Bilateral Active Contours

14-05-2018

Flinders University
BY
Noura Alhawiti

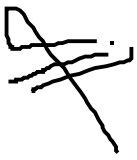
Supervisor Murk Bottema

Submitted to the College of Science and Engineering at Flinders University in partial fulfilment of the requirements for the degree of Master of Science (Mathematics) - Adelaide, Australia.

Contents

Declaration of originality

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

sign:  date: 14-05-2018

Abstract

Active contours (deformable curves), also known as "snakes", were introduced as a general tool for delineating boundaries in computational image analysis. Snakes are considered one of the most successful variational models used in image segmentation. The aim of this method is to segment an image into a finite number of important regions. The theoretical basis of the method is drawn from the calculus of variations theory. Many adaptations of the original method have been introduced over the years using mathematical properties and efficient numerical schemes, usually at the implementation level to improve performance on particular types of problems. Snakes are energy minimising and they balance internal forces that oppose deformation and image forces that pull it towards object contours. This thesis introduces snakes specialised to incorporate bilateral symmetry. The purpose of the study is to develop a theoretical basis for the method of active contours specific to the biological and medical application that directly takes into account first order bilateral symmetry. Human-made objects are often constructed with exact bilateral symmetry; however, in biology, bilateral symmetry is common as a first approximation but not necessarily at all levels of detail. Most vertebrates exhibit bilateral symmetry, as do, for example, the leaves of many plants. This theoretical study proceeds by constructing definitions of functionals and deriving properties analytically. First, an analytical solution are derived from first principles. Second, a discrete solution is derived from the analytical solution to implement the method on real data. Next, a computer program is written in Matlab to implement the bilateral snake. Finally, the computer code is validated on an example image of a human face.

Acknowledgment

I would first like to thank my advisor, Murk Bottema, of the College of Science and Engineering/Faculty of Science and Engineering at Flinders University. The door to Professor Bottema's office was always open whenever I ran into trouble or had a question about my research or writing. He allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it. I would also like to thank my family for supporting me during difficult times. I am truly thankful for all that they have done and sacrificed for me.

1 Introduction

Automatic segmentation of medical images has captured the attention of many researchers. Over the past 35 years, many methods have been developed for the automatic segmentation of medical images. Typically, medical images such as X-rays, magnetic resonance imaging (MRI), positron emission tomographic (PET) images and ultrasound, as well as images related to biological problems, differ substantially from images encountered in areas such as industrial applications, security, non-destructive evaluation and military applications of computational image analysis. Accordingly, there is scope to re-evaluate the basic theoretical basis for active contours to develop variations that are better suited for biological and medical applications. It is possible to segment images manually, but this often takes a long time and is subject to operator variability. Therefore, the manual method is not considered viable in situations where large throughput is required or where objectivity and consistency are important. However, significant problems must be overcome to achieve good segmentation [30].

Snakes can solve the problem of segmenting images through the minimisation of energy involving region and boundary functionals. Using the standard methods of the calculus of variations and the corresponding Euler-Lagrange equations, we can study a large class of region functionals with appropriate boundary conditions. Snakes segment the images by accurately localising the lines and edges. They provide a unified account of a number of visual problems, including detection of edges, lines and subjective contours [13]. In addition, snakes cover objects with poorly defined boundaries by using the technique of curve evolution to find objects in a given image. The main idea of this model is to develop a curve (contour) that starts around the object to be detected, moves towards the interior of the object and stops on its boundary [5].

In general, there are two main types of active contour models: edge-based and region-based. Edge-based models apply the image gradient information to stop the evolving contours on the boundaries of the object. These models have an edge-based stopping term to control the motion of the contour. The disadvantage of these models is that they are very sensitive to the initial curve and noise. As a result, they have limited applications in practice. Region-based models apply the region information of the image to segment different regions. They provide better performance for images with weak object boundaries, and they are less sensitive to initial contours. The disadvantage of these models is that they depend on intensity homogeneity, so the segmentation of images fails if there is intensity inhomogeneity. To overcome this problem, different models have been introduced that use local information, for example, the region-scalable fitting (RSF) model in [17] and the local Gaussian distribution fitting (LGDF) model in [28].

To guide the active contour, a variety of image features are applied. These typically include region statistics, image gradient, colour and texture [31]. Further, many techniques have been proposed in the literature to extract the features in images. For example, in [24], Sarti and Malladi used the concept of level-set methods for the filtering and segmentation of echocardiographic images. This approach uses regularisation to fill in the edge-gaps and improve edge fidelity. In [22], Paragios, Mellina-Gottardo and Ramesh combined the integration of the gradient vector flow and the geodesic active contour with a geometric boundary-based flow for boundary extraction and image segmentation. When such a flow is implemented using a stable numerical method that has real-time performance, it can deal naturally with topological change, it exhibits certain desirable characteristics and it is relatively free of the initial conditions. In [7], Cohen made the curve model behave like a balloon by introducing a pressure force. To obtain more stable results, Cohen modified the definition of external forces deriving from the gradient of the image.

The outline of this paper is as follows. Chapter 2 provides the background on active contour

models (e.g., parametric models) and includes an example of the behaviour of traditional deformable contours and geometric models. Chapter 3 presents a brief review of the calculus of variations needed to understand snakes. Formulas for snakes are derived in Chapter 3, and the original snake as introduced by Kass is described in Chapter 4, along with examples-in particular, demonstrating the effectiveness of original snakes. Chapter 5 presents the main contribution of this thesis—namely, the development of the bilateral snake. This technique is demonstrated by showing the results of boundary detection in a face image. Chapter 6 presents brief concluding remarks and discusses directions for future work.

2 Snakes: Active Contour Models

2.1 Background

Since image processing began, many researchers have addressed the issue of image segmentation, which consists of dividing an image into disjoint regions. Previously, filters were used to detect edges as approaches for image segmentation (e.g., Sobel filter [27] and Canny filter [1]). Using pixel classification, there are two types of object boundaries: edge and non-edge pixels based on a threshold [12]. Data of the image can be easily analysed using the locating boundaries of objects and any method of variation-for example, variations that include sensor noises, variations in photometric and background conditions or perspective distortion or certain distortion, concealment of target objects by other objects, and motion or deformation of target objects. If there are problems with models that have geometric properties, these problems can be solved more easily. However, in many practical situations, traditional methods such as rigid models are unsuitable because objects of the same class are not identical and even the objects' shape can vary over time. To overcome this problem, different models have been introduced in the past few decades, including Fourier series shape models, finite element models, statistical shape models and deformable template models [14].

In general, there are two types of active contours–parametric active contour and geometric active contour–both of which are based on the theory of curve evolution implemented through level-set techniques. Many modifications and enhancements have been made to improve their performance and change their behaviour. Parametric active contours are formulated by minimising an energy functional that takes a minimum when contours are smooth and match the object boundaries. Solving the energy minimisation problem results in a dynamic equation that has both internal and external forces. The external forces resulting from this formulation are deemed conservative because they can be written as gradients of scalar potential functions. Many types of external forces have been developed over the years, including the well-known pressure force and the Gaussian potential force. Geometric active contours were introduced as the solution to the problem of required topological changes during curve evolution. In geometric models, the evolution of the level-set function is first computed, followed by the parametric representations of the curves themselves [6]. The geometric contour can split and incorporate the contour to detect multiple objects in the image, and to model the features of complex shapes, such as sharp corners [20]. Geometric active contours have been deemed the most significant tool in computer vision, and it is widely applied in motion analysis, visual tracking and object recognition [35].

Many researchers have examined the relationship between parametric and geometric active contours. For example, in [4], Caselles et al. found that their geometric active contours were equivalent to a special class of classical parametric active contours. Further, in [6], Chenyang et

al. found that the general formulations of parametric active contours were equivalent to those of geometric active contours. This equivalence included rigidity and highlighted the Lagrangian nature of parametric active contours and the Eulerian nature of geometric active contours.

2.2 Parametric models

Snakes are considered a special case of deformable model. The deformable contour model is mapped as follows

$$r : \Omega = [0, 1] \rightarrow R^2$$
$$s \longmapsto r(s) = (x(s), y(s)).$$

A deformable model is defined as a space of admissible deformations A and a functional E. This functional represents the energy of the model that will be minimised, and it has the following form $E : A \rightarrow R$.

$$E(r)_{snake} = \int_0^1 \left(\frac{\alpha}{2}||r_s||^2 + \frac{\beta}{2}||r_{ss}||^2 + E_{ext}(r(s)) \right) ds, \tag{2.2.1}$$

where α and β are positive control parameters [8]. Here, $r_s(s)$ and $r_{ss}(s)$ are the first and second derivatives of $r(s)$ with respect to s.
To emphasise that E represents both internal and external energy, Equation (2.2.1) may be written as a force balance equation

$$E = E_{int} + E_{ext}, \tag{2.2.2}$$

where

$$E_{int} = \int_0^1 (\frac{\alpha}{2}||r_s||^2 + \frac{\beta}{2}||r_{ss}||^2)ds,$$

and E_{ext} has a small value in the homogeneous region of the image and a large value near the edge. The internal force makes the snake smooth, and the external force pulls the snake towards the desired image features [18].
Agrey-level image $F(x, y)$ is viewed as a function of continuous position variables (x, y). User-imposed constraint forces guide the snake near features of interest. Typical external energies that are modelled to lead an active contour towards a boundary are

$$E^1_{ext} = -|\nabla F(x, y)|^2,$$
$$E^2_{ext} = -|\nabla (G_\sigma(x, y) * F(x, y))|^2, \tag{2.2.3}$$

where ∇ is the gradient operator and G_σ is a two-dimensional Gaussian function with standard deviation σ, and $(*)$ denotes convolution. If the image is a line drawing (black on white), then appropriate external energies include

$$E^3_{ext}(x, y) = F(x, y),$$
$$E^4_{ext} = G_\sigma(x, y) * F(x, y), \tag{2.2.4}$$
$$E^i_{ext} = v(x). \tag{2.2.5}$$

Here the edge map

$$f(x, y) = E^i_{ext}$$

where $i = 1, 2, 3$ or 4. The gradient vector flow (GVF) field $v(x)$ is an equilibrium solution to the following vector diffusion equation

$$u_t = \mu\nabla^2 u - |\nabla f|^2(u - \nabla f),$$
$$u(x, 0) = \nabla f(x).$$

To minimise $E(r)$, the snake should satisfy the Euler-Lagrange equation

$$\alpha r''(s,t) - \beta r''''(s,t) - \nabla E_{ext}(r) = 0. \tag{2.2.6}$$

To solve (2.2.6), the snake needs to be treated as a function of time t and solved in dynamic system. Thus, Equation (2.2.6) becomes

$$-\frac{\partial r(s,t)}{\partial t} = \alpha r''(s,t) - \beta r''''(s,t) - \nabla E_{ext}. \tag{2.2.7}$$

The term $\frac{\partial r(s,t)}{\partial t}$ vanishes when the solution of (2.2.6) stabilises, and the solution of (2.2.7) is achieved. Further, a numerical solution can be found by solving (2.2.7) iteratively. Most implementations of deformable contour use either a parameter that multiplies ∇E_{ext} in order to separate control of the external force strength, or a parameter that multiplies x_t, which controls the step-size temporarily [33].

2.3 Geometric models

Geometric models define the contour as the zero-level set of a higher dimensional function. The evolution of the function is then computed as an initial value problem [9].
To solve (2.2.1) by using Casella's formulation, let O be an object defined as functions of t. The geometrical active contours based on the level-set evolution curvetake the following form

$$\frac{\partial u}{\partial t} = g(x)|\nabla u|(div(\frac{\nabla u}{|\nabla u|}) + v), \tag{2.3.1}$$
$$(t, x) \in [0, \infty[\times R^2,$$
$$u(0, x) = u_0(x), x \in R^2. \tag{2.3.2}$$
$$g(x) = \frac{1}{1 + |\nabla(G_\sigma(x, y) * F(x, y))|^2}, \tag{2.3.3}$$

where u_0 indicates the distance of the initialised active contour curve, $|\nabla u|div(\frac{\nabla u}{|\nabla u|})$, $(t, x) \in [0, \infty[\times R^2$ describes the level sets of the mean curvature motion of the function u, $\{u(t, x) = k, k \in R\}$, which evolve following the normal direction with speed depending on the mean curvature. $v > 0$ is a constant that works to attract curve towards the boundary by increasing the speed of evolution. G represents a Gaussian kernel. O is an object contour with the Gaussian $G_\sigma = C\sigma^{-\frac{1}{2}}exp(-|x|^2/4\sigma)$. $G_\sigma * F$ is the convolution of the image F. u_0 is the initial data taken from the function $1 - x_c$ as a smoothed version, where x_c is the characteristic function of a set, C, containing O. $g(x)$ is a function that is used to stop the level set to reproduce the desired snake [2]. Thus, the magnitude of the high gradient that denotes the possible existence of an edge is mapped by g to small values, whereas flat regions in the image are mapped by g to one [15]. Object edges boundaries can reach boundaries by calculating just one iteration in Equation (2.3.3) at the beginning. This keeps the step iteration unchanged. Moreover, the edge map function $g(x)$ is 1 inside or outside a homogeneous region, and 0 on the boundary of the edge. The term $(\nabla G_\sigma * F)$ is 0 everywhere, except where the gradient of

the image changes to a higher value. Thus, $g(x) \approx 1$ is away from boundaries, and it declines to 0 near sharp changes in the image gradient [10].

To interpret the model in (2.3.1) geometrically, let k be a real number and the boundary of the zero-level set of $u(t, x, y) \in C^2$

$$\partial c(t) = \{(x, y) : u(t, x, y) = k\},$$

also consider

$$\partial C = \{(x(s), y(s)) : s \in [0, L(\partial C)]\},$$

where $L(\partial C)$ is the length of C (where C is sets that contains the object), as $u(t, x(s), y(s)) = k$. Thus

$$\begin{aligned} u_x^{'} x_s^{'} + u_y^{'} y_s^{'} &= 0, \\ y_s^{'} &= \lambda u_x^{'}, \\ x_s^{'} &= \lambda u_y^{'}. \end{aligned} \tag{2.3.4}$$

In case $\lambda > 0$, taking the derivative to (2.3.4) with respect to s gives that

$$u_{xx}^{''}(x_s^{'})^2 + 2u_{xy}^{''} x_s^{'} y_s^{'} + u_{yy}^{''}(y_s^{'})^2 + u_x^{'} x_s^{''} + u_y^{'} y_s^{''} = 0$$

The curvature ρ of ∂C at the point $(x(s), y(s))$ can be written as

$$\rho(s) = \frac{x_s^{'} y_s^{''} - y_s^{'} x_s^{''}}{((x_s^{'})^2 + (y_s^{'})^2)^{\frac{1}{2}}}$$

Written in terms of u, this becomes

$$\rho(x) = \lambda^2 \frac{u_{xx}^{''}(u_y^{'})^2 - 2u_{xy}^{''} u_x^{'} u_y^{'} + u_{yy}^{''}(u_x^{'})^2}{((x_s^{'})^2 + (y_s^{'})^2)^{\frac{1}{2}}}$$

Lastly, since $(x_s^{'})^2 + (y_s^{'})^2 = 1 = \lambda^2((u_x^{'})^2 + (u_y^{'})^2)$, the curvature ρ of ∂C at the point $(x(s), y(s))$ becomes [2]

$$\rho = div(\frac{\nabla u}{|\nabla u|})$$

.The geometrical interpretation of this model is detailed below

1. The term $|\nabla u|(div(\frac{\nabla u}{|\nabla u|})$ guarantees that the grey-level set(A grey-level set is a set of the form $\{p : X(p) = c\}$ for a fixed c.) at a point in ∂C will evolve proportionally towards the boundary curvature at this point.
2. The constant v is a correction term that is selected to keep the term $|\nabla u|(div(\frac{\nabla u}{|\nabla u|}) + v)$ always positive. Hence, the grey-level at a point at ∂C will increase (going from black to grey), and it can be expected that the ∂C will transcribe the boundary of object O. Further, when the curvature of ∂C becomes null or negative, the constant v can work as a force pushing ∂C towards 0.
3. The term $|\nabla u|$ is accountable for the regularising effect of the model. It guarantees that the function u barely changes its grey-level, except on the neighbourhood of ∂C.
4. The term $g(x)$ controls ∂C moves by controlling the speed. When ∂C nears the boundary of object O, $|\nabla G_\sigma * F|$ is large and ∂C stops.
5. The image F is convolved to remove the effect of noise on the motion of ∂C. This coefficient reduces the growth of the function u near the boundary of the object O, and it stops just within the boundary if the boundary is a regular curve [3].

2.4 Basic Snake Behaviour

Since active contour were introduced by Kass, Witkin and Terzopoulos, many researchers have been studied their behaviour. This section discusses some of researchers who addressed a problem with U-shaped objects.

The traditional snake performs well if the boundary of the object is convex or nearly so. The standard snake often does not follow boundaries well into concavities of objects. An example of this problem in a traditional deformable contour is displayed in Fig. 1, where Fig. 1(a) displays a (64×64) pixel line-drawing of a U-shaped object (in grey) that has concave boundary at the top of the shape. The figure also displays a sequence of curves (in black) that iteratively describes the progression of a traditional snake towards the boundary with $\alpha = 0.6$ and with no pressure forces ($\beta = 0.0$). Fig. 1(b) presents a potential force field $-\nabla E_{ext}^4$ and $\sigma = 1.0$ pixels. Thus, the final solution resolves the Euler equations of the snake formulation, but the split remains across the concave boundary.

There are two main reasons for the poor convergence in Fig. 1(a). First, the capture range is limited in Fig. 1(b), and the magnitude of the external forces decreases rapidly away from the object's boundary. Increasing σ in (2.2.4) will increase this range; however, the localization of the boundary will become less accurate and different, and the concavity will therefore become too large. Second, a close-up of the external force field within the boundary concavity causes poor convergence of the deformable contour. The boundary concavity of the forces points horizontally in opposite directions to the object boundary, whereas the external forces point towards the object boundary. Therefore, the deformable contour is pulled away towards each edge of the U-shape, but not made to progress downward into the concave boundary. There are no values of α and β that will correct this problem [34].

In [8], Cohen and Cohen proposed an external force model that critically raised the capture

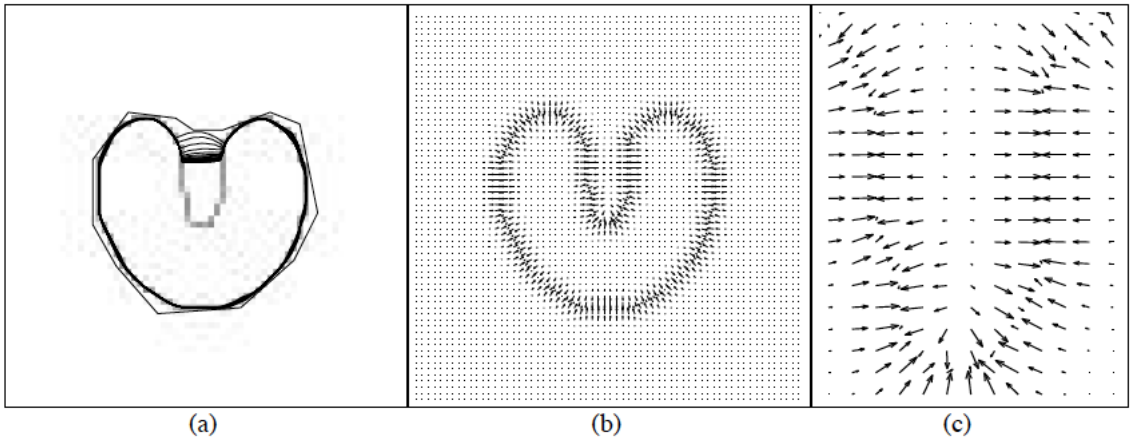


Figure 1: Standard snakes and concavity. (a) shows a concave object and a sequence of iterations of a standard snake. The snake does not enter deeply into the concavity. (b) shows the magnitudes of the external forces. (c) shows a close-up view of the external forces within the concavity. Sourced from [34].

range of a traditional deformable model. These forces will be referred to as distance potential forces to distinguish them from traditional potential forces. These forces are the negative gradient of a potential function that is computed using a Euclidean or chamfer distance map.
Fig. 2(a) displays both the U-shaped object (in grey) and a sequence of contours (in black)

to describe the progression of the snake from its initialisation far from the object to its final configuration. Fig. 2(b) presents distance potential forces that have vectors with large magnitudes away from the object, describing why the range of capture is large for this external-force model. Fig. 2(c) displays distance potential forces, which are similar to traditional potential forces. These forces also point horizontally in opposite directions to the object boundary, and they pull the snake apart, but not downward into the concave boundary. Cohen and Cohen's modification applies a nonlinear transformation to the distance map. This changes the forces' magnitudes, but not the direction of the forces. As a result, the snake fails to converge to the boundary concavity. This can be explained by checking the magnified portion of the distance potential forces in Fig. 2(c).

In [34], Xu and Prince proposed a novel external force model for deformable models called the

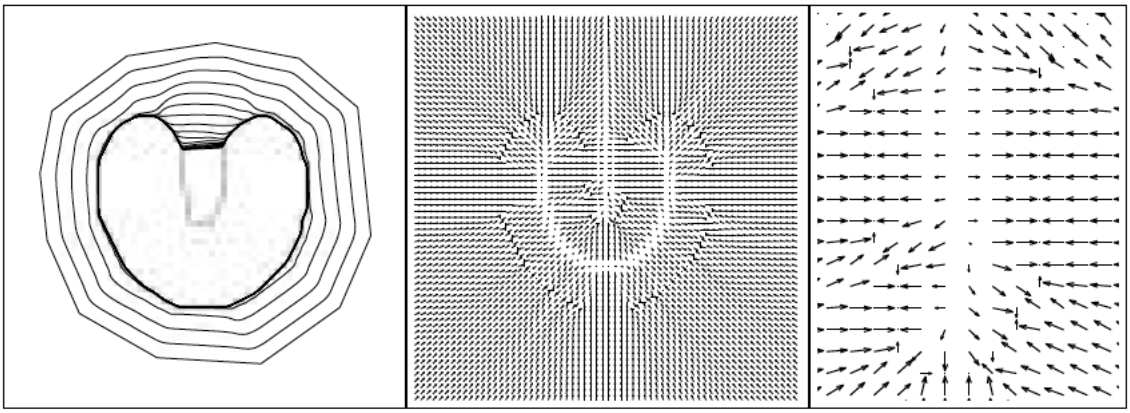


Figure 2: snakes with distance potential forces and concavity. (a) shows a concave object and a sequence of iterations of a standard snake. The snake does not enter deeply into the concavity. (b) the magnitudes of the distance potential forces. (c) a closeup view of the distance potential forces within the concavity. A snake cannot converge to concave regions with distance potential forces field. Sourced from [33].

GVF field, which allows for flexible initialisation of the deformable model.
Fig. 3(a) displays the U-shaped object (in grey-level) corrupted by additional white Gaussian noise; the signal-to-noise ratio is 6 dB. Fig. 3(b) displays an edge map that was computed using $f^2(x,y) = |\nabla(G_\sigma(x,y) * F(x,y))|^2$ with $\sigma = 1.5$ pixels. Fig. 3(c) shows the computed GVF field. The weaker gradients are smoothed out and the stronger edge-map gradients are retained. Fig. 3(d) displays the GVF deformable contours sequence (in a shade of grey) and the result of the GVF deformable contour (in white). Despite the initialisation from far away, as well as the image noise and the boundary concavity, the result shows excellent convergence to the boundary.

A different example of problem is displayed in Fig. 4 with $\alpha = 1.5$, $\beta = 0$, $\gamma = 1$, $\tau = 0.5$, $\lambda = 1.5$, $\lambda_1 = \lambda_2 = 1$. This example uses different parameter values than others example presented in this thesis.
In [16], Li and Acton proposed a vector field convolution (VFC) model. A parametric active contour model replaces the original external force in Equation (2.2.3), which is a potential force, with a non-potential force

$$f_{VFC}(x,y) = [u_{VFC}(x,y), v_{VFC}(x,y)], \tag{2.4.1}$$

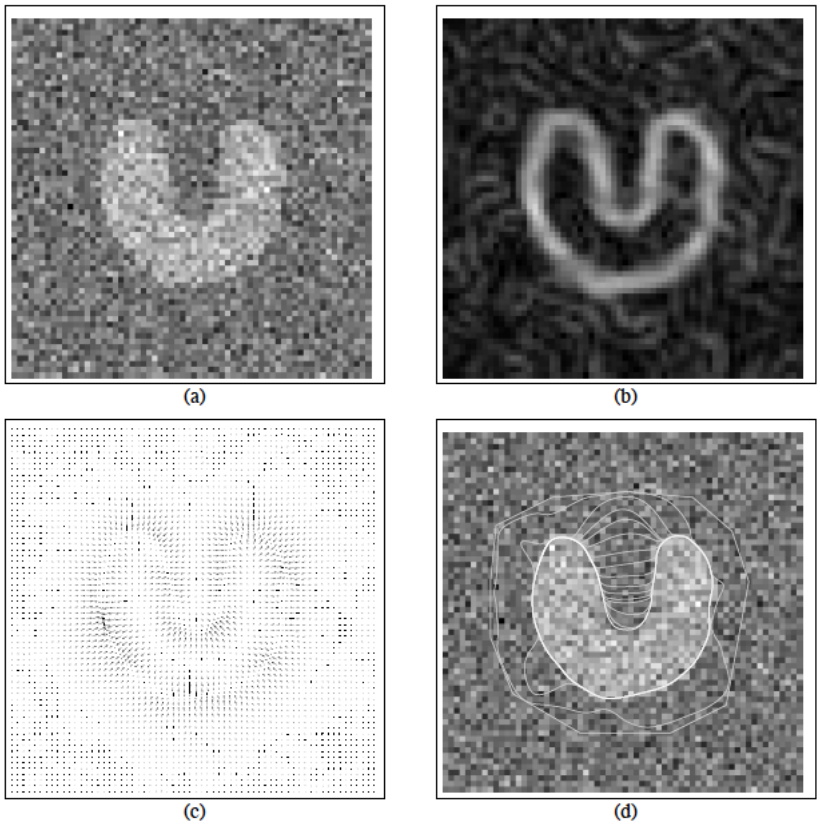


Figure 3: The GVF snake and concavity (a) shows a noisy 64×64-pixel image of the object. (b) shows the edge-map $|\nabla(G_\sigma * F)|^2$ with $\sigma = 1.5$ pixels. (c) shows a close-up view of the GVF external force field. (d) the GVF forces within the concavity. A snake converges perfectly to concave regions with the GVF external force field. Sourced from [34].

which is computed by convolving the vector field kernel $k(x, y)$ and the edge map of the image is $E^i_{ext}(x, y)$, as follows

$$\begin{aligned} f_{VFC}(x, y) &= E^i_{ext}(x, y) * k(x, y) \\ &= [E^i_{ext}(x, y) * u_k(x, y), E^i_{ext}(x, y) * v_k(x, y)], \end{aligned} \tag{2.4.2}$$

where $k(x, y) = [u_k(x, y), v_k(x, y)]$, and $k(x, y) = m(x, y)n(x, y)$. Here $m(x, y)$ is the magnitude of the vector field kernel $k(x, y)$ and is defined as a decreasing positive function of distance from the kernel origin. $n(x, y)$ is the unit vector pointing to the kernel origin $(0, 0)$, and it is calculated by

$$n(x, y) = [-x/r, -y/r],$$

where the distance from the origin is $r = \sqrt{x^2 + y^2}$. The final evolution function of the VFC snake is

$$\frac{\partial v(s, t)}{\partial t} = \alpha r''(s, t) - \beta r''''(s, t) + f_{VFC}(r(s, t)). \tag{2.4.3}$$

Fig. 4(a) shows that VFC is still sensitive to noise and can be easily trapped in cluttered regions of an image because it only considers edge information.

In [5], Chan and Vese proposed fitting energy based on the techniques of curve evolution,

Mumford–Shah functional for segmentation and level sets. This model uses the geometric active contour framework and is not based on an edge function to stop the evolving curve on the desired boundary. It can detect object whose boundaries are not necessarily defined by the gradient, or very smooth. This model can be written as

$$F_1(r) + F_2(r) = \int_{\Omega} |F(x,y) - c_1|^2 dxdy + \int_{\overline{\Omega}} |F(x,y) - c_2|^2 dxdy, \tag{2.4.4}$$

where Ω is inside curve r and $\overline{\Omega}$ is outside r. c_1 and c_2 are the average intensities of the regions inside and outside contour r respectively. The fitting energy definition is used to define a region information map for a point on the contour r as $-\lambda_1(F - c_1)^2 + \lambda_2(F - c_2)^2$. To generate a valid force field in order to be embedded into a parametric active contour model, its outward normal is multiplied by the value of the map at this point. Thus, the region force becomes

$$\begin{aligned} f_{reg}(x,y) &= n(x,y)(f_1(x,y) + f_2(x,y)) \\ &= n(x,y)(-\lambda_1(F(x,y) - c_1)^2 + \lambda_2(F(x,y) - c_2)^2) \end{aligned} \tag{2.4.5}$$

where $n(x,y)$ is the normal vector at point (x,y), $f_1(x,y) = -\lambda_1(F(x,y) - c_1)^2$ and $f_2(x,y) = \lambda_2(F(x,y) - c_2)^2$. For a point (x,y) in the domain of image F, if this point is inside the desired object, $F(x,y)$ is similar to c_1. However, there is a relatively large difference between $F(x,y)$ and c_2. Hence, $f_1(x,y)$ is approximately 0 and $f_2(x,y)$ is larger than 0, assuming $\lambda_1 = \lambda_2$. $f_1 + f_2$ is positive, and when multiplied by the outward normal, the direction of the region force points outwards to pull this point towards the object boundary. If the point (x,y) is outside the object, $f_2(x,y)$ is approximately 0 and $f_1(x,y)$ is smaller than 0. $f_1 + f_2$ is negative and the direction of the force points inwards. λ_1 and λ_2 are positive weighting parameters that control the strength of shrinking or expanding the contour.

The Chan-Vese model could segment the U-shaped object as two separate regions, considering only the region force. The evolution function then becomes

$$\frac{\partial r(s,t)}{\partial t} = \alpha r''(s,t) - \beta r''''(s,t) + \lambda f_{reg}(r(s,t)) \tag{2.4.6}$$

Fig. 4(b) demonstrates that because of a lack of edge information and weak boundary information, the contour leaks through the right side.

In [32], Xie and Mirmehdi proposed the region-aided geometric snake (RAGS). This model is independent of any particular segmentation technique, but it is dependent on the quality of the regions produced. The model integrates the gradient flow forces with region constraints, which are composed of the image region vector flow forces obtained through the diffusion of the region segmentation map. The final level-set representation of RAGS is as follows

$$u_t = g(|\nabla F|)(\rho + \alpha)|\nabla u| + \nabla g(|\nabla F|) \cdot \nabla u - \beta \bar{R} \cdot \nabla u, \tag{2.4.7}$$

where $\bar{R}(z) = (w(z), v(z)), z = (x,y)$ is a two-dimensional vector field. α and β work as trade-offs between gradient forces and region forces.

Fig. 4(c) is the result of the RAGS model. It shows that the noise is so small that it is merges into the background. As a result of the region segmentation map, it would not be influenced by the noisy spot. However, the contour also fails to converge into the concavity. In this case, the contour would first leak through the right side of the U-shape, even when the weight of the region force increases.

In [25], Sun, Ray and Zhang proposed a new model that has also been implemented in the parametric active framework. They added the region force with the VFC force to generate the final external force, or the final evolution function, as follows

$$\frac{\partial r(s,t)}{\partial t} = \alpha r''(s,t) - \beta r''''(s,t) + \gamma f_{VFC}(r(s,t)) + \lambda f_{reg}(r(s,t)), \tag{2.4.8}$$

where γ and λ are weighting parameters that control the strength of the VFC force and region force respectively. The numerical solution can be applied using the finite difference method to solve Equation (2.4.8). Updating of the contour can be computed using the following equation

$$(F - \tau A)r^{t+1} = r + \tau(\gamma f_{VFC}^t + \lambda f_{reg}^t), \tag{2.4.9}$$

where A is an $n \times n$ pentadiagonal banded matrix used to compute the internal force, f_{VFC} and f_{reg} are $n \times 2$ matrices denoting the forces at points on the contour. τ is constant.

Fig. 4(d) shows the novel active contour model combining edge and regional information called VFCCV method, which converges to the correct boundary of U-shape and is strong to the noise. For the edge force to control the evolution, γ must be large and λ must be small in Equation (2.4.9). This model not only has a large capture range and the ability to converge into concavities, but it is also robust to noise and a cluttered image background.

Figure 4: Comparison using image with U-shaped gradient object. A snake convergence to concave regions, (a) with the VFC which is easily affected by noise, (b) Chan-Vese snake, the contour leaks through the right side with weak boundary information, (c) RAGS, the contour leak through the right side of the U shape even the noise is so small that it is merged into the background, and (d) VFCCV, the contour converges to the correct U-shape boundary and is robust to the noise. Sourced from [25].

Note that, on all the past work there are no methods that dealt with bilateral symmetry specific with active contour.

3 A brief review of the calculus of variations

The theory of snakes is based on the calculus of variations. There are many variations of the Euler-Lagrange equations, including the number of derivatives in the functional, the number of functions and the number of independent variables. This chapter reviews three variations in detail and follows work by Sagan [23], Van Brunt [26], Logan [19], Gelfand et al. [11], Weinstock [29] and Morrey Jr [21].

3.1 The extrema of functionals

Definition:
Let $A = \{y \in C^2[x_0, x_1] : y(x_0) = y_0, y(x_1) = y_1\}$ be a set of admissible functions where (C^2 means that the function is twice continuously differentiable). Suppose J is a functional defined on A. J has a local minimum at $y \in A$ if there exists a number $\epsilon > 0$ such that
$J(y) \leqslant J(\hat{y})$, for all functions $\hat{y} \in A$, such that $\|\hat{y} - y\| < \epsilon$.
Local maxima are defined in a similar way.

3.2 The simplest case of the Euler-Lagrange Equation

Consider $J : A \to R$ defined by

$$J(\varphi) = \int_{x_0}^{x_1} f(x, \varphi, \varphi^{'})dx, \quad \text{for} \quad \varphi \in A. \tag{3.2.1}$$

Here $\varphi^{'} = \frac{d\varphi}{dx}$, and f is a smooth function that includes three variables, $x, \varphi, \varphi^{'}$, explicitly, but the arguments of f are the variable x and the functions φ and $\varphi^{'}$ viewed as independent variables.
Suppose the set H is defined by

$$H = \{\eta \in C^2[x_0, x_1] : \eta(x_0) = 0, \eta(x_1) = 0\}. \tag{3.2.2}$$

Proposition(1)
Let y be the function that minimises Equation (3.2.1). For all $\hat{y} \in A$, there exists $\eta \in H$, such that

$$\hat{y} = y + \epsilon\eta.$$

Proof. Set $\eta = (\hat{y} - y)/\epsilon$. Since $y \in C^2[x_0, x_1]$ and $\hat{y} \in C^2[x_0, x_1], \eta \in C^2[x_0, x_1]$.
Since $y(t_0) = y_0$ and $\hat{y}(t_0) = y_0, \eta(t_0) = (y_0 - y_0)/\epsilon$; thus, $\eta(y_0) = 0$.
Similarly, $\eta(t_1) = 0$. Hence, $\eta \in H$. □

Using Taylor's theorem to find a formula for $f(x, \hat{y}, \hat{y}^{'})$ expanded about $a = (x, y, y^{'})$, results in the following expression

$$\begin{aligned} f(x, \hat{y}, \hat{y}^{'}) &= f(x, \hat{y} + \epsilon\eta, \hat{y}^{'} + \epsilon\eta^{'}) \\ &= f(x, y, y^{'}) + \frac{\partial f}{\partial x}|_a(\hat{x} - x) + \frac{\partial f}{\partial y}|_a(\hat{y} - y) + \frac{\partial f}{\partial \hat{y}}|_a(\hat{y}^{'} - y^{'}) \\ &+ \text{.....} \end{aligned} \tag{3.2.3}$$

Given that:

$$\hat{y} - y = (y + \epsilon\eta) - y$$

and

$$\hat{y}^{'} - y^{'} = (\hat{y} + \epsilon\eta^{'}) - y^{'}$$

Equation (3.2.3) becomes

$$f(x, \hat{y}, \hat{y}^{'}) = f(x, y, y^{'}) + \frac{\partial f}{\partial y}|_a(\epsilon\eta) + \frac{\partial f}{\partial \hat{y}}|_a(\epsilon\eta^{'}) + O(\epsilon^2).$$

Thus:

$$J(\hat{y}) - J(y) = \int_{x_0}^{x_1} f(x, \hat{y}, \hat{y}')dx - \int_{x_0}^{x_1} f(x, y, y')dx$$

$$= \int_{x_0}^{x_1} \left(f(x, y, y') + \epsilon\{\eta \frac{\partial f}{\partial y} + \eta' \frac{\partial f}{\partial y'}\} + O(\epsilon^2) - f(x, y, y') \right) dx$$

$$= \int_{x_0}^{x_1} \epsilon\{\eta \frac{\partial f}{\partial y} + \eta' \frac{\partial f}{\partial y'}\}dx + O(\epsilon^2). \tag{3.2.4}$$

The first variation of J, is defined by

$$\delta J(\eta, y) = \int_{x_0}^{x_1} \{\eta \frac{\partial f}{\partial y} + \eta' \frac{\partial f}{\partial y'}\}dx. \tag{3.2.5}$$

Thus

$$J(\hat{y}) - J(y) = \epsilon \delta J(\eta, y) + O(\epsilon^2). \tag{3.2.6}$$

Proposition(2)
If $\eta \in H$, then $-\eta \in H$.

Proof. Since $\eta \in C^2[x_0, x_1], -\eta \in C^2[x_0, x_1]$. Since $\eta(x_0) = 0, -\eta(x_0) = 0$ and since $\eta(x_1) = 0, -\eta(x_1) = 0$, thus $-\eta \in H$. □

For sufficiently small ϵ, the sign of the righ-hand side of Equation (3.2.6) is determined by the first term (linear term) unless this term is zero. The left-hand side of Equation (3.2.6) is nonnegative since J has a local minimum at y. If the right side of (3.2.6) is nonnegative for some $\eta \in H$, then it is negative for $-\eta$. This shows that the right-hand side is psitive for all η. The conclusion is that the linear term in (3.2.6) is zero. Hence

$$0 = \int_{x_0}^{x_1} \{\eta \frac{\partial f}{\partial y} + \eta' \frac{\partial f}{\partial y'}\}dx. \tag{3.2.7}$$

To solve the above equation, the integration is first applied by part to the second term. Thus

$$\int_{x_0}^{x_1} \eta' \frac{\partial f}{\partial y'}dx = \eta \frac{\partial f}{\partial y'}\Big|_{x_0}^{x_1} - \int_{x_0}^{x_1} \eta \frac{d}{dx}\frac{\partial f}{\partial y'}dx$$

$$= -\int_{x_0}^{x_1} \eta \frac{d}{dx}\frac{\partial f}{\partial y'}dx.$$

Thus

$$0 = \int_{x_0}^{x_1} \eta \frac{\partial f}{\partial y}dx - \int_{x_0}^{x_1} \eta \frac{d}{dx}\frac{\partial f}{\partial y'}dx.$$

This equation can be written as

$$0 = \int_{x_0}^{x_1} \eta\{\frac{\partial f}{\partial y} - \frac{d}{dx}\frac{\partial f}{\partial y'}\}dx. \tag{3.2.8}$$

This can be written as

$$0 = \left\langle \eta, \frac{\partial f}{\partial y} - \frac{d}{dx}\frac{\partial f}{\partial y'} \right\rangle.$$

This equation holds for all $\eta \in H$.
Generally, if $\langle \eta, g \rangle = 0$, then η and g are orthogonal, but does not say much about either η or g individually. The situation is different if $\langle \eta, g \rangle = 0$ for all $\eta \in H$.

The zero inner product lemma
Assume that g is a continuous function on $[x_0, x_1]$ and that $\langle \eta, g \rangle = 0$ for every $\eta \in H$. Then $g(x) = 0$ for all $x \in [x_0, x_1]$.

Proof. Suppose $g(c) > 0$ for some $c \in [x_0, x_1]$. Since g is continuous, there exist a and b such that $x_0 \leqslant a < b \leqslant x_1$ and $g(x) > 0$ for $x \in (a, b)$. Function η is defined by

$$\eta(x) = \begin{cases} 0 & x \in [x_0, a] \\ [(x-a)(b-x)]^3 & x \in (a, b) \\ 0 & x \in [b, x_1] \end{cases}$$

The objective is to show that $\eta \in H$. Obviously, $\eta(x_0) = \eta(x_1) = 0$. Computing $\eta^{''}$ on (a, b) is tedious, but it can be simplified as follows

$$\eta^{'} = 3[(x-a)(b-x)]^2(-2x+b+a) = [(x-a)(b-x)]^2 P_1(x),$$

where $P_1(x)$ is a polynomial (of degree 1). Further

$$\begin{aligned} \eta^{''} &= [(x-a)(b-x)]^2 P_1^{'}(x) + 2(x-a)(b-x)(-2x+(b+a))P_1(x) \\ &= (x-a)(b-x)P_2(x), \end{aligned}$$

where $P_2(x)$ is a polynomial (of degree 2).

This shows that $\lim_{x \downarrow a} \eta'' = \lim_{x \uparrow b} \eta'' = 0$. Obviously $\eta^{''}$ is zero outside (a, b), and so $\eta^{''}$ is continuous on all of $[x_0, x_1]$. This proves that $\eta \in H$. Note that $\frac{a+b}{2}$ is the only critical point for η and that $\eta(x) > 0$ on (a, b). Consider that

$$\langle \eta, g \rangle = \int_{x_0}^{x_1} \eta(x) g(x) dx = \int_a^b \eta(x) g(x) dx > 0.$$

But $\langle \eta, g \rangle = 0$ for all $\eta \in H$; thus, there is a contradiction. This means that the assumption of the existence of a number $c \in [x_0, x_1]$, for which $g(c) > 0$, must be wrong.
A similar contradiction is found if the assumption is made that $g(c) < 0$ for some $c \in [x_0, x_1]$. Thus the only option is that $g(x) = 0$ for all $x \in [x_0, x_1]$.

□

Applying the zero inner product lemma to Equation (3.2.8) shows that

$$0 = \frac{\partial f}{\partial y} - \frac{d}{dx}\frac{\partial f}{\partial y'}. \tag{3.2.9}$$

Equation (4.2.4) is known as the Euler-Lagrange equation.

3.3 Functionals comprising higher derivatives

Consider the functional

$$J(y) = \int_{x_0}^{x_1} f(x, y, y^{'}, y^{''}, ..., y^{(n)})dx. \tag{3.3.1}$$

The set of admissible functions is
$A = \{y \in C^{2n}[x_0, x_1] : y(x_0) = y_0, y(x_1) = y_1, y^{'}(x_0) = y_0^{'}, y^{'}(x_1) = y_1^{'}, ..., y^{(n-1)}(x_0) = y_0^{(n-1)}, y^{(n-1)}(x_1) = y_1^{(n-1)}\}$.

Suppose that the set H is defined by
$H = \{\eta \in C^{2n}[x_0, x_1] : \eta(x_0) = 0, \eta(x_1) = 0, \eta^{'}(x_0) = 0, \eta^{'}(x_1) = 0, ...\eta^{(n-1)}(x_0) = 0, \eta^{(n-1)}(x_1) = 0\}$.

The proof follows the steps of the proof for the n = 1 case, and gives the following equation

$$\frac{d^{(n)}}{dx^{(n)}}\frac{\partial f}{\partial y^{(n)}} - \frac{d^{(n-1)}}{dx^{(n-1)}}\frac{\partial f}{\partial y^{(n-1)}} + \frac{d^{(n-2)}}{dx^{(n-2)}}\frac{\partial f}{\partial y^{(n-2)}} - ... + (-1)^n\frac{\partial f}{\partial y} = 0.$$

In particular, for $n = 2$, the Euler-Lagrange equation is

$$\frac{d^2}{dx^2}\frac{\partial f}{\partial y^{''}} - \frac{d}{dx}\frac{\partial f}{\partial y^{'}} + \frac{\partial f}{\partial y} = 0.$$

3.4 Functionals comprising multiple functions

A functional may depend on more than one function. To be specific, consider n separate functions y_i, $i = 1, 2, ..., n$. These may be viewed as components of a function. Let Y denote the vector valued function $Y(x) = (y_1(x), y_2(x), ..., y_n(x)) : R \to R^n$. Here, we consider functionals of the form

$$J(Y) = \int_{x_0}^{x_1} f(x, y_1, y_1^{'}, y_2, y_2^{'}, ..., y_n, y_n^{'})dx. \tag{3.4.1}$$

With set of admissible functions A

$$A = \{Y \in C^2([x_0, x_1]) : y_i(x_0) = a_i, y_i(x_1) = b_i, i = 1, 2, ..., n\} \tag{3.4.2}$$

If Y is a local minimum on A for J, then there exists $\epsilon > 0$ such that

$$J(\hat{Y}) - J(Y) \geq 0$$

for all $\hat{Y} \in A$ with $|\hat{Y} - Y| < \epsilon$.

Suppose that the set H is defined by

$$H = \{\eta = (\eta_1, \eta_2, ..., \eta_n) \in C^2([x_0, x_1]^n) : \eta_i(x_0) = \eta_i(x_1) = 0, i = 1, 2, ..., n\}. \tag{3.4.3}$$

Using Taylor's theorem

$$f(x, \hat{y}_1, \hat{y}_1^{'}, \hat{y}_2, \hat{y}_2^{'}, ..., \hat{y}_n, \hat{y}_n^{'}) = f(x, y_1, y_1^{'}, y_2, y_2^{'}, ..., y_n, y_n^{'})$$

$$+\epsilon\{\eta_1\frac{\partial f}{\partial y_1} + \eta_1^{'}\frac{\partial f}{\partial y_1^{'}} + \eta_2\frac{\partial f}{\partial y_2} + \eta_2^{'}\frac{\partial f}{\partial y_2^{'}} + ... + \eta_n\frac{\partial f}{\partial y_n} + \eta_n^{'}\frac{\partial f}{\partial y_n^{'}}\} + O(\epsilon^2)$$

Thus

$$J(\hat{Y})-J(Y)=\epsilon\int_{x_0}^{x_1}\{\eta_1\frac{\partial f}{\partial y_1}+\eta_1^{'}\frac{\partial f}{\partial y_1^{'}}+\eta_2\frac{\partial f}{\partial y_2}+\eta_2^{'}\frac{\partial f}{\partial y_2^{'}}+...+\eta_n\frac{\partial f}{\partial y_n}+\eta_n^{'}\frac{\partial f}{\partial y_n^{'}}\}dx+O(\epsilon^2). \quad (3.4.4)$$

This equation holds for all $\eta=(\eta_1,\eta_2,...,\eta_n)\in H$.

For sufficiently small ϵ, the sign of the right-hand side of Equation (3.4.4) is determined by the first term (linear term) unless this term is zero.
By setting $\eta_1=0,\eta_2=0,...,\eta_{i-1}=0,\eta_{i+1}=0,...,\eta_n=0$, Equation (3.4.4) becomes

$$J(\hat{Y})-J(Y)=\epsilon\int_{x_0}^{x_1}\{\eta_i\frac{\partial f}{\partial y_i}+\eta_i^{'}\frac{\partial f}{\partial y_i^{'}}\}dx+O(\epsilon^2). \quad (3.4.5)$$

If this integral is positive for η_i, then it is negative for $-\eta_i$. This contradicts the fact that the left-hand side of Equation (3.4.5) is always nonnegative. This shows that

$$0=\int_{x_0}^{x_1}\{\eta_i\frac{\partial f}{\partial y_i}+\eta_i^{'}\frac{\partial f}{\partial y_i^{'}}\}dx.$$

Applying the integration by part to the second term gives

$$0=\int_{x_0}^{x_1}\eta_i\{\frac{\partial f}{\partial y_i}-\frac{d}{dx}\frac{\partial f}{\partial y_i^{'}}\}dx.$$

Using the zero inner product lemma

$$0=\frac{\partial f}{\partial y_i}-\frac{d}{dx}\frac{\partial f}{\partial y_i^{'}},$$

for $i=1,2,3,...,n$.

4 Kass's Snake

In 1987, Kass, Witkin and Terzopouloi first presented the idea of snakes being used to extract features in images using the idea of energy-minimising curves. The first and most famous model of active contour is edges-based object extraction, which uses the magnitude of the image gradient to derive the external forces. These characterise a boundary as the maximal value of the magnitude of the gradient of the image [5]. In a broad sense, snakes are an example of using the means of energy minimisation as a technique for linking an image to a deformable model. If we present the basic mathematical behaviour of snakes, we find that the basic snake model is a directed continuity spline under the effect of external constraint forces and image forces. Internal spline forces thrust the snake towards salient image features such as edges, lines and subject contours. Forces of external constraint can, for example, come from automatic attentional mechanisms, a user interface or high-level interpretations. These forces are responsible for setting the snake near the required local minimum [13].

4.1 The Euler-Lagrange equations for Kass's snake

A snake may be represented as a parametrised curve $r(s)=(x(s),y(s))$ in R^2. The curve modelled as a physical object with elasticity is given by $\alpha||r_s||^2$, and flexibility is given by $\beta||r_{ss}||^2$. These two quantities are referred to in the literature as the internal properties of the snake.

In addition, the snake is meant to find edges in an image F; thus, the snake is designed to minimise an edge image.

Many other characteristics may be included, such as terms to find lines instead of edges and terminal points of boundaries, etc. In addition, various edge images may be considered. Only the basic snake is of interest here; therefore, these extensions will not be developed here, and only the simple edge image $||\nabla F(x(s), y(s))||^2$ will be considered.

The basic curve minimises the functional

$$J(r) = \int_0^1 (\frac{\alpha}{2}||r_s||^2 + \frac{\beta}{2}\|r_{ss}||^2 - ||\nabla F(r)||^2)ds, \tag{4.1.1}$$

where $r_s = \frac{dr}{ds}$ and $r_{ss} = \frac{d^2r}{ds^2}$. Here, $\alpha||r_s||^2$ is elasticity, $\beta||r_{ss}||^2$ is flexibility, F is the image and ∇F is the edge image.

Equation (4.1.1) can be simplified as

$$J(x, y) = \int_0^1 (\frac{\alpha}{2}(x_s^2 + y_s^2) + \frac{\beta}{2}(x_{ss}^2 + y_{ss}^2) - F_x^2(x, y) - F_y^2(x, y))ds. \tag{4.1.2}$$

The functional $J(x, y)$ is a functional of one independent variable s and two dependent functions x, y. The boundary conditions depend on the problem to be treated. In this case, the boundary is $r(0) = r(1) = r_0$ because closed curves are needed. Here functionals are of the form

$$J(x, y) = \int_{s_0}^{s_1} f(s, x, x_s, x_{ss}, y, y_s, y_{ss})ds.$$

Note that

$$f = \frac{\alpha}{2}(x_s^2 + y_s^2) + \frac{\beta}{2}(x_{ss}^2 + y_{ss}^2) - F_x^2(x, y) - F_y^2(x, y).$$

The Euler-Lagrange equations (3.2) for finding the extrema for J are

$$\frac{\partial f}{\partial x} - \frac{d}{ds}\frac{\partial f}{\partial x_s} + \frac{d^2}{ds^2}\frac{\partial f}{\partial x_{ss}} = 0, \tag{4.1.3}$$

$$\frac{\partial f}{\partial y} - \frac{d}{ds}\frac{\partial f}{\partial y_s} + \frac{d^2}{ds^2}\frac{\partial f}{\partial y_{ss}} = 0. \tag{4.1.4}$$

Written in terms of x and y, Equations (4.1.3) and (4.1.4) become

$$\beta x_{ssss} - \alpha x_{ss} - \frac{\partial}{\partial x}(F_x^2 + F_y^2) = 0, \tag{4.1.5}$$

$$\beta y_{ssss} - \alpha y_{ss} - \frac{\partial}{\partial y}(F_x^2 + F_y^2) = 0. \tag{4.1.6}$$

4.2 Numerical Solutions

To find numerical solutions, Kass and others researches recommended viewing the Euler-Lagrange equations as the stable limit of a dynamic system. Thus, x and y are viewed as functions of both s and t.
The dynamic versions are given by

$$-\frac{\partial x(s,t)}{\partial t} = \beta x_{ssss}(s,t) - \alpha x_{ss}(s,t) - \frac{\partial}{\partial x}(F_x^2(s,t) + F_y^2(s,t)), \tag{4.2.1}$$

$$-\frac{\partial y(s,t)}{\partial t} = \beta y_{ssss}(s,t) - \alpha y_{ss}(s,t) - \frac{\partial}{\partial y}(F_x^2(s,t) + F_y^2(s,t)). \tag{4.2.2}$$

Note that $x(s)$ and $y(s)$ form an equilibrium for Equations (4.2.1) and (4.2.2). This means that they are solutions to (4.1.5) and (4.1.6). Hence, computing the limit $t \to \infty$ or limit $t \to -\infty$ of (4.2.1) and (4.2.2) leads to finding solutions to (4.1.5) and (4.1.6).

Note that, the negative signs on the left-hand side of (4.2.1) and (4.2.2) are used because the equilibrium appears stable as $t \to -\infty$ and unstable as $t \to \infty$.

This concept is applied to the functions $x(s)$ and $y(s)$ at time t, and the functions $x(s)$ and $y(s)$ are viewed as vectors $x_t = (x_1, x_2, x_3, ..., x_n)$ and $y_t = (y_1, y_2, y_3, ..., y_n)$, where n denotes the number of points used to discretise the snake $r(s,t)$.

The discrete derivatives along the curve $r(x,y)$ are as follows
$x_s = x_{i+1} - x_i$,
$x_{ss} = (x_{i+1} - x_i) - (x_i - x_{i-1}) = x_{i+1} - 2x_i + x_{i-1}$,
$x_{sss} = (x_{i+2} - 2x_{i+1} + x_i) - (x_{i+1} - 2x_i + x_{i-1}) = x_{i+2} - 3x_{i+1} + 3x_i - x_{i-1}$,
$x_{ssss} = (x_{i+2} - 3x_{i+1} + 3x_i - x_{i-1}) - (x_{i+1} - 3x_i + 3x_{i-1} - x_{i-2})$,
$= x_{i+2} - 4x_{i+1} + 6x_i - 4x_{i-1} + x_{i-2}$.

Similarly for y

$y_s = y_{i+1} - y_i$,
$y_{ss} = (y_{i+1} - y_i) - (y_i - y_{i-1}) = y_{i+1} - 2y_i + y_{i-1}$,
$y_{sss} = (y_{i+2} - 2y_{i+1} + y_i) - (y_{i+1} - 2y_i + y_{i-1}) = y_{i+2} - 3y_{i+1} + 3y_i - y_{i-1}$,
$y_{ssss} = (y_{i+2} - 3y_{i+1} + 3y_i - y_{i-1}) - (y_{i+1} - 3y_i + 3y_{i-1} - y_{i-2})$,
$= y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}$.

Thus, the discretised versions of $\beta x_{ssss}(s,t) - \alpha x_{ss}(s,t)$ and $\beta y_{ssss}(s,t) - \alpha y_{ss}(s,t)$ are

$$\begin{aligned}
&\beta(x_{i+2} - 4x_{i+1} + 6x_i - 4x_{i-1} + x_{i-2}) - \alpha(x_{i+1} - 2x_i + x_{i-1}) \\
&= \beta x_{i+2} + (-\alpha - 4\beta)x_{i+1} + (2\alpha + 6\beta)x_i + (-\alpha - 4\beta)x_{i-1} + \beta x_{i-2} \\
&= \beta y_{i+2} + (-\alpha - 4\beta)y_{i+1} + (2\alpha + 6\beta)y_i + (-\alpha - 4\beta)y_{i-1} + \beta y_{i-2}.
\end{aligned}$$

The discretised versions of $-\frac{dx(s,t)}{dt}$ and $-\frac{dy(s,t)}{dt}$ are

$$-\frac{x_{t+1} - x_t}{\triangle t},$$

and

$$-\frac{y_{t+1} - y_t}{\triangle t}.$$

Taking this discretisation into account, Equations (4.2.1) and (4.2.2) become

$$-\frac{x_{t+1}-x_t}{\triangle t} = Ax_t - \frac{\partial}{\partial x}(F_x^2(s,t) + F_y^2(s,t)), \tag{4.2.3}$$

$$-\frac{y_{t+1}-y_t}{\triangle t} = Ay_t - \frac{\partial}{\partial y}(F_x^2(s,t) + F_y^2(s,t)), \tag{4.2.4}$$

where A is $n \times n$ five-banded wrap around the matrix defined as follows. Let $a = 2\alpha + 6\beta$ and $b = -\alpha - 4\beta$. Then

$$A = \begin{pmatrix} a & b & \beta & 0 & 0 & 0 \cdots 0 & 0 & 0 & \beta & b \\ b & a & b & \beta & 0 & 0 \cdots 0 & 0 & 0 & 0 & \beta \\ \beta & b & a & b & \beta & 0 \cdots 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & b & a & b & \beta \cdots 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \cdots \beta & b & a & b & \beta \\ \beta & 0 & 0 & 0 & 0 & 0 \cdots 0 & \beta & b & a & b \\ b & \beta & 0 & 0 & 0 & 0 \cdots 0 & 0 & \beta & b & a \end{pmatrix} \tag{4.2.5}$$

Equations (4.2.3) and (4.2.4) become

$$x_{t+1} = x_t - A\triangle t x_t + \frac{\partial}{\partial x}(F_x^2(s,t) + F_y^2(s,t))_t. \tag{4.2.6}$$

$$y_{t+1} = y_t - A\triangle t y_t + \frac{\partial}{\partial y}(F_x^2(s,t) + F_y^2(s,t))_t. \tag{4.2.7}$$

4.3 Experimental Results for Kass's snake

The formulas in Equations (4.2.6) and (4.2.7) were implemented successfully, as shown in Figure 5. These formulas are not the same as those used by Kass, who applied the matrix A to x_{t+1} instead of x_t and seemed to have misplaced the stepsize $\triangle t$. In any event, attempts to implement Kass's version exactly were not successful. All computations and programs were performed in Matlab $^{\text{TM}}$.

The program **MBasicSnake.m** was written to implement Equations (4.2.6) and(4.2.7). The program was tested on an image of a pear downloaded from the internet, as shown in Figure 5. The program was run within the program **KassPearDemo.m**.

The program **KassPearDemo.m** loaded a grayscale image called Pear.mat, which was previously adapted from a colour image downloaded from the internet. The image was padded to allow room for a circular initial snake. The image was blurred using a Gaussian filter size 241×241 with $\sigma = 20$. An initial snake was created that comprised 100 points evenly spaced on a circle of radius 140 pixels and roughly centred on the pear. The blurred image and the coordinates of the initial snake were passed to the program **MBasicSnake.m**.

Equations in (4.2.6) and (4.2.7) were used to update the snake coordinates, and 500 time iterations were computed with $\alpha = 0.2$ and $\beta = 0.2$. Many initial attempts failed because the edge image did not contribute significantly to the updated snake coordinates. Finally an artificial factor of 10 was used to boost the contribution of the edge image to obtain the results shown in Figure 5.

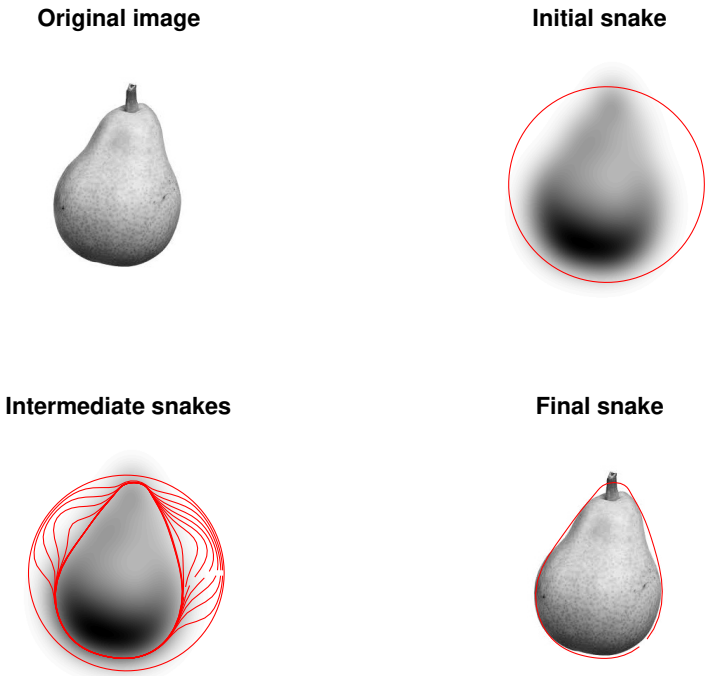


Figure 5: Kass's snake

5 Bilateral snake

This section introduces a new snake called Bilateral Snake. This snake can be achieved from five steps.
1. Two snakes are considered r_1 and r_2. To encourage bilateral symmetry, two terms are added to Kass's version. Thus, the minimising calculation includes four functions: x, y, u and v.
2. Four Euler-Lagrange equations are computed.
3. From the Euler-Lagrange equations to the dynamic system, new vectors $w(s)$ and $z(S)$ are constructed. The first vector contains x and y, and the second vector contains y and v.
4. The discrete derivatives along the curve $r(x, y))$ and $r(u, v)$ are computed to obtain two matrices. The first matrix M is not wrap around, as in the Kass's version, and the second matrix N is an identity matrix.
5. Updating the snake at iteration i can be achieved using a new formula, that obtained from the previous step.

The above steps will be calculated manually. MATLAB will be used, then to confirm the results.

5.1 The Euler-Lagrange equations for Bilateral Snake

Consider the parametric position of two snakes by the curves $r_1(t) = (x(t), y(t))$ and $r_2(t) = (u(t), v(t))$ in R^2. To encourage bilateral symmetry about the axis $x = 0$, penalty terms are introduced for $x(t)$ to be different from $-u(t)$ and for $y(t)$ to be differet from $v(t)$. The bilateral snake minimises the functional

$$J(r) = \int_0^1 (\frac{\alpha}{2}(||r_{1s}||^2 + ||r_{2s}||^2) + \frac{\beta}{2}(||r_{1ss}||^2 + \beta||r_{2ss}||^2) +$$

$$\frac{\gamma}{2}(||x+u||^2+||y-v||^2)-||\nabla F(r_1)||^2-||\nabla F(r_2)||^2)ds. \tag{5.1.1}$$

Numbers 1 and 2 are used to indicate $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial y}$ respectively, thus to avoid confusion between coordinates x and y as a placeholdesr in the image and the functions $x(s)$ and $y(s)$, which specify the coordinates of the snake at parameter s.

Equation (5.1.1) can be viewed as a functional of the four functions x, y, u and v. It may be written as

$$\begin{aligned} J(x,y,u,v) = \int_0^1 (&\frac{\alpha}{2}(x_s^2+y_s^2+u_s^2+v_s^2)+ \\ &\frac{\beta}{2}(x_{ss}^2+y_{ss}^2+u_{ss}^2+v_{ss}^2)+ \\ &\frac{\gamma}{2}((x+u)^2+(y-v)^2) \\ -(F_1^2(x,y)+F_2^2(x,y))&-(F_1^2(u,v)+F_2^2(u,v)))ds. \end{aligned} \tag{5.1.2}$$

The character F_1 denotes partial differentiation in the first component, and F_2 denotes partial differentiation in the second component.

Here, functionals are considered of the form

$$J(x,y,u,v) = \int_{s_0}^{s_1} f(s,x,x_s,x_{ss},y,y_s,y_{ss},u,u_s,u_{ss},v,v_s,v_{ss})ds.$$

Note that

$$\begin{aligned} f = \frac{\alpha}{2}(x_s^2+y_s^2+u_s^2+v_s^2)+\frac{\beta}{2}(x_{ss}^2+y_{ss}^2+u_{ss}^2+v_{ss}^2)+ \\ \frac{\gamma}{2}((x+u)^2+(y-v)^2)-(F_1^2(x,y)+F_2^2(x,y))-(F_1^2(u,v)+F_2^2(u,v)). \end{aligned}$$

Note that $F_x = F_u$ and $F_y = F_v$, so it is enough to compute F_x and F_y.

The four Euler-Lagrange equations for finding the extrema for J are

$$\begin{aligned} \frac{\partial f}{\partial x}-\frac{d}{ds}\frac{\partial f}{\partial x_s}+\frac{d^2}{ds^2}\frac{\partial f}{\partial x_{ss}} &= 0, \\ \frac{\partial f}{\partial y}-\frac{d}{ds}\frac{\partial f}{\partial y_s}+\frac{d^2}{ds^2}\frac{\partial f}{\partial y_{ss}} &= 0, \\ \frac{\partial f}{\partial u}-\frac{d}{ds}\frac{\partial f}{\partial u_s}+\frac{d^2}{ds^2}\frac{\partial f}{\partial u_{ss}} &= 0, \\ \frac{\partial f}{\partial v}-\frac{d}{ds}\frac{\partial f}{\partial v_s}+\frac{d^2}{ds^2}\frac{\partial f}{\partial v_{ss}} &= 0. \end{aligned}$$

In terms of x, y, u and v, the Euler-Lagrange equations are

$$\begin{aligned} \beta x_{ssss}-\alpha x_{ss}-D_1(F_1^2(x,y)+F_2^2(x,y))+\gamma(x+u) &= 0, \\ \beta u_{ssss}-\alpha u_{ss}-D_1(F_1^2(u,v)+F_2^2(u,v))+\gamma(x+u) &= 0, \\ \beta y_{ssss}-\alpha y_{ss}-D_2(F_1^2(x,y)+F_2^2(x,y))+\gamma(y-v)) &= 0, \\ \beta v_{ssss}-\alpha v_{ss}-D_2(F_1^2(u,v)+F_2^2(u,v))-\gamma(y-v) &= 0. \end{aligned}$$

Here, D_1 and D_2 indicate partial differentiation in the first and second coordinates x and y respectively. Note that the partial derivatives of the image F with respect to x are the same as partial derivatives with respect to u, and the partial derivatives of the image F with respect to y are the same as the partial derivative with respect to v. Hence, notation is reduced by setting

$$G_1 = D_1(F_1^2 + F_2^2)$$

and

$$G_2 = D_2(F_1^2 + F_2^2).$$

The quantities G_1 and G_2 are fixed throughout the iterations. They are just evaluated at different points as the iterations take place.

5.2 Numerical Solutions for Bilateral Snake

Viewing x, y, u and v as functions of both s and t, the dynamical system is given by

$$\begin{aligned}
-\frac{\partial x(s,t)}{\partial t} &= \beta x_{ssss}(s,t) - \alpha x_{ss}(s,t) - G_1(x(s,t)) + \gamma(x(s,t) + u(s,t)),\\
-\frac{\partial u(s,t)}{\partial t} &= \beta u_{ssss}(s,t) - \alpha u_{ss}(s,t) - G_1(u(s,t)) + \gamma(x(s,t) + u(s,t)),\\
-\frac{\partial y(s,t)}{\partial t} &= \beta y_{ssss}(s,t) - \alpha y_{ss}(s,t) - G_2(y(s,t)) + \gamma(y(s,t) - v(s,t)),\\
-\frac{\partial v(s,t)}{\partial t} &= \beta v_{ssss}(s,t) - \alpha v_{ss}(s,t) - G_2(v(s,t)) - \gamma(y(s,t) - v(s,t)).
\end{aligned}$$

The first two equations are related because the functions x and u appear explicitly in both equations. Similarly, the last two equations are related because the functions y and v, appear explicitly in both equations. This means that the functions x and u, as well as y and v, must be updated at the same time.
To implement this strategy, construct new vectors

$$w = \begin{pmatrix} x \\ u \end{pmatrix}$$

and

$$z = \begin{pmatrix} y \\ v \end{pmatrix}$$

By comparing n points at each snake (the (x, y) snake and the (u, v) snake), the updating of the snakes at iteration i can be computed using the following equations

$$w_{t+1} = w_t - \triangle t(-P_1 * w_t + G_1(w_t)), \tag{5.2.1}$$

$$z_{t+1} = z_t - \triangle t(-P_2 * z_t + G_2(z_t)), \tag{5.2.2}$$

where P_1 and P_2 are matrices of size $2n \times 2n$ given by

$$P_1 = \begin{pmatrix} M & N \\ N & M \end{pmatrix}, P_2 \begin{pmatrix} M & -N \\ -N & M \end{pmatrix}$$

and M and N are matrices of size $n \times n$ given by

$$N = \begin{pmatrix} \gamma & 0 & 0 & \cdots & 0 \\ 0 & \gamma & 0 & \cdots & 0 \\ 0 & 0 & \gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \gamma \end{pmatrix}$$

$$M = \begin{pmatrix} a & b & \beta & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ b & a & b & \beta & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \beta & b & a & b & \beta & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & b & a & b & \beta & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \beta & b & a & b & \beta \\ \beta & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \beta & b & a & b \\ b & \beta & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \beta & b & a \end{pmatrix}.$$

Here the matrix N is γ times the $n \times n$ identity matrix. The matrix M is not a wrap-around matrix, otherwise, M is similar to matrix A in Equation (4.2.5).

5.3 Experimental Results for Bilateral Snake

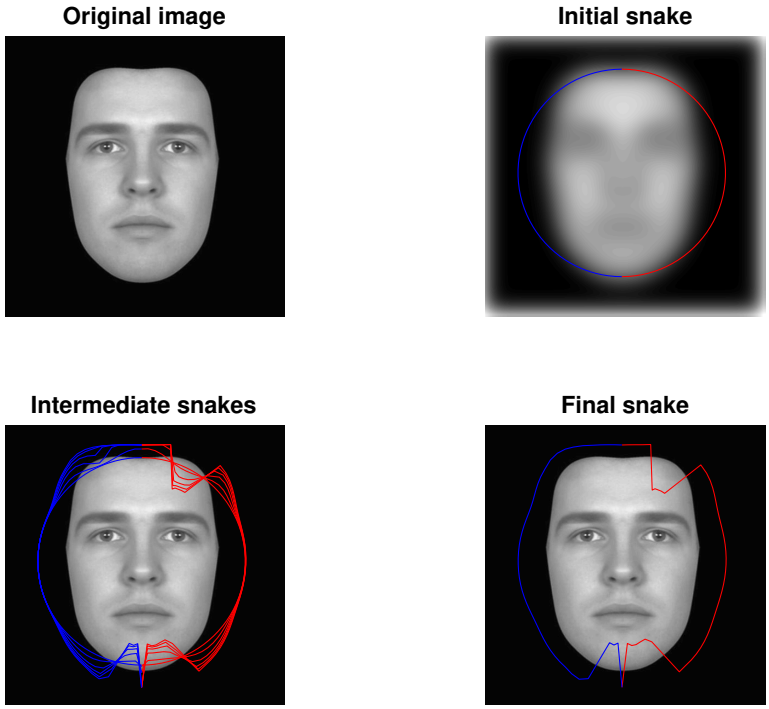


Figure 6: Bilateral Snake

The formulas in Equations (5.2.1) and (5.2.2) were implemented, as shown below.

MBasicSnake1.m was written to implement Equations (5.2.1) and (5.2.2). The program was

tested on an image of a human face downloaded from the internet Figure 6. The program was run within the programme **KassPearDemo1.m**.

KassPearDemo1.m loaded a grayscale image called face.mat, which was previously adapted from a colour image downloaded from the internet. The image was padded to allow room for a circular initial snake. The image was blurred using a Gaussian filter size 241×241 with $\sigma = 20$. An initial snake comprising 50 points evenly spaced on a circle of radius 190 pixels and roughly centred on the face was created. The blurred image and the coordinates of the initial snake were passed to the program **MBasicSnake1.m**.

Equations in (5.2.1) and (5.2.2) were used to update the snake coordinates, and 500 time iterations were computed with $\alpha = 0.0001$, $\beta = 0$, and $\gamma = 0.0001$. Many initial attempts failed because the edge image did not contribute significantly to the updated snake coordinates. Finally, an artificial factor of 10 was used to boost the contribution of the edge image to obtain the results shown in Figure 6.

Figure 6. shows that the curves are not correct (lower right panel), and many attempts were made to discover the cause of this. However, no explanation was found for this. The values of the parameters were changed many times, including time step and boost. Different pictures and different-sized pictures were examined, and x, y, u and v were computed separately, as in a previous version. Further, terms such as $x + u$ and $y + v$ were changed many times.

6 Conclusion and future work

The external force in a GVF model were designed and then the snake was implemented using diffusion equations. GVF model can offer advantages over balloons and distance-based snakes, including a larger capture range and the ability to capture concavities by diffusing the gradient vectors of an edge map generated from the image. Symmetry in biology is the balanced distribution of duplicate body parts or shapes within the body of an organism. In nature and biology, symmetry is always approximate (e.g., plants, freshwater algae).Scientists believe that bilateralism might occur in complex central nervous systems, such as the human spinal cord and its vast network of nerves funnelling into the brain. Bilateralism also fosters the development of improved sensory organs, such as ears and eyes, that help to support the intentional and focused movements of which these types of symmetrical bodies are capable. For example, a starfish has one axis about which it can be equally divided. This paper presented a new active contour model, called the bilateral snake, for image segmentation. The main goal of performing projections is to find a method that can help find the missing symmetrical part of any image or shape in cases where the part is damaged or does not appear. For example, if a lizard loses one of its legs, it can be found using the bilateral snake method. This method can make contributions in many medical and biology areas. The bilateral snake is calculated as two snakes, which are connected by two new terms that encourage bilateral symmetry. Examples on (simulated) and real images were presented. The experimental results indicated that either the equation statement or the code displayed contained an error. In future research, the GVF snake should be implemented to avoid the problem of the snake not 'seeing' the edge. Further, future research should develop a method to take into account several boundaries at once. For example, in automatic delineating of a cell on a microscope slide, artefacts and other components in the sample may lead to a false positive identification of a cell, or the outline of a true cell may be missed. If the cells in question are known to contain a single nucleus, for example, then the simultaneous delineation of both the cell boundary and the nucleus boundary may

reduce the number of errors in each of these steps alone.
The GVF model is designed

References

[1] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[2] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.

[3] Vicent Caselles and B Coll. Snakes in movement. *SIAM Journal on Numerical Analysis*, 33(6):2445–2456, 1996.

[4] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.

[5] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[6] A. Chenyang Xu, J.L. Yezzi, and J.L. Prince. On the relationship between parametric and geometric active contours. volume 1, pages 483–489. IEEE Publishing, 2000.

[7] Laurent D Cohen. On active contour models and balloons. *CVGIP: Image understanding*, 53(2):211–218, 1991.

[8] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1131–1147, November 1993.

[9] Foued Derraz, Mohamed Beladgham, and M'hamed Khelif. Application of active contour models in medical image segmentation. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 675–681. IEEE, 2004.

[10] Foued Derraz, Abdelmalik Taleb-Ahmed, Azzeddine Chikh, and Fethi Bereksi-Reguig. Improved edge map of geometrical active contour model based on coupling to anisotropic diffusion filtering. In *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, pages 1097–1101. IEEE, 2007.

[11] Izrail Moiseevitch Gelfand, Richard A Silverman, et al. *Calculus of Variations*. Courier Corporation, 2000.

[12] Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29(1):100–132, 1985.

[13] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[14] Jae Seon Kim, Kyoung Chul Koh, and Hyung Suck Cho. An active contour model with shape regulation scheme. *Advanced Robotics*, 14(6):495–514, 2000.

[15] Ron Kimmel. *Numerical Geometry of Images: Theory, algorithms, and applications*. Springer Science & Business Media, 2012.

[16] Bing Li and Scott T Acton. Active contour external force using vector field convolution for image segmentation. *IEEE transactions on image processing*, 16(8):2096–2106, 2007.

[17] Chunming Li, Chiu-Yen Kao, John C Gore, and Zhaohua Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE transactions on image processing*, 17(10):1940–1949, 2008.

[18] Lixiong Liu and Alan Bovik. Active contours with neighborhood-extending and noise-smoothing gradient vector flow external force. *EURASIP Journal on Image and Video Processing*, 2012(1):1–6, December 2012.

[19] J David Logan. *Applied mathematics*. John Wiley & Sons, 2013.

[20] Pauline Merveilleux, Ouiddad Labbani-Igbida, and El Mustapha Mouaddib. Real-time free space detection and navigation using omnidirectional vision and parametric and geometric active contours. pages 6312–6317. IEEE Publishing, May 2011.

[21] Charles Bradfield Morrey Jr. *Multiple integrals in the calculus of variations*. Springer Science & Business Media, 2009.

[22] N. Paragios, O. Mellina-Gottardo, and V. Ramesh. Gradient vector flow fast geometric active contours. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(3):402–407, March 2004.

[23] Hans Sagan. *Introduction to the Calculus of Variations*. Courier Corporation, 1969.

[24] A. Sarti and R. Malladi. A geometric level set model for ultrasounds analysis. *International Journal of Computer Vision*, 46(3), 1999.

[25] Jiuyu Sun, Nilanjan Ray, and Hong Zhang. Vfccv snake: A novel active contour model combining edge and regional information. pages 927–931. IEEE, October 2014.

[26] B Van Brunt. *The calculus of variations*. Universitext The calculus of Variations. 2004.

[27] O Rebecca Vincent and Olusegun Folorunso. A descriptive algorithm for sobel image edge detection. In *Proceedings of Informing Science & IT Education Conference (InSITE)*, volume 40, pages 97–107. Informing Science Institute California, 2009.

[28] Li Wang, Lei He, Arabinda Mishra, and Chunming Li. Active contours driven by local gaussian distribution fitting energy. *Signal Processing*, 89(12):2435–2447, 2009.

[29] Robert Weinstock. *Calculus of Variations : with applications to physics and engineering*. International series in pure and applied mathematics. McGraw-Hill, New York, 1952.

[30] Daniel J Withey and Zoltan J Koles. A review of medical image segmentation: methods and available software. *International Journal of Bioelectromagnetism*, 10(3):125–148, 2008.

[31] James S. Xiaowei Zhou, James S. Xiaojie Huang, James S. Duncan, and James S. Weichuan Yu. Active contours with group similarity. pages 2969–2976. IEEE, June 2013.

[32] Xianghua Xie and M. Mirmehdi. Rags: region-aided geometric snake. *Image Processing, IEEE Transactions on*, 13(5):640–652, May 2004.

[33] Chenyang Xu and Jerry L Prince. Gradient vector flow: A new external force for snakes. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 66–71. IEEE, 1997.

[34] Chenyang Xu and Jerry L. Prince. Gradient vector flow deformable models-10. In *Handbook of Medical Imaging*, pages 159–169. 2000.

[35] Zheng Ying, Li Guangyao, Sun Xiehua, and Zhou Xinmin. Geometric active contours without re-initialization for image segmentation. *Pattern Recognition*, 42(9):1970–1976, 2009.