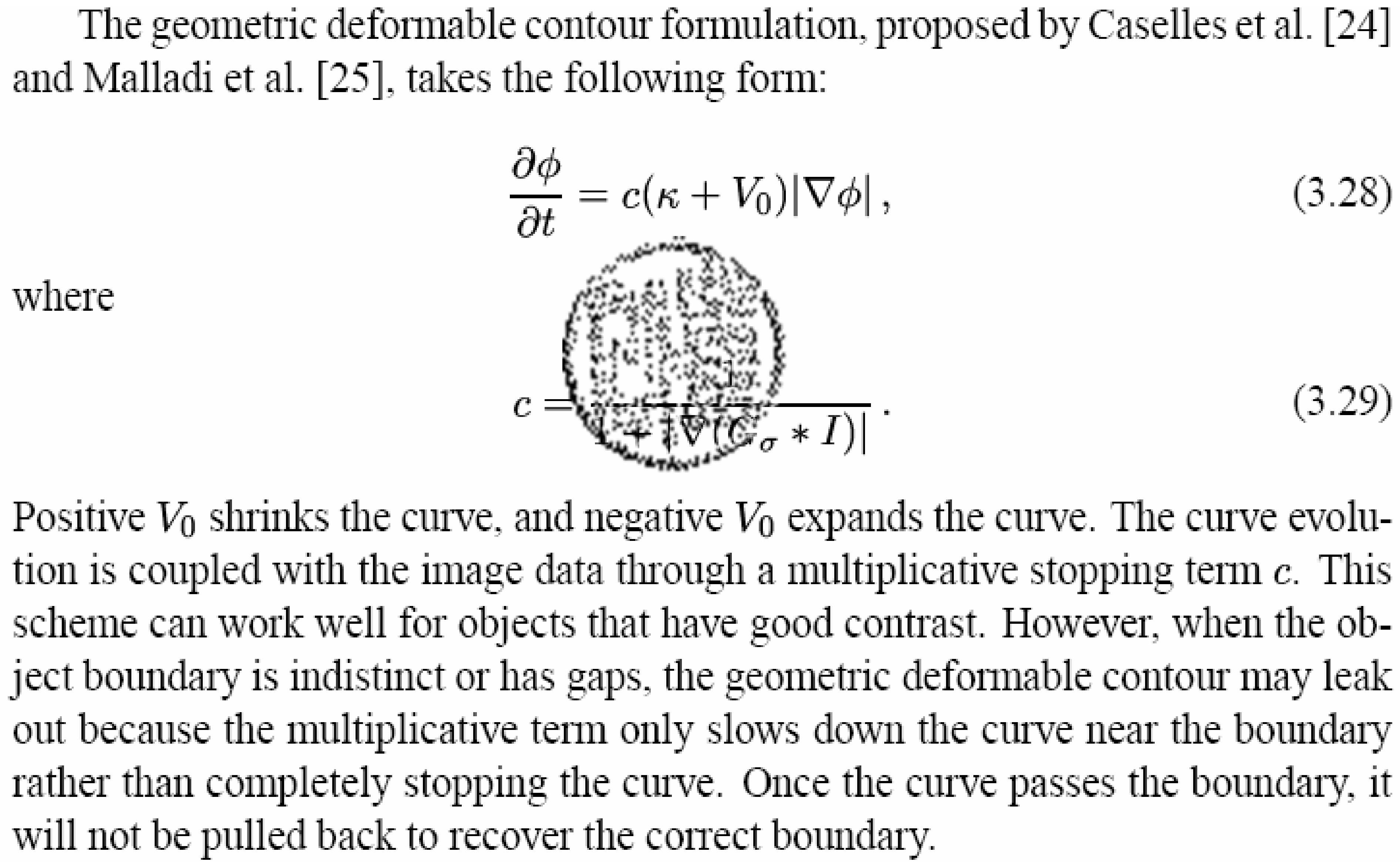
Three issues need to be considered in order to implement geometric deformable contours:

1. An initial function $\phi(x, y, t = 0)$ must be constructed such that its zero level set corresponds to the position of the initial contour. A common choice is to set $\phi(x, y, 0) = D(x, y)$, where $D(x, y)$ is the signed distance from each grid point to the zero level set. The computation of the signed distance for an arbitrary initial curve is expensive. Recently, Sethian and Malladi developed a method called the fast marching method, which can construct the signed distance function in $O(N \log N)$, where N is the number of pixels.

2. Since the evolution equation (3.26) is derived for the zero level set only, the speed function $V(\kappa)$, in general, is not defined on other level sets. Hence, we need a method to extend the speed function $V(\kappa)$ to all of the level sets. We note that the expressions for the unit normal and the curvature, however, hold for all level sets. Many approaches for such extensions have been developed (see [33] for a detailed discussion on this topic). However, the level set function that evolves using these extended speed functions can lose its property of being a signed distance function, causing inaccuracy in curvature and normal calculations. As a result, reinitialization of the level set function to a signed distance function is often required for these schemes. Recently, a method that does not suffer from this problem was proposed by Adalsteinsson and Sethian [61]. This method casts the speed extension problem as a boundary value problem, which can then be solved efficiently using the fast marching method.

3. In the application of geometric contours, constant deformation is often used to account for large-scale deformation and narrow boundary indentation and protrusion recovery. Constant deformation, however, can cause the formation of sharp corners from an initial smooth zero level set. Once the corner is developed, it is not clear how to continue the deformation, since the definition of the normal direction becomes ambiguous. A natural way to continue the deformation is to impose the so-called *entropy condition* originally proposed in the area of interface propagation by Sethian [62]. In Section 3.3.5, we describe an entropy satisfying numerical scheme, proposed by Osher and Sethian [32], which implements geometric deformable contours.

Speed Functions



The geometric deformable contour formulation, proposed by Caselles et al. [24] and Malladi et al. [25], takes the following form:

$$\frac{\partial \phi}{\partial t} = c(\kappa + V_0)|\nabla \phi| \,, \tag{3.28}$$

where

$$c = \frac{1}{1 + |\nabla (G_\sigma * I)|} \,. \tag{3.29}$$

Positive V_0 shrinks the curve, and negative V_0 expands the curve. The curve evolution is coupled with the image data through a multiplicative stopping term c. This scheme can work well for objects that have good contrast. However, when the object boundary is indistinct or has gaps, the geometric deformable contour may leak out because the multiplicative term only slows down the curve near the boundary rather than completely stopping the curve. Once the curve passes the boundary, it will not be pulled back to recover the correct boundary.

To remedy the latter problem, Caselles et al. [26,64] and Kichenassamy et al. [63, 65] used an energy minimization formulation to design the speed function. This leads to the following geometric deformable contour formulation:

$$\frac{\partial \phi}{\partial t} = c(\kappa + V_0)|\nabla \phi| + \nabla c \cdot \nabla \phi \, . \tag{3.30}$$

Note that the resulting speed function has an extra stopping term $\nabla c \cdot \nabla \phi$ that can pull back the contour if it passes the boundary. This term behaves in similar fashion to the Gaussian potential force in the parametric formulation. An example of using this type of geometrical deformable contours is shown in Fig. 3.11.

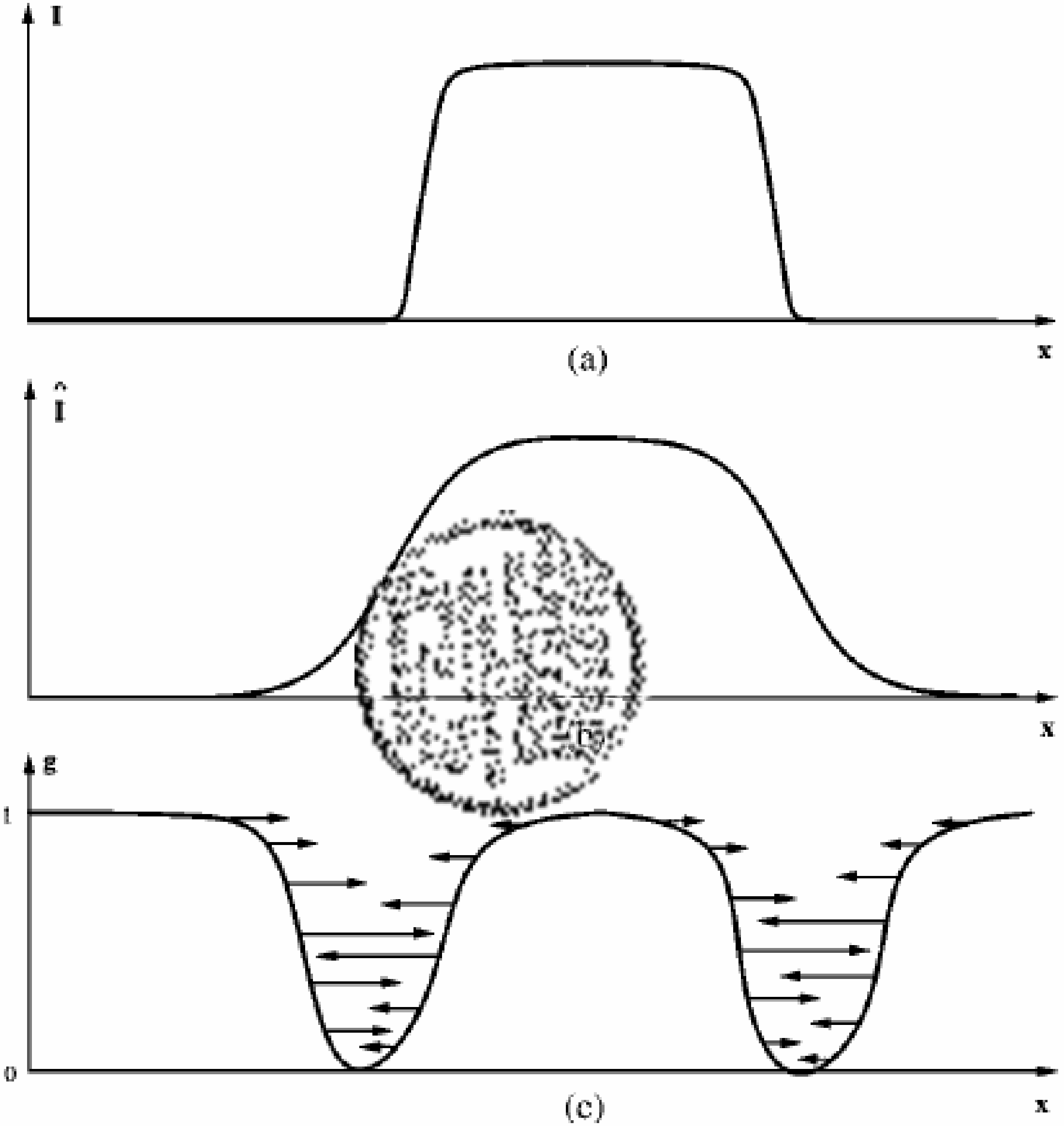


Figure 1. Geometric interpretation of the attraction force in 1D. The original edge signal I, its smoothed version $\hat{I}$, and the derived stopping function g are given. The evolving contour is attracted to the valley created by $\nabla g \cdot \nabla u$ (see text).

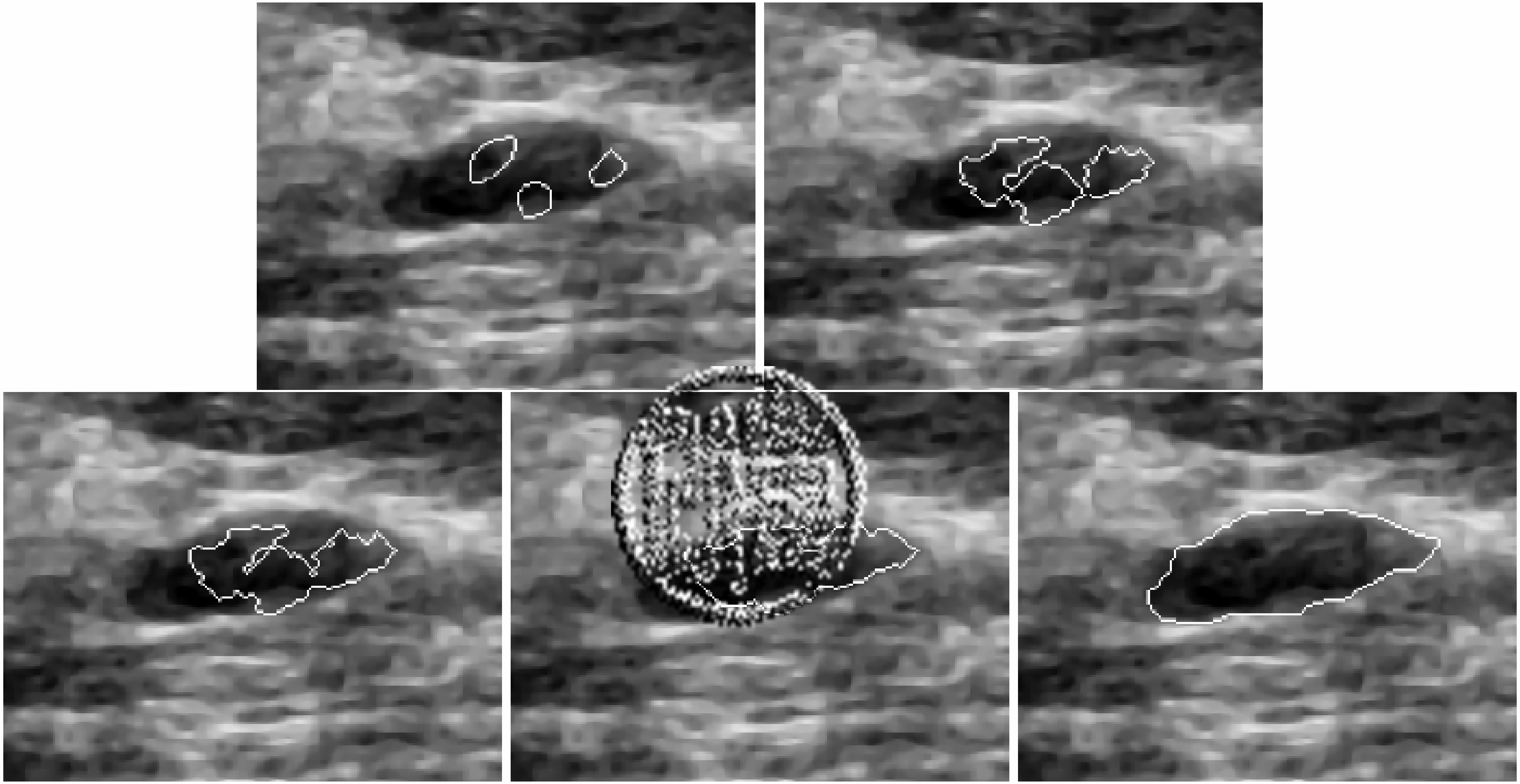


Figure 3.11: Contour extraction of cyst form ultrasound breast image via merging multiple initial level sets. Images courtesy of Yezzi [63], Georgia Institute of Technology.

The latter formulation can still generate curves that pass through boundary gaps. Siddiqi et al. [66] partially address this problem by altering the constant speed term through energy minimization, leading to the following geometric deformable contour:

$$\frac{\partial \phi}{\partial t} = \lambda(c\kappa|\nabla\phi| + \nabla c \cdot \nabla\phi) + (c + \frac{1}{2}\boldsymbol{X} \cdot \nabla c)|\nabla\phi| \, . \tag{3.31}$$

In this case, the constant speed term V_0 in Eq. (3.30) is replaced by the second term, and the term $\frac{1}{2}\boldsymbol{X} \cdot \nabla c$ provides additional stopping power that can prevent the geometrical contour from leaking through small boundary gaps. The second term can be used alone as the speed function for shape recovery as well. Figure 3.12 shows an example of this deformable contour model. Although this model is robust to small gaps, large boundary gaps can still cause problems.

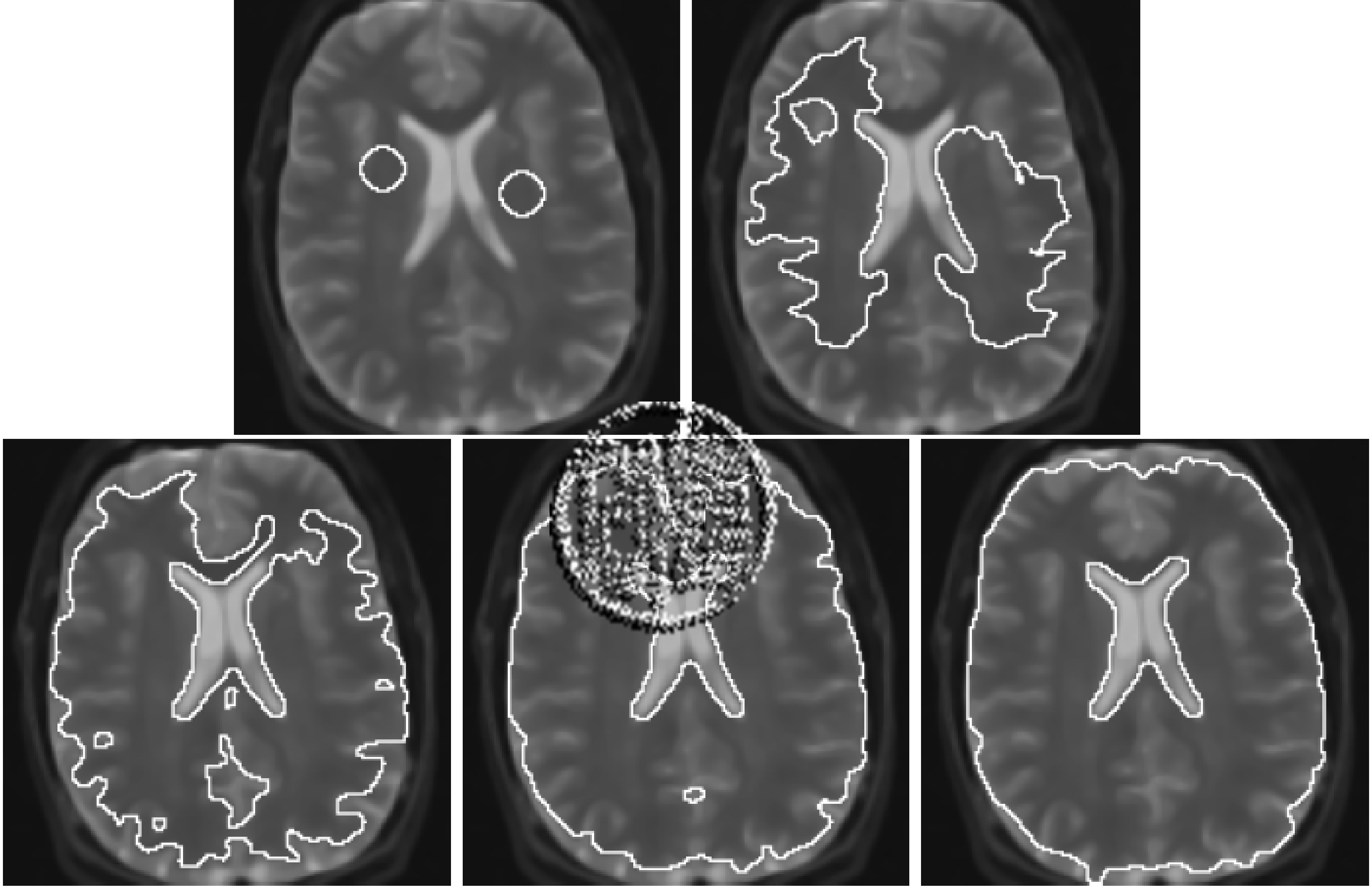
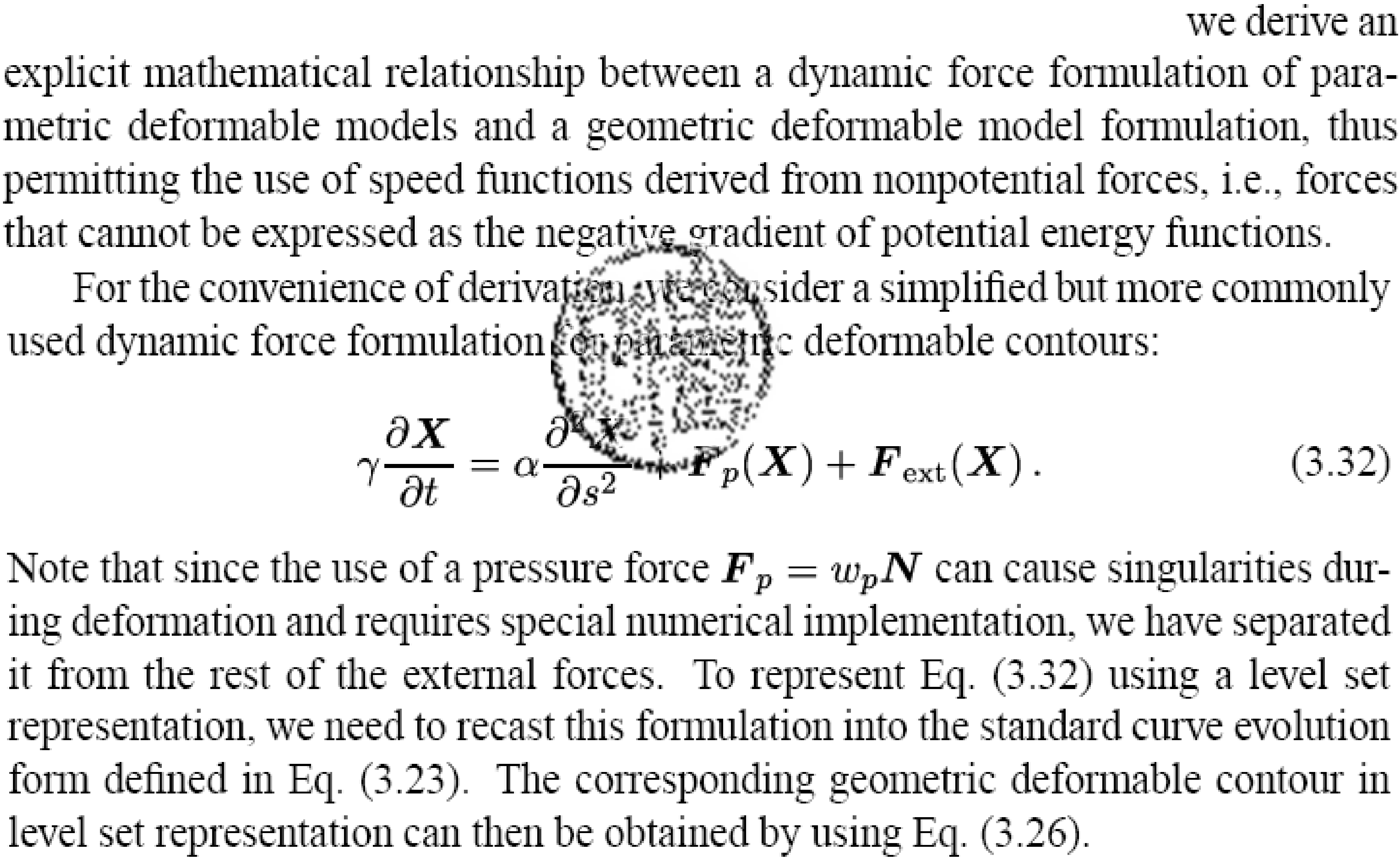


Figure 3.12: Segmentation of the brain using only the second term in (3.31). Left to right and top to bottom: iterations 1, 400, 800, 1200, and 1600. Images courtesy of Siddiqi [66],

Relationship to PACM



we derive an explicit mathematical relationship between a dynamic force formulation of parametric deformable models and a geometric deformable model formulation, thus permitting the use of speed functions derived from nonpotential forces, i.e., forces that cannot be expressed as the negative gradient of potential energy functions.

For the convenience of derivation, we consider a simplified but more commonly used dynamic force formulation for parametric deformable contours:

$$\gamma \frac{\partial \boldsymbol{X}}{\partial t} = \alpha \frac{\partial^2 \boldsymbol{X}}{\partial s^2} + \boldsymbol{F}_p(\boldsymbol{X}) + \boldsymbol{F}_{\text{ext}}(\boldsymbol{X}) . \qquad (3.32)$$

Note that since the use of a pressure force $\boldsymbol{F}_p = w_p \boldsymbol{N}$ can cause singularities during deformation and requires special numerical implementation, we have separated it from the rest of the external forces. To represent Eq. (3.32) using a level set representation, we need to recast this formulation into the standard curve evolution form defined in Eq. (3.23). The corresponding geometric deformable contour in level set representation can then be obtained by using Eq. (3.26).

Since the contour's tangential motion only affects its parameterization but not its geometry, we modify Eq. (3.32) by considering only the normal components of internal and external forces. Given a parameterized curve $\boldsymbol{X}(s,t)$, where s is the arc-length parameterization of the curve, its inward unit normal $\boldsymbol{N}$ and curvature κ, we can use the fact that $\partial^2 \boldsymbol{X}/\partial s^2 = \kappa \boldsymbol{N}$ to rewrite Eq. (3.32) as follows:

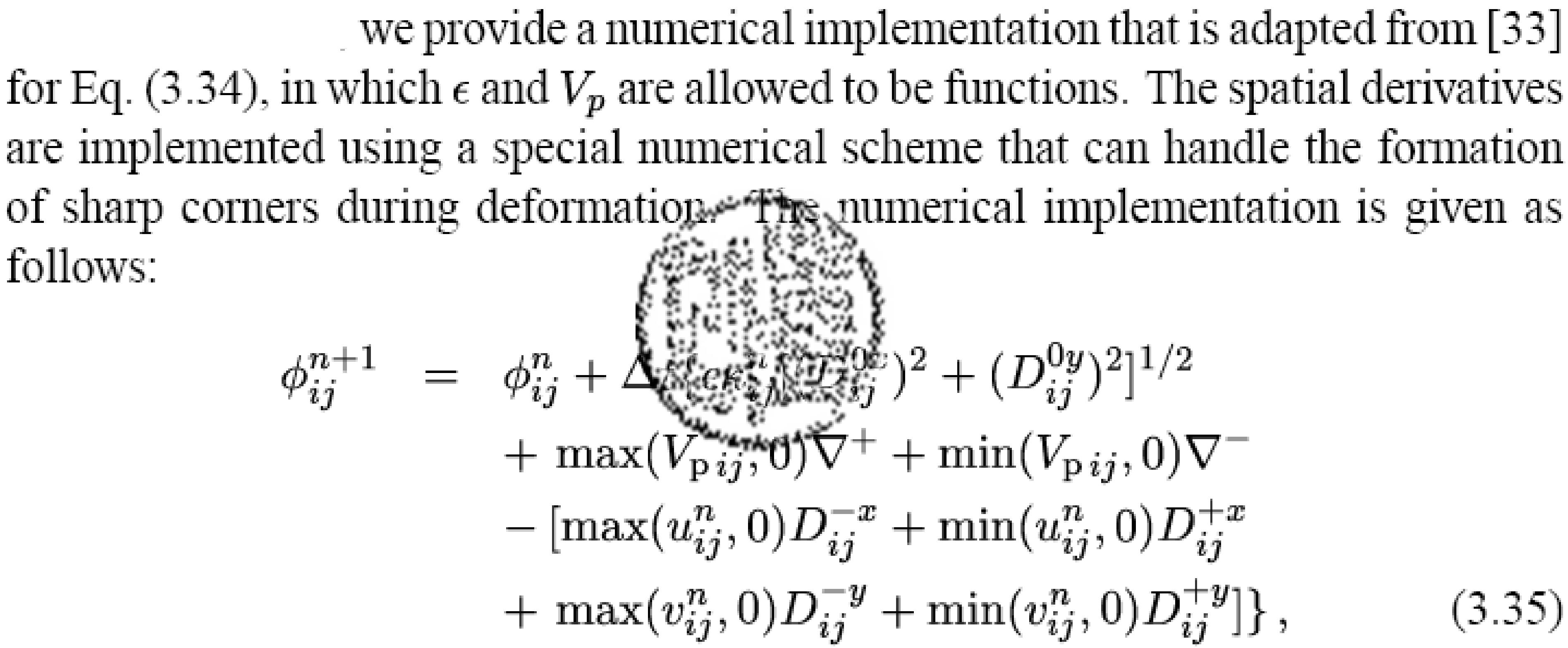
$$\frac{\partial \boldsymbol{X}}{\partial t} = (\epsilon\kappa + V_p + \boldsymbol{V}_{\text{ext}} \cdot \boldsymbol{N})\boldsymbol{N}\,, \tag{3.33}$$

where $\epsilon = \alpha/\gamma$, $V_p = w_p/\gamma$, and $\boldsymbol{V}_{\text{ext}} = \boldsymbol{F}_{\text{ext}}/\gamma$. Here, we have divided through by γ so that both sides have units of velocity. If we let $V(\kappa) = \epsilon\kappa + V_p + \boldsymbol{V}_{\text{ext}} \cdot \boldsymbol{N}$, where $\boldsymbol{N}$ is given by Eq. (3.25), and substitute $V(\kappa)$ into Eq. (3.26), we obtain the following geometric deformable contour evolution equation:

$$\frac{\partial \phi}{\partial t} = V(\kappa)|\nabla\phi| = (\epsilon\kappa + V_p)|\nabla\phi| - \boldsymbol{V}_{\text{ext}} \cdot \nabla\phi\,. \tag{3.34}$$

If we allow both ϵ and V_p to be functions defined on the image domain, then Eq. (3.34) generalizes Eq. (3.31) and can be used to implement almost any parametric deformable model as a geometric deformable model.

Numerical Implementation



we provide a numerical implementation that is adapted from [33] for Eq. (3.34), in which ϵ and V_p are allowed to be functions. The spatial derivatives are implemented using a special numerical scheme that can handle the formation of sharp corners during deformation. The numerical implementation is given as follows:

$$
\begin{aligned}
\phi_{ij}^{n+1} \quad = \quad & \phi_{ij}^{n} + \Delta t \{ \epsilon \kappa_{ij}^{n} [(D_{ij}^{0x})^2 + (D_{ij}^{0y})^2]^{1/2} \\
& + \max(V_{\mathrm{p}\,ij}, 0)\nabla^{+} + \min(V_{\mathrm{p}\,ij}, 0)\nabla^{-} \\
& - [\max(u_{ij}^{n}, 0) D_{ij}^{-x} + \min(u_{ij}^{n}, 0) D_{ij}^{+x} \\
& + \max(v_{ij}^{n}, 0) D_{ij}^{-y} + \min(v_{ij}^{n}, 0) D_{ij}^{+y}]\} \,, \qquad (3.35)
\end{aligned}
$$

where $\boldsymbol{V}_{\text{ext}} = (u, v)$, and $\kappa_{i,j}^{n}$ is the central difference approximation to the curvature expression given in Eq. (3.27). The first-order numerical derivatives and the gradient of the level set function ϕ are given by

$$\begin{aligned}
D_{ij}^{-x} &= \frac{\phi_{ij}^{n} - \phi_{i-1,j}^{n}}{\Delta x}, \quad D_{ij}^{+x} = \frac{\phi_{i+1,j}^{n} - \phi_{i,j}^{n}}{\Delta x}, \\
D_{ij}^{-y} &= \frac{\phi_{ij}^{n} - \phi_{i,j-1}^{n}}{\Delta y}, \quad D_{ij}^{+y} = \frac{\phi_{i,j+1}^{n} - \phi_{i,j}^{n}}{\Delta x}, \\
D_{ij}^{0x} &= \frac{\phi_{i+1,j}^{n} - \phi_{i-1,j}^{n}}{2\Delta x}, \quad D_{ij}^{0y} = \frac{\phi_{i,j+1}^{n} - \phi_{i,j-1}^{n}}{2\Delta y}, \\
\nabla^{+} &= [\max(D_{ij}^{-x}, 0)^2 + \min(D_{ij}^{+x}, 0)^2 + \max(D_{ij}^{-y}, 0)^2 + \\
&\quad \min(D_{ij}^{+y}, 0)^2]^{1/2}, \\
\nabla^{-} &= [\max(D_{ij}^{+x}, 0)^2 + \min(D_{ij}^{-x}, 0)^2 + \max(D_{ij}^{+y}, 0)^2 + \\
&\quad \min(D_{ij}^{-y}, 0)^2]^{1/2}.
\end{aligned}$$

A detailed description of the principle behind this numerical method is described in [33]. We note that more efficient implementations of geometric deformable models have been developed, including the particularly noteworthy narrow-band level set method described in [25, 67].

Demos and Coffee Break

References

[13] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: active contour models,” *Int’l J. Comp. Vis.*, vol. 1, no. 4, pp. 321–331, 1987.

[21] A. A. Amini, T. E. Weymouth, and R. C. Jain, “Using dynamic programming for solving variational problems in vision,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 12, no. 9, pp. 855–867, 1990.

[22] L. D. Cohen, “On active contour models and balloons,” *CVGIP: Imag. Under.*, vol. 53, no. 2, pp. 211–218, 1991.

[23] T. McInerney and D. Terzopoulos, “A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis,” *Comp. Med. Imag. Graph.*, vol. 19, no. 1, pp. 69–83, 1995.

[24] V. Caselles, F. Catte, T. Coll, and F. Dibos, “A geometric model for active contours,” *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.

[25] R. Malladi, J. A. Sethian, and B. C. Vemuri, “Shape modeling with front propagation: a level set approach,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, 1995.

[26] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” in *Proc. 5th Int’l Conf. Comp. Vis.*, pp. 694–699, 1995.

[27] R. T. Whitaker, "Volumetric deformable models: active blobs," Tech. Rep. ECRC-94-25, European Computer-Industry Research Centre GmbH, 1994.

[28] G. Sapiro and A. Tannenbaum, "Affine invariant scale-space," *Int'l J. Comp. Vis.*, vol. 11, no. 1, pp. 25–44, 1993.

[29] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space," *Int'l J. Comp. Vis.*, vol. 15, pp. 189–224, 1995.

[30] R. Kimmel, A. Amir, and A. M. Bruckstein, "Finding shortest paths on surfaces using level sets propagation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 6, pp. 635–640, 1995.

[31] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Archive for Rational Mechanics and Analysis*, vol. 123, no. 3, pp. 199–257, 1993.

[32] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Computational Physics*, vol. 79, pp. 12–49, 1988.

[33] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge, UK: Cambridge University Press, 2nd ed., 1999.

[34] I. Cohen, L. D. Cohen, and N. Ayache, “Using deformable surfaces to segment 3-D images and infer differential structures,” *CVGIP: Imag. Under.*, vol. 56, no. 2, pp. 242–263, 1992.

[35] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1953.

[39] H. Tek and B. B. Kimia, “Volumetric segmentation of medical images by three-dimensional bubbles,” *Comp. Vis. Imag. Under.*, vol. 65, pp. 246–258, 1997.

[40] L. D. Cohen and I. Cohen, “Finite-element methods for active contour models and balloons for 2-D and 3-D images,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1131–1147, 1993.

[41] P. E. Danielsson, “Euclidean distance mapping,” *Comp. Graph. Imag. Proc.*, vol. 14, pp. 227–248, 1980.

[42] G. Borgefors, “Distance transformations in arbitrary dimensions,” *Comp. Vis. Graph. Imag. Proc.*, vol. 27, pp. 321–345, 1984.

[43] C. Xu and J. L. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Trans. Imag. Proc.*, vol. 7, no. 3, pp. 359–369, 1998.

[46] D. J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Imag. Under.*, vol. 55, no. 1, pp. 14–26, 1992.

[47] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: deformable superquadrics," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, pp. 703–714, 1991.

[48] A. Gupta, L. von Kurowski, A. Singh, D. Geiger, C.-C. Liang, M.-Y. Chiu, L. P. Adler, M. Haacke, and D. L. Wilson, "Cardiac MR image segmentation using deformable models," in *Proc. IEEE Conf. Computers in Cardiology*, pp. 747–750, 1993.

[49] H. Delingette, "Adaptive and deformable models based on simplex meshes," in *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 152–157, 1994.

[50] S. Kumar and D. Goldgof, "Automatic tracking of SPAMM grid and the estimation of deformation parameters from cardiac MR images," *IEEE Trans. Med. Imag.*, vol. 13, pp. 122–132, 1994.

[51] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, pp. 294–402, 1995.

[52] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," *IEEE Trans. Med. Imag.*, vol. 14, pp. 12–24, 1995.

[53] C. Nastar and N. Ayache, "Frequency-based nonrigid motion analysis: application to four dimensional medical images," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, pp. 1067–1079, 1996.

[54] R. Durikovic, K. Kaneda, and H. Yamashita, "Dynamic contour: a texture approach and contour operations," *The Visual Computer*, vol. 11, pp. 277–289, 1995.

[55] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *Proc. 5th Int'l Conf. Comp. Vis.*, pp. 840–845, 1995.

[56] B. B. Kimia, *Conservation Laws and a Theory of Shape.* Ph.D. thesis, McGill Centre for Intelligent Machines, McGill University, Montreal, Canada, 1990.

[57] M. A. Grayson, "Shortening embedded curves," *Annals of Mathematics*, vol. 129, pp. 71–111, 1989.

[58] J. A. Sethian, “Curvature and evolution of fronts,” *Commun. Math. Phys.*, vol. 101, pp. 487–499, 1985.

[59] J. A. Sethian, “A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed,” *J. Differential Geometry*, vol. 31, pp. 131–161, 1989.

[60] G. Sapiro, “Geometric partial differential equations in image analysis: past, present, and future,” in *Proc. IEEE Int’l Conf. Imag. Proc.*, vol. 3, pp. 1–4, 1995.

[61] D. Adalsteinsson and J. A. Sethian, “The fast construction of extension velocities in level set methods,” *J. Computational Phys.*, vol. 148, pp. 2–22, 1999.

[62] J. A. Sethian, *An Analysis of Flame Propagation*. Ph.D. thesis, Dept. of Mathematics, University of California, Berkeley, CA, 1982.

[63] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tennenbaum, “A geometric snake model for segmentation of medical imagery,” *IEEE Trans. Med. Imag.*, vol. 16, pp. 199–209, 1997.

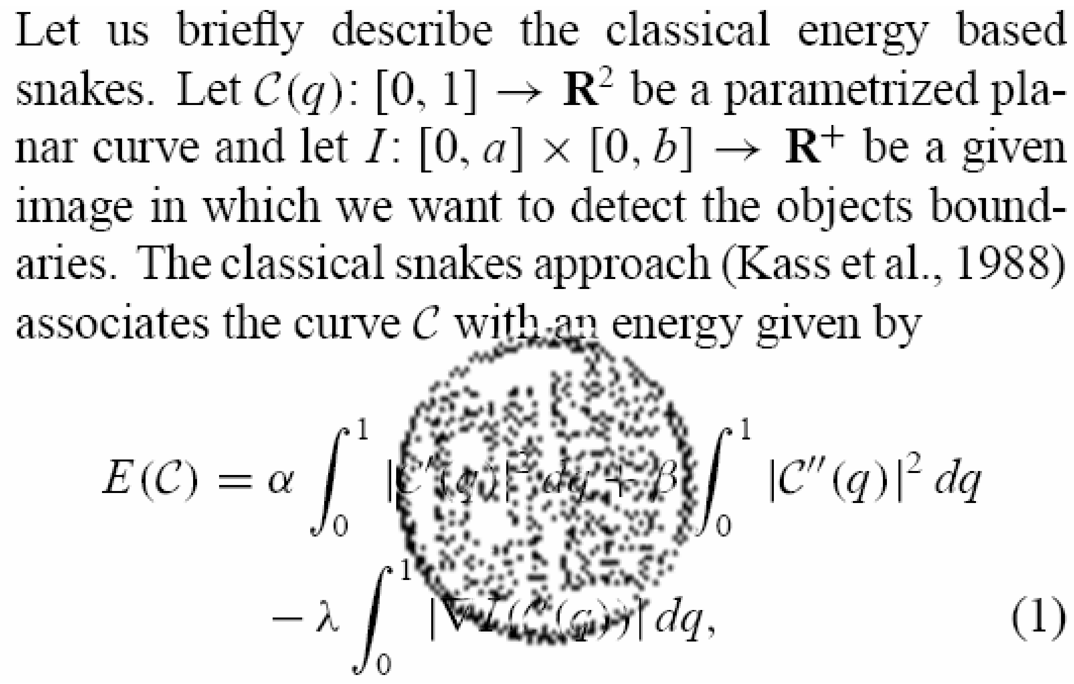
[64] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *Int’l J. Comp. Vis.*, vol. 22, pp. 61–79, 1997.

[65] S. Kichenassamy, A. Kumar, P. Olver, A. Tennenbaum, and A. Yezzi, “Conformal curvature flows: from phase transitions to active vision,” *Arch. Rational Mech. Anal.*, vol. 134, pp. 275–301, 1996.

[66] K. Siddiqi, Y. B. Lauzière, A. Tannenbaum, and S. W. Zucker, “Area and length minimizing flows for shape segmentation,” *IEEE Trans. Imag. Proc.*, vol. 7, pp. 433–443, 1998.

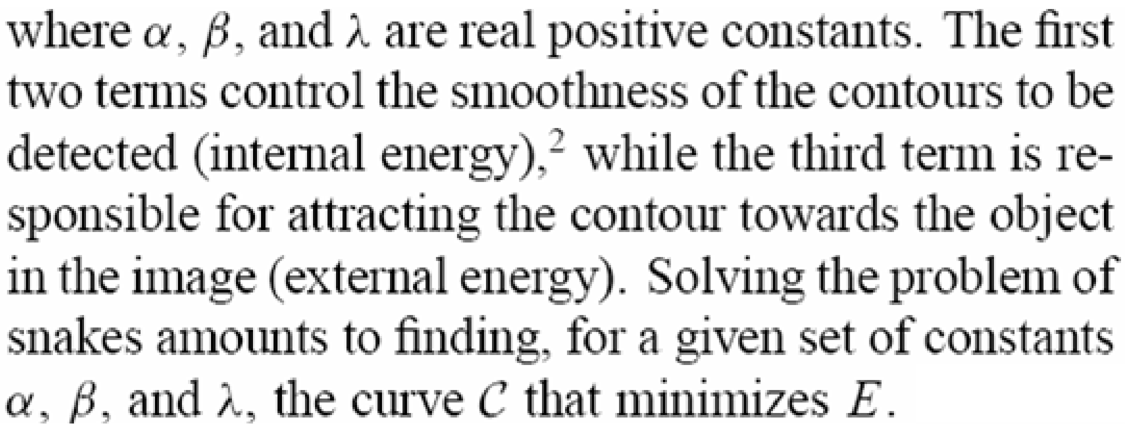
[67] D. L. Chopp, “Computing minimal surfaces via level set curvature flow,” *J. of Comp. Phys.*, vol. 106, pp. 77–91, 1993.

Geodesic Active Contour



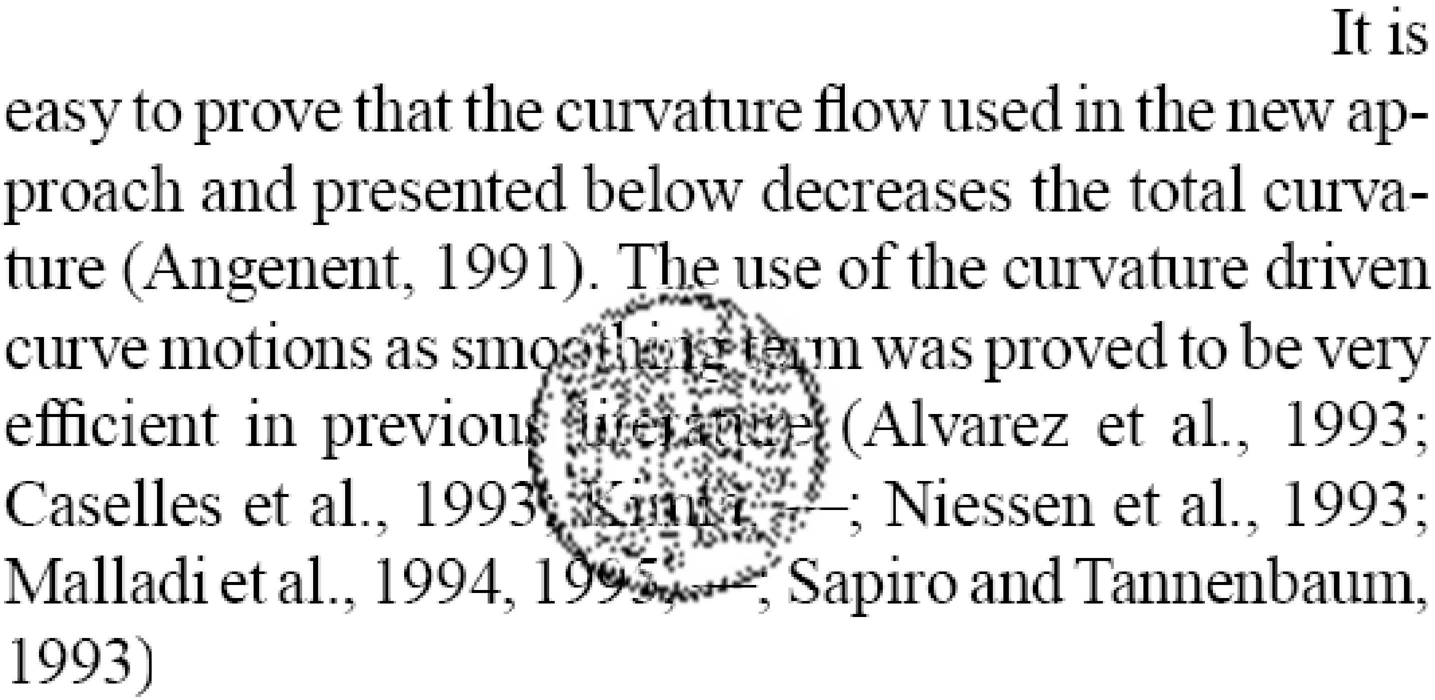
Let us briefly describe the classical energy based snakes. Let $\mathcal{C}(q)\colon [0, 1] \rightarrow \mathbf{R}^2$ be a parametrized planar curve and let $I\colon [0, a] \times [0, b] \rightarrow \mathbf{R}^+$ be a given image in which we want to detect the objects boundaries. The classical snakes approach (Kass et al., 1988) associates the curve $\mathcal{C}$ with an energy given by

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'(q)|^2 \, dq + \beta \int_0^1 |\mathcal{C}''(q)|^2 \, dq - \lambda \int_0^1 |\nabla I(\mathcal{C}(q))| \, dq, \qquad (1)$$



where α, β, and λ are real positive constants. The first two terms control the smoothness of the contours to be detected (internal energy),[2] while the third term is responsible for attracting the contour towards the object in the image (external energy). Solving the problem of snakes amounts to finding, for a given set of constants α, β, and λ, the curve $\mathcal{C}$ that minimizes E.

Validity of $\beta = 0$



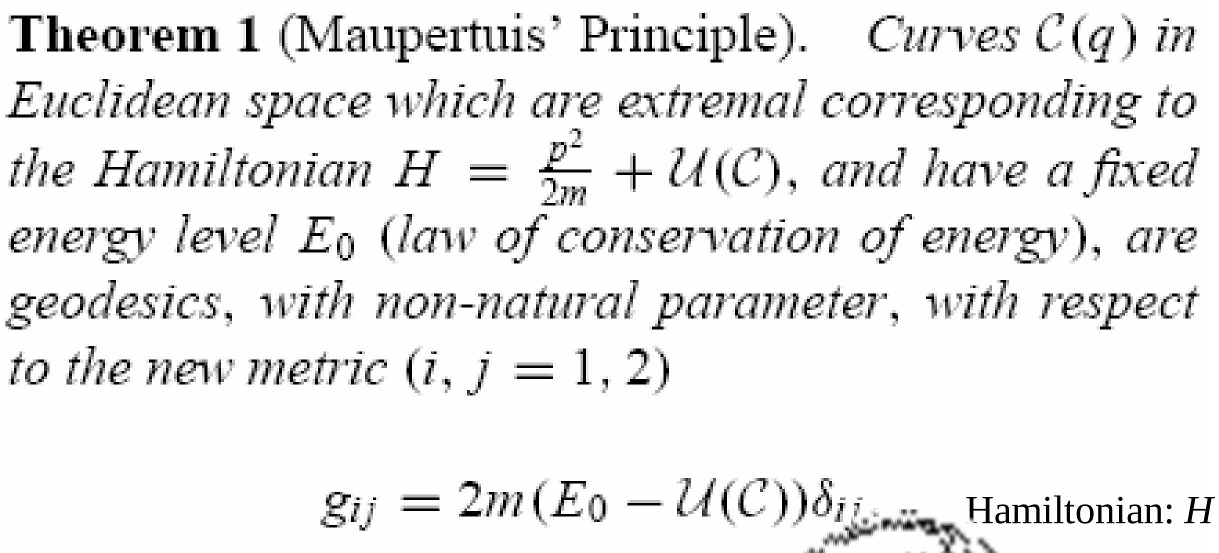
It is easy to prove that the curvature flow used in the new approach and presented below decreases the total curvature (Angenent, 1991). The use of the curvature driven curve motions as smoothing term was proved to be very efficient in previous literature (Alvarez et al., 1993; Caselles et al., 1993; Kimia, —; Niessen et al., 1993; Malladi et al., 1994, 1995, —; Sapiro and Tannenbaum, 1993)

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'(q)|^2 \, dq - \lambda \int_0^1 |\nabla I(\mathcal{C}(q))| \, dq. \quad (2)$$

Observe that, by minimizing the functional (2), we are trying to locate the curve at the points of maxima $|\nabla I|$ (acting as "edge detector"), while keeping certain smoothness in the curve (object boundary). This is actually the goal in the general formulation (1) as well. The tradeoff between edge proximity and edge smoothness is played by the free parameters in the above equations.

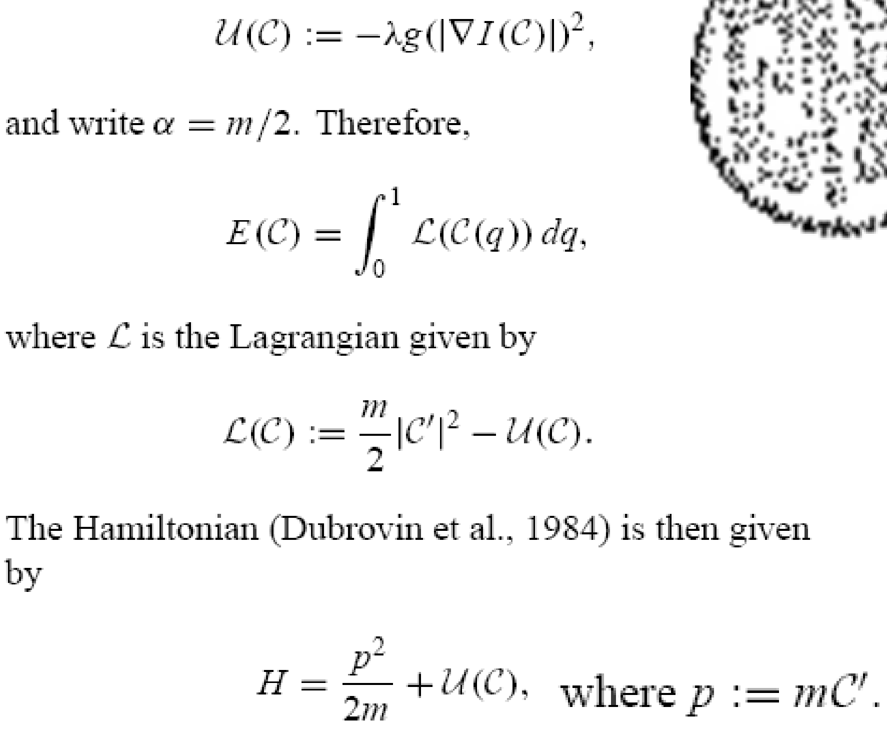
Equation (2) can be extended by generalizing the edge detector part in the following way: Let g: $[0, +\infty[\rightarrow \mathbf{R}^+$ be a strictly decreasing function such that $g(r) \rightarrow 0$ as $r \rightarrow \infty$. Hence, $-|\nabla I|$ can be replaced by $g(|\nabla I|)^2$, obtaining a general energy functional given by

$$\begin{aligned} E(\mathcal{C}) &= \alpha \int_0^1 |\mathcal{C}'(q)|^2 \, dq + \lambda \int_0^1 g(|\nabla I(\mathcal{C}(q))|)^2 \, dq \\ &= \int_0^1 (E_{\text{int}}(\mathcal{C}(q)) + E_{\text{ext}}(\mathcal{C}(q))) \, dq. \end{aligned} \qquad (3)$$



Theorem 1 (Maupertuis' Principle). *Curves $\mathcal{C}(q)$ in Euclidean space which are extremal corresponding to the Hamiltonian $H = \frac{p^2}{2m} + \mathcal{U}(\mathcal{C})$, and have a fixed energy level E_0 (law of conservation of energy), are geodesics, with non-natural parameter, with respect to the new metric $(i, j = 1, 2)$*

$$g_{ij} = 2m(E_0 - \mathcal{U}(\mathcal{C}))\delta_{ij}.$$



$$\mathcal{U}(\mathcal{C}) := -\lambda g(|\nabla I(\mathcal{C})|)^2,$$

and write $\alpha = m/2$. Therefore,

$$E(\mathcal{C}) = \int_0^1 \mathcal{L}(\mathcal{C}(q))\, dq,$$

where $\mathcal{L}$ is the Lagrangian given by

$$\mathcal{L}(\mathcal{C}) := \frac{m}{2}|\mathcal{C}'|^2 - \mathcal{U}(\mathcal{C}).$$

The Hamiltonian (Dubrovin et al., 1984) is then given by

$$H = \frac{p^2}{2m} + \mathcal{U}(\mathcal{C}), \quad \text{where } p := m\mathcal{C}'.$$

Hamiltonian: $H = p\dot{x} - L$

momentem: $p = mass \times velocity = m\dot{x}$

Action: $S = \int L dt = \int \left(p\dot{x} - H \right) dt$

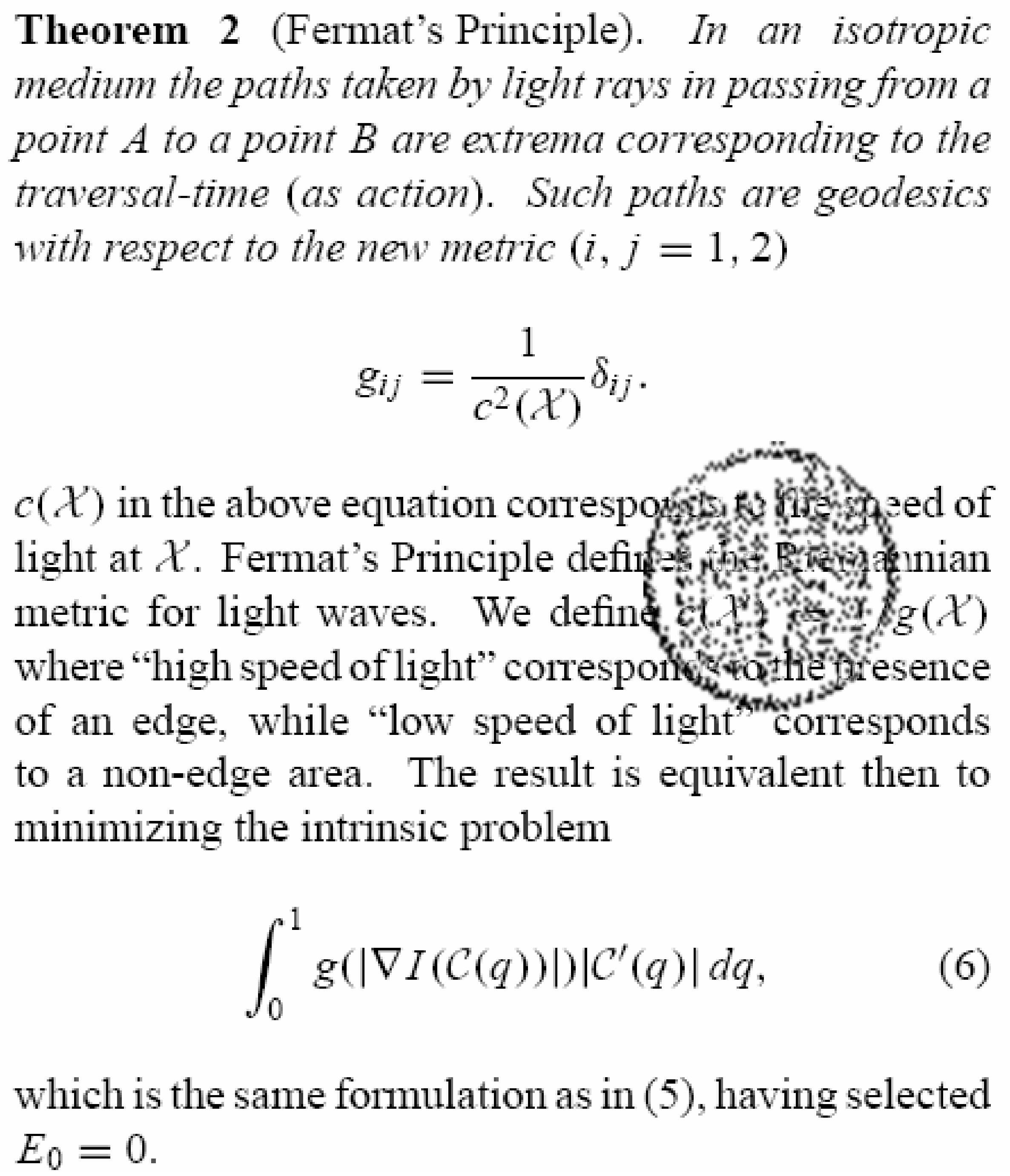
Truncated Action: $S_0 = \int p\dot{x} dt$ with $H = constant = E_0$

Thus, $p = \sqrt{2m\left(E_0 - U(C)\right)}$, and

Recall arclength ds=$\sqrt{g_{ij}\dot{x}^i\dot{x}^j}\,dt$

Thus, $S_0 = \int p\dot{x}dt = \int \dot{x}\sqrt{2m\left(E_0 - U(C)\right)}dt$

$= \int ds = \int \sqrt{g_{ij}\dot{x}^i\dot{x}^j}\,dt$



Theorem 2 (Fermat's Principle). *In an isotropic medium the paths taken by light rays in passing from a point A to a point B are extrema corresponding to the traversal-time (as action). Such paths are geodesics with respect to the new metric* $(i, j = 1, 2)$

$$g_{ij} = \frac{1}{c^2(\mathcal{X})}\delta_{ij}.$$

$c(\mathcal{X})$ in the above equation corresponds to the speed of light at $\mathcal{X}$. Fermat's Principle defines the Riemannian metric for light waves. We define $c(\mathcal{X}) = 1/g(\mathcal{X})$ where "high speed of light" corresponds to the presence of an edge, while "low speed of light" corresponds to a non-edge area. The result is equivalent then to minimizing the intrinsic problem

$$\int_0^1 g(|\nabla I(\mathcal{C}(q))|)|\mathcal{C}'(q)|\,dq, \qquad (6)$$

which is the same formulation as in (5), having selected $E_0 = 0$.

$$\begin{aligned} S_0 &= \int p|\dot{x}|dt \\ &= E_0\int \frac{|\dot{x}|}{c(x)}dt \\ &= E_0\int \sqrt{g_{ij}\dot{x}^i\dot{x}^j}\,dt \end{aligned}$$

Therefore, with $E_0 = 0$, and $g_{ij} = 2m\lambda g(|\nabla I(C)|)^2 \delta_{ij}$, Eq. (4) becomes

$$\text{Min} \int_0^1 \sqrt{2m\lambda}\, g(|\nabla I(C(q)|)|C'(q)|\, dq. \qquad (7)$$

Since the parameters above are constants, without loss of generality we can set now $2\lambda m = 1$ to obtain

$$\text{Min} \int_0^1 g(|\nabla I(C(q))|)|C'(q)|\, dq. \qquad (8)$$

We have transformed the problem of minimizing (2) into a problem of geodesic computation in a Riemannian space, according to a new metric.

Since $|\mathcal{C}'(q)|\, dq = ds$, we obtain

$$L_R := \int_0^{L(\mathcal{C})} g(|\nabla I(\mathcal{C}(q)|)\, ds. \tag{12}$$

$$\frac{\partial \mathcal{C}(t)}{\partial t} = g(I)\kappa \vec{N} - (\nabla g \cdot \vec{N})\vec{N}, \tag{13}$$

Therefore, based on (8) and embedding (13) in u, we obtain that solving the geodesic problem is equivalent to searching for the steady state solution ($\frac{\partial u}{\partial t} = 0$) of the following evolution equation ($u(0, \mathcal{C}) = u_0(\mathcal{C})$):

$$\begin{aligned} \frac{\partial u}{\partial t} &= |\nabla u| \mathrm{div} \left(g(I) \frac{\nabla u}{|\nabla u|} \right) \\ &= g(I) |\nabla u| \mathrm{div} \left(\frac{\nabla u}{|\nabla u|} \right) + \nabla g(I) \cdot \nabla u \\ &= g(I) |\nabla u| \kappa + \nabla g(I) \cdot \nabla u, \end{aligned} \tag{14}$$

Area and Length Minimizing Flows

1. Length Gradient Flow
   Geometric Heat Equation: curvature motion

Let $\mathcal{C} = \mathcal{C}(p,t)$ be a smooth family of closed curves where t parametrizes the family and p the given curve, say $0 \leq p \leq 1$. (Note we assume that $\mathcal{C}(0,t) = \mathcal{C}(1,t)$ and similarly for the first derivatives.) Define the length functional

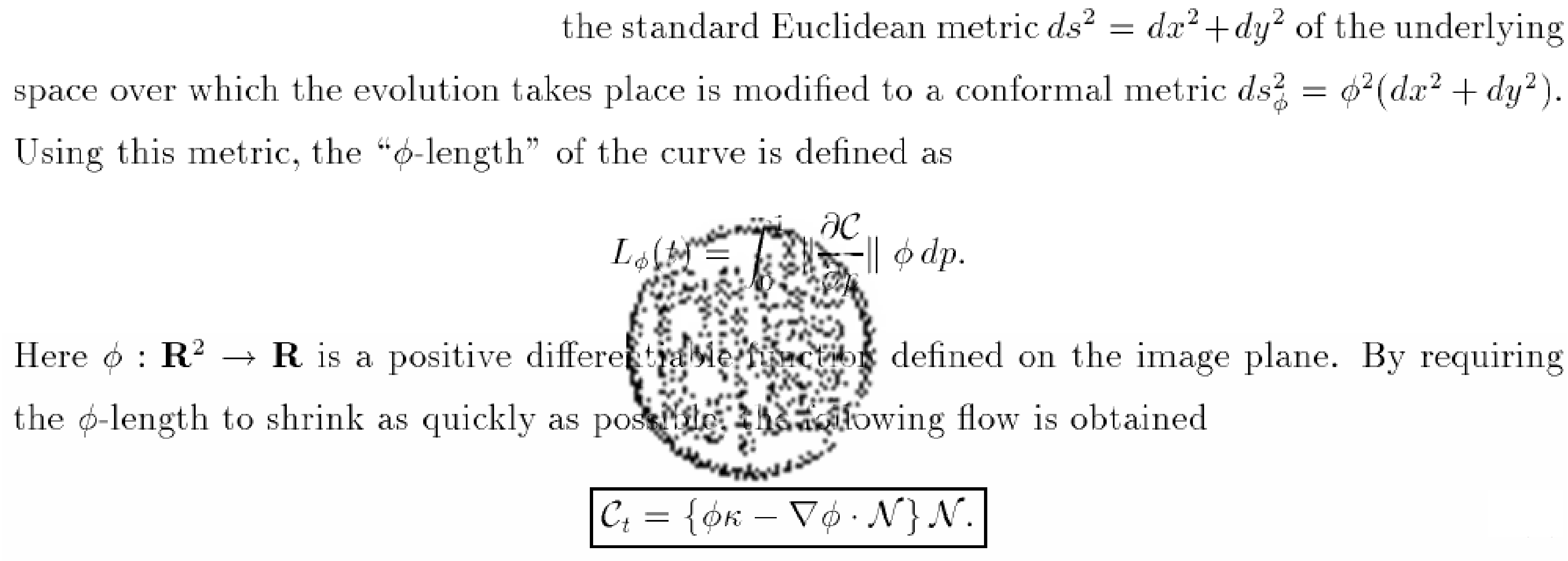
$$L(t) = \int_0^1 \|\frac{\partial \mathcal{C}}{\partial p}\| \, dp.$$

Differentiating (taking the "first variation" with respect to t), and using integration by parts, one can show that

$$L'(t) = -\int_0^{L(t)} \langle \frac{\partial \mathcal{C}}{\partial t}, \kappa \mathcal{N} \rangle \, ds \ ,$$

where $ds = \|\frac{\partial \mathcal{C}}{\partial p}\| \, dp$ denotes arc-length. Thus the direction in which $L(t)$ is decreasing most rapidly is when $\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}$.

2. Weighted Length Gradient Flow



the standard Euclidean metric $ds^2 = dx^2 + dy^2$ of the underlying space over which the evolution takes place is modified to a conformal metric $ds_\phi^2 = \phi^2(dx^2 + dy^2)$. Using this metric, the "ϕ-length" of the curve is defined as

$$L_\phi(t) = \int_0^1 \|\frac{\partial \mathcal{C}}{\partial p}\| \, \phi \, dp.$$

Here $\phi : \mathbf{R}^2 \to \mathbf{R}$ is a positive differentiable function defined on the image plane. By requiring the ϕ-length to shrink as quickly as possible, the following flow is obtained

$$\boxed{\mathcal{C}_t = \{\phi\kappa - \nabla\phi \cdot \mathcal{N}\}\,\mathcal{N}.}$$

3. Area Gradient Flow

In analogy to the geometric heat equation which minimizes Euclidean length, this evolution may be derived as the gradient flow which locally minimizes area. Indeed, for the family of closed curves defined above, the area functional is given by

$$A(t) = -\frac{1}{2}\int_0^L \langle \mathcal{C}, \mathcal{N} \rangle \, ds = -\frac{1}{2}\int_0^1 \langle \mathcal{C}, \begin{pmatrix} -y_p \\ x_p \end{pmatrix} \rangle \, dp.$$

Again, taking the first variation

$$A'(t) = -\frac{1}{2}\int_0^1 \langle \mathcal{C}_t, \begin{pmatrix} -y_p \\ x_p \end{pmatrix} \rangle \, dp - \frac{1}{2}\int_0^1 \langle \mathcal{C}, \begin{pmatrix} -y_{pt} \\ x_{pt} \end{pmatrix} \rangle \, dp.$$

Using integration by parts for the second integral and changing to arc-length parametrization

$$A'(t) = -\int_0^L \langle \mathcal{C}_t, \mathcal{N} \rangle ds.$$

Thus the direction in which $A(t)$ is decreasing most rapidly (locally) is when $\mathcal{C}_t = \mathcal{N}$ and (5) also defines a *gradient flow.*

4. Weighted Area Gradient Flow

$$A_\phi(t) = -\frac{1}{2}\int_0^{L(t)} \phi\langle \mathcal{C}, \mathcal{N}\rangle\, ds = -\frac{1}{2}\int_0^1 \phi\,\langle \mathcal{C}, \begin{pmatrix} -y_p \\ x_p \end{pmatrix}\rangle\, dp.$$

Here $\phi : \mathbf{R}^2 \to \mathbf{R}$ is a positive differentiable function defined on the image plane. We now derive the flow associated with the ϕ-area, A_ϕ. As above, differentiating the functional with respect to t will give us the evolution equation for the curve:

$$-2A_\phi'(t) = I_1 + I_2 + I_3$$

with

$$\begin{aligned} I_1 &= \int_0^L \langle \nabla\phi, \mathcal{C}_t\rangle\,\langle \mathcal{C}, \mathcal{N}\rangle\, ds, \\ I_2 &= \int_0^L \phi\,\langle \mathcal{C}_t, \mathcal{N}\rangle\, ds, \\ I_3 &= \int_0^1 \langle \phi\,\mathcal{C}, \begin{pmatrix} -y_{pt} \\ x_{pt} \end{pmatrix}\rangle\, dp. \end{aligned}$$

For I_3 using integration by parts, we get

$$I_3 = -\int_0^1 \langle (\phi \mathcal{C})_p, \begin{pmatrix} -y_t \\ x_t \end{pmatrix} \rangle \, dp.$$

We will use the following notation: let $V = (a, b)$ be a vector, its "perp" is defined by

$$V^\perp = (-b, a).$$

With respect to the scalar product, we have the following properties

$$\begin{aligned} \langle V_1, V_2^\perp \rangle &= -\langle V_1^\perp, V_2 \rangle \\ \langle V_1^\perp, V_2^\perp \rangle &= \langle V_1, V_2 \rangle. \end{aligned} \tag{6}$$

Using this, we rewrite I_3 as follows

$$\begin{aligned} I_3 &= -\int_0^1 \langle (\phi \mathcal{C})_p, (\mathcal{C}_t)^\perp \rangle \, dp \\ &= \int_0^1 \langle \mathcal{C}_t, (\phi \mathcal{C})_p^\perp \rangle \, dp. \end{aligned}$$

But

$$\begin{aligned}(\phi\mathcal{C})_p &= \langle\nabla\phi,\mathcal{C}_p\rangle\mathcal{C}+\phi\,\mathcal{C}_p\\ \Rightarrow(\phi\mathcal{C})_p^\perp &= \langle\nabla\phi,\mathcal{C}_p\rangle\mathcal{C}^\perp+\phi\,\mathcal{C}_p^\perp,\end{aligned}$$

hence

$$\begin{aligned}I_3 &= \int_0^1\langle\mathcal{C}_t,\langle\nabla\phi,\mathcal{C}_p\rangle\mathcal{C}^\perp+\phi\,\mathcal{C}_p^\perp\rangle\,dp\\ &= \int_0^1\langle\nabla\phi,\mathcal{C}_p\rangle\langle\mathcal{C}_t,\mathcal{C}^\perp\rangle\,dp+\int_0^1\phi\langle\mathcal{C}_t,\mathcal{C}_p^\perp\rangle\,dp.\end{aligned}$$

Using equation (6) this can be rewritten as

$$I_3=\int_0^1\langle\mathcal{C}_t,\mathcal{C}^\perp\rangle\langle\nabla\phi^\perp,\mathcal{C}_p^\perp\rangle\,dp+\int_0^1\phi\langle\mathcal{C}_t,\mathcal{C}_p^\perp\rangle\,dp,$$

and finally changing to arc length parametrization

$$I_3=\int_0^L\langle\mathcal{C}_t,\mathcal{C}^\perp\rangle\langle\nabla\phi^\perp,\mathcal{N}\rangle\,ds+\int_0^L\langle\mathcal{C}_t,\phi\mathcal{N}\rangle\,ds.$$

Grouping everything together, we get

$$-2A'_\phi(t)=\int_0^L\langle\mathcal{C}_t,\langle\mathcal{C},\mathcal{N}\rangle\nabla\phi+2\phi N+\langle\nabla\phi^\perp,\mathcal{N}\rangle\mathcal{C}^\perp\rangle\,ds.$$

and finally changing to arc length parametrization

$$I_3 = \int_0^L \langle \mathcal{C}_t, \mathcal{C}^\perp \rangle \langle \nabla \phi^\perp, \mathcal{N} \rangle \, ds + \int_0^L \langle \mathcal{C}_t, \phi \mathcal{N} \rangle \, ds.$$

Grouping everything together, we get

$$-2A'_\phi(t) = \int_0^L \langle \mathcal{C}_t, \langle \mathcal{C}, \mathcal{N} \rangle \nabla \phi + 2\phi N + \langle \nabla \phi^\perp, \mathcal{N} \rangle \mathcal{C}^\perp \rangle \, ds.$$

Therefore for A_ϕ to decrease as fast as possible, take

$$\mathcal{C}_t = \phi \mathcal{N} + \frac{1}{2} \left[\langle \mathcal{C}, \mathcal{N} \rangle \nabla \phi + \langle \nabla \phi^\perp, \mathcal{N} \rangle \mathcal{C}^\perp \right].$$

Now decomposing $\nabla \phi$ and $\mathcal{C}^\perp$ in the Frenet frame $\{T, \mathcal{N}\}$, and dropping the tangential terms, which can always be done by reparametrizing the curve, we end up with

$$\mathcal{C}_t = \left\{ \phi + \frac{1}{2} \left[\langle \nabla \phi, \mathcal{N} \rangle \langle \mathcal{C}, \mathcal{N} \rangle + \langle \nabla \phi^\perp, \mathcal{N} \rangle \langle \mathcal{C}^\perp, \mathcal{N} \rangle \right] \right\} \mathcal{N}.$$

The last result can be simplified further. Writing $\nabla\phi = (\phi_x, \phi_y)$, $\mathcal{C}(p,t) = (x(p,t), y(p,t))$ and expanding the scalar products we obtain

$$\langle \nabla\phi, \mathcal{N}\rangle\langle \mathcal{C}, \mathcal{N}\rangle + \langle \nabla\phi^{\perp}, \mathcal{N}\rangle\langle \mathcal{C}^{\perp}, \mathcal{N}\rangle = \langle \mathcal{C}, \nabla\phi\rangle.$$

Hence the ϕ-area minimizing evolution equation takes on the following simple form

$$\boxed{\mathcal{C}_t = \left\{\phi + \frac{1}{2}\langle \mathcal{C}, \nabla\phi\rangle\right\}\mathcal{N}.}$$

$$\mathcal{C}_t = \psi \mathcal{N}, \tag{8}$$

may be derived by minimizing a weighted area functional of the form

$$\int \psi \; dxdy.$$

Via Green's theorem, this is equivalent to minimizing the modified area function $A_\phi(t)$ proposed above. In fact using Green's theorem, we have that

$$A_\phi(t) = -\frac{1}{2}\int_0^{L(t)} \phi\langle \mathcal{C}, \mathcal{N}\rangle ds = -\frac{1}{2}\int\int \text{div}\left(\begin{pmatrix} x \\ y \end{pmatrix} \phi\right) dxdy.$$

Thus for given ψ, we may find ϕ, by solving

$$\psi = \frac{1}{2}\text{div}\left(\begin{pmatrix} x \\ y \end{pmatrix} \phi\right),$$